

Wilhelm Hasselbring (Hrsg.)

WISENT:

Wissensnetz Energiemeteorologie

Schlussbericht

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01C5968 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Inhaltsverzeichnis

1. Einleitung	3
2. Kurzdarstellung	4
3. Eingehende Darstellung	6

Lieferobjekte zu den Arbeitspaketen

CSCW-Anforderungen der Community Energiemeteorologie (AP 1.1)	13
Grid-Anforderungen der Community Energiemeteorologie (AP 1.1)	23
CSCW-Erfahrungen der Community Energiemeteorologie (AP 1.2)	29
Erfahrungen mit Grid-Technologien im Projekt WISENT (AP 1.2)	35
Bericht zu den Auswirkungen der DLR-IT-Sicherheitsstrukturen beim Datentransfer mit Partnern außerhalb des DLR (AP 2.1)	85
Überwachung des Serverparks und der Datenströme mit Nagios (AP 2.1)	89
Das MSG-Tailor System zur Extraktion von Meteosat Second Generation value adding Datenprodukten (AP 2.2 / 3.3)	99
Integration energiemeteorologischer Daten in Geographische Informationssysteme (AP 2.3)	109
Anforderungskatalog für Qualitätssicherungsmetadaten im Datenaustausch zwischen Universität Oldenburg und meteocontrol GmbH (AP 3.1)	123
Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten im Projekt WISENT (AP 3.1)	127
Qualitätssicherungsmetadaten und Validierungen im Datenaustausch zwischen Universität Oldenburg und meteocontrol GmbH (AP 3.1)	163
GridFTP-Client (AP 3.2)	173
Betriebskonzept des Grid-Speicherelements des DFD (AP 3.2)	191
Einsatz von und Erfahrungen mit Web-Service-Technologien (AP 3.3)	201
Anbindung von meteocontrol-Applikationen an einen Web-Service (AP 3.3)	235
Effiziente Interaktion mit großen Datenobjekten mit Hilfe von Quicklooks (AP 4)	247
Erfahrungen mit visuellen Datenformaten (AP 4)	283
Strahlungstransportrechnung – Erfahrungsbericht Condor (AP 5.1)	289
Mesoskalige Modellierung (AP 5.2)	297
Parallele Simulation von Solarstromanlagen (AP 5.3)	357
Verarbeitung energiemeteorologischer Daten in Geographischen Informationssystemen (AP 5.4)	371

Weitere Lieferobjekte

WISENT-Verwertungsplan	385
WISENT-Geschäftsmodell	401

1 Einleitung

Erneuerbare Energiequellen gewinnen eine immer größere Bedeutung für unsere Energieversorgung und für eine von fossilen Energieträgern unabhängige Wirtschaft. Die zukünftige Energieversorgung als einer der wesentlichen Bereiche staatlicher Vorsorge macht daher gerade in dem noch jungen Bereich der erneuerbaren Energiequellen eine intensive und langfristig angelegte Zusammenarbeit in der Forschung über die Grenzen von Disziplinen und Organisationen hinweg notwendig. Der Wissenschaftliche Beirat der Bundesregierung Globale Umweltveränderungen (WBGU) stellt dazu fest, dass die Energieversorgung der Zukunft erhebliche vorausschauende Forschungsanstrengungen und Investitionen sowie die Einführung neuartiger Instrumente erfordert.

Der Schnittstelle zwischen Meteorologie und Energiesystemforschung kommt dabei eine besondere Bedeutung zu, da die Verfügbarkeit dieser neuen Energiequellen außerordentlich von den meteorologischen Randbedingungen beeinflusst sind. Die Notwendigkeit energiemeteorologischer Forschung für einen großskaligen Einsatz erneuerbarer fluktuierender Energiequellen wird auch vom WBGU hervorgehoben.

Die Entwicklung von energiespezifischen Methoden in der Fernerkundung und der Atmosphärenphysik und die Bereitstellung entsprechender Informationen für den Energiesektor stellt ein neues Anwendungsfeld mit einem langfristig hohen ökonomischen Wert dar.

Die Verbundpartner DLR, Universität Oldenburg und meteocontrol GmbH arbeiten seit Jahren in verschiedenen Projekten zusammen. Im Projekt ENVISOLAR (<http://www.envisolar.com>) wurden beispielsweise im Rahmen des Earth Observation Market Development Programs der Europäischen Weltraumorganisation die bisherigen Forschungsergebnisse in den Markt der Solarenergie-Branche eingeführt.

Um der wachsenden Bedeutung dieses Forschungsgebiets Rechnung zu tragen, haben zwei der antragstellenden Partner (DLR und Universität Oldenburg) 2004 mit Förderung der Helmholtz-Gemeinschaft Deutscher Forschungszentren das "virtuelle Institut für Energiemeteorologie" (vIEM) (<http://www.viem.de>) gegründet. Das Institut soll die ergänzenden Kompetenzen der Partner bündeln und ein bisher einzigartiges Zentrum für die wissenschaftliche Bearbeitung energiemeteorologischer Fragestellungen bilden. Mit dieser Konstruktion einer verteilten und interdisziplinär ausgerichteten Forschungseinrichtung ist bereits eine institutionelle Basis geschaffen, die im IT-Bereich eine Ergänzung durch innovative gridbasierte Techniken für den Zugriff auf verteilte und vielfach sehr heterogene Ressourcen und deren verteilte Bearbeitung notwendig macht.

Merkmal der Arbeiten im vIEM ist die gemeinsame Nutzung und Bearbeitung großer Datenbestände (in der Größenordnung von einigen Terabyte), für die zu Beginn des Vorhabens noch keine ausreichend leistungsfähigen Instrumente und Dienste verfügbar waren. Die Nutzung von Grid-Ressourcen ist daher als vielversprechender Lösungsweg für dieses Problem eingeschlagen worden.

Die Kooperationspartner beginnen mit einer verstärkten und langfristig angelegten Zusammenarbeit. Das Ziel ist eine an den Erfordernissen der zukünftigen Energieversorgung orientierte Forschung im Bereich der Energiemeteorologie. Dieses Ziel wird in Abstimmung, Kooperation und Vernetzung auch mit Partnern außerhalb des virtuellen Instituts auf nationaler und internationaler Ebene und insbesondere auch durch die Einbindung von Industriepartnern im Rahmen des Technologietransfers verfolgt.

Die neue Form des verteilten wissenschaftlichen Arbeitens innerhalb virtueller Organisationen (speziell dem vIEM) erfordert neue innovative Infrastrukturen und Dienstleistungen im Bereich der wissenschaftlichen Kommunikation, Information und Publikation. Wesentliches Augenmerk liegt hierbei auf einer effizienten Aufbereitung und Verfügbarkeit relevanter wissenschaftlicher Informationen durch entsprechende gridbasierte Informationszugänge.

2 Kurzdarstellung

2.1 Aufgabenstellung

Die Aufgabenstellung des Vorhabens ist es, gridbasierte Instrumente und Verfahren bereitzustellen, um die Anforderungen der zukünftigen Energieversorgung an die Energiemeteorologie zu erfüllen und die sich daraus ergebenden komplexen informationstechnischen Aufgaben in einem interdisziplinären Forschungsumfeld zu lösen. Konkret wird die wissenschaftliche Zusammenarbeit im bereits interdisziplinär ausgerichteten Helmholtz-Verbund „Virtuelles Institut für Energiemeteorologie“ (vIEM) durch die Nutzung verteilter Ressourcen und darauf aufbauender Grid-Dienste unterstützt.

2.2 Voraussetzungen

Das Vorhaben setzt auf Vorarbeiten der beteiligten Projektpartner auf, die bereits innovative Anwendungen im energiemeteorologischen Bereich entwickeln, die durch die Kombination von Daten aus mehreren Quellen und durch Verarbeitungsdienste unterschiedlicher Anbieter gekennzeichnet sind.

Das vIEM wurde bereits vor Beginn des WISENT-Vorhabens von einigen der Verbundpartner gegründet und repräsentiert bereits eine virtuelle Organisation, die im Rahmen des Vorhabens zu einer Grid-Community erweitert wird. Die entstehende Grid-Community „Energiemeteorologie“ wird in die D-Grid-Initiative (<http://www.d-grid.de>) eingebettet, deren Ziel es ist, eine nachhaltige Grid-Infrastruktur in Deutschland aufzubauen.

2.3 Planung und Ablauf des Vorhabens

Die Laufzeit des WISENT-Projektes erstreckt sich von Oktober 2005 bis September 2008. Das Projekt gehört damit zu den ersten Community-Projekten, die auf der D-Grid-Infrastruktur aufsetzen. Das Vorhaben ist in sieben Arbeitspakete unterteilt, deren zeitlicher Verlauf in Abbildung 1 dargestellt ist. Bei den gezeigten Laufzeiten gibt es leichte Abweichungen gegenüber der ursprünglichen Planung, was jedoch keine Auswirkungen auf die Gesamtlaufzeit oder die Projektergebnisse mit sich bringt.

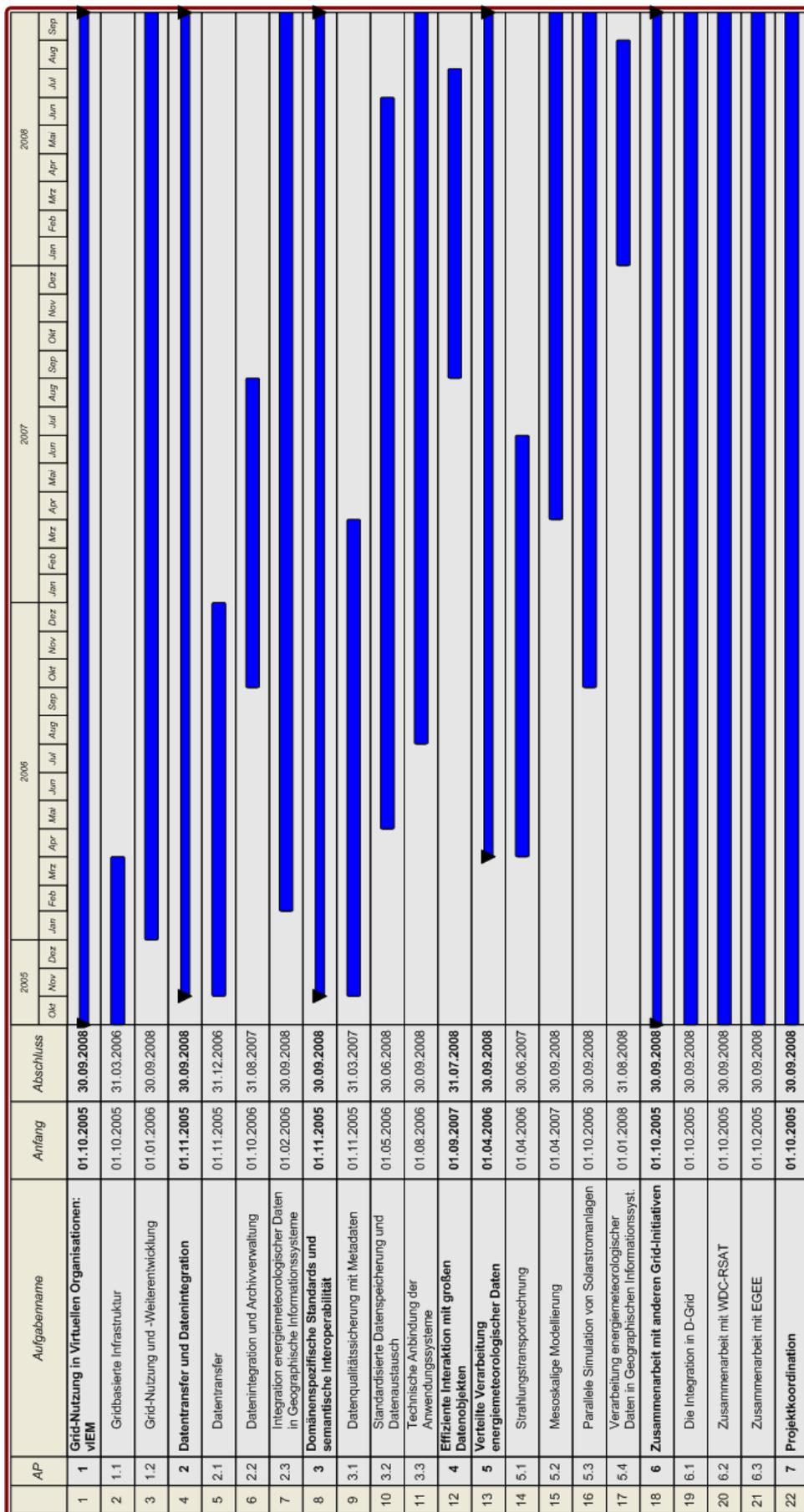


Abbildung 1: Laufzeit der Arbeitspakete

3 Eingehende Darstellung

3.1 Erzielte Ergebnisse

Im Folgenden wird auf die in den Arbeitspaketen erzielten Ergebnisse eingegangen. Dies schließt neben den Lieferobjekten, die Bestandteil dieses Dokumentes sind und auf die im Rahmen der Ergebnisdarstellung verwiesen wird, auch technische Umsetzungen mit ein, die im Folgenden ebenfalls beschrieben werden. Abbildung 2 zeigt eine Übersicht der Arbeitspakete des WISENT-Projektes, deren Ziele und Ergebnisse im Folgenden zusammengefasst werden.

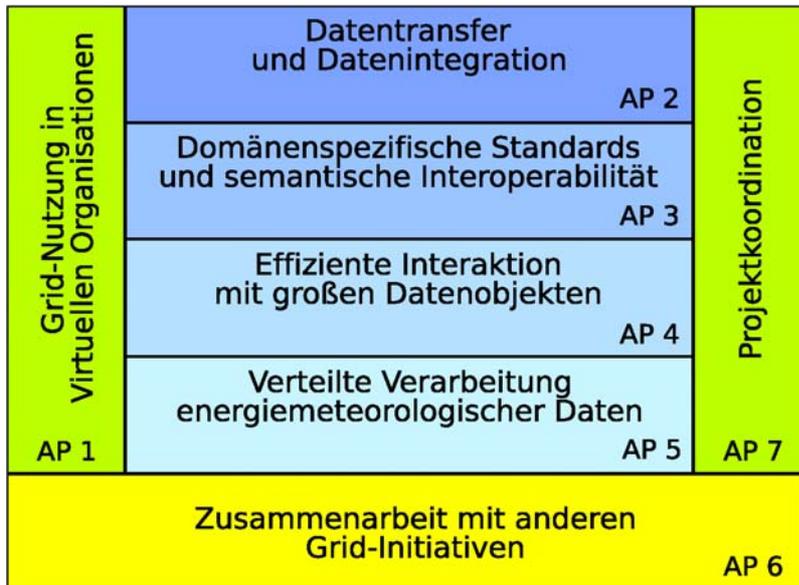


Abbildung 2: Arbeitspakete des WISENT-Projektes

AP 1: Grid-Nutzung in Virtuellen Organisationen

In diesem projektbegleitenden Arbeitspaket werden die organisatorischen Abläufe im Virtuellen Institut für Energiemeteorologie (vIEM) an die kollaborative Nutzung von Grid-Technologien angepasst und die notwendige Grid-Infrastruktur aufgebaut. Insbesondere werden Softwarewerkzeuge zum vernetzten Wissens- und Informationsmanagement evaluiert und eingeführt sowie Rahmenbedingungen für die gemeinsame Ressourcennutzung ausgehend von bisherigen Kooperationen erarbeitet. Dieses Arbeitspaket unterteilt sich in zwei Unterarbeitspakete, wobei sich AP 1.1 auf den Aufbau der grundlegenden Community-spezifischen Infrastruktur konzentriert und AP 1.2 darauf aufbauend die Nutzung und Weiterentwicklung dieser Infrastruktur fokussiert.

AP 1.1: Gridbasierte Infrastruktur

Die Ziele dieses Arbeitspaketes sind der Aufbau einer gridbasierten Infrastruktur für die Grid-Community „Energiemeteorologie“ und der Aufbau einer Infrastruktur zur Kommunikation, Koordination und Kooperation innerhalb dieser Community. Die Ergebnisse dieses Arbeitspaketes werden in den Lieferobjekten „CSCW-Anforderungen der Community Energiemeteorologie“ (ab Seite 13) und „Grid-Anforderungen der Community Energiemeteorologie“ (ab Seite 23) zusammengefasst. Die gesteckten Ziele konnten vollständig umgesetzt werden. Aufbauend auf diesen Ergebnissen erfolgt die Nutzung und Weiterentwicklung der Infrastruktur im nachfolgenden Arbeitspaket 1.2.

AP 1.2: Grid-Nutzung und –Weiterentwicklung

Die in Arbeitspaket 1.1 erarbeitete Community-spezifische Infrastruktur wird in diesem Arbeitspaket genutzt und weiterentwickelt. Die Lieferobjekte „*CSCW-Erfahrungen der Community Energiemeteorologie*“ (ab Seite 29) und „*Erfahrungen mit Grid-Technologien im Projekt WISENT*“ (ab Seite 35) fassen die gesammelten Erfahrungen zusammen. Details zur Infrastruktur und gesammelte Erfahrungen und Richtlinien für die gemeinsame Ressourcennutzung wurden zudem im TikiWiki-System in elektronischer Form festgehalten, um eine gemeinschaftliche Arbeit an diesen Texten zu vereinfachen. Durch die enge Integration mit der D-Grid-Infrastruktur (siehe AP 6.1) ergeben sich weitere Rahmenbedingungen (z.B. im Bereich der verwendeten Grid-Middleware und des VO-Managements), die auch für die Community-spezifische Grid-Infrastruktur gelten. Anforderungen aus der Community wurden wiederum an das D-Grid herangetragen. Die gesteckten Ziele der Nutzung und Weiterentwicklung der Community-spezifischen Infrastruktur werden als erfolgreich angesehen, jedoch ist dieser Prozess nicht auf die Projektlaufzeit beschränkt, da auch nach Projektende noch Änderungen abzusehen sind, die sich alleine schon durch zu erwartende Änderungen der D-Grid-Infrastruktur ergeben.

AP 2: Datentransfer und Datenintegration

Dieses Arbeitspaket fokussiert die Automatisierung, Vereinheitlichung und Überwachung der Datenübertragung im operationellen Betrieb der Projektpartner. Es bildet damit eine Grundlage für den Einsatz von Grid-Technologien für verteilte Berechnungen.

AP 2.1: Datentransfer

Die Ergebnisse dieses Arbeitspaketes werden in zwei Lieferobjekten zusammengefasst. Das erste Lieferobjekt ist ein „*Bericht zu den Auswirkungen der DLR-IT-Sicherheitsstrukturen beim Datentransfer mit Partnern außerhalb des DLR*“ (ab Seite 85). Das zweite Lieferobjekt betrifft die „*Überwachung des Serverparks und der Datenströme mit Nagios*“ (ab Seite 89) bei meteocontrol.

AP 2.2: Datenintegration und Archivverwaltung

Das „Data Information and Management System“ DIMS des DLR bietet umfassende Funktionalität zum Management großer heterogener digitaler Datenbestände mit Raum-Zeit-Bezug. Zur Extraktion von Raum-Zeit-Kuben existiert für die Daten der Meteosat-Satelliten-Serie ein Tailor-Processing-System, das einheitliche Datensätze aus mehreren, einzeln archivierten Meteosat-Datensätzen automatisch zusammensetzt. In diesem Arbeitspaket wird ein einheitlicher Zugriff auf die am DFD archivierten Produkte der Meteosat-Satelliten ermöglicht. Es geht vor allem darum, den Umgang mit evolvierenden Datenformaten zu verfeinern, mit denen sich zukünftige Datenintegrationsaufgaben ähnlicher Art bewältigen lassen. Die Ergebnisse dieses Arbeitspaketes sind im Lieferobjekt „*Das MSG-Tailor System zur Extraktion von Meteosat Second Generation value adding Datenprodukten*“, (ab Seite 99) zusammengefasst.

AP 2.3: Integration energiemeteorologischer Daten in Geographische Informationssysteme

Die ursprünglichen Planungen zu diesem Arbeitspaket waren stark an der Nutzung des DLR-internen Werkzeuges STEPS (Expert System for Solar Thermal Power Stations) orientiert. Im Rahmen einer Analyse der notwendigen Arbeiten innerhalb dieses Arbeitspaketes wurde von DLR-TT jedoch der Schluss gefasst, von der geplanten Weiterentwicklung des vorhandenen STEPS-Programmcodes abzusehen und eine Neuentwicklung auf Grundlage bestehender Erfahrungen anzustreben. Die Ergebnisse der Entwicklungsarbeiten sind in dem Lieferobjekt „*Integration energiemeteorologischer Daten in Geographische Informationssysteme*“ (ab Seite 109) zusammengefasst.

AP 3: Domänenspezifische Standards und semantische Interoperabilität

Die Erkenntnisse aus Arbeitspaket 2 werden in diesem Arbeitspaket auf die Problematik des unregelmäßigen Datenaustausches in energiemeteorologischen Entwicklungsprojekten übertragen. Außerdem werden Maßnahmen zur Verfolgung der Datenqualität innerhalb von Prozessierungsketten entwickelt und Schnittstellen der Dienste vereinheitlicht. Das Gesamtziel des Pakets besteht in der Erreichung der standardgestützten Interoperabilität zwischen Anwendungen der Projektpartner und, in weiterer Perspektive, auch mit externen Projekten.

AP 3.1: Datenqualitätssicherung mit Metadaten

In diesem Arbeitspaket geht es vor allem um die Definition relevanter Metadaten zur Sicherung der Datenqualität von Datenprodukten in operationellen und forschungsgetriebenen Verarbeitungsketten. Die Ergebnisse dieses Arbeitspaketes werden innerhalb der folgenden Lieferobjekte dargestellt:

- „Anforderungskatalog für Qualitätssicherungsmetadaten im Datenaustausch zwischen Universität Oldenburg und meteocontrol GmbH“ (ab Seite 123)
- „Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten im Projekt WISENT“ (ab Seite 127)
- „Qualitätssicherungsmetadaten und Validierungen im Datenaustausch zwischen Universität Oldenburg und meteocontrol GmbH“ (ab Seite 163)

AP 3.2: Standardisierte Datenspeicherung und Datenaustausch

Im Gegensatz zu Arbeitspaket 2.1 behandelt dieses Arbeitspaket den nichtoperationellen bzw. nichtregelmäßigen Datenaustausch. Zudem geht es um Monitoring-Verfahren für die Datenübertragung und Qualitäts- und Fehlerursacheinformationen. Ebenfalls wird der Einsatz von Verschlüsselungsverfahren geprüft. In den folgenden Lieferobjekten werden die Ergebnisse des Arbeitspakets zusammengefasst:

- „GridFTP-Client“ (ab Seite 173)
- „Betriebskonzept des Grid-Speicherelements des DFD“ (ab Seite 191)

Gesammelte Erfahrungen mit GridFTP und dCache wurden dem Dokument „Erfahrungen mit Grid-Technologien im Projekt WISENT“ (ab Seite 35) hinzugefügt.

AP 3.3: Technische Anbindung der Anwendungssysteme

In diesem Arbeitspaket geht es um den Einsatz von Web-Service-Technologien, um externen Nutzern den Zugriff auf Community-spezifische Dienste über einheitliche und standardisierte Schnittstellen anbieten zu können. Die Ergebnisse dieses Arbeitspaketes werden in den Lieferobjekten „Einsatz von und Erfahrungen mit Web-Service-Technologien“ (ab Seite 201) und „Anbindung von meteocontrol Applikationen an einen Web-Service“ (ab Seite 235) ausführlich beschrieben.

AP 4: Effiziente Interaktion mit großen Datenobjekten

Arbeitspaket 4 behandelt das Problem der Visualisierung von großen verteilten Datenmengen bei knapper Übertragungsbandbreite. Es werden hier zum Einen fachliche Lösungen durch neuartige Visualisierungsformen und zum Anderen technische Lösungen durch die vorausschauende Datenverteilung im Grid verfolgt. Die Ergebnisse des Arbeitspaketes sind in den Lieferobjekten „Effiziente Interaktion mit großen Datenobjekten mit Hilfe von Quicklooks“ (ab Seite 247) und „Erfahrungen mit visuellen Datenformaten“ (ab Seite 283) zusammengefasst.

AP 5: Verteilte Verarbeitung energiemeteorologischer Daten

Dieses Arbeitspaket beinhaltet die Nutzung von Gridtechnologien zur Parallelisierung von Rechenvorgängen. Bei der Durchführung werden erstens Skalierbarkeitsvorteile für vorhandene Systeme erreicht, zweitens aber auch allgemeine Erfahrungen gesammelt, die für die Energiemeteorologie-Community den Weg zur flexiblen Nutzung von externen Gridressourcen öffnen.

AP 5.1: Strahlungstransportrechnung

Ziel dieses Arbeitspaketes ist die exemplarische Erschließung vorhandener verteilter Rechenkapazität für die Strahlungstransportsimulation. Die Ergebnisse des Arbeitspakets werden im Lieferobjekt „*Strahlungstransportrechnung – Erfahrungsbericht Condor*“ (ab Seite 289) beschrieben.

AP 5.2: Mesoskalige Modellierung

Die Modellierung der zeitlich und räumlich stark variablen Energieerzeugung aus Solar- und Windkraftwerken erfordert zunehmend den Einsatz meteorologischer numerischer Modelle auf überwiegend regionaler Skala (mesoskalige Modellierung). Die Ergebnisse dieses Arbeitspakets werden durch das Lieferobjekt „*Mesoskalige Modellierung*“ (ab Seite 297) zusammengefasst.

AP 5.3: Parallele Simulation von Solarstromanlagen

In diesem Arbeitspaket wird die Nutzung verteilter Rechenkapazitäten durch parallele Simulation auf Basis von Grid-Technologien ermöglicht. Die meteocontrol GmbH kann auf diesem Wege in Zukunft die Simulation spezieller Anwendungen unter Nutzung von Grid für unterschiedliche Anwendungsbereiche anbieten. Die Ergebnisse des Arbeitspakets sind im Lieferobjekt „*Parallele Simulation von Solarstromanlagen*“ (ab Seite 357) zusammengefasst.

AP 5.4: Verarbeitung energiemeteorologischer Daten in Geographischen Informationssystemen

Einer der Schwerpunkte bei der Verarbeitung energiemeteorologischer Daten bei DLR-TT ist die Prozessierung von Fernerkundungsdaten, um Karten der solaren Einstrahlung zu erstellen. Die Rohdaten werden im DIMS des DFD geholt und bei DLR-TT verarbeitet. Dabei geht es oft auch um große Regionen, wie z.B. den Mittelmeerraum. Auf einem einzelnen Rechner würde die Verarbeitung der Daten mehrere Jahre dauern. Eine Verteilung auf mehrere Rechner ist daher zwingend notwendig, um derartige Anfragen beantworten zu können. Auf Basis der Arbeiten in AP 2.3 wird hier ein System entwickelt, welches die Quelldaten aufteilt, die Prozesse verteilt und anschließend die Ergebnisse wieder zusammenfügen kann. Die entwickelte Systemarchitektur soll insbesondere ausreichend zukunftsorientiert sein, um die Nutzung von D-Grid-Ressourcen über WISENT hinaus zu ermöglichen. Die Ergebnisse dieses Arbeitspaketes werden in dem Lieferobjekt „*Verarbeitung energiemeteorologischer Daten in Geographischen Informationssystemen*“ (ab Seite 371) zusammengefasst.

AP 6: Zusammenarbeit mit anderen Grid-Initiativen

Während des gesamten Projekts werden im Rahmen des Arbeitspakets 6 Kontakte zu anderen Grid-Initiativen aufgebaut. Im Einzelnen findet die Zusammenarbeit mit D-GRID, EGEE und WDC-RSAT statt. Das Ziel dieser Zusammenarbeit ist die Einbindung der Energiemeteorologie-Community mit ihren Anforderungen und fachspezifischem Wissen in nationale und internationale gridbezogene Forschungsprojekte und die Vermeidung redundanter Bearbeitungen bereits erforschter Probleme.

AP 6.1: Die Integration in D-Grid

Bereits Ende 2005 gab es Gespräche mit UNICORE-Entwicklern im Forschungszentrum Jülich bzgl. der Einsatzmöglichkeiten der Grid-Middleware UNICORE. Auch die Grid-Middleware Globus Toolkit findet in Arbeitspaket 1.1. und 1.2 besondere Berücksichtigung. Dies ist für die Integration ins D-Grid relevant, weil diese Middlewares Bestandteil der D-Grid-Infrastruktur sind.

Im Juni 2006 fand ein Gespräch mit einem Vertreter des D-Grid-Integrationsprojektes bzgl. der Einsatzmöglichkeiten von GAT und GridSphere statt. Zudem gab es regelmäßige Treffen mit den D-Grid-Projekten C3-Grid und GDI-Grid aufgrund sich überschneidender Themengebiete.

Die Anforderungen der „Energiemeteorologie“-Community an die D-Grid-Infrastruktur wurden u.a. auf Workshops des D-Grids zur Nachhaltigkeit präsentiert. Zudem wurde ein Geschäftsmodell für die D-Grid-Community „Energiemeteorologie“ auf Grundlage der vom D-Grid gelieferten Vorlage erstellt und an das D-Grid geliefert (siehe Seite 403).

Die Integration ins D-Grid wird durch die im Aufbau befindliche Hardware aus den BMBF-Sonderinvestitionen weiter gestärkt. Im OFFIS wurde ein High-Performance-Cluster mit 192 Rechenkernen und einem vier Terabyte umfassenden Festplattenspeicher beschafft und in die D-Grid-Infrastruktur integriert. Beim DLR wurde eine Storage-Einheit beschafft und ebenfalls in die D-Grid-Infrastruktur integriert.

Darüber hinaus ist Prof. Dr. Hasselbring als wissenschaftlicher Leiter von WISENT Mitglied des D-Grid-StA. Aufgrund der regelmäßigen Sitzungen des D-Grid-StA besteht somit ein enger Kontakt zum D-Grid-Integrationsprojekt und zu den Community-Projekten.

AP 6.2: Zusammenarbeit mit WDC-RSAT

Eine enge Kooperation zwischen WISENT und dem Weltdatenzentrum für Fernerkundung der Atmosphäre (WDC-RSAT) besteht, da sowohl die Beteiligung an WISENT als auch die Funktion des Managing Directors des Zentrums in der Abteilung Klima- und Atmosphärenprodukte des DLR-DFD angesiedelt sind.

In diesem Arbeitspaket lag der Schwerpunkt auf der Integration der MSG-Produkte in den WDC-RSAT-Webauftritt und in die Datenbibliothek sowie auf der Abstimmung von Metadatenstrukturen zwischen WISENT und C3-Grid.

AP 6.3: Zusammenarbeit mit EGEE

Ziele der Kooperation der Energiemeteorologie-Community mit dem EU-Projekt „Enabling Grids for E-science in Europe“ (EGEE) sind der Austausch von Erfahrungen und die Berücksichtigung der Adaptierbarkeit der entwickelten Verfahren und Dienste in dieses europäische Grid.

Eine Mitarbeiterin der AG Energiemeteorologie an der Universität Oldenburg hat als ehemaliges CERN-Mitglied das Datenmanagement- und Datenvisualisierungspaket ROOT eingeführt, bei dem Weiterentwicklungen in Richtung Parallelisierung und Grid-Computing gemacht werden. Bei einem Besuch des OFFIS auf der GridKa-Schule 2006 in Karlsruhe, die von EGEE-Mitgliedern veranstaltet wurde, gab es neben einem Vortrag über WISENT und den Einsatz der Middleware Condor einen Informationsaustausch über die Querverbindungen, insbesondere die Einbindung von ROOT in einen Grid-Kontext.

Bei einem Besuch von Jens Rehn vom CERN wurde das EGEE-Projekt Phedex vorgestellt, insbesondere das Verfahren der verteilten Verarbeitung sehr großvolumiger Experimentdaten. Es wurden die Unterschiede und Gemeinsamkeiten sowie die Anwendungsmöglichkeiten in WISENT diskutiert. Außerdem wurde die Nutzung von ROOT (s.o.) thematisiert.

Eine Zusammenarbeit mit EGEE erfolgte auch indirekt durch die inzwischen enge Integration in das D-Grid. Dabei kommt u.a. die in EGEE entwickelte Grid-Middleware gLite zum Einsatz.

AP 7: Projektkoordination

Die Rolle des Projektkoordinators wurde von OFFIS übernommen. Dies beinhaltet die fachliche und administrative Koordination der Teilprojekte, die Repräsentation gegenüber dem Projektträger und beteiligten Institutionen, die Zusammenführung und Präsentation von Ergebnissen in Berichts- und Vortragsform als auch Öffentlichkeitsarbeit zur Verbreitung der Projektidee wie die professionelle Gestaltung einer Website und die Durchführung von Veranstaltungen.

Änderungen im Projektverlauf wurden stets mit den Projektpartnern und dem Projektträger abgestimmt. Des Weiteren wurden Berichte, Dokumentationen, Software u. a., die als Ergebnisse der einzelnen Arbeitspakete ausgewiesen sind, gesammelt. Die Erstellung von Zwischenberichten, die Überarbeitung des Verwertungsplans und die Erstellung eines Geschäftsmodells wurden koordiniert. Ebenfalls wurden die Projekttreffen am Projektbeginn und jeweils zu den Meilensteinen organisiert. Wichtiges Ergebnis dieses Arbeitspaketes ist die Dokumentation des jeweiligen Projektstandes und der zu den Meilensteinen fertig gestellten Ergebnisse gegenüber dem Projektträger. Die Erstellung dieses Schlussberichtes gehört ebenfalls zu den Tätigkeiten der Projektkoordination.

3.2 Verwertbarkeit des Ergebnisses

Der voraussichtliche Nutzen und die Verwertbarkeit der Ergebnisse werden im WISENT-Verwertungsplan ab Seite 385 ausführlich beschrieben.

3.3 Veröffentlichungen

Im Projektverlauf sind die folgenden Veröffentlichungen entstanden:

WISENT: eScience for Energy Meteorology

Hasselbring, Wilhelm; Heinemann, Detlev; Hurka, Johannes; Scheidsteger, Thomas; Bischofs, Ludger; Mayer, Christoph; Ploski, Jan; Scherp, Guido; Lohmann, Sina; Hoyer-Klick, Carsten; Erbertseder, Thilo; Gesell, Gerhard; Schroedter-Homscheidt, Marion
In: Pre-conference proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing. Amsterdam, Netherlands, Dezember 4-6, 2006. IEEE

Grid-based Processing of High-volume Meteorological Data Sets

Scherp, Guido; Ploski, Jan; Hasselbring, Wilhelm
In: Euro-Par 2006: Parallel Processing. Dresden, Germany, August 30-31, 2006. Lecture Notes in Computer Science

Introducing Version Control to Database-Centric Applications in a Small Enterprise

Ploski, Jan; Hasselbring, Wilhelm; Rehwinkel, Jochen; Schwierz, Stefan
In: Software Engineering Challenges in Small Software Organizations. IEEE Software, 25(1):38-44, Januar/Februar, 2007.

Grid-based modeling in "Wissensnetz Energiemetereologie"

Ploski, Jan; Petroliagis, Thomas; Heinemann, Detlev; Scheidsteger, Thomas; Hasselbring, Wilhelm
DACH Meteorologentagung, Hamburg, 2007

Implementing the Weather Research & Forecasting Model on the German Grid

Ploski, Jan; Petroliagis, Thomas; Hasselbring, Wilhelm
7th EMS Annual Meeting, San Lorenzo, Spanien, 2007

The WISENT Grid Architecture: Coping with Firewalls and NAT

Scherp, Guido; Hasselbring, Wilhelm; Ploski, Jan
German e-Science Conference 2007, Baden-Baden, Mai 2007

Software Fault Diagnosis for Grid Middleware with Bayesian Networks

Ploski, Jan; Hasselbring, Wilhelm
In: Tagungsband Software Engineering 2007, Gesellschaft für Informatik, Hamburg, März 2007.

Grid-based Deployment and Performance Measurement of the Weather Research & Forecasting Model

Ploski, Jan; Scherp, Guido; Petroligis, Thomas I.; Büchner, Otto; Hasselbring, Wilhelm
Future Generation Computer Systems, 2008

3.4 Open-Source-Software

Im Verlauf des Projektes sind die im Folgenden beschriebenen drei Werkzeuge entwickelt worden, die als Open-Source-Software frei verfügbar sind. Weitere Informationen zu diesen Werkzeugen sind auch auf den WISENT-Webseiten (<http://wisent.d-grid.de>) zu finden. Neben diesen Eigenentwicklungen sind auch Erweiterungen und Korrekturen in andere Entwicklungsprojekte eingeflossen. Beispielsweise sind Teile des an die Bedürfnisse der Energiemeteorologie-Community angepassten TikiWiki-Systems in das offizielle Release eingeflossen.

3.4.1 Grid-FTP-Client

Im Rahmen des Arbeitspakets "Standardisierte Datenspeicherung und Datenaustausch" wurde ein interaktiver, benutzerfreundlicher und mit dCache kompatibler GridFTP-Client entwickelt (siehe Lieferobjekt „GridFTP-Client“ ab Seite 173). Die Software steht unter <http://bi.offis.de/gridftp/> zum Download bereit.

3.4.2 GIS-Metadaten-Manager

Der GIS-Metadaten-Manager entstand im Rahmen des Arbeitspakets "Integration energiemeteorologischer Daten in Geographische Informationssysteme" und wird im gleichnamigen Lieferobjekt ab Seite 109 beschrieben. Der GIS-Metadaten-Manager ist unter SourceForge verfügbar (<http://sourceforge.net/projects/gis-manager/>).

3.4.3 Super-Overlay-Generator

Der Super-Overlay-Generator entstand innerhalb des Arbeitspakets „Effiziente Interaktion mit großen Datenobjekten“ und wird im Lieferobjekt „Effiziente Interaktion mit großen Datenobjekten mit Hilfe von Quicklooks“ ab Seite 247 beschrieben. Das Werkzeug steht auf SourceForge zum Download bereit unter <http://sourceforge.net/projects/superoverlaygen/>).



CSCW-Anforderungen der Community Energiemeteorologie

Arbeitspaket 1.1

Ludger Bischofs
OFFIS – Institut für Informatik
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de>

1 Einleitung

Das Arbeitspaket 1 hat zum Ziel, den kooperativen Dialog zwischen den im virtuellen Institut für Energiemeteorologie (vIEM) zusammenarbeitenden Projektpartnern zu fördern und technische Mittel bereitzustellen, um die Vision des vIEM Wirklichkeit werden zu lassen. Der Einsatz von Werkzeugen des Computer Supported Cooperative Work (CSCW) steht nicht in unmittelbarem Zusammenhang zu Grid-Technologien, ist jedoch eine wesentliche Voraussetzung zur Bildung und nachhaltigen Festigung einer jungen bzw. in der Entstehung befindlichen Grid-Community. Zudem unterstützen die zum Einsatz kommenden Werkzeuge des kooperativen Arbeitens das gemeinsame Management von Wissen und den Austausch von Erfahrungen im Wissensnetz Energiemeteorologie.

Dieses Dokument fasst die Gründe für die bisherige sehr geringe Nutzung bereits eingerichteter Werkzeuge im vIEM zusammen, um in einem zweiten Schritt die Anforderungen an eine die Forschungskollaboration wirksam unterstützende CSCW-Plattform zu beschreiben. In einem dritten Schritt werden die auf Basis der Anforderungen ausgewählten Werkzeuge beschrieben.

2 Vorgehen zur Erhebung der Anforderungen

Die Aufnahme des Ist-Zustands und der Anforderungen an den CSCW-Einsatz erfolgten auf einem vIEM-Treffen am 15. November 2005 beim DLR-DFD in Oberpfaffenhofen. Auf Grundlage dieses Treffens wurde durch OFFIS eine Dokument-Vorlage erstellt, in der alle Projektpartner ihre Anforderungen an den CSCW-Einsatz nochmals schriftlich darlegen konnten.

Das von allen Projektpartnern ausgefüllte Dokument zur Bestimmung des Ist-Zustands und zur Aufnahme der Anforderungen besitzt die folgende Struktur mit Fragestellungen zu jedem Punkt:

- Ausgangssituation und Zielsetzung
 - Wie ist der aktuelle Stand bzgl. des CSCW-Einsatzes?
 - Welche Werkzeuge werden für was eingesetzt?
 - Wo liegen die Probleme? Was könnte verbessert werden?
 - Was soll sich mit der Einführung neuer CSCW-Werkzeuge ändern?
 - Was sind die Erwartungen?
- Funktionale Anforderungen
 - Welche Funktionen sollten von CSCW-Werkzeugen unterstützt werden? Asynchrone vs. synchrone Kommunikation, Dateiablage, Kalender, Wiki, Videokonferenz, gemeinsames Arbeiten an Dokumenten, Chat, Glossar usw.
 - Welche Funktionen sind auf jeden Fall erforderlich und welche sind optional („nice-to-have“)? Eine Prioritätenliste wäre hilfreich. Randbedingungen wie technische Probleme mit freien Ports sollten hier ebenfalls mit aufgenommen werden.
- Nicht-Funktionale Anforderungen
 - Nicht-funktionale Anforderungen sind beispielsweise eine leichte Bedienbarkeit, 100%ige Zuverlässigkeit, hohe Performanz beim Dateizugriff oder ähnliches. Weitere Anforderungen könnten sein: geringer Administrationsaufwand, gute Änderbarkeit, hohe Skalierbarkeit, gute Ergonomie, hohe Sicherheit usw.
 - Die Anforderungen sollten genauer erläutert werden. Natürlich möchte man immer Systeme mit den besten nicht-funktionalen Anforderungen haben. Da dies in der Realität schwer zu erreichen ist, sollte hier eine Priorisierung

stattfinden.

- Risikominimierung
 - Was muss getan werden, damit die CSCW-Werkzeuge auch eingesetzt werden?
 - Worin liegen die größten Risiken, dass die Werkzeuge abgelehnt werden, und wie kann man diese minimieren?
- Skizze des Lebenszyklus und der Gesamtarchitektur
 - Wie können die neuen Technologien eingeführt werden?
 - In welcher Reihenfolge sollte dies geschehen (Bezug nehmen auf die in 1.2/1.3 erarbeitete Prioritätenliste)? Schrittweise oder alles auf ein mal?
 - Wie fügen sich die Technologien in die bestehende Systemlandschaft ein? Wie kann die Nachhaltigkeit gewährleistet werden (auch nach Beendigung des WISENT-Projektes)?

3 Ausgangssituation und Zielsetzung

Bereits auf dem WISENT-Kickoff-Treffen am 20. Oktober 2005 wurde festgehalten, dass in erster Linie Störfaktoren erhoben werden sollen, die die Nutzung von CSCW-Werkzeugen im vIEM erschweren und für die geringe Nutzung verantwortlich sind. Auf dem vIEM-Treffen am 15. November 2005 beim DLR-DFD in Oberpfaffenhofen wurde deshalb der Ist-Zustand zum CSCW-Einsatz aufgenommen.

Zum Zeitpunkt der Befragung existierten im vIEM(-Umfeld) folgende CSCW-Werkzeuge:

1. BSCW-Server mit einem für das WISENT-Projekt eingerichteten Arbeitsbereich (Administration: OFFIS/Uni OL).
2. TeamSite für vIEM; Teil der Microsoft SharePoint-Software bei DLR (Administration: DLR-TT/DLR-IT).
3. TWiki, das unabhängig von vIEM im Projekt von Dr. Ebertseder genutzt wird.

Der BSCW-Server wird seit dem WISENT-Kickoff-Treffen als Dateiablage genutzt. Die vIEM-TeamSite wurde bisher nur sehr wenig genutzt. Die Anwesenden äußerten keine starken Präferenzen für die eine oder andere Software. Als Grund der Nichtnutzung wurde eher ihre Unbekanntheit genannt.

3.1 Ist-Zustand des CSCW-Einsatzes

Folgende Werkzeuge werden derzeit für die Kommunikation, Koordination und Kooperation im vIEM verwendet.

- E-Mail: Hauptwerkzeug zur Kommunikation und zum Austausch von Dokumenten und Daten kleineren Ausmaßes.
 - Probleme: "CC-Katastrophe", Textakkumulation erzeugt Unübersichtlichkeit ("Re-Katastrophe")
- FTP: Austausch von (Satelliten-)Daten und Datenprodukten größeren Ausmaßes nach vorheriger Absprache per Telefon oder E-Mail.
- Telefon: bilaterale Kommunikation
- CVS: Versionsmanagement für Quellcode innerhalb EHF begonnen; verteiltes Entwickeln macht noch Mühe und ist ungewohnt.
- Sharepoint/TeamSite: Diese Software wird kaum genutzt und es sind nur wenige Dokumente vorhanden. Das System wirkt auf den Benutzer zum Teil träge und der

Zugriff auf die eigentlichen Dateien ist aufgrund der tiefen Verschachtelung zu umständlich.

- BSCW: Der Server steht erst seit kurzem zur Verfügung und es wurden zahlreiche Dokumente als Grundstock eingepflegt.
- Physische Treffen finden alle 4-6 Monate an einem der drei Standorte statt. Der Aufwand und Zeiteinsatz der Anreisenden ist groß.
- Papierprotokoll der wöchentlichen Sitzung der Oldenburger Solargruppe:
 - + Informationen direkt verschriftlicht
 - + keine zeitaufwändige Überarbeitung nötig
 - + nachhaltige Referenz
 - + Protokollant alternierend -> Arbeitsverteilung
 - - eigentlich zusammenhängende, aufeinander aufbauende Informationen sind verstreut
 - - Keine Erinnerungsfunktion an Fristen und Aufgaben
 - - Keine Wiedervorlage, keine Index -> häufige Redundanzen bzw. vergessene Vor-Informationen

4 Funktionale Anforderungen

Die folgenden funktionalen Eigenschaften sind von den Projektpartnern gewünscht:

- Gemeinsame (zentrale) Dokumentenablage inkl. URLs; konfigurierbare Rechte für Verzeichnisse; öffentliche Bereiche einzurichten und über Website zu verlinken erspart Doppelarbeit.
- Gemeinsame synchrone und asynchrone Dokumentenbearbeitung (zzgl. einfach anwendbarem web-basierten Versionskontrollsystem).
- Glossar bzw. Wissensbasis (Wiki)
- Gemeinsamer Terminkalender: als Referenz geeignet; zum Abstimmen weniger, da Problem der Kompatibilität mit anderen Kalenderformaten.
- Peer-to-Peer-Telefonie (z.B. Skype) für kleine spontane Konferenzen/Telefonate
- Telefonkonferenz: bisher nur einmal zur Vorbereitung des WISENT-Antrages: geringer technischer Aufwand, keine Reisen.
- Videokonferenz mit 2-4 Partnern recht einfach mit Webcams an Desktops durchzuführen.
- Whiteboard-Funktion bei Videokonferenzen
- Workflow-Kontrolle innerhalb des Projektes
- Content-Management-System für Webseiten: verteilte Bearbeitung; Aktualität; Arbeitsteilung
- Diskussionsforum: viele Teilnehmer können gezielt nach Themen sortiert diskutieren; Upload von Dateien möglich; Archiv; auch Externe einladbar; Zugang konfigurierbar
- EDV-gestütztes Sitzungsprotokoll:
 - Direkt mit Notebook schreiben

- "Threads" verfolgen (Wiedervorlage und Referenz)
- Abgleich mit Server (nachträglich oder bei WLAN-Anbindung sofort)
- Gültigkeit alter Informationen und Absprachen gewährleisten? (gegen Überschreiben schützen)

5 Nichtfunktionale Anforderungen

Die folgenden nicht-funktionalen Eigenschaften wurden besonders hervorgehoben:

- Einfache Bedienbarkeit
- Schneller Zugriff auf Dokumente
- Integration in normale Arbeitsabläufe
- Übersichtlichkeit der Anwendung
- Verfügbarkeit (jederzeit zugreifbar)
- Aktualität (und damit einhergehend bis zu einem gewissen Grad die Verbindlichkeit)
- Flexibilität bei der Anwendung weiterführender Funktionen (Anpassungsmöglichkeiten)
- Einfache Administration, um die Nachhaltigkeit der Lösungen zu unterstützen.
- Vertraulichkeit, soweit nötig und gewünscht.
- Geringe Kosten und Aufwand für Anschaffung, Einrichtung und Administration
- Einhaltung offener Standards und Interoperabilität mit anderen Lösungen
- Datensicherheit (Security, Safety, Backups)
- Nachhaltigkeit
- Zugreifbarkeit
- Verbindlichkeit, d.h. belastbar referenzierbar mit von allen Partnern anerkannten Inhalten.
- Klar definierte Workflows, wie von allen anerkannte Inhalte erzeugt werden
- Berücksichtigung des Datenschutzes

6 Risikominimierung

Als Risiken für den Einstieg in neue Kommunikationsmöglichkeiten werden gesehen:

- Zu große Komplexität oder Unzuverlässigkeit von Werkzeugen, die die erste Anwendung erschweren.
- Geringe Aktualität der Daten und Dokumente durch geringe Nutzung
- Systemwechsel in der Einstiegsphase
- Entstehung von „Parallelwelten“, wie sie im Moment für vIEM/WISENT bestehen (vIEM Homepage, vIEM-TeamSite, WISENT-BSCW, WISENT-Homepage/Wiki).
- Zu hohe Sicherheitshürden
- Geringe Vertraulichkeit
- Objektnamen, z.B. für Dateien, die nicht selbsterklärend sind.

- Fehlen eines Administrators, der ab und zu aufräumt und auf die Einhaltung der definierten Regeln achtet
- Die Verbindlichkeit der Inhalte im Projekt ist nicht allgemein anerkannt.

Alle Partner drängen darauf, möglichst schnell eine Lösung zu finden, um den genannten Punkten vorzubeugen. Es sollte für alle WISENT-Partner eine Pflicht zur regelmäßigen Nutzung (z.B. Berichterzeugung mit Versionskontrollsystemen statt via Mail) und Weiterentwicklung der gemeinsamen Arbeitsumgebung geben (z.B. Aufbau eines Glossars, Entwicklung der Homepage).

7 Skizze des Lebenszyklus und der Gesamtarchitektur

Für den Betrieb und die Nachhaltigkeit sind unter anderem der Standort und die Administration von Bedeutung. Die Regelung von Zugriffsberechtigungen wird ebenfalls als zentraler Punkt angesehen.

7.1 Standort und Zuständigkeit für die Administration der CSCW-Software

Für die Nachhaltigkeit ist es wichtig, dass die CSCW-Plattform für vIEM nach dem Ablauf der WISENT-Projektlaufzeit bestehen bleibt. Andererseits ist es im Rahmen von WISENT günstig, wenn die verwendeten Werkzeuge bei OFFIS installiert und administriert werden, weil dort sowohl Expertise als auch Fördermittel vorhanden sind. Insbesondere sollte das Ausprobieren von Alternativen und Softwareerweiterungen im Rahmen von WISENT nicht durch interne administrative Regelungen oder Mittelknappheit der einzelnen beteiligten Forschungsinstitute behindert werden.

OFFIS schlägt vor, die Einrichtung und Pflege der CSCW-Werkzeuge während der Projektlaufzeit von WISENT zu übernehmen und zum Ende dieser Laufzeit für eine Übergabe an ein vIEM-Mitglied zu sorgen.

7.2 Regelung der Zugangsberechtigungen, Vertraulichkeit der Inhalte

Die CSCW-Software soll zum Einen im vIEM zur Verbreitung von forschungsbezogenen Informationen genutzt werden. Zum Anderen dient sie als Ablage für projektinterne (ggf. vertrauliche) Dokumente des WISENT-Konsortiums. Diese Tatsache sollte bei der Einrichtung der Zugangs- und Administrationsrechte für vIEM- und Projektmitglieder berücksichtigt werden.

8 Auswahl von CSCW-Werkzeugen

Durch OFFIS wurden den Projektpartnern unterschiedliche Möglichkeiten des CSCW-Einsatzes präsentiert. Auf einem gemeinsamen Treffen wurde daraufhin die Nutzung einzelner Werkzeuge diskutiert.

8.1 CVS, Subversion etc.

Versionsverwaltungswerkzeuge wurden bisher nicht institutsübergreifend eingesetzt und an der Uni OL nur beschränkt genutzt. Da jedoch gemeinsame Entwicklungsvorhaben sowohl vorstellbar als auch erwünscht sind, besteht in diesem Kontext Bedarf an solcher Software.

8.2 BSCW und TeamSite

Die beiden eingerichteten Softwarepakete BSCW und TeamSite sollten demnächst als Dokumentablage für WISENT bzw. vIEM parallel genutzt werden. Dies ist zwar mit

zusätzlichem Aufwand verbunden, bietet aber die Möglichkeit, informierte Entscheidungen bzgl. einer nachhaltigen Auswahl zu treffen.

8.3 Wiki

Ein Wiki-System wurde von allen Projektpartnern als geeignetes System zur Umsetzung der Ziele im vIEM angesehen. Z.B. wäre die Verwaltung eines Online-Glossars für energiemeteorologische Begriffe möglich, das von allen Projektpartnern angelegt und gepflegt werden könnte. Ein Wiki-System könnte das in der Community vorhandene bzw. entstehende Wissen dokumentieren bzw. als Plattform zur Erarbeitung neuen Wissens dienen.

Folgende Funktionen werden als nützlich angesehen: - leichte verteilte Bearbeitung - jeder kann beitragen - jeder kann korrigieren/ergänzen - überall (online) verfügbar - auch ein Download für den Offline-Gebrauch bei einigen Implementationen möglich (z.B. Wikipedia)

Ein Wiki ermöglicht eine unkomplizierte Bearbeitung von miteinander verknüpften Dokumentausschnitten. Für WISENT wird demnächst bei OFFIS ein Wiki installiert und den vIEM-Mitgliedern zur Verfügung gestellt. Es wurden folgende Ideen zu den ersten Einsatzmöglichkeiten eines Wiki geäußert:

1. Erstellung eines gemeinsamen vIEM-Glossars
2. Erstellung des vIEM-Forschungsplans
3. Erstellung einer Liste von vorhandenen und gemeinsam im vIEM nutzbaren Softwarepaketen und Nutzungsbedingungen

Das Wiki sollte auch unter dem Gesichtspunkt evaluiert werden, inwiefern es eine zentrale Dokumentablage ergänzen oder ersetzen und die Langlebigkeit bzw. Nachvollziehbarkeit der Forschungsergebnisse verbessern kann, z. B. als Ausgangspunkt zur Vorbereitung von Präsentationen.

Als Kriterien zur Auswahl einer konkreten Wiki-Software wurden genannt:

1. Möglichkeit der Offline-Nutzung, z. B. auf Reisen
2. Möglichkeit der Eingabe mathematischer Formeln

8.4 Content-Management-Systeme (CMS)

Sowohl für die WISENT-Webseite als auch für die vIEM-Webseite ist die Nutzung eines CMS erwünscht, um eine dezentrale Pflege der Webseiteninhalte durch Projektmitglieder zu ermöglichen. Als Hauptkandidat wurde von OFFIS Plone genannt, da sich diese Software bereits im Einsatz in der Software-Engineering-Abteilung der Uni OL bewährt hat.

8.5 Internet-Telefonkonferenz / Remote Desktop

Es besteht der Wunsch nach der Einführung einer Telekonferenzlösung. Zum einen sollte sie die Anzahl von erforderlichen bilateralen (Telefon)Absprachen senken. Zum anderen sollte sie nicht nur Sprachübertragung, sondern auch eine gleichzeitige Übermittlung und ggf. eine gemeinsame Bearbeitung einer Präsentation erlauben. Als erster Kandidat wurde Microsoft NetMeeting erwähnt. Bei dieser Software ist die Firewall-Problematik (DLR-DFD) zu beachten.

9 Einsatz von CSCW-Werkzeugen

9.1 Versionsverwaltungssysteme CVS und Subversion

Es erfolgte die Einführung von Entwicklungswerkzeugen für Teams bei den Projektpartnern durch OFFIS. Dies betrifft insbesondere die Einführung von Versionsverwaltungssystemen wie CVS und Subversion, die eine verteilte Zusammenarbeit an Quelltexten unterstützen (siehe auch [Ploski u.a. 2007]). Inzwischen werden diese Systeme bei allen Projektpartnern eingesetzt. Sie bilden eine notwendige Voraussetzung, um in einer verteilten Community effektiv Grid-Anwendungen entwickeln zu können.

9.2 Wiki-System TikiWiki

Das Wiki-System TikiWiki wurde auf einem OFFIS-Server installiert und an die Bedürfnisse der Community angepasst. Dies beinhaltet unter anderem die Einrichtung einer hierarchischen Dateiablage, um die zuvor eingesetzten und wenig genutzten Lösungen BSCW und TeamSite ablösen zu können, und Anpassungen des Berechtigungskonzepts. Über das TikiWiki wurde ein Glossar eingerichtet (siehe Abbildung 1), das für die Community relevante Einträge wie "Strahlungstransport", "Strahlungssimulation", "Atmosphäre, Wolken und Boden" etc. beinhaltet. Das TikiWiki bildet bereits jetzt eine intensiv genutzte Plattform für die Zusammenarbeit im WISENT-Projekt. Neben dem Projekt WISENT und dem vIEM nutzt auch die AG EnMet (Energiemeteorologie) das TikiWiki und liefert zusätzliche Anforderungen aus der Community. Die AG EnMet beabsichtigt die Nutzung des TikiWiki zum Aufbau einer verteilten Wissensdatenbasis, die später auch dem vIEM zur Verfügung stehen soll. Das TikiWiki wurde so erweitert, dass es mandantenfähig ist, also eine Server-Installation, die gleichzeitig mehrere Teil-Communitys mit teilweise überlappenden Datenbeständen bedient (hier WISENT, vIEM und EnMEt). Das TikiWiki-System wird ebenfalls für die Internet-Präsenz des WISENT-Projektes eingesetzt und dient somit auch als Content-Management-System.

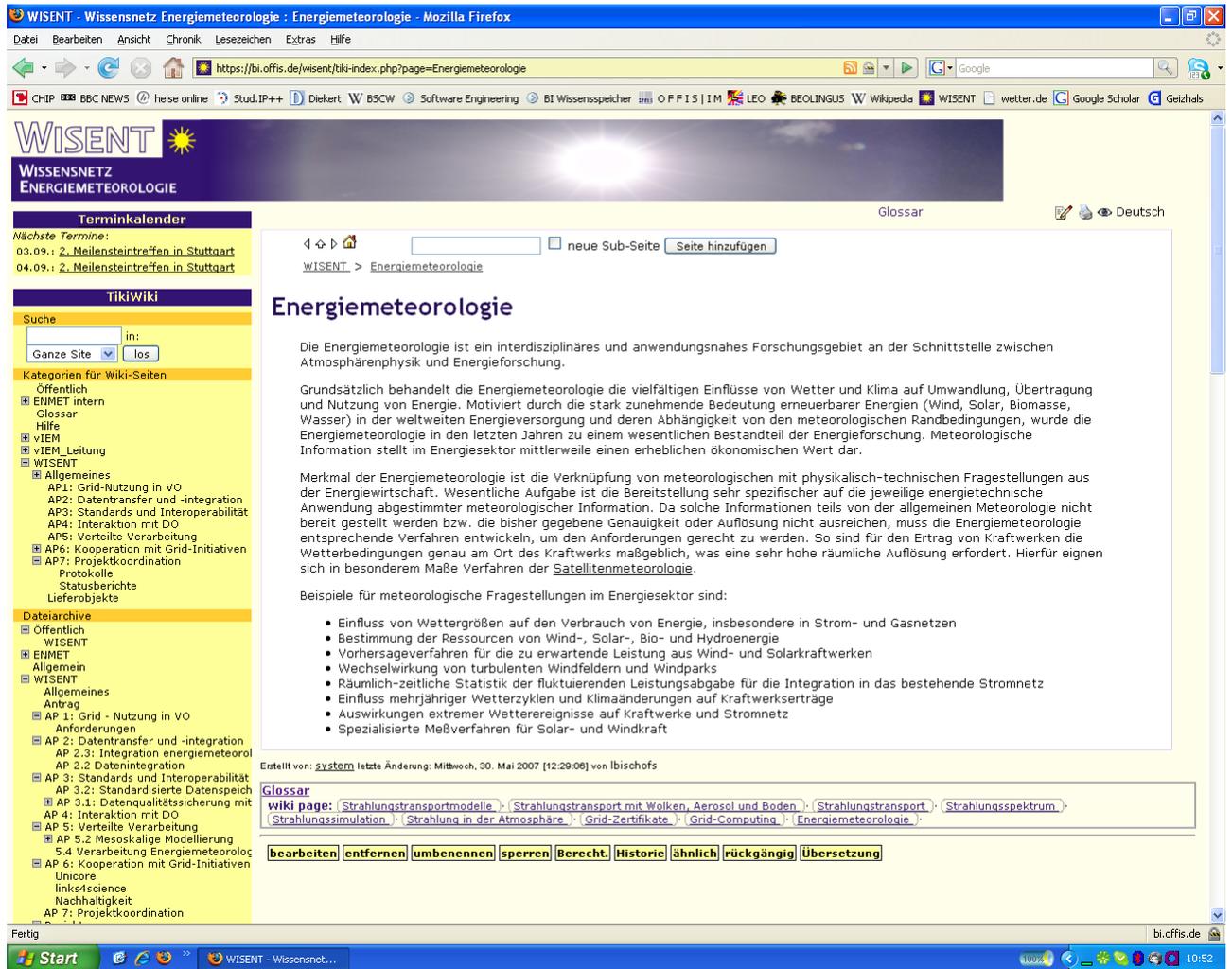


Abbildung 1: TikiWiki-System des Wissensnetzes Energiemeteorologie

9.3 Videokonferenzsysteme

Die Kommunikation zwischen den Projektpartnern wurde durch die Videokonferenzanwendungen Skype und Festoon unterstützt. Inzwischen ist die Verwendung von Skype im DLR jedoch aufgrund von Sicherheitsbedenken nicht mehr zugelassen. Der DFN-Service zu Adobe Connect stellte sich als geeignete Alternative heraus. VNC-Werkzeuge (Virtual Network Computing) wurden als Unterstützung bei Telefongesprächen zu technischen Themen (bspw. Eclipse-Nutzung) erfolgreich erprobt. Sie ermöglichen die gleichzeitige und gemeinsame Bearbeitung von Dokumenten und können auch für Präsentationen genutzt werden. Ein Tool zur Erstellung von Schulungs-/Diskussionsunterlagen in Form von Screencasts (vnc2swf) wurde bei OFFIS erfolgreich getestet.

10 Weiteres Vorgehen

Die in AP 1.1 untersuchten und bereits zum Teil eingesetzten CSCW-Werkzeuge werden in AP 1.2 genutzt und entsprechend der Community-Anforderungen angepasst und weiterentwickelt. Es ist beabsichtigt, alle eigenen TikiWiki-Erweiterungen in das offizielle TikiWiki-Projekt einfließen zu lassen.

Referenzen

[Ploski u.a. 2007] Ploski, Jan; Hasselbring, Wilhelm; Rehwinkel, Jochen; Schwierz, Stefan: Introducing Version Control to Database-Centric Applications in a Small Enterprise. In: Software Engineering Challenges in Small Software Organizations. IEEE Software, 25(1): S. 38-44, Januar/Februar, 2007.



Grid-Anforderungen der Community Energiemeteorologie

Arbeitspaket 1.1

Guido Scherp
OFFIS – Institut für Informatik
Escherweg2
26121 Oldenburg
Internet: <http://www.offis.de>

1 Einleitung

Ein Ziel des Arbeitspakets 1 im Projekt WISENT ist der Aufbau einer Grid-Infrastruktur, die im Laufe des Projektes genutzt und auch erweitert wird. Um diesen Aufbau besser planen zu können, wurde im Rahmen von Arbeitspaket 1.1 ("Gridbasierte Infrastruktur") zunächst ein im OFFIS erstellter Fragebogen von den übrigen Projektpartnern (Universität Oldenburg, meteocontrol, DLR-DFD, DLR-IPA, DLR-TT) beantwortet, mit dem die Anforderungen an diese Grid-Infrastruktur erfasst wurden.

In diesem Dokument sind die Ergebnisse dieser Befragung dargestellt. Im Folgenden wird dieser Fragebogen in Abschnitt 2 zunächst selbst vorgestellt, worauf in Abschnitt 3 die Auswertung der Ergebnisse folgt. Bei der Auswertung werden insbesondere die größten Gemeinsamkeiten aller Projektpartner vorgestellt und nur bei Bedarf wird gezielt auf einzelne Anforderungen eines Projektpartners eingegangen, wenn diese für den Projektpartner eine besonders hohe Relevanz haben.

2 Struktur des Fragebogens

Der von den Projektpartnern beantwortete Fragebogen hat die im Folgenden vorgestellte Struktur. Als Hilfestellung wurden für jeden Punkt Fragen vorgegeben, an denen sich die Projektpartner inhaltlich orientieren konnten.

2.1 Ausgangssituation und Zielsetzung

- Wie ist der aktuelle Stand bzgl. des Einsatzes von Grid-Technologien?
- Falls in diesem Bereich noch nichts eingesetzt wird, stellt sich die Frage, welche anderen Techniken derzeit verwendet werden, um verteilte Rechen- oder Speicherressourcen zu nutzen.
- Gibt es z.B. bestimmte Programmbibliotheken für die Parallelisierung von Simulationsdurchläufen oder ähnliches?
- Welche Werkzeuge werden für was eingesetzt? Wo liegen die Probleme? Was könnte verbessert werden? Was soll sich mit der Einführung von Grid-Technologien ändern? Was sind die Erwartungen?

2.2 Funktionale Anforderungen

- Welche Funktionen sollten von der Grid-Infrastruktur unterstützt werden? Nutzung verteilter, ungenutzter Rechenressourcen oder ungenutzten Festplattenspeichers?
- Welche Funktionen sind auf jeden Fall erforderlich und welche sind optional (nice-to-have)? Eine Prioritätenliste wäre hilfreich.
- Randbedingungen wie technische Probleme mit freien Ports sollten hier ebenfalls mit aufgenommen werden.

2.3 Nicht-Funktionale Anforderungen

- Nicht-funktionale Anforderungen sind beispielsweise eine leichte Bedienbarkeit, 100%ige Zuverlässigkeit, hohe Performanz beim Dateizugriff oder ähnliches. Weitere Anforderungen könnten sein: geringer Administrationsaufwand, gute Änderbarkeit, hohe Skalierbarkeit, gute Ergonomie, hohe Sicherheit usw.

- Die Anforderungen sollten genauer erläutert werden. Natürlich möchte man immer Systeme mit den besten nicht-funktionalen Anforderungen haben. Da dies in der Realität nur schwer zu erreichen ist, sollte hier eine Priorisierung stattfinden.

2.4 Risikominimierung

- Was muss getan werden, damit die Grid-Infrastruktur auch eingesetzt wird?
- Worin liegen die größten Gefahren, dass die Infrastruktur abgelehnt wird?
- Wie kann man dieses Risiko minimieren?

2.5 Skizze des Lebenszyklus und der Gesamtarchitektur

- Wie können die neuen Technologien eingeführt werden?
- In welcher Reihenfolge sollte dies geschehen (Bezug nehmen auf die in 2.2/2.3 erarbeitete Prioritätenliste)? Schrittweise oder alles auf ein mal?
- Wie fügen sich die Technologien in die bestehende Systemlandschaft ein?
- Wie kann die Nachhaltigkeit gewährleistet werden (auch nach Beendigung des WISENT-Projekts)?

3 Auswertung des Fragebogens

Die Auswertung des Fragebogens zur Erhebung der Anforderungen an eine Grid-Infrastruktur orientiert sich an der Struktur des Fragebogens selbst. Es wird darauf verzichtet, die einzelnen Antworten der Projektpartner detailliert darzustellen. Vielmehr wurde ein gemeinsamer Konsens herausgearbeitet und nur bei Bedarf werden spezielle Aspekte hoher Relevanz für einen einzelnen Projektpartner dargestellt.

3.1 Ausgangssituation und Zielsetzung

Bei allen befragten Projektpartnern wurden bisher keine Grid-Technologien genutzt. Aber es werden derzeit andere Technologien zur Nutzung von bestehenden Rechen- und Speicherressourcen eingesetzt.

Es werden bereits sowohl interne als auch externe Rechen-Ressourcen genutzt, um Berechnungen durchführen zu lassen, die auch Parallelisierungstechnologien wie MPI und OpenMP einsetzen. Interne Ressourcen sind dabei Arbeitsplatzrechner sowie (operationelle) Server (zum Teil Multiprozessorsysteme). DLR-IPA besitzt zudem einen eigenen Cluster mit 16 Knoten. Externe Ressourcen sind verschiedene Hochleistungsrechenzentren. Berechnungen auf internen Ressourcen werden über einen Shell-Zugang manuell gestartet. DLR-IPA betreibt auf deren Cluster eine Job-Verteilung über pvm/ppmake. Externe Hochleistungsrechenzentren werden über einen Shell-Zugang und das dort vorhandene Batch-System (z.B. PBS) genutzt.

Die Parallelität der verwendeten Anwendungen wird nicht selbst implementiert, sondern es werden entsprechende anwendungsspezifische Bibliotheken genutzt, die bereits OpenMP oder MPI verwenden. Zudem verwendet die Universität Oldenburg den kommerziellen Compiler PGC (Portland Group Compiler), der die Parallelisierung mit OpenMP zur Beschleunigung von Anwendungen auf einem Multiprozessorsystem unterstützt. Dieser Compiler hat die Option, die Schleifen eines „seriellen“ Programms zu analysieren, um dieses Programm automatisch für ein Multiprozessorsystem zu „parallelisieren“.

Bei allen Projektpartnern sind kleinere Festplatten-Speichersysteme oder auch SAN-Systeme vorhanden, die entweder per NFS eingebunden werden, oder die Prozessierung

wird direkt auf dem System ausgeführt. Das DLR-DFD administriert zusätzlich das Band-Archiv DIMS (Data Information and Management System). Der Datentransfer zwischen diesen einzelnen Systemen bzw. den Projektpartnern erfolgt mit rsync, scp, FTP, HTTP, SOAP oder E-Mail. Die verwendeten Datenformate sind z.T. proprietär und orientieren sich an keinem Standard.

Die Nutzung von Grid-Technologien soll in erster Linie Einheitlichkeit schaffen bzgl. des Zugangs zu Rechen- und Speicherressourcen sowie den dazwischen ausgeführten Datentransfers. Es sollen einheitliche und standardisierte Schnittstellen und Protokolle eingeführt und verwendet werden. Der Mangel an dieser Einheitlichkeit wird als Hauptgrund angesehen, warum eine effiziente Nutzung verteilter Ressourcen derzeit nicht machbar ist. Des Weiteren ist eine gleichmäßige Auslastung der Rechen-Ressourcen durch manuelle Verteilung oder mit pvm/ppmake (bei DLR-IPA) nicht zufriedenstellend umsetzbar. Auch hier sollen entsprechende Grid-Technologien eingeführt werden.

Ein weiteres Ziel, das mit der Einführung von Grid-Technologien verbunden wird, ist die Parallelisierung von bestehenden Anwendungen und dessen Ausführung auf den in der aufzubauenden Grid-Infrastruktur vorhandenen Ressourcen. Eine Nutzung „öffentlicher“ Grids wird nicht ausgeschlossen, um bei Bedarf kurzfristig Zugang zu großen Rechenkapazitäten zu bekommen. Bezüglich der Speicher-Ressourcen sollte die Einrichtung verteilter Datenbestände (z.B. Wetterdaten) in Betracht gezogen werden.

Für DLR-DFD steht die Nutzung von Grid-Technologien, die einem Data-Grid zuzuordnen sind, im Vordergrund. Für die Universität Oldenburg, meteocontrol, DLR-IPA und DLR-TT sind auch Grid-Technologien eines Data-Grid relevant, im Vordergrund stehen aber Grid-Technologien, die dem Bereich des Computing-Grid zuzuordnen sind.

Für meteocontrol als Wirtschaftsunternehmen spielen zudem Aspekte wie Ausfallsicherheit, eine mögliche Verwendung von Grid-Technologien für Datentransfers mit Kunden sowie die Integration in das bestehende Überwachungssystem (Nagios) eine wichtige Rolle. Im DLR-Institut für Simulations- und Softwaretechnik (DLR-SISTEC) gibt es bereits Erfahrungen mit Grid-Technologien, die für den Aufbau der Grid-Infrastruktur im Projekt genutzt werden sollten.

3.2 Funktionale Anforderungen

Die von Projektpartnern gemachten Angaben zu den funktionalen Anforderungen werden im Folgenden den Prioritätstufen „Hoch“ sowie „Mittel und Niedrig“ zugeordnet.

Hohe Priorität

- Nutzung von „Idle“-Zyklen verteilter Rechen-Ressourcen
- Verwendung von „single sign-on“ (SSO) zur Authentifizierung
- Nutzung von GridFTP für Datentransfers
- Monitoring und Accounting. Integration des Monitoring in ein bestehendes Überwachungssystem (z.B. Nagios bei meteocontrol)
- Suche nach Rechen- und Speicher-Ressourcen (freie CPUs, freier Speicher). Suche nach bestimmten Daten auf den Speicher-Ressourcen
- Nutzung von Grid-Technologien zur Einrichtung einer Plattform zur Auslieferung von Datenprodukten (DLR)
- Verschlüsselung von Datentransfers kommerzieller Daten

Mittlere und niedrige Priorität

- Temporäre Ausnutzung von ungenutztem, lokal vorhandenem Festplattenspeicher (Universität Oldenburg)
- Integration bestehender Web Services in die Grid-Infrastruktur oder Erweiterung von bestehenden Web Services als Grid Services (DLR)
- Nutzung großer externer Speicher-Ressourcen

3.3 Nicht-Funktionale Anforderungen

Die von Projektpartnern gemachten Angaben zu den nicht-funktionalen Anforderungen werden im Folgenden den Prioritätsstufen „Hoch“ sowie „Mittel und Niedrig“ zugeordnet.

Hoch

- Effizienz: Gleichmäßige Nutzung und Auslastung der verteilten Rechen- und Speicher-Ressourcen
- Fairness: Alle Benutzer sind gleichberechtigt bzgl. der Belegung von CPU-Zeit und Festplattenspeicher (Job-Priorisierung, Quotas). Benutzer von Arbeitsplatzrechnern werden in der Ausführung ihrer Arbeit nicht durch im Hintergrund laufende Jobs beeinträchtigt.
- Ausfallsicherheit und Zuverlässigkeit; Robustheit gegenüber Hardwaredefekten sowie Korrektheit der Ergebnisse
- Einfacher und standardisierter Zugang zu und Benutzung von Rechen- und Speicher-Ressourcen; Authentifizierung und Autorisierung bzw. Ausführung von Berechnungen und Datentransfers
- Sicherheit; Berücksichtigung von Firewall-Einstellungen und Sicherheitsrichtlinien beim DLR (DLR-DFD, DLR-IPA, DLR-TT)
- Klärung rechtlicher Aspekte wie Softwarelizenzen und Nutzungsrechte von Daten
- Einheitliche Arbeitsumgebung auf allen Ressourcen inklusive der benötigten Bibliotheken
- Nutzung von Metadaten

Mittel und Niedrig

- Grid-Software erzeugt geringen Overhead
- Einfache Administration der Grid-Ressourcen
- Datendurchsatz von Datentransfers

3.4 Risikominimierung

Zur Risikominimierung bei der Einrichtung einer Grid-Infrastruktur wurden folgende kritische Punkte genannt.

- Eine einfache und intuitive Benutzung ist von essentieller Bedeutung. Die zu überwindenden Hürden zur Nutzung der Grid-Infrastruktur müssen so gering wie möglich gehalten werden bzw. die Benutzer an schwierigen Punkten entsprechend unterstützt werden.

- Es muss eine deutliche Verbesserung gegenüber den bisherigen angewendeten Methoden erkennbar sein.
- Es muss Vertrauen in die Grid-Infrastruktur geschaffen werden, indem die Stabilität und Sicherheit gewährleistet wird. Der Benutzer darf sich bei der Zuteilung von Rechen- oder Daten-Ressourcen nicht ungerecht behandelt fühlen und nicht zu lange warten.
- Zunächst sollten einfache Anwendungsfälle realisiert werden, um frühzeitig die generelle Funktionsfähigkeit der Grid-Infrastruktur zu zeigen.
- Es muss eine gute Dokumentation über die Grid-Infrastruktur und deren Benutzung erstellt werden.
- Ein oder mehrere Administratoren müssen bei den Projektpartnern die Wartung der Grid-Infrastruktur übernehmen und Fehlerquellen schnell und zuverlässig beheben.

3.5 Skizze des Lebenszyklus und der Gesamtarchitektur

Im Wesentlichen wird ein schrittweises Vorgehen vorgeschlagen, um nach und nach die Grid-Infrastruktur aufzubauen und die Verwendung von Grid-Technologien einzuführen. Dabei sind zwei große Schritte festzustellen, die sich bei jedem Projektpartner in mehrere Unterschritte, die hier nicht im einzelnen aufgeführt werden, aufteilen.

Der erste Schritt ist, dass jeder Projektpartner bei sich eine Grid-Testumgebung aufbaut und die notwendigen Unterschritte dazu durchführt. Die bei den Projektpartnern vorhandenen Rechen-Ressourcen werden entsprechend intern vernetzt, um schon die ersten Erfahrungen mit der Nutzung von Grid-Technologien zu sammeln. Bei DLR-IPA wären das beispielsweise der interne Cluster sowie mehrere Arbeitsplatzrechner. Parallel dazu wird schon die Schnittstelle zur externen Integration geschaffen. Projektpartner wie DLR-DFD, die in erster Linie an Datentransfers interessiert sind, konzentrieren sich auf die Einrichtung entsprechender Grid-Technologien und führen Transfertests mit einem Projektpartner durch. Wichtig ist, dass frühzeitig erste, wenn auch einfache, Tests durchgeführt werden können, um Erfahrungen im Umgang mit den Grid-Technologien zu sammeln.

Im zweiten Schritt folgt die Vernetzung der einzelnen Projektpartner. Dies beinhaltet die Einrichtung einer Public-Key-Infrastruktur (PKI), um den Zugang der Benutzer zu Grid-Ressourcen mittels Authentifizierung und Autorisierung regeln zu können. Projektpartner mit Bedarf an Rechen-Ressourcen nutzen die Ressourcen der Projektpartner oder anderer externer Einrichtungen, die einen Zugang über Grid-Technologien ermöglichen. Die Testdatentransfers aus dem ersten Schritt werden durch operationelle Datentransfers ersetzt. Wichtig ist, dass ein sanfter Übergang von Nicht-Grid-Technologien zu Grid-Technologien gewährleistet ist, d.h. zunächst werden beide Lösungen parallel betrieben, um bei evtl. auftretenden Problemen mit der Grid-Lösung noch die alte, bereits bestehende Nicht-Grid-Lösung im operationellen Betrieb nutzen zu können. Auf dieser Ebene spielen zudem Monitoring sowie Billing und Accounting eine wesentliche Rolle.



CSCW-Erfahrungen der Community Energiemeteorologie

Arbeitspaket 1.2

Ludger Bischofs
OFFIS – Institut für Informatik
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de>

1 Einleitung

Die in Arbeitspaket 1.1 erhobenen Anforderungen der Community Energiemeteorologie an die Unterstützung der Zusammenarbeit und die Vernetzung von Wissen mit Hilfe von CSCW-Werkzeugen werden im Rahmen von Arbeitspaket 1.2 umgesetzt. Ziel ist es, die nachhaltige Festigung der jungen Community Energiemeteorologie durch den Einsatz von Werkzeugen zur Kollaboration zu stützen. Dieses Dokument beschreibt die bisher gesammelten Erfahrungen mit dem eingesetzten TikiWiki-System. Neben dem TikiWiki-System wurden auch Versionsverwaltungssysteme wie CVS und Subversion sowie Videokonferenzsysteme erfolgreich eingesetzt.

2 TikiWiki

Das TikiWiki-System wurde angepasst und erweitert, um die Anforderungen der Community besser abzudecken. Zu den wichtigsten Erweiterungen gehört die Mandanten-Fähigkeit, wodurch eine Nutzung des TikiWiki-Systems durch Teil-Communitys bzw. Teilgruppen ermöglicht wird. Derzeit wird das TikiWiki-System nicht nur von WISENT allein genutzt, sondern ebenfalls vom vIEM (Virtuelles Institut für Energiemeteorologie) und von der Arbeitsgruppe EnMet (Energiemeteorologie) der Universität Oldenburg. Eine weitere wesentliche Erweiterung stellt das hierarchische Dateiarchiv dar. Dazu kommen ein Benachrichtigungsdienst, der über inhaltliche Änderungen informiert sowie eine Kalenderfunktionalität. Die Ergänzungen am TikiWiki sind bereits teilweise in die offizielle Version des Open-Source-Systems eingeflossen, so dass auch andere Communitys davon profitieren können.

Das TikiWiki-System wird vor allem für die folgenden Aktivitäten genutzt:

- Projektmanagement (Berichtswesen)
- Glossar
- Beschreibung von Datentransfers
- Beschreibung von Datenformaten und Metadaten-Standards
- Beschreibung von Software-Architekturen
- Installationsanleitungen
- Dokumentation zur Ausführung von Jobs im Grid
- Erfahrungsberichte zu Grid-Middleware und CSCW-Werkzeugen

3 TikiWiki-Nutzung

Das TikiWiki-System wird bereits intensiv von der Community genutzt. Die anfängliche Sorge, dass sich *ein* Werkzeug für die kollaborative Arbeit nicht durchsetzen könnte, war unbegründet. Das TikiWiki-System stellt die zentrale Plattform dar, über die aktuell eine Zusammenarbeit in der Community auch über weite Entfernungen hinweg erfolgt. Es ist zu einer soliden Basis geworden, auf deren Grundlage schrittweise weitere innovative Erweiterungen zur Wissensvernetzung eingebracht werden können. Diese schrittweise Erweiterung reduziert voraussichtlich die Gefahr der mangelnden Akzeptanz aufgrund zu großer Umstellungen.

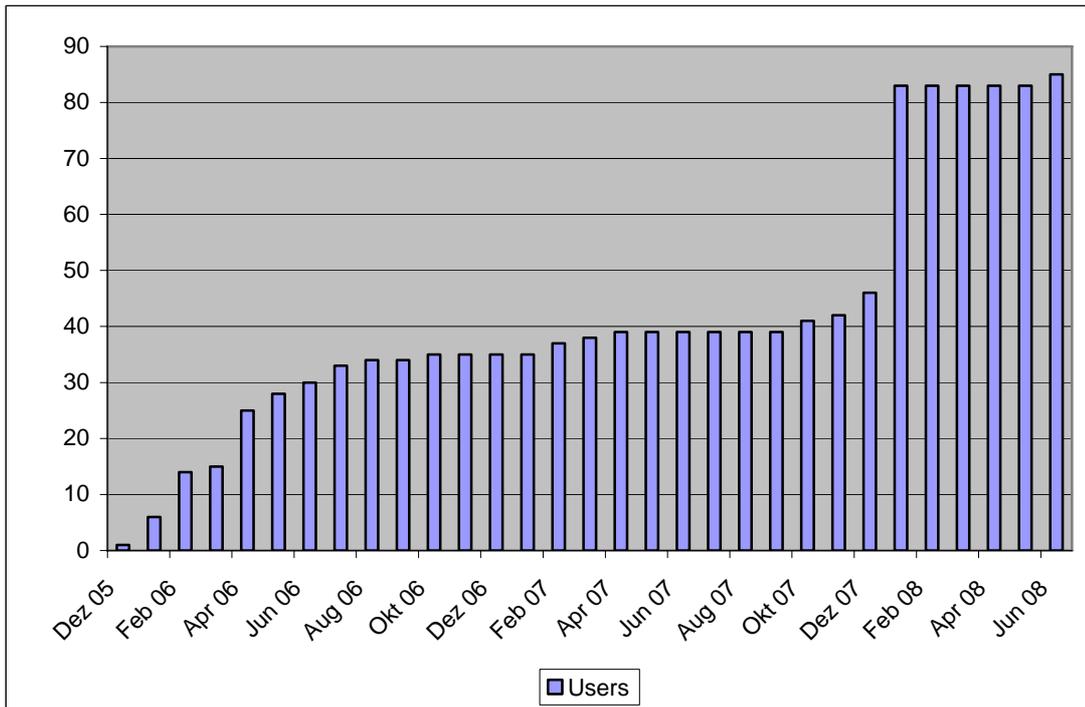


Abbildung 1: Anzahl der Benutzer des TikiWiki-Systems der Community Energiemeteorologie

Abbildung 1 zeigt die Anzahl der am TikiWiki-System angemeldeten Benutzer im Laufe des Projektes. Seit der Installation und Bereitstellung des Systems im Dezember 2005 hat sich die Nutzerzahl bis auf über 80 erhöht. Der starke Anstieg im Januar 2008 ist auf das Hinzukommen von Benutzern aus dem Zentrum für Energieforschung ForWind (<http://www.forwind.de/>) zurückzuführen.

Abbildung 2 zeigt die Aufteilung der Nutzer in die fünf derzeit eingerichteten Gruppen EnMet, ForWind, vIEM, vIEM_Leitung und WISENT.

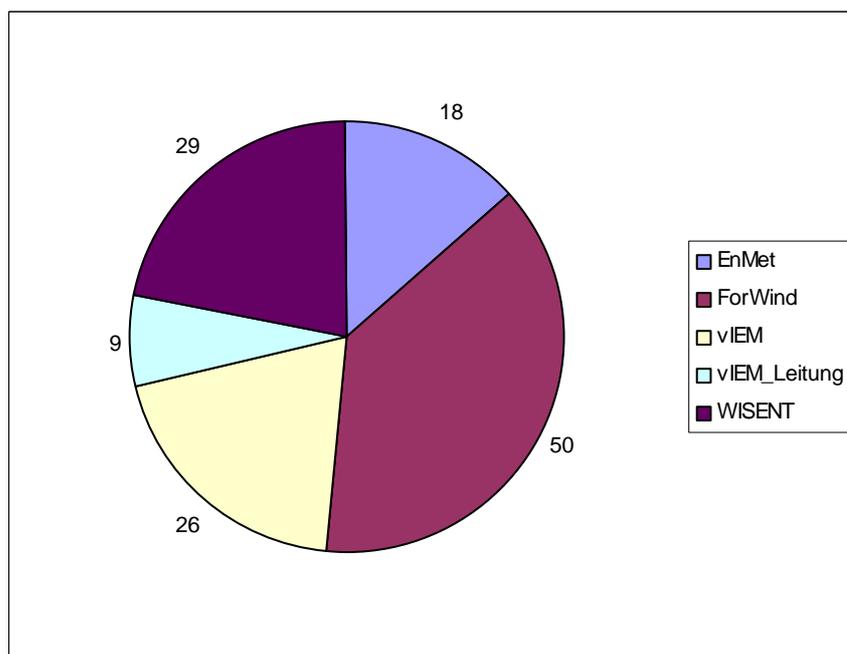


Abbildung 2: Anzahl der Benutzer in den Gruppen

An Abbildung 3 ist zu erkennen, dass die Anzahl der Wiki-Seiten im TikiWiki seit der Einführung des Systems stark angestiegen ist und derzeit bei über 600 liegt. An der Häufigkeit der Änderungen ist zu erkennen, dass Seiten nicht nur hinzugefügt werden, sondern das auch mit bzw. an ihnen gearbeitet wird. Auch die Erweiterung der hierarchischen Dateiablage wird intensiv genutzt. Es wurden bereits über 650 Dateien in über 100 Verzeichnissen abgelegt (siehe Abbildung 4).

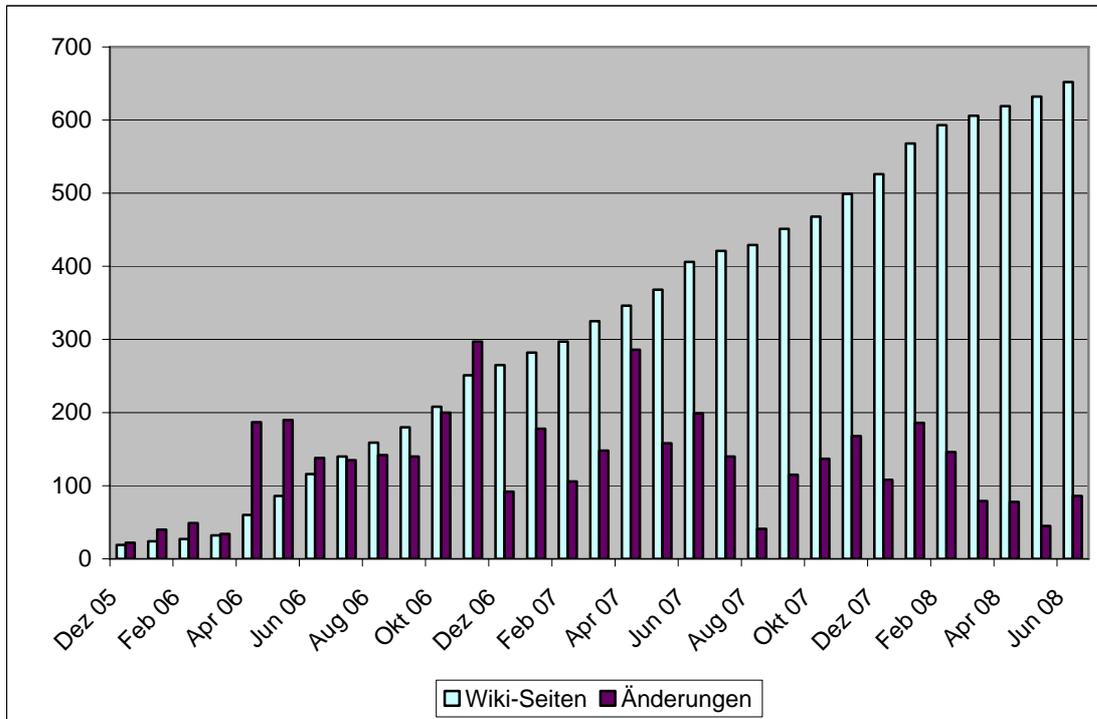


Abbildung 3: Anzahl neuer Wiki-Seiten und Änderungen an Wiki-Seiten

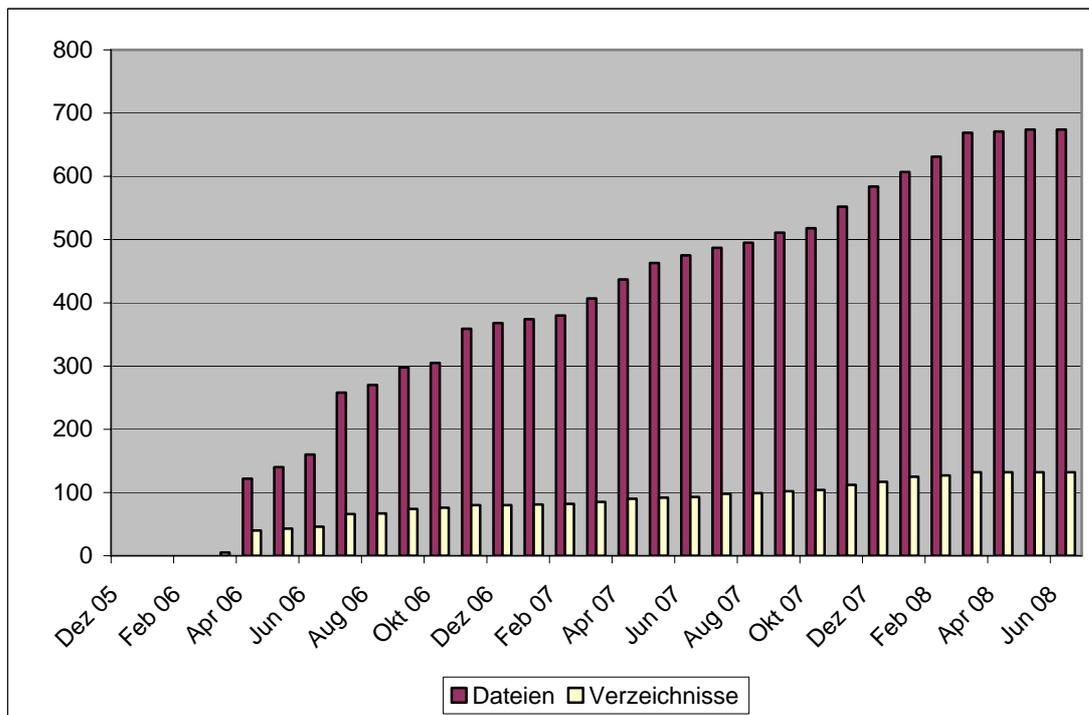


Abbildung 4: Anzahl Dateien und Verzeichnisse

4 Ausblick

Das TikiWiki-System wird derzeit von OFFIS gehostet und bis zum Projektende an die Universität Oldenburg übergehen. Das TikiWiki-System bleibt der Community Energiemeteorologie damit erhalten. Es bleibt noch abzuwarten, ob sich die Community Energiemeteorologie in eine größere Community Energie integrieren wird und ob das TikiWiki-System auch in der Energie-Community zum Einsatz kommen wird, um die Wissensvernetzung in einer noch größeren Community zu unterstützen.



Erfahrungen mit Grid-Technologien im Projekt WISENT

Arbeitspaket 1.2

Sönke Brummerloh
Luca Bugliaro
Gerhard Gesell
Johannes Hurka
Sina Lohmann
Jan Ploski
Guido Scherp

Sönke Brummerloh, Jan Ploski, Guido Scherp
OFFIS – Institut für Informatik
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de>

Johannes Hurka
AG Energiemeteorologie - Energie- und Halbleiterforschung Institut für Physik Universität
Oldenburg
26111 Oldenburg

Internet: <http://www.energiemeteorologie.de>

Gerhard Gesell
Deutsches Zentrum für Luft- und Raumfahrt DLR, Deutsches
Fernerkundungsdatenzentrum DFD
Münchner Straße 20
82234 Weßling

Internet: <http://www.caf.dlr.de>

Sina Lohmann, Luca Bugliaro
DLR Oberpfaffenhofen
Institut für Physik der Atmosphäre
82234 Weßling

Internet: <http://www.dlr.de/pa/>

1. Einleitung

Im Rahmen des Arbeitspakets 1.1 wurde eine Grid-basierte Infrastruktur aufgebaut, die im Laufe des Projekts in Arbeitspaket 1.2 genutzt und auch erweitert wurde. Aus Software-Sicht werden dazu Batch-Systeme und Grid-Middleware für den Zugang zu Rechenressourcen eingesetzt, sowie verschiedene Technologien zum Datenmanagement. Die Erfahrungen, die im Rahmen von WISENT bzgl. des Einsatzes von Grid-Technologien gemacht werden, sind in diesem Dokument festgehalten worden. Dabei stammen die Erfahrungen sowohl aus dem Arbeitspaket 1, welches im Projekt eine übergeordnete Rolle gespielt hat, als auch aus den einzelnen fachlichen Arbeitspaketen, in denen Grid-Technologien für ein konkretes Szenario eingesetzt wurde.

Dieses Dokument ist wie folgt aufgeteilt. In Abschnitt 2 werden die Erfahrungen mit den Batch-Systemen Condor, Sun Grid Engine 6 und TORQUE/Maui dargestellt. Darauf folgen in Abschnitt 3 die Erfahrungen mit den Grid-Middlewares Globus Toolkit 4, UNICORE 5 und gLite. Die Erfahrungen mit den Grid-Technologien GridFTP und dCache zum Datenmanagement befinden sich in Abschnitt 4. Am Ende wird das Dokument mit einem kurzen Ausblick in Abschnitt 5 abgeschlossen.

2. Batch-Systeme

Bereits vor der Entstehung des Konzepts eines Computing-Grids wurde für die Verwaltung von lang dauernden, nicht-interaktiven und ggf. parallel ablaufenden Rechenaufträgen dedizierte Software entwickelt, die meistens in Rechenzentren und auf Supercomputern Einsatz fand. Diese Softwarekategorie, bekannt als Batch-Systeme, stellt auch heute einen wesentlichen Teil jeder Computing-Grid-Infrastruktur dar. Sie wird zur effizienten Lastverteilung, Ausführung und Überwachung von Anwendungen innerhalb der einzelnen zum Grid gehörenden Computing-Ressourcen verwendet, während die eigentliche Grid-Middleware sich um die Abwicklung entsprechender Aufgaben zwischen den Computing-Ressourcen konzentriert bzw. eine einheitliche Schnittstelle zu diesen nach außen anbietet.

Im Rahmen des Projekts WISENT wurde eine Auswahl von geeigneten Batch-Systemen für die bei den Projektpartnern vorliegenden Rechenressourcen unter Berücksichtigung ihrer besonderen Merkmale vorgenommen. Nach dieser Auswahl wurden die Batch-Systeme Condor und TORQUE produktiv eingeführt. Das Batch-System Sun Grid Engine 6 wurde zum Vergleich im Rahmen einer Diplomarbeit getestet.

Die Erfahrungen mit den Batch-Systemen Sun Grid Engine 6, Condor und TORQUE/Maui sind in den Abschnitten 2.1 bis 2.3 dargestellt. Auf Grund der umfassenderen Erfahrungen mit Condor und TORQUE/Maui im Produktivbetrieb auf dem Compute-Cluster im OFFIS, wird in Abschnitt 2.4 ein Vergleich zwischen diesen beiden Batch-Systemen vorgenommen. Abschließend wird in Abschnitt 2.5 umfangreich über die Erfahrungen mit Condor bei DLR PA berichtet.

2.1 Sun Grid Engine 6

Von der Sun Grid Engine (SGE) gibt es zum einen die kommerzielle Sun N1 Grid Engine 6 und zum anderen das Open-Source-Projekt unter der Sun Industry Standards Source Licence. Aufbauend auf der Open-Source-Version stehen auch aktualisierte Versionen der kommerziellen Version zur Verfügung. Die kommerzielle Version verfügt zusätzlich über einige Erweiterungen, die in der Open-Source-Version nicht verfügbar sind. Dazu gehören z. B. Programme zum Verwalten des Batch-Systems, die Möglichkeit, Computer mit

Microsoft Windows in das Grid zu integrieren und ARCo, eine Software, um das Grid zu überwachen und zu analysieren. Inzwischen ist auch die kommerzielle Version kostenlos verfügbar.

Im Rahmen einer Diplomarbeit wurde die kommerzielle Variante der SGE in der Version 6.0 Update 8 eingesetzt und auf ihre Leistungsfähigkeit hin untersucht.

Die Sun Grid Engine wurde ursprünglich von Gridware in Regensburg in Deutschland entwickelt und hatte den Namen CODINE (Computing for Distributed Network Environments). Die Entwicklung von CODINE begann im Jahre 1992. Nach dem Kauf von Gridware im Jahre 2000 durch Sun Microsystems, wurde CODINE in Sun Grid Engine umbenannt und der Quelltext veröffentlicht.

Die Sun Grid Engine bietet von sich aus keine explizite Möglichkeiten für die Zusammenarbeit mit anderen Batch-Systemen oder Grid-Middleware. Über einen kommerziellen Adapter von GridwiseTech soll es möglich sein, dass die SGE mit dem Globus Toolkit und anderen Grid-Middleware-Lösungen zusammenarbeitet.

Die SGE benötigt ein Verzeichnis, z.B. ein über das Network File System (NFS) eingebundenes Verzeichnis, auf das alle Computer zugreifen können. Hier werden die Konfigurationen der einzelnen Hosts und des Batch-Systems gespeichert. Zusätzlich benötigt jeder Computer einen eindeutigen Hostnamen. Dieser wird während der Installation zur Computer-Identifikation genutzt. Die direkte Angabe einer IP wird vom Installationsprogramm abgelehnt und kann daher nicht verwendet werden.

Die SGE kann auf mehrere Arten installiert werden:

- Manuell
- Automatisiert
- Mit zusätzlichen Sicherheitsfunktionen

Bei der manuellen Installation wird zuerst das Installations-Skript für den Master Node (dem zentralen Computer, der die SGE-Installation verwaltet) auf dem entsprechenden Rechner gestartet. Nachdem der Master-Node installiert ist, wird das Installations-Skript für die so genannten Execution Hosts (Worker Nodes) auf den einzelnen Rechnerknoten gestartet. Sobald die ersten Execution Hosts eingerichtet sind, kann die SGE Jobs auf diese verteilen.

Je mehr Computer eingerichtet werden sollen, desto aufwändiger wird die manuelle Installation. Daher gibt es die Möglichkeit, die SGE automatisiert über eine Konfigurationsdatei zu installieren. Die automatisierte Installation installiert automatisch die verschiedenen Hosts auf den Computern und richtet bei Bedarf auch eine Datenbank zur Speicherung von Daten aus dem operativen Betrieb ein.

In der Hilfedatei zur automatisierten Installation steht, dass die Installation mit erweiterten Sicherheitsfunktionen nicht durch die automatisierte Installation unterstützt wird. In den Konfigurationsdateien können jedoch die Parameter für die Installation mit erweiterten Sicherheitsfunktionen gesetzt werden. Versuche, über die automatisierte Installation, die erweiterten Sicherheitsfunktionen zu installieren, schlugen allerdings fehl.

Die SGE kann automatisch wieder deinstalliert werden. Ein Deinstallieren über das Installations-Skript ist bei der manuellen und der automatisierten Installation möglich.

Gewöhnlich werden die Konfigurationen der Execution Hosts im globalen NFS-Verzeichnis gespeichert. Alternativ können sie auch jeweils lokal verwaltet werden. Das globale Verzeichnis muss aber in jedem Fall für die Konfiguration des Batch-Systems verfügbar sein.

Positiv bei der SGE ist zunächst die Installationsanleitung. In dieser wird Schritt für Schritt durch die Installationsmöglichkeiten der SGE geführt. Des Weiteren ist die automatisierte Installation und Deinstallation sehr hilfreich, vor allem, wenn mehrere Konfigurationen automatisch nacheinander getestet werden sollen.

Für die Administration der SGE kann die Administratoranleitung viele vertiefende Informationen liefern. Was bei der Dokumentation allerdings teilweise fehlt, sind Erklärungen für Einstellungen und Parameter. Es wird erklärt, wie etwas eingestellt werden kann, jedoch fehlen Informationen über die Bedeutung der einzelnen Parameter. Trotz vieler gemeldeter Fehler wurde die offizielle Dokumentation seit 2005 nicht aktualisiert. Für spezielle Informationen und die Bedeutung der einzelnen Parameter ist daher das aktuelle Handbuch im CVS-Repository die beste Quelle.

Sehr positiv viel auf, dass durch die SGE automatisch die Prozesskerne der Rechenknoten im Compute-Cluster im OFFIS erkannt und dann das Batch-System entsprechend eingerichtet wurde. So kann pro Prozessorkern ein Job ausgeführt werden und nicht nur ein Job pro Rechenknoten. Diese Einstellungen können für jeden Execution Host bei Bedarf manuell geändert werden.

Für die Diplomarbeit wurden mit der SGE zwei Batch-Systeme eingerichtet und mit jeweils zwei Anwendungen getestet. Durch die automatisierte Installation war ein schnelles und unkompliziertes Installieren der Sun Grid Engine möglich. Bei den Tests stellte sich heraus, dass die Sun Grid Engine für die untersuchten Anwendungen nahezu optimal die Last verteilt. Allerdings kann es bei einem ungünstigen Scheduling-Intervall in Bezug zur Jobdauer zu deutlichen Verzögerungen bei der Gesamtlaufzeit kommen. Je länger die Jobs sind, desto weniger fallen diese Verzögerungen durch die Wahl des Scheduling-Profiles ins Gewicht. Bei einem Vergleich der verschiedenen Konfigurationen stellte sich heraus, dass die Wahl des Scheduling-Profiles den größten Einfluss auf die Gesamtlaufzeit hat. Hier war die Einstellung „max“ die beste Wahl. Dies liegt daran, dass das Scheduling-Intervall bei dieser Einstellung entsprechend kurz ist und nicht zu unnötigen Verzögerungen führt.

Im Rahmen der Diplomarbeit wurde gezeigt, dass die Sun Grid Engine ein Batch-System ist, das auch für den praktischen Einsatz in Desktop-Grids verwendet werden kann. Sie bietet eine ausführliche und gut strukturierte Dokumentation. Bei speziellen Problemen kann die aktive Mailingliste für Lösungshinweise genutzt werden. Durch die automatisierte Installation auf Basis von Konfigurationsdateien kann die Sun Grid Engine schnell und unkompliziert auf mehreren Computern installiert werden. Das Verwenden von Zertifikaten und verschlüsselten Verbindungen führt allerdings zu einem hohen Mehraufwand bei der Installation und Konfiguration, da diese Sicherheitsfunktionalität nicht mit der automatisierten Installation eingerichtet werden kann.

2.2 Condor

Condor ist ein Softwarepaket zur Realisierung des so genannten High-Throughput-Computing. Unter Nutzung vorhandener verteilter Rechenkapazitäten soll damit der Berechnungsdurchsatz (Anzahl von erfüllten, unabhängigen Aufträgen pro Zeiteinheit, z. B. pro Monat) in einer Organisation erhöht werden. Dieser Ansatz unterscheidet sich vom besser etablierten High-Performance-Computing. Im letzteren Fall wird die erforderliche Rechenkapazität innerhalb kürzester Zeit gefordert, und ihre effektive Nutzung erfordert Anpassungen in dem Anwendungscode. In vielen Anwendungskontexten, einschließlich dem Projekt WISENT, spielt jedoch HPC keine vorrangige Rolle.

Das Projekt Condor startete bereits 1988. Heute werden durch die Software alleine am Standort des Entwicklers (Computer Science Department, University of Wisconsin) mehrere hundert Rechner verwaltet. Weltweit existieren mehr als 1700 Installationen mit über 85000 Rechnern. Ursprünglich wurden für Condor lediglich Binaries angeboten, mittlerweile wurden aber auch der dazugehörige Quelltext veröffentlicht.

Der ursprüngliche Fokus des Projekts Condor lag auf der Lastverteilung in verteilten Systemen. Im Laufe der Zeit wurde das Thema etwas eingeschränkt und die Aufmerksamkeit dem „distributedly owned computing“ gewidmet. Das Augenmerk ist die Durchführung von Berechnungsaufträgen mit Hilfe von vorhandenen Desktop-Arbeitsplätzen unter strenger Berücksichtigung von Nutzungsrichtlinien, die von jedem Anwender autonom festgelegt werden können. Die Idee ist, die Berechnungen nur dann ablaufen zu lassen, wenn sie die interaktive Nutzung der Maschine nicht beeinträchtigen (nachts, während Kaffeepausen usw.). Um dies zu gewährleisten, verfügt Condor über einen ausgefeilten Mechanismus zur Beschreibung der Verfügbarkeit von Rechenressourcen und zur Überwachung (und Unterbrechung) von Berechnungsaufträgen. Diese Flexibilität erlaubt Condor, heterogene Rechner zu einem Verbund zusammenzufassen: Aspekte wie Rechnerarchitektur, Betriebssystem, Softwareausstattung, bevorzugte Nutzungsart usw. lassen sich administrativ beschreiben und für die Auftragsverteilung nutzen.

Im Projekt WISENT wurde die Condor-Einführung vor allem durch die Anforderungen bezüglich der einheitlichen Nutzung von verfügbaren Arbeitsplatzrechnern und des Computing-Cluster beim Projektpartner DLR PA motiviert. Vor dem Projektanfang wurden darauf lang dauernde Strahlungstransportrechnungen mit Hilfe eines ausgefeilten Lastverteilungsverfahrens (PVM/ppmake) auf die zum Rechencluster gehörenden Rechenknoten verteilt oder auf die zur Verfügung stehenden Arbeitsplatzrechner manuell kopiert und dort mit Hilfe von Shell-Skripten ausgeführt. Diese Vorgehensweisen hatten Nachteile technischer und organisatorischer Art. Beispielsweise war die Zuverlässigkeit von „ppmake“ eingeschränkt, und beim Fehlschlagen eines einzelnen Teilauftrags musste evtl. die gesamte Berechnung wiederholt werden, wodurch ggf. Rechen- und Arbeitszeit verschwendet wurden. Bei der Nutzung von Desktop-Arbeitsplatzrechnern für Berechnungen bestand die Gefahr, dass diese Aktivitäten die normale interaktive Verwendung der Rechner behindern. Außerdem konnten die einmal gestarteten Rechenaufträge nicht während ihrer Ausführung von einem Rechner auf einen anderen übergehen oder ihre Zwischenergebnisse sichern, wodurch wieder die Gefahr des Datenverlustes bei Abbruch gegeben war. Alle diese Probleme werden durch die eingebauten Condor-Funktionen behandelt, da Condor speziell für die Einrichtung von Desktop-Grids entworfen wurde.

Die Einführung von Condor bei DLR PA verlief erfolgreich mit Hilfe einer Installationsanleitung, die von OFFIS an Hand der offiziellen, umfangreichen Dokumentation erstellt wurde. Der Einsatz von Condor ermöglichte eine bequemere

Ausführung von Strahlungstransportrechnungen einschließlich der Rechenzeit-aufwändigen 3D-Simulationen und ersetzte die vorherigen, suboptimalen Verfahren, besonders bei der Verwendung des internen Clusters. Insbesondere wurde auch das Checkpointing-Feature von Condor ausgenutzt, um die Ausführung von lang dauernden Jobs ohne Datenverlustgefahr zu realisieren. Die Erfahrungen von DLR PA, die noch genauer Abschnitt 2.5 beschrieben sind, führten dazu, dass die Software ebenfalls beim Projektpartner Universität Oldenburg für die Nutzung von Desktop-Rechnern installiert wurde.

Als problematisch stellt sich bei Condor vor allem die sehr umfangreiche Menge von Konfigurationseinstellungen dar, die die Funktionsweise der Software beeinflussen. Die meisten von ihnen werden zwar bereits bei der Installation mit sinnvollen Werten belegt und sind gut dokumentiert. Allerdings müssen zur Optimierung des Ressourcenmanagements und zur Problembhebung die Zusammenwirkung von Konfigurationseinstellungen und die von Condor verwendeten Algorithmen verstanden werden. Da diese durch Anwender nicht einfach intuitiv nachvollziehbar sind, wurden die Abweichungen zwischen dem tatsächlichen und erwarteten Verhalten von ihnen als Fehler empfunden, obwohl sich Condor gemäß der konfigurierten Vorgaben verhielt. Beispielsweise wurden die laufenden Rechenaufträge unerwartet bzw. zu oft suspendiert; dieses Problem ergab sich aus Konfigurationseinstellungen und konnte durch einen Eingriff des Administrators behoben werden. Ein weiteres Problem ergab sich durch die Erschöpfung des freien Festplattenspeichers auf der Maschine, von der die Jobs abgeschickt wurden. Dieses Problem äußerte sich darin, dass die Rechenknoten als belegt erschienen, obwohl die Jobs ausdrücklich gestoppt wurden. In dieser Situation zeigte Condor leider unzureichende Fehlertoleranz und mangelhafte Fehlerberichterstattung. Die Problembhebung war dagegen nach der Feststellung seiner Ursache einfach.

Eine Anforderung, die erst nach der Condor-Einführung und einiger Verwendungszeit deutlich wurde und nur zum Teil zufriedenstellend erfüllt werden konnte, ist die relative Priorisierung von Benutzern bzw. Benutzergruppen. Condor versucht zwar, eine faire Zuweisung von Rechenressourcen nach den vorgegebenen Prioritäten zu erreichen. So kann man z. B. administrativ festlegen, dass ein Benutzer doppelt so viele Rechner wie ein anderer Benutzer bekommt. Leider werden solche Vorgaben durch Condor basierend auf der Gesamtanzahl aller Rechner umgesetzt, anstatt auf der aktuell verfügbaren, die Job-Anforderungen erfüllenden Anzahl. So kann sich im Fall der Ressourcenknappheit ergeben, dass dem Benutzer mit der höheren Priorität alle verfügbaren Rechner zugeordnet werden und dem Benutzer mit der niedrigeren Priorität gar keine, weil der bevorzugte Benutzer noch nicht die ihm zustehende Gesamtanzahl von Rechnern erschöpft hat.

Insgesamt ist Condor auf die Optimierung des Gesamtdurchsatzes in einer Umgebung mit autonom verwalteten Rechnern ausgelegt und für eine feingranulare, zentralisierte Festlegung von globalen Nutzungsrichtlinien eher ungeeignet. Dieser Mangel ging aus der Condor-Dokumentation nicht hervor, sondern konnte erst durch Experimente und Diskussionen in der Mailing-Liste verstanden werden. Es handelt sich hierbei leider nicht um ein fehlendes Feature, das möglicherweise in der Zukunft nachimplementiert wird, sondern vielmehr um grundsätzliche Beschränkungen, die sich aus dem verteilten, dezentralen Charakter des Scheduling-Algorithmus und aus fehlenden Informationen über den genauen Zustand aller Ressourcen ergeben.

2.3 TORQUE/Maui

Mit der Anschaffung eines Computing-Clusters im Rahmen der D-Grid-Sonderinvestition bei OFFIS musste ebenfalls ein geeignetes Batch-System installiert werden. Die vorherigen Erfahrungen mit Condor haben gezeigt, dass dieses zum Zeitpunkt der Cluster-Anschaffung bereits bei DLR PA und der Universität Oldenburg im Einsatz befindliche Batch-System zwar auch für OFFIS ausreichend wäre, aber aufgrund seiner Merkmale nicht optimal für die Verwaltung eines zentral administrierten D-Grid-Clusters mit einer fixen Anzahl von Rechenknoten erscheint. Die wichtigsten Argumente für die Auswahl einer Alternative zu Condor waren:

- Relativ geringe Verbreitung und kleine Erfahrungsbasis für den Einsatz von Condor im HPC-Bereich
- Fehlende Unterstützung für Condor seitens des D-Grid-Integrationsprojekts (DGI) und Wunsch nach einer Vereinheitlichung der Batch-Systeme im D-Grid
- Fehlende Condor-Features, die ein besseres zentrales Cluster-Management ermöglichen würden (z. B. Vorab-Reservierung von Rechenzeit für bestimmte Benutzergruppen)
- Bessere Integrationsaussichten mit zukünftigen D-Grid-Accounting-Systemen für TORQUE/Maui.
- Überflüssige Condor-Features, wie z. B. Erkennung der interaktiven Rechnernutzung oder die Migration von laufenden Jobs zwischen Rechnern. Diese Features werden im Kontext eines Rechenclusters (im Gegensatz zum Desktop-Grid) nicht benötigt und tragen zur Komplexität bei.

TORQUE ist ein Batch-System, das als Nachfolger von PBS eine lange Entwicklungsgeschichte im High-Performance-Computing und infolgedessen auch eine entsprechend hohe Stabilität aufweist. In Kombination mit dem Scheduler Maui bietet es eine weit verbreitete und offene Lösung in seinem Einsatzbereich. Als Option wird von den Entwicklern auch der Migrationsweg zum kommerziellen Cluster-Management-Produkt Moab angeboten.

Die Installation von TORQUE/Maui verlief bei OFFIS nach der Anleitung, die vom DGI geliefert wurde. Die offizielle Dokumentation für die Produkte ist im Vergleich zu Condor weniger umfassend und schlechter gepflegt. Zum Teil wird auf die nicht mehr aktuelle Dokumentation von PBS bzw. OpenPBS verwiesen.

Bei der Installation von Maui traten Probleme auf, die durch eine mangelhafte Verpackung der Binary-Version in ein RPM-Paket für das Betriebssystem SuSE Linux Enterprise Server 10.0 verursacht wurden. Für diese Probleme wurde in WISENT eine Lösung erarbeitet und an DGI berichtet. Dort wurden entsprechende Korrekturmaßnahmen ergriffen, von denen auch D-Grid-Projekte profitieren konnten.

Während des initialen Cluster-Betriebs ist in TORQUE ein weiteres Problem aufgetreten, das durch den Festplattenplatzmangel auf der Root-Partition von Rechenknoten verursacht wurde und sich dadurch manifestierte, dass von den ausgeführten Jobs keine Ausgaben auf den zentralen Rechner übertragen wurden. Die sichtbaren Symptome dieses Problems, wie es auch bei dem ähnlichen Problem mit Condor bei DLR PA der Fall war, wiesen nicht auf seine Ursache hin. Die Problembehebung war nach der Feststellung der Ursache jedoch leicht.

Da der OFFIS-Cluster verschiedenartige Rechenknoten enthält (mit 4 oder 2 Prozessoren, mit oder ohne InfiniBand-Netzwerkkarte), wurden die Rechenknoten mit Hilfe von

zusätzlichen Attributen kategorisiert. Diese Attribute können in einer Job-Beschreibung verwendet werden, um den Job entsprechend zu einem Knoten mit der gewünschten Ausstattung zu lenken. Allerdings fiel dabei auf, dass die Job-Beschreibungssprache von TORQUE im Vergleich zu Condor viel beschränkter ist, da ausschließlich statische, administrativ definierte Attribute und ausschließlich einfache Vergleiche unterstützt werden.

Neben der Ausführung von sequentiellen Jobs unterstützt TORQUE in Kombination mit dem Programm „mpiexec“ die Ausführung von parallelen Jobs, die mit Hilfe der Bibliothek MPI miteinander kommunizieren. Im OFFIS-Cluster wurde die MPICH-Implementierung aus dem Lieferumfang des kommerziellen PGI-Compilers installiert. Zusätzlich wurde für die Nutzung der InfiniBand-Netzwerkkarten, die eine deutlich geringere Latenzzeit und höhere Bandbreite als das von MPICH verwendete Gigabit Ethernet aufweisen, die MVAPICH-Implementierung aus dem Treiberpaket OFED (OpenFabrics Enterprise Distribution) der Firma Mellanox installiert.

Die Nutzung von MVAPICH mit TORQUE stellte sich zum Anfang als unmöglich heraus. Der Grund dafür war ein Fehler im Hilfsprogramm „mpiexec“. Dieser konnte in Zusammenarbeit mit seinem Entwickler behoben werden. Mit dieser Fehlerkorrektur leistete WISENT einen Beitrag für die internationale HPC-Community, die ebenfalls an der Nutzung von TORQUE in Kombination mit der aktuellen Version der InfiniBand-Treiber interessiert ist.

2.4 Vergleich zwischen Condor und TORQUE/Maui

Aufgrund von gesammelten Erfahrungen mit Condor und TORQUE wurde ein Vergleich der beiden Batch-Systeme möglich, der einen Beitrag zum besseren Verständnis und einer fundierten Entscheidung zwischen diesen Systemen leistet. Die allgemein verfügbare Dokumentation beschränkt sich stets auf Merkmale des einen oder anderen Batch-Systems, ohne dass ein solcher Vergleich aus der Sicht der Softwaretechnik angeboten wird. Um diese Lücke zu schließen, werden an dieser Stelle die Unterschiede zwischen Condor und TORQUE knapp zusammengefasst. Sie lassen sich gut in zwei Kategorien unterteilen: Entwurfsunterschiede und Annahmen über den Verwendungskontext.

Die Entwurfsunterschiede bestehen in der Art der Trennung des Ressourcenmanagement (Feststellung der Rechnerzustände, sowie die Job-Steuerung und -Ausführung) und Scheduling (Festlegung der Ausführungsreihenfolge) in Condor und TORQUE.

Condor besitzt eine monolithische Architektur, in der die Aspekte des Ressourcenmanagement und Scheduling zusammenwachsen. Im Umfang von Condor werden sowohl der Ressourcenmanager (zuständig für die Job-Steuerung und -Ausführung) als auch der Scheduler (zuständig für Entscheidungen über die Ausführungsreihenfolge) geliefert, und es gibt keine Möglichkeit für den Administrator, eine dieser Softwarekomponenten zu ersetzen.

Die Trennung zwischen dem Ressourcenmanagement und Scheduling ist dagegen sehr deutlich im Fall von TORQUE/Maui. TORQUE übernimmt in erster Linie die Aufgaben des Ressourcenmanagers. Das Softwarepaket enthält zwar einen rudimentären First-In-First-Out-Scheduler und ist damit auch alleinstehend verwendbar. Allerdings wird auch eine Scheduler-Schnittstelle angeboten, die eine Ersetzung des einfachen Schedulers durch eine ausgefeiltere Variante – Maui oder Moab erlaubt.

Die unterschiedlichen Annahmen über den Verwendungskontext von Condor und TORQUE/Maui lassen sich folgendermaßen charakterisieren:

Condor nimmt ein verteiltes Eigentum und Management von Ressourcen an (typisch für Desktop-Grids). Es werden folgende Rollen unterschieden: Condor-Administratoren, Ressourceneigentümer und Ressourcennutzer, wobei die letzten beiden Rollen oft von einzelnen Personen gemeinsam wahrgenommen werden. Unter diesen Annahmen werden Entwurfsziele definiert, die variierende Verfügbarkeit von Ressourcen zu tolerieren und die maximale Flexibilität bezüglich der Bestimmung der Ressourcennutzungsart für die Ressourceneigentümer zu gewährleisten. Charakteristisch ist, dass die Ressourceneigentümer jedoch keine Verpflichtungen für die Sicherstellung von Qualitätseigenschaften gegenüber Nutzern eingehen und nicht als Dienstleister auftreten. Die jeweils freien Rechenkapazitäten werden lediglich vermittelt und den Nutzern ohne Garantie angeboten.

TORQUE/Maui nimmt ein zentralisiertes Eigentum und Management von Ressourcen an (typisch für HPC-Cluster in Rechenzentren). Es werden folgende Rollen unterschieden: Cluster-Administratoren, Cluster-Nutzer und Geschäftsführung. TORQUE/Maui hat zum Ziel, die maximale Flexibilität für einen Cluster-Administrator in der Festlegung von Nutzungsrichtlinien für wohl definierte Gruppen von Nutzern zu erreichen. Gleichzeitig sind die Administratoren eines Clusters dafür verantwortlich, vorhersehbare Qualitätseigenschaften für die Nutzergruppe zu gewährleisten und die allgemeinen Richtlinien und Prioritäten entsprechend den vorgegebenen Geschäftszielen durchzusetzen.

2.5 Einsatz von Condor bei DLR PA

2.5.1 Motivation

In der Abteilung für Fernerkundung der Atmosphäre bei DLR-PA werden unterschiedliche Aspekte der Wolken-Strahlung-Wechselwirkungen untersucht. Einer davon, der eine wichtige Rolle in vIEM spielt, ist die Entwicklung, Bewertung und Validierung von Verfahren zur Bestimmung von Solarstrahlung am Erdboden aus Satellitenbeobachtungen. Wolken weisen nämlich die höchste zeitliche und räumliche Variabilität auf und haben den größten Einfluss auf die Solarstrahlung am Boden. Dazu werden detaillierte Strahlungstransfersimulationen mithilfe des libRadtran (library for radiative transfer [Mayer und Kylling 2005]) Softwarepakets durchgeführt. Wolken- und Atmosphärendaten des LM (Lokalmodell) Vorhersagemodells des Deutschen Wetterdienstes (DWD) und Zusatzdaten zur Beschreibung des Bodenreflexionsvermögens werden als Input dafür benutzt, um Satellitenmessungen von MSG/SEVIRI und einfallende Solarstrahlung am Boden zu simulieren. Nach Anwendung der Solarstrahlungsalgorithmen auf diese berechneten Satellitenmessungen können die Resultate mit den ebenfalls berechneten Solarstrahlungsdaten verglichen werden, um die Genauigkeit der Retrievalverfahren zu bestimmen. Außerdem hat man dann die Möglichkeit, die Fälle im Detail zu untersuchen, welche erhebliche Abweichungen zeigen, weil alle Inputfaktoren, d.h. alle atmosphärischen Parameter, bekannt sind. Damit können diese Algorithmen verbessert werden.

Zur Berechnung der Satellitenbeobachtungen sowie der einfallenden Solarstrahlung am Boden für eine gewisse Region gibt es grundsätzlich zwei Möglichkeiten. Üblicherweise wird die Gesamtsimulation in einzelne Atmosphärensäulen zerlegt und angenommen, dass diese horizontal homogen und unabhängig voneinander sind. Unter diesen Annahmen können sogenannte eindimensionale (1D) Strahlungstransferrechnungen durchgeführt werden. Auf libRadtran aufbauend ist zu diesem Zweck bei DLR-PA das

SIMSAT (Satellite Simulator) Modul geschrieben worden. Wenn hingegen keine Vereinfachungen gemacht werden und Photonen auf ihrem Weg durch die Atmosphäre genau verfolgt werden, sind diese sogenannten dreidimensionalen (3D) Strahlungstransferrechnungen aufwändiger und werden mit dem MYSTIC (Monte Carlo code for the physically correct tracing of photons in cloudy atmospheres [Mayer 2000]) Modul, auch auf libRadtran basierend, durchgeführt.

Wenn Satellitenbeobachtungen für ausgedehnte Gebiete simuliert werden, sind schon 1D Rechnungen sehr rechenzeitintensiv. Bisher wurden solche Arbeiten entweder auf einzelnen Standard-Arbeitsplatzrechnern unter Linux/UNIX oder auf einem 32 x 2 GHz PC-Cluster durchgeführt. Für einzelne Anwendungen wurden auch die gerade außerhalb der täglichen Kernarbeitszeiten wenig genutzten Arbeitsplatzrechner für verteiltes Rechnen genutzt. Allerdings wurde dies bisher durch personalintensives einzelnes Ansprechen der Rechner vorgenommen, eine automatische technische Lösung bestand nicht. Deswegen wurde im Rahmen von WISENT die Vernetzung der einzelnen Knoten des Linux-Clusters sowie die Vernetzung von 30 Arbeitsplatz-PCs mit Hilfe der Condor-Software vorgenommen.

2.5.2 Condor Installation und MYSTIC

Die Installation des Condor-Paketes mit der vorbereiteten Anleitung von OFFIS bereitete keine Probleme. Testläufe mit einfachen kleinen Beispielprogrammen, die im Paket enthalten sind, verliefen ebenfalls ohne Störungen.

Zum Testen der Berechnung von rechenzeitintensiveren Anwendungen wurden 25 Jobs mit dreidimensionalen Strahlungstransferrechnungen vorbereitet, wobei diese sich in einigen atmosphärischen Parametern unterschieden, um die Berechnung eines realistischen Wolkenfeldes zu simulieren. Alle Jobs wurden auf einer Maschine mit 3 GHz CPU bereits im Vorfeld berechnet. Die Dauer der Berechnungen variierte zwischen 4 Stunden und 85 Stunden.

Die Jobs wurden zunächst unter Verwendung des „vanilla“-Universums an das Condor-Netzwerk übergeben. Die Verteilung der Jobs auf die vorhandenen Knoten lief problemlos. Folgende Beobachtungen konnten während der Tests gemacht werden:

- Bei Simulation von mehreren Knoten unter Verwendung von hyperthreading auf Servern wird der Gesamtspeicher auf die entstehenden virtual machines aufgeteilt. Jede virtual machine erscheint als unabhängiger Knoten mit einer begrenzten Kapazität an Speicher im Condor-Netzwerk. Dies kann dazu führen, dass bei Angabe von einem Mindestspeicher in den Condor-requirements der betreffende Rechner nicht mehr verwendet wird, obwohl er eigentlich verfügbar wäre und auch entsprechende Voraussetzungen bzgl. CPU und Speicher hätte. Dieses Problem wurde gelöst, indem nur eine begrenzte Anzahl an virtual machines pro Rechner in das Condor-Netzwerk eingetragen wurden.
- Die Definition von „idle“ innerhalb Condors erwies sich während der Testläufe als zu weit gefasst, da Mausbewegungen beispielsweise beim Öffnen eines Browsers die Condor-Jobs nicht stoppen. Erst durch Tastatureingaben oder ssh-Zugriff werden Jobs angehalten. Dies konnte dazu führen, dass ein Rechner auf die Eingaben des Users sehr schwerfällig reagierte.
- Startet ein User sehr viele Jobs, ist das gesamte Netzwerk solange für andere User blockiert, bis diese Jobs abgearbeitet sind. Es können zwar über „user priorities“ einzelne User vorrangig behandelt werden, eigentlich sollte dies aber nicht der Fall sein. Des Weiteren können über die Angabe von Rank Einstellungen vorgenommen

werden, wie genau Jobs zu verteilen sind. Nachteil ist aber, dass dies vom User im Condor command file angegeben werden muss. Der User muss also einerseits wissen, wie er die Einstellungen am besten konfiguriert und andererseits auch bereit sein, seinen eigenen Job nicht mit maximaler Ausdehnung im Netz laufen zu lassen. Wünschenswert wären Einstellungen, die vom Administrator bei der Einrichtung des Netzwerkes vorgenommen werden können, z.B. maximale Anzahl von Jobs pro User.

- Jobs auf Rechnern, die als Arbeitsplatzrechner dienen, wurden oft in kurzen Abständen suspended und wieder unsuspending. Zum Teil wurden die Jobs schließlich nach mehreren Stunden abgebrochen und neu verteilt, wobei sie jedoch häufig wieder auf den Rechner geschoben wurden, auf dem sie gerade abgebrochen worden waren. Die Berechnung dieser Jobs dauerte dementsprechend sehr lange. Um die Verteilung der Jobs zu verbessern und auch dem suspend/unsuspend vorzubeugen, wurde die ContinueldleTime auf 5 Minuten hochgesetzt und ein rank=kflops in die command-Datei eingefügt, um bei der Neuverteilung schnelle Rechner zu bevorzugen.

Um bei Benutzung des DLR-PA-Netzwerkes zu gewährleisten, dass nach stundenlanger Bearbeitung eines Jobs nicht alle Ergebnisse verloren gehen, wenn ein User an seinen Arbeitsplatzrechner zurückkehrt, wurde die Verwendung des „standard“-Universums und damit die Möglichkeit des Checkpointing in Betracht gezogen. Kleinere C-Programme konnten unter Verwendung von condor_compile kompiliert und verlinkt werden. Für das Gesamtpaket von libRadtran erwies sich dies jedoch als äußerst schwierig. Um überhaupt einen fehlerfreien Kompilierungsvorgang mit condor_compile durchführen zu können, mussten einige Anpassungen in der betreffenden libRadtran-library vorgenommen werden. Als Hürde erwies sich auch die Einbeziehung der netcdf library. Der Pfad für diese kann zwar beim Ausführen des entsprechenden configure-Skriptes explizit angegeben werden, so dass sich libRadtran unter normalen Umständen problemlos kompilieren lässt. Unter Einbeziehung von Condor führte dies jedoch nicht zum Erfolg. Eine vollständige Kompilierung konnte nur erreicht werden, wenn die netcdf-library im System statisch verlinkt war, aber auch dann nicht auf jedem System. So enthalten SuSE-Versionen ab 10.x zwar netcdf, führen aber trotzdem zu Fehlermeldungen und Abbruch der Kompilierung. Funktionierte die Ausführung des make-Befehls, war die Dauer des Kompilierungsvorganges sehr stark erhöht. Gegenüber Kompilation ohne Einbeziehung von Condor verlängerte sich diese mindestens um einen Faktor 5.

Noch ein Faktor musste berücksichtigt werden: wenn libRadtran mit Condor kompiliert wird und im „standard“-Universum arbeitet, hat MYSTIC keinen Zugriff auf den Rechner, wo der Job tatsächlich läuft, weil die Umgebung des Rechners simuliert wird, wo der Job eingereicht wurde. Das macht die Abfrage von host id und ähnlichem unmöglich, was aber einen Einfluss auf die Erzeugung der Zufallszahl hat, die zum Initialisieren des Monte Carlo Prozesses gebraucht wird. Fangen zwei MYSTIC-Rechnungen mit derselben Zufallszahl an, sind die Resultate identisch. Das Problem kann man umgehen, indem man diese Zufallszahl an MYSTIC explizit übergibt.

Auf dem DLR-PA-Cluster war die Kompilation von libRadtran nach den oben genannten Veränderungen problemlos möglich. Auch die Bearbeitung von Jobs unter Verwendung des „standard“-Universums funktionierte auf diesem System. Im Abstand von 3 Stunden wurden Checkpoints angelegt und bei Bedarf der Job auf einen anderen Knoten verlagert.

2.5.3 SIMSAT

Als zusätzliche Anwendung wurde das SIMSAT Paket zur Simulation von Satellitenstrahldichten mit eindimensionalen Strahlungstransferlösern mit den Condor-

Funktionalitäten erweitert. Der technische Hauptunterschied zu den oben erwähnten dreidimensionalen Rechnungen besteht darin, dass SIMSAT kein C-Programm ist, sondern ein C-Shell-Skript, das unter Linux läuft und unterschiedliche C-Programme und weitere Shell-Skripte aufruft. Unter den C-Programmen befindet sich natürlich auch libRadtran zur Lösung der Strahlungstransportgleichung. Besondere Aufmerksamkeit musste hier den temporären Dateien gewidmet werden, die von SIMSAT während eines Laufes erzeugt werden und Gefahr laufen, von konkurrierenden SIMSAT-Läufen überschrieben zu werden. Mit einer sauberen Namensgebung und den mit MYSTIC gesammelten Erfahrungen konnte SIMSAT mit Condor im „vanilla“-Universum auf den DLR-PA-Arbeitsplatzrechnern und auf dem DLR-PA-Cluster benutzt werden. Unterschiedliche Tests wurden dabei erfolgreich durchgeführt.

Schließlich wurde die Nutzung des von OFFIS betreuten Compute-Clusters mit Condor-GT4 angestrebt und realisiert. Nach erfolgreicher Beantragung eines Grid-Zertifikats erwies sich die DLR-Firewall als das Hauptproblem. Mit Hilfe von OFFIS konnten diese Schwierigkeiten beseitigt werden, indem

- Ein Rechner außerhalb der eigentlich Firewall in einer so genannten demilitarisierten Zone (DMZ) als „submission host“ eingesetzt wurde.
- GridFTP die erlaubten Ports für den benötigten Datentransfer explizit mitgeteilt wurden.
- Die genaue Identifikation des DMZ-Rechners seitens des OFFIS-Clusters sichergestellt wurde.

Die für das SIMSAT-Paket notwendigen Veränderungen lassen sich wie folgt zusammenfassen:

- Der „submission host“ und die CPUs, die die Strahlungstransferrechnungen durchführen, verfügen nicht über einen gemeinsamen Speicher, auf den sie zugreifen könnten. Deswegen musste auf der einen Seite gewährleistet werden, dass alle nötigen Files dem OFFIS-Cluster zur Verfügung stehen; und auf der anderen Seite, dass die unterschiedlichen Verzeichnisstrukturen auf dem DMZ- und den OFFIS-Rechnern berücksichtigt wurden.
- SIMSAT wurde so angepasst, dass Grid-Ressourcen wie das oben erwähnte GridFTP-Client zum Einsatz kommen, wenn Input- und Output-Daten von und zu dem einreichenden DMZ-Rechner transferiert werden.
- Wegen der kurzen Rechenzeit eines einzelnen Jobs und der gleichzeitig hohen Anzahl solcher Jobs, erwies es sich als sehr nützlich, das von OFFIS entwickelte Perl-Modul Multijob zu benutzen, das mehrere Jobs bündelt und sie auf eine beschränkte Anzahl Prozessoren verteilt. Damit konnte erreicht werden, dass die NFS-Last auf dem OFFIS-Cluster begrenzt blieb.

Das so modifizierte SIMSAT-Paket wurde zusammen mit libRadtran letztendlich „per Hand“ auf dem OFFIS-Cluster installiert und dort mittels Condor-GT4 vom „submission host“ angesprochen.

3.Grid-Middleware

Neben Batch-Systemen, die ein wichtiger Bestandteil einer Grid-Infrastruktur sind, basiert die eigentliche Konstruktion eines Grid im Wesentlichen auf dem Einsatz von Grid-Middleware. Eine Grid-Middleware abstrahiert von den Ressourcen und den unterschiedlichen Zugriffsmöglichkeiten, z.B. bei Compute-Ressourcen von den

unterschiedlichen Batch-Systemen, und bietet eine einheitliche Zugriffsschnittstelle an. Zudem finden Sicherheitsaspekte, z.B. Zugriffssicherheit durch eine „Authentication and Authorisation Infrastructure“ (AAI), eine besondere Berücksichtigung.

Im Rahmen des Projektes WISENT wurden zunächst verschiedene Grid-Middleware gesichtet und grob verglichen. Die Entscheidung fiel aus unterschiedlichen Gründen recht schnell auf Globus Toolkit 4. Hauptgründe waren der hohe Bekanntheitsgrad, die vielfältigen Funktionen für das Datenmanagement, und die Verwendung von Grid-Standards wie das „Web Services Resource Framework“ (WSRF). Später wurde auf dem Compute-Cluster im OFFIS neben Globus Toolkit 4 noch UNICORE 5 und gLite installiert, um eine Vorgabe des DGI zum Betrieb der Sonderinvestitionen umzusetzen.

Die Erfahrungen mit Globus Toolkit 4, UNICORE 5 und gLite werden in den Abschnitten 3.1 bis 3.3 vorgestellt.

3.1 Globus Toolkit 4

Im Jahre 1995 wurden in den USA im Rahmen des Projektes I-Way 17 Hochleistungsrechenzentren miteinander vernetzt. Dies gilt als Startpunkt für die Entwicklung des Globus Toolkit zur Vernetzung von verteilten Ressourcen, dessen Entwicklung von Ian Foster und Carl Kesselman, die den Begriff „Grid“ im Jahre 1998 prägten, vorangetrieben wurde. Seit dem Jahr 2002, das Globus Toolkit 2 hatte schon einen hohen Verbreitungsgrad erreicht, wird die Einführung von Standards im Grid-Umfeld fokussiert, um eine Interoperabilität mit anderer Grid-Middleware zu erreichen bzw. auf bereits bestehenden und etablierten Standards aufzusetzen. Diese Bestrebung wurde in der so genannten „Open Grid Services Architecture“ (OGSA) festgehalten, die insbesondere auf Web Service-Technologien aufbaut. Eine der daraus resultierenden Standards ist das „Web Services Resource Framework“ (WSRF), das in Globus Toolkit 4 (GT4) implementiert wurde. GT4 wurde als Grid-Middleware im Projekt WISENT ausgewählt. Dazu waren folgende Gründe ausschlaggebend.

- Hoher Bekanntheitsgrad
- Viele Dienste zum Datenmanagement z.B. GridFTP
- Verwendung von Standards wie WSRF
- Verfügbarkeit als Open-Source-Software
- Erweiterbarkeit
- Anbindung von Condor als Batch-System möglich

GT4 sollte im Projekt WISENT in erster Linie zur Vereinheitlichung von Datentransfers genutzt werden sowie als Schnittstelle zur Nutzung verteilter Compute-Ressourcen. Initial wurde das GT4.0.2 verwendet, worauf sich der folgende Text auch im Wesentlichen bezieht. Später wurde dann auf neuere Versionen gewechselt, insbesondere auf dem Compute-Cluster im OFFIS.

Für die erste Installation wurde das Red Book „Introduction to Grid Computing“ von IBM genutzt, weil darin sehr gut Schritt für Schritt die Installation des GT4 mit drei Rechnern (einer dient als CA) sowie einfache Tests beschrieben werden. Später wurde die offizielle Dokumentation auf www.globus.org konsultiert. Allerdings sind die dort angebotenen Dokumentationen der einzelnen Dienste zum Teil unvollständig bzw. recht knapp gehalten. Auf Basis dieser Quellen wurde eine eigene Installationsanleitung für die WISENT-Projektpartner erstellt. Bei jedem Projektpartner wurde das GT4 installiert. Zur Konfiguration wurden später die Vorgaben des DGI umgesetzt.

Die Installation kann entweder mit der Binary-Version oder der Source-Version durchgeführt werden. Binary-Installer gibt es für zahlreiche Linux-Betriebssysteme wie Red Hat, den Red Hat-Ableger Fedora, SuSE und Debian. Für alle anderen Unix/Linux-Systeme steht dann der Source-Installer zur Verfügung. Der Source-Installer hat allerdings den Nachteil, dass eine vollständige Installation gut eine Stunde dauern kann, je nach Performanz des Zielsystems. Der Binary-Installer hingegen benötigt ein paar Minuten. Der Java-WS-Core, im Prinzip der WSRF-Grid Service-Container und mitgelieferte Grid Services, ist auch unter Windows lauffähig, weil dieser komplett in Java implementiert wurde. Weitere Komponenten wie der GridFTP-Server sind in C implementiert und laufen nur auf Unix/Linux-Systemen.

Die Installation des GT4 (4.0.2) hat einige Systemvoraussetzungen, wie Java SDK ab Version 1.4.2, Apache ANT ab Version 1.5.1, PostgreSQL ab Version 7.1, gcc ab Version 2.95, zlib ab 1.1.4, GNU tar/sed/make und sudo. Diese wurde mit Folgeversionen des GT4 jeweils angepasst. Unter Sun Solaris 9, was beim DLR eingesetzt wird, gab es dafür zum Teil keine vorgefertigten Pakete, so dass eine Installation bzw. Kompilierung vom Quelltext notwendig war, zum Teil mit weiteren Abhängigkeiten. Dies erzeugte einen größeren Mehraufwand zur Erfüllung der Systemvoraussetzungen, bevor die eigentliche Installation des GT4 durchgeführt werden konnte. Der Installationsvorgang selbst lief dann ohne Probleme ab. Mittlerweile ist aber auch eine Binary-Version für Solaris 9 und 10 verfügbar.

Die Einrichtung der PostgreSQL-Datenbank für den „Reliable File Transfer“-Dienst hat zum Teil Probleme verursacht. Zum einen lag das daran, dass sich zwischen den PostgreSQL-Versionen 7 und 8 die Konfigurationseinstellung geändert hat, mit der die Netzwerkschnittstelle der Datenbank aktiviert werden kann. Zum anderen sind Probleme mit Datenbankberechtigungen aufgetreten, weil die Datenbank mit einer falschen Benutzerkennung angelegt wurde, was auf eine schlechte Vorgehensweise zur Einrichtung einer Datenbank in der IBM-Anleitung zurückzuführen war.

Allgemein fiel bei der Einrichtung verschiedener Dienste negativ auf, dass für jeden Dienst, der eine Datenbank benötigt, eine eigene Datenbankverbindung konfiguriert wird. Globale „Connection-Pools“ sind nicht vorgesehen. So nutzt der RFT-Dienst (Reliable File Transfer) beispielsweise eine Datenbank (z.B. PostgreSQL oder MySQL) über JDBC und der RLS-Dienst (Replica Location Service) über ODBC.

Bezüglich des Datenmanagements wurden mehrere Dienste getestet. Dazu gehören insbesondere GridFTP und der Reliable File Transfer (RFT).

Bezüglich GridFTP stehen im WISENT-Kontext drei Optionen im Vordergrund:

- Unterstützung der Grid Security Infrastructure (GSI), so dass eine einmalige (single-sign on - SSO), auf X.509-Zertifikaten basierende Anmeldung möglich ist.
- Unterstützung von so genannten „Third Party Transfers“, d.h. ein Client-Rechner baut eine Kontrollverbindung zu zwei GridFTP-Servern auf, woraufhin zur Datenübertragung eine direkte Verbindung zwischen den beiden GridFTP-Servern aufgebaut wird. Der Datenstrom geht somit nicht über den Client selbst.
- Nutzung von mehreren, parallelen Datenkanälen, um höhere Durchsatzraten zu ermöglichen.

Erste Tests mit GridFTP haben gezeigt, dass strenge Firewall-Regeln dessen Nutzung stark beeinträchtigen bzw. gänzlich verhindern können. Grund ist die Verwendung von dynamischen Ports zur Öffnung von Datenkanälen, wobei vom DGI die pauschale Öffnung des Portbereichs 20000-25000 empfohlen wurde. Das war insbesondere mit den Sicherheitsrichtlinien beim DLR nicht kompatibel. Weitere Informationen dazu sind im

Dokument „AP 2.1 - Bericht zu den Auswirkungen der DLR-IT-Sicherheitsstrukturen beim Datentransfer mit Partnern außerhalb des DLR“ zu finden.

Üblicherweise sind bei GridFTP der Kontrollkanal verschlüsselt und die Datenkanäle unverschlüsselt. Der Aufbau des Datenkanals wird aber über eine so genannte „data channel authentication“ (DCAU) abgesichert. Über entsprechende Optionen kann der Datenkanal zusätzlich gegenüber Manipulation (Checksumme) abgesichert oder gar komplett verschlüsselt (Checksumme und Verschlüsselung) werden, was aber bei Tests zu der erwarteten starken Reduzierung (um mehr als die Hälfte) des Datendurchsatzes geführt hat.

Im Rahmen kleinerer Datentransfertests mit GridFTP wurde für ein Teil der in WISENT verwendeten Datentransferwege überprüft, inwieweit die Nutzung paralleler Datenkanäle den Durchsatz erhöhen können. Die Ergebnisse dazu befinden sich in Abschnitt 4.1.1.

Der „Reliabe File Transfer“ (RFT) des GT4 kann dazu genutzt werden, mehrere Datentransfers durchzuführen und überwachen zu lassen. Der Zustand der einzelnen Datentransfers wird dabei in einer Datenbank persistiert. Mit dem RFT-Dienst können nur so genannte „Third-Party-Transfers“ mittels GridFTP durchgeführt werden. Bis auf die zusätzliche Absicherung der Datenkanäle sind alle GridFTP-Optionen nutzbar. Allerdings verwendet der RFT-Dienst eine umständlich aufgebaute Eingabe-Datei mit fester Struktur, in der am Anfang Werte für bestimmte Optionen in einer bestimmten Reihenfolge untereinander stehen müssen, gefolgt von Quell-Ziel-Paaren für die zu übertragenden Dateien. Ein einzelner RFT unterstützt dabei nur Transfers mehrerer Dateien zwischen einem Quell- und einem Ziel-Rechner. Daher ist eine direkte manuelle Nutzung inklusive Bearbeitung der Eingabe-Datei zu umständlich, was durch entsprechende Skripte abgefangen werden muss.

Das GT4 bietet die Möglichkeit an, Jobs an das Batch-System Condor zu schicken. Um diese Möglichkeit zu nutzen, muss bei der Installation die Source-Version genommen werden und eine entsprechende Option (`--enable-wsgram-condor`) aktiviert werden. Bei ersten Tests konnten zwar Jobs abgeschickt werden, diese wurden aber nur auf dem lokalen (GT4-)System ausgeführt, insofern dieses im Condor-Pool eingetragen war. Eine nähere Betrachtung der Vorgänge beim Einreichen eines Jobs über das GT4 an Condor hat ergeben, dass die Fehlerursache in dem in Perl geschriebenen Condor-Adapter lag, der bestimmte Condor-Optionen falsch gesetzt hatte. Der Condor-Adapter wurde entsprechend modifiziert. Im Verlauf des Projektes wurde ebenfalls der PBS-Adapter für TORQUE angepasst, weil dieser für die Ausführung eines Jobs gewisse Unzulänglichkeiten besaß. So wurde durch eine mangelhafte Implementierung beispielsweise vor jeder Job-Ausführung ein ssh-Login (wurde in WISENT durch rsh ersetzt) auf den Zielknoten gemacht. GT4 bietet über die Option „count“ die Möglichkeit an, den selben Jobs mehrmals zu starten. Das wird über ein Wrapper im Job-Adapter gesteuert, der eine Knotenliste bekommt und den Job dann der ssh-Login auf den entsprechenden Zielknoten startet. Das macht aus zwei Gründen keinen Sinn: 1) Bei `count=1` wird immer ein ssh-Login auf den lokalen Knoten gemacht, der bereits vom Batch-System als Zielknoten ausgewählt wurde. Das führte u.a. dazu, dass Jobs nicht richtig suspendiert werden konnten, d.h. die Prozesse liefen weiter und der Jobs-Zustand war „suspend“. Im DGI führte dies ebenfalls zu Seiteneffekten, denn im Accounting konnte dadurch die CPUtime eines Jobs nicht richtig erfasst werden. Der Adapter wurde daher entsprechend für `count=1` geändert. 2) Bei mehrfacher Ausführung des gleichen Jobs sollte das Batch-System das Starten jedes einzelnen Jobs übernehmen, und nicht der Job-Adapter selbst. Letztlich war durch das einfache, auf Perl-Skripten basierende Job-Adapter-Konzept die Möglichkeiten gegeben, sehr schnell und einfach eigene Anpassungen vorzunehmen.

Insgesamt kann das GT4 als eine einsatzfähige Grid Middleware bezeichnet werden, die nutzbare Basisdienste anbietet. Dennoch ist das GT4 dem Namen nach wie ein „Toolkit“ zu verstehen, auf dem eigene, komplexere Dienste aufgebaut werden können. Insbesondere sind erweiterte Datenmanagementdienste wie der RLS (Replication Location Service) und der DRS (Data Replication Service) direkt nicht wirklich nutzbar. Die Benutzerfreundlichkeit der angebotenen Kommandozeilenwerkzeuge steht ohnehin nicht im Vordergrund, was für den Endnutzer zumindest über einfache Skripte abgefangen werden sollte. Die Entwicklung eigener Dienste bzw. Grid Services ist mit dem GT4 allerdings noch sehr komplex und es fehlt an ausgereifter Werkzeugunterstützung. Eine weitere Schwäche von GT4 ist auch die Performanz. Es wurden dahingehend zwar keine umfangreichen Tests durchgeführt, aber der Betrieb des Compute-Clusters hat gezeigt, dass insbesondere bei Einreichung mehrerer Jobs (>100) die Performanz stark leidet und GT4 viel Hauptspeicher (z.Z. 2 GB) zur Verfügung gestellt werden muss. Mittlerweile wurde GT4.2 veröffentlicht, was einige Neuerungen beinhaltet. Diese wurde aber im Rahmen von WISENT nicht betrachtet.

3.2 UNICORE 5

UNICORE steht für „Uniform Interface to Computing Resources“ und startete 1997 als ein BMBF-gefördertes Projekt und wurde später auch auf EU-Ebene gefördert. UNICORE hat sich besonders im Bereich der Hochleistungsrechenzentren verbreitet. Weil UNICORE bzw. UNICORE 5 eine unterstützte Grid-Middleware in D-Grid ist, wurde UNICORE 5 auch auf dem Compute-Cluster im OFFIS installiert. Vorher wurden bereits kleine Tests mit UNICORE 5 auf Testsystemen durchgeführt, um dieses mit dem GT4 zu vergleichen.

UNICORE 5 bietet als Einstieg ein Quickstart-Bundle an, mit dem schnell und einfach eine lauffähige UNICORE 5-Installation durchgeführt werden kann. Danach können Jobs allerdings nur auf dem lokalen System gestartet werden, eine Anbindung an ein Batch-System fehlt noch. Eine initiale, separate Installation und Einrichtung der einzelnen UNICORE-Komponenten (Gateway, NJS, TSI, UDDB) war sehr schwer durchzuführen, weil dazu genaue Kenntnisse der einzelnen Konfigurationseinstellungen notwendig sind. Es gibt aber keine zusammenhängende Dokumentation, sondern lediglich Einzeldokumente zu den einzelnen Komponenten. Dies erschwert eine manuelle Konfiguration mit geringen Vorkenntnissen.

Das DGI bietet ein modifiziertes Quickstart-Bundle sowie eine Installationsanleitung an, mit dem UNICORE 5 schnell und einfach auf dem Compute-Cluster installiert werden konnte. Ein kleiner Fehler, der sich im Installationsskript verbarg, wurde an die entsprechenden UNICORE-Ansprechpartner berichtet. Mit Hilfe dieser Ansprechpartner wurden auch kleinere Konfigurationseinstellungen überarbeitet und die UNICORE 5-Installation für D-Grid-Benutzer zugreifbar gemacht.

Ein Vorteil von UNICORE 5 ist der grafische Client, mit dem Jobs erstellt und an eine Compute-Ressource geschickt werden können. Zudem können einfache Workflows erstellt werden. Nachteilig erschien aber die notwendige manuelle Zuteilung eines Jobs an eine bestimmte Compute-Ressource. Eine Art Scheduling fehlt an dieser Stelle noch. Besonders umständlich ist dies bei größeren Workflows, bei dem jeder Teilschritt ein Job ist und entsprechend eine Compute-Ressource manuell zugeordnet werden muss. UNICORE 5 verfolgt allerdings eher den Ansatz, dass der Nutzer sich selbst seine präferierte Ressource aussucht.

Ein großer Vorteil von UNICORE 5 ist, dass der Gateway als Schnittstelle nach außen einen einzigen Port benötigt, was sehr Firewall-freundlich ist. Dies bringt aber zugleich einen sehr großen Nachteil mit sich. Denn alle Datentransfers müssen zum einen durch

den Gateway geleitet werden, zum anderen wird ein eigenes Datenübertragungsprotokoll verwendet. Die maximal erreichbare Durchsatzrate liegt bei ca. 200KB/s, was zur Übertragung der Datenmengen im WISENT-Kontext nicht geeignet ist. Allerdings geht UNICORE 5 eher davon aus, dass sich benötigte Daten schon auf der Computere-Ressource befinden, die vorher über andere, schnellere Datentransferprotokolle dorthin übertragen wurden.

Insgesamt ist UNICORE 5 die Grid-Middleware, die am schnellsten konfiguriert und installiert werden konnte. Allerdings hat UNICORE 5 noch einige Schwächen bzgl. der Unterstützung von Standards (es werden keinerlei Standards aus dem Grid-Umfeld unterstützt) und schneller Datentransfers. Der Nachfolger UNICORE 6 ist eine komplette Neuimplementierung der UNICORE 5-Komponenten in Java und bringt einige wesentliche Neuerungen mit sich. Eine der wichtigsten Neuerungen ist die Unterstützung von etlichen Standards wie WSRF, JSDL, usw., die zum Teil noch nicht vom GT4 (in den Versionen 4.0.x) unterstützt werden. UNICORE 6 kann generell als Vorreiter in der Aufnahme von Grid-relevanten Standards bezeichnet werden und hat damit einen großen Schritt in Richtung Interoperabilität vollzogen. Zusätzlich hat UNICORE 6 sowohl Client- als auch Server-seitig eine ausgereifte Plugin-Funktionalität, um eigene Erweiterungen einzubringen. UNICORE 6 hat sich grundlegend weiter entwickelt und wird sich voraussichtlich als Grid-Middleware in Zukunft stärker etablieren können.

3.3 gLite

gLite ist eine sehr umfangreiche Grid-Middleware, die im Projekt EGEE (Enabling Grids for E-Science) unter der besonderen Berücksichtigung von Anforderungen der Hochenergiephysikanwendungen entwickelt wurde. Aufgrund der Größe des EGEE-Projekts ist gLite die am weitesten verbreitete produktiv eingesetzte Grid-Middleware überhaupt. Im Rahmen der vom BMBF finanzierten D-Grid-Sonderinvestition wird gLite 3.0 neben GT4 und UNICORE als Zugangsmechanismus für die angeschafften Cluster verwendet. Für WISENT bietet die Auseinandersetzung mit gLite eine Möglichkeit der zukünftigen Ressourcennutzung über das D-Grid hinaus.

Die Installation von gLite wurde mit Hilfe einer durch das DGI-Projekt erstellten Anleitung und eines vorgefertigten Festplattenimage vorgenommen. Trotz dieser Hilfestellungen war die Installation von gLite im Vergleich zu den beiden anderen Middleware-Paketen bei weitem die aufwändigste.

gLite 3.0 ist nur eingeschränkt portabel zwischen verschiedenen Linux-Versionen und erfordert die Verwendung von Scientific Linux 3.0.8, einer beim CERN speziell für das Grid entwickelten Linux-Distribution. Da die zentrale IT-Administration in OFFIS ausschließlich SuSE Linux unterstützt (insb. was die Aktualisierung mit Patches betrifft), und da der Cluster in einer Single-System-Image-Konfiguration betrieben wird (die Software wird auf dem Master-Knoten installiert und ist dann automatisch auf allen Rechenknoten verfügbar), wurde im gesamten Cluster SuSE Linux Enterprise 10 installiert. Aus diesem Grund musste gLite in einer virtuellen Maschine (Xen), ähnlich wie in der Referenzinstallation des DGI, untergebracht werden.

Die ersten Probleme ergaben sich bei der Xen-Installation. Die vom DGI gelieferte Anleitung erklärte nicht die Details der notwendigen Netzwerk- und Routingkonfiguration für die zwei installierten Netzwerkkarten und die Interaktion von Einstellungen mit der Firewall auf dem Master-Knoten. Zudem musste zum Hochfahren der virtuellen Maschine eine neue Ram-Disk erstellt werden, was ebenfalls erst im Laufe von gescheiterten Versuchen herausgefunden werden konnte.

Nach der erfolgreichen Einrichtung der virtuellen Maschine musste die gLite-Konfiguration und die Installation von zusätzlichen auf dem Festplattenimage nicht mitgelieferten Modulen durchgeführt werden. Dabei wurden einige Unklarheiten in der Anleitung des DGI entdeckt, die von den DGI-Ansprechpartnern beseitigt und in eine aktualisierte Version der Anleitung eingebracht wurden. Die Tests von bereits installierten Paketen wurden dadurch erschwert, dass die für gLite notwendige zentrale Infrastruktur in Karlsruhe zeitgleich mit der WISENT-Installation aufgesetzt wurde. Deswegen war bei manchen aufgetretenen Problemen nicht sofort klar, ob ihre Ursache auf der WISENT- oder DGI-Seite lag.

Im Rahmen der Installation wurde ein Fehler in einem der Job-Manager von gLite entdeckt, der die Berechnung des freien Arbeitsspeichers auf dem Master-Knoten betraf und zur Ablehnung von abgeschickten Jobs führte. Dieser Fehler musste im vorliegenden Perl-Code manuell korrigiert werden.

Es war während der Installation nicht klar, ob gLite mit der auf den Rechenknoten installierten Version von SuSE Enterprise Linux überhaupt zusammenarbeiten kann. In der DGI-Referenzinstallation wurde stattdessen Scientific Linux für die Knoten verwendet, was in OFFIS aus den oben genannten Gründen nicht durchgesetzt wurde. Durch die erfolgreiche Einrichtung der gewünschten Konfiguration in WISENT wurden Erfahrungen gesammelt, die an DGI berichtet wurden und in die gemeinsame Installationsanleitung einfließen.

Eine Reihe von Problemen wurde durch die Ausschaltung des Job-Managers „fork“ verursacht. Dieser Job-Manager ermöglicht Grid-Nutzern die Ausführung von beliebigen Programmen auf dem Master-Knoten, was generell als Sicherheitslücke einzustufen ist. Trotzdem muss er bei gLite 3.0 eingeschaltet bleiben, wie unsere Erfahrungen zeigten. Es fiel auf, dass während der gescheiterten Job-Ausführung Fehlermeldungen sichtbar wurden, die allerdings nichts mit dem ausgeschalteten Job-Manager zu tun hatten und alleine die Job-Ausführung nicht beeinträchtigten. Trotzdem wurde sehr viel Zeit in die genaue Diagnose der Ursachen dieser Fehler investiert, die nicht nur Eingriffe in verschiedene von gLite verwendete Skripte, sondern auch das Herunterladen und die Übersetzung des gLite-Quellcode erforderte (zum Debugging). Zusammenfassend litt der gesamte gLite-Installationsprozess darunter, dass das beobachtete Verhalten der Software nur mühsam in korrekt, anscheinend korrekt, falsch oder anscheinend falsch klassifiziert werden konnte. Die Verständlichkeit von jeglichen Fehlermeldungen, die von gLite berichtet wurden, war sehr schlecht.

Trotz Schwierigkeiten bei der Installation war WISENT eins der ersten mit der Sonderinvestition finanzierten Projekte, die den Zugriff auf die angeschafften Ressourcen mit gLite gemäß Anforderungen des DGI ermöglichten. Nach der erfolgreichen Einrichtung von gLite 3.0 wurden Tests mit verschiedenen Jobs durchgeführt, die sowohl an den lokalen Cluster in OFFIS als auch an andere bereits mit gLite erreichbare D-Grid-Cluster abgeschickt wurden. Unter anderem wurden erfolgreich virtuelle Maschinen (realisiert mit User Mode Linux) zum Rechenzentrum in Hannover verschickt, um dort in einer von WISENT festgelegten Softwareumgebung einige Testprogramme auszuführen. Beim Abschicken von gLite-Jobs fiel negativ auf, dass der gesamte Vorgang sehr langsam ist – es verliefen stets mehrere Minuten von der Job-Einreichung bis zur Ausführung bzw. dem Einreichen des Jobs an ein Batch-System. Die Zeitverzögerung beteiligter Batch-Systeme hat auf Grund hinreichend verfügbarer Ressourcen keine wesentliche Rolle gespielt.

4. Datenmanagement

Neben dem Zugriff den Rechenressourcen und der Ausführung von Jobs spielt das Datenmanagement für Input bzw. Output-Daten eine zentrale Rolle. Zum Teil wurde dies

schon in Abschnitt 3 bei der jeweiligen Grid-Middleware erwähnt. Dieser Abschnitt konzentriert sich auf das in WISENT produktiv eingesetzte GridFTP als Datentransferprotokoll (Abschnitt 4.1) sowie dCache als Schnittstelle zu großen Speicherressourcen (Abschnitt 4.2).

4.1 GridFTP

Neben der Durchführung von Datentransfertests (Abschnitt 4.1.1), werden in diesem Abschnitt zwei Einsatzszenarien des GridFTP vorgestellt, in denen sicherheitsrelevante Problemstellungen fokussiert wurden. Dazu gehört der Umgebung mit strengen Sicherheitsrichtlinien und daraus resultierenden strengen Firewalls bei DLR DFD (Abschnitt 4.1.2) sowie die Einrichtung eines GridFTP-Servers in einer „chroot“-Umgebung an der Universität Oldenburg und dessen Einbindung in einen komplexeren Workflow mit Nutzung von Rechenressourcen und einer dCache-Ressource des D-Grid (Abschnitt 4.1.3).

4.1.1 Datendurchsatztests

GridFTP bietet die Option an, für einen Datentransfer mehrere Datenkanäle gleichzeitig zu nutzen. Um den Nutzen dieser Option zu bestimmen, wurden kleinere Tests durchgeführt. An den Testszenarien waren drei GridFTP-Server beteiligt:

- srvgrid.offis.uni-oldenburg.de - OFFIS
- eridanus.caf.dlr.de - DLR
- juggle-glob.fz-juelich.de - Jülich

Als Testdatei wurde eine 1GB-Datei (1073741824 Bytes) verwendet. In einem Durchgang wurde die Datei 10 mal übertragen. Der erste Transfer in einem Durchgang begann mit einem Datenkanal, für jeden weiteren Transfer wurde die Anzahl um 1 inkrementiert. Der 10te Transfer in einem Durchgang hat demnach 10 Datenkanäle genutzt. Jeder Durchgang wurde für jeden möglichen Transferweg (DLR->Jülich, Jülich->DLR, OFFIS->DLR, DLR->OFFIS, OFFIS->Jülich, Jülich->OFFIS) 3 mal durchgeführt. Dabei wurden die benötigte Zeit für einen Transfer sowie die von GridFTP gemessene mittlere Durchsatzrate protokolliert und am Ende pro Transferweg ein Mittelwert gebildet. Der anschließend errechnete „Speedup“ gibt den Beschleunigungsfaktor im Verhältnis zu einem Datenkanal an. Ein paar Werte wurden als Ausreißer aus der Wertung genommen (vermutlich eine kurzzeitig sehr stark ausgelastete Leitung), weil sie sonst die Mittelwerte zu stark beeinflusst hätten. Im Prinzip können in der Realität insbesondere durch ausgelastete Leitungen immer Abweichungen von den (unter „optimalen“ Bedingungen gemessenen) Mittelwerten auftreten. Von den insgesamt 180 durchgeführten Transfers schlugen zwei fehl. Alle Transfers waren so genannte Third-Party-Transfers, die von einem OFFIS-Rechner aus initiiert wurden.

In einem weiteren Test wurden die Transferwege OFFIS->DLR, DLR->OFFIS, OFFIS->Jülich, Jülich->OFFIS nochmals durchlaufen, weil der OFFIS-Cluster eine Gigabit-Anbindung bekommen hatte (vorher waren es 100MBit).

Auf weitere Tests wurde verzichtet, weil auf Basis der Ergebnisse die Performanz von GridFTP gut eingeschätzt werden konnte. Im Wesentlichen ging es bei diesem Test darum, herauszufinden, welche Übertragungsraten die einzelnen Verbindungswege ermöglichen und welche Anzahl paralleler Datenkanäle dabei empfohlen werden kann. Weitere Tests könnten z.B. die Übertragung kleinerer Dateien beinhalten oder die Nutzung von mehreren Daten-Knoten.

Erfahrungen mit Grid-Technologien im Projekt WISENT (AP 1.2)

Die folgenden Tabellen enthalten die protokollierten Durchsätze und Beschleunigungsfaktoren in Abhängigkeit von der Anzahl der Datenkanäle. In den angefügten Abbildungen ist die Abhängigkeit zwischen der Anzahl der Datenkanäle und dem (mittleren) Durchsatz dargestellt bzw. der Vergleich von Upload und Download für eine Verbindung.

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	107,108	9,78	105,278	9,93	109,867	9,5	107,42	9,74	
2	79,298	13,3	82,927	12,67	90,231	11,68	84,15	12,55	1,29
3	72,76	14,52	66,217	16,03	68,345	15,44	69,11	15,33	1,57
4	60,525	17,56	66,46	15,98	70,836	14,97	65,94	16,17	1,66
5	60,356	17,59	64,176	16,6	68,866	15,38	64,47	16,52	1,7
6	61,017	17,44	74,592	14,22	81,568	12,95	72,39	14,87	1,53
7	59,457	17,87	62,702	16,98	68,574	15,47	63,58	16,77	1,72
8	60,024	17,72	62,166	17,1	76,225	13,8	66,14	16,21	1,66
9	60,942	17,44	86,539	12,15	64,814	16,41	70,77	15,33	1,57
10	66,934	15,85	55,315	19,39	59,887	17,81	60,71	17,68	1,82

Tabelle 1: Datenübertragungstest DLR->Jülich am 02.08.2007

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	82,244	12,74	74,61	15,13	77,803	14,42	78,22	14,1	
2	56,642	19,07	55,501	20,28	53,676	20,77	55,27	20,04	1,42
3	49,025	22,6	42,529	26,39	44,12	26,26	45,22	25,08	1,78
4	43,008	26,39	43,282	26,6	42,41	27,38	42,9	26,79	1,9
5	40,904	29,26	40,197	28,6	48,99	24,91	43,36	27,59	1,96
6	48,979	23,98	40,998	28,76	40,733	29,2	43,57	27,31	1,94
7	45,071	26,67	41,489	29,17	39,203	29,26	41,92	28,37	2,01
8	41,906	29,17	43,895	27,9	38,999	29,68	41,6	28,92	2,05
9	40,006	29,26	42,726	29,34	41,265	29,68	41,33	29,43	2,09
10	41,482	28,37	44,569	28,21	42,956	26,06	43	27,55	1,95

Tabelle 2: Datenübertragungstest Jülich->DLR am 02.08.2007

DLR -> Jülich

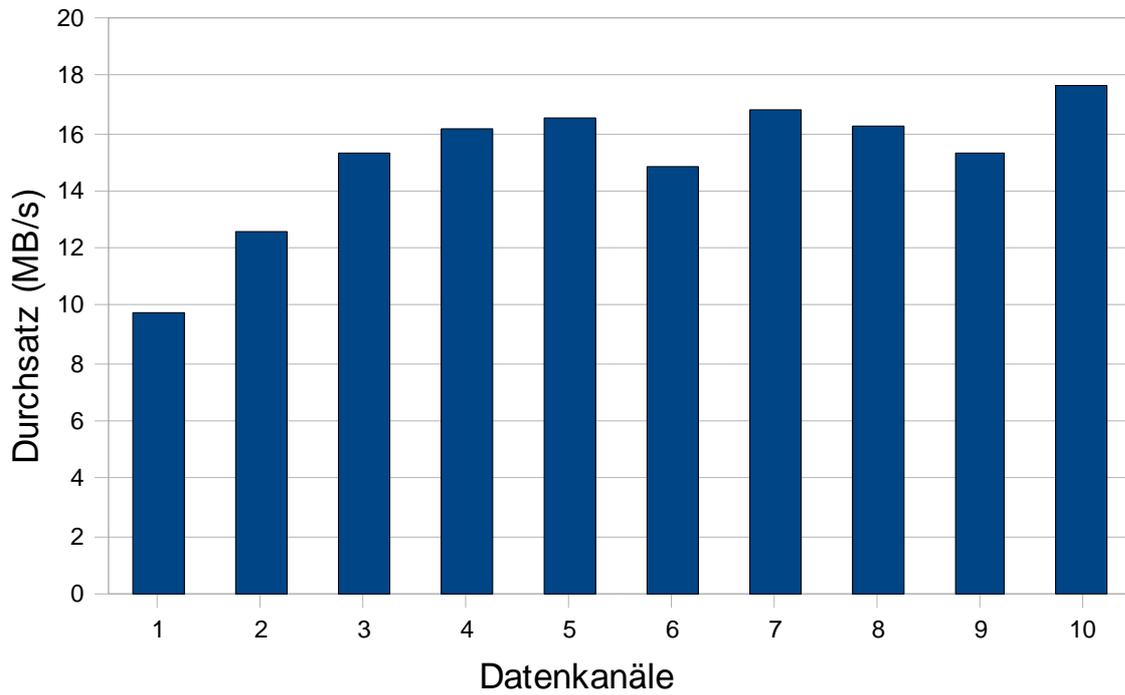


Abbildung 1: Datenübertragungstest DLR->Jülich

Jülich -> DLR

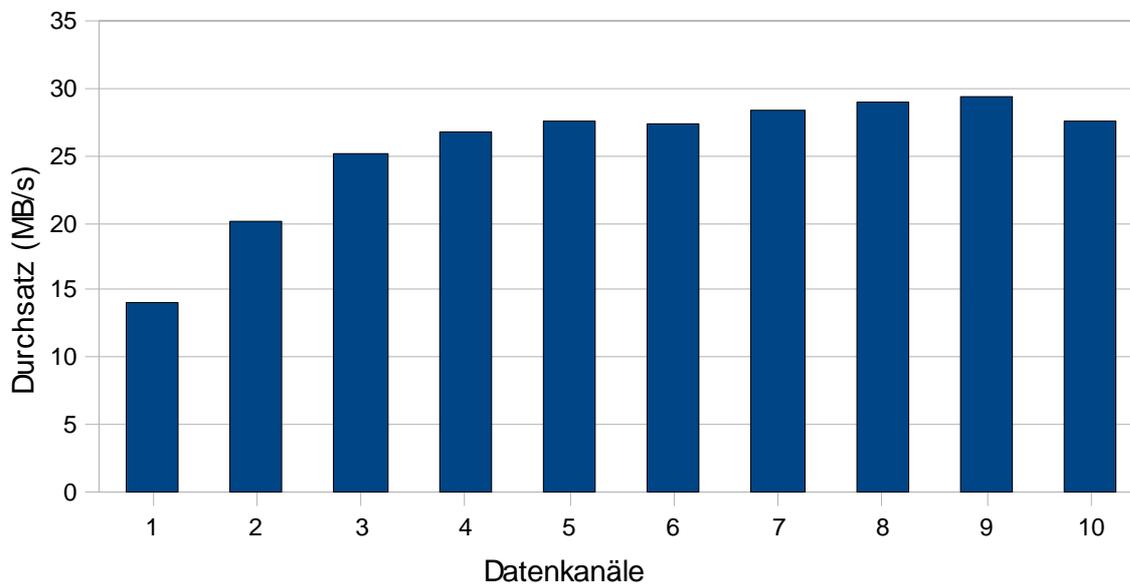


Abbildung 2: Datenübertragungstest Jülich->DLR

DLR <-> Jülich

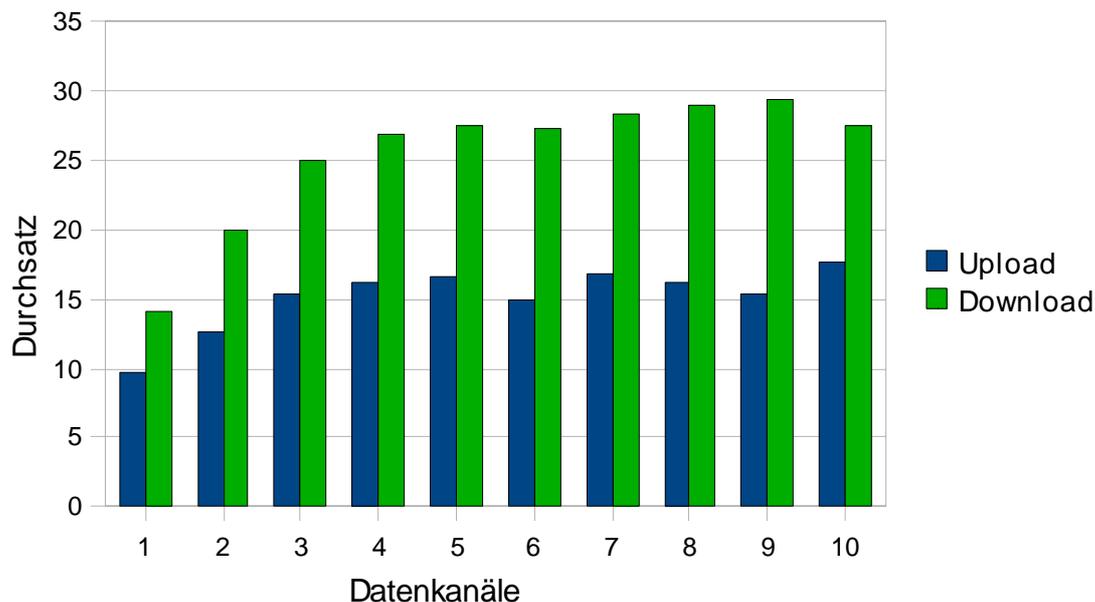


Abbildung 3: Vergleich Upload/Download

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	414,693	2,48	426,513	2,41	418,028	2,47	419,74	2,45	
2	208,291	4,98	200,965	4,69	213,19	4,85	207,48	4,84	1,98
3	153,392	6,87	169,916	6,16	152,562	6,81	158,62	6,61	2,7
4	119,971	8,74	125,537	8,33	125,821	8,28	123,78	8,45	3,45
5	107,276	9,87	106,953	9,89	105,083	9,95	106,44	9,9	4,04
6	101,868	10,26	103,926	10,13	106,411	9,83	104,07	10,07	4,11
7	101,016	10,49	100,311	10,51	101,422	10,32	100,92	10,44	4,26
8	97,158	10,88	97,004	10,89	100,771	10,5	98,31	10,76	4,39
9	97,654	10,82	98,561	10,66	99,563	10,64	98,59	10,71	4,37
10	98,943	10,73	100,569	10,58	98,645	10,63	99,39	10,65	4,35

Tabelle 3: Datenübertragungstest OFFIS->DLR am 03.08.2007

Erfahrungen mit Grid-Technologien im Projekt WISENT (AP 1.2)

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	897,476	1,15	590,05	1,75	529,404	1,88	672,31	1,59	
2	842,815	1,22	356,942	2,89	307,54	3,55	502,43	2,55	1,6
3	511,44	2,02	301,986	3,43	227,851	4,66	347,09	3,37	2,12
4	206,284	5,03	205,436	5,05	184,214	5,89	198,64	5,32	3,35
5	168,031	6,19	154,58	6,9	145,245	7,05	155,95	6,71	4,22
6	145,782	7,24	161,604	6,42	148,075	7,35	151,82	7	4,4
7	137,786	7,87	127,025	8,22	129,839	8,49	131,55	8,19	5,15
8	123,659	8,52	117,562	9,1	116,774	8,87	119,33	8,83	5,55
9	111,17	9,43	108,635	9,73	104,012	10,73	107,94	9,96	6,26
10	107,375	9,82	107,158	9,93	108,013	10,81	107,52	10,19	6,41

Tabelle 4: Datenübertragungstest OFFIS->DLR am 21.02.2008, mit Gigabit Ethernet

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	1886,14	0,54	1740,87	0,59	1779,76	0,58	1802,26	0,57	
2	995,024	1,03	986,066	1,04	982,825	1,04	987,97	1,04	1,82
3	676,439	1,52	703,284	1,46	719,451	1,43	699,72	1,47	2,58
4	587,402	1,75	562,21	1,83	567,164	1,81	572,26	1,8	3,16
5	460,131	2,23	463,629	2,22	460,359	2,23	461,37	2,23	3,91
6	410,08	2,51	420,214	2,45	427,471	2,41	419,25	2,46	4,32
7	406,521	2,54	382,219	2,69	363,176	2,83	383,97	2,69	4,72
8	371,15	2,78	353,607	2,91	328,256	3,14	351	2,94	5,16
9	331,983	3,1	318,283	3,23	312,415	3,3	320,89	3,21	5,63
10	299,991	3,44	289,47	3,44	287,211	3,59	292,22	3,49	6,12

Tabelle 5: Datenübertragungstest DLR->OFFIS am 03.08.2007

Erfahrungen mit Grid-Technologien im Projekt WISENT (AP 1.2)

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	448,703	2,32	462,906	2,23	461,927	1,48	457,85	2,01	
2	255,204	4,07	264,824	3,93	266,192	2,59	262,07	3,53	1,76
3	179,288	5,79	191,219	5,46	167,527	3,35	179,34	4,87	2,42
4	168,03	6,19	153,163	6,79	154,59	4,39	158,59	5,79	2,88
5	144,374	7,22	139,291	7,53	133,695	5,06	139,12	6,6	3,28
6	127,775	8,16	133,186	7,94	109,803	5,02	123,59	7,04	3,5
7	144,133	7,56	118,233	8,8	140,33	5,19	134,23	7,18	3,57
8	117,461	8,97	109,068	10,36	119,192	4,73	115,24	8,02	3,99
9	128,597	8,32	73,693	14,34	93,321	5,14	98,54	9,27	4,61
10	105,769	10,19	102,452	10,76	95,444	5,54	101,22	8,83	4,39

Tabelle 6: Datenübertragungstest DLR->OFFIS am 21.02.2008, mit Gigabit Ethernet

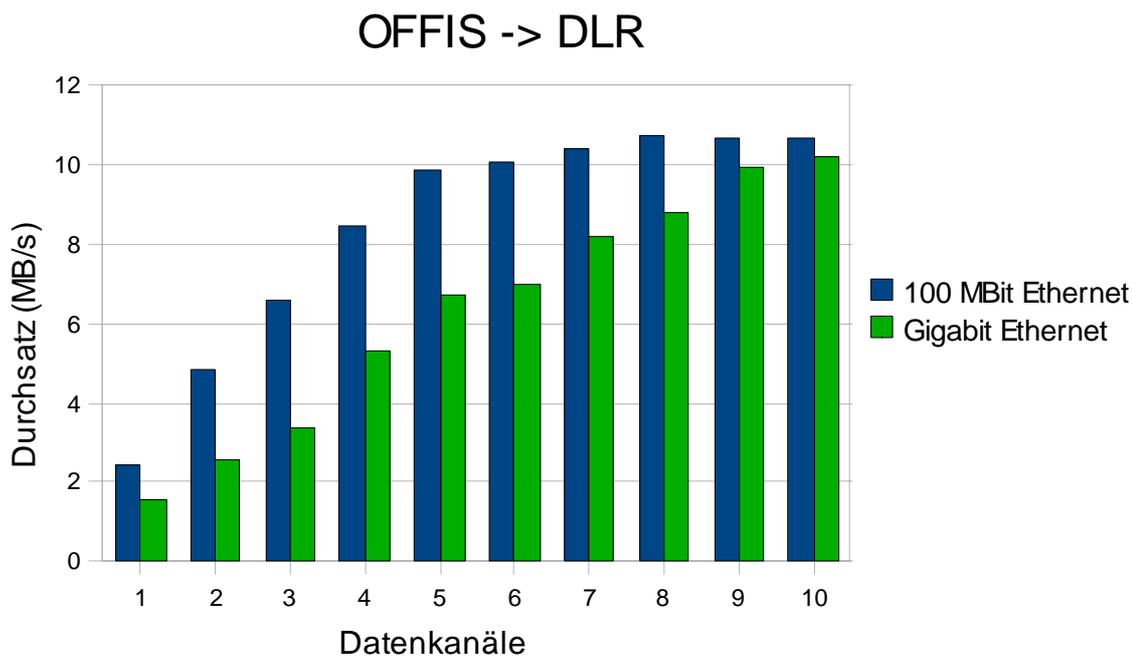


Abbildung 4: Datenübertragungstest OFFIS->DLR

DLR -> OFFIS

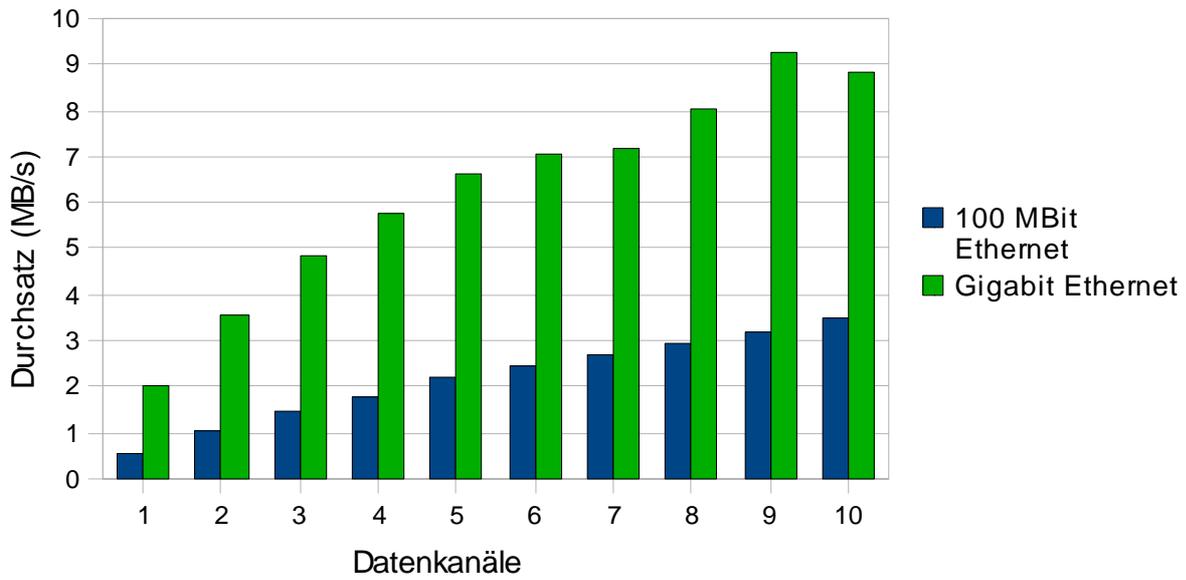


Abbildung 5: Datenübertragungstest OFFIS->DLR

OFFIS <-> DLR 100 MBit Ethernet

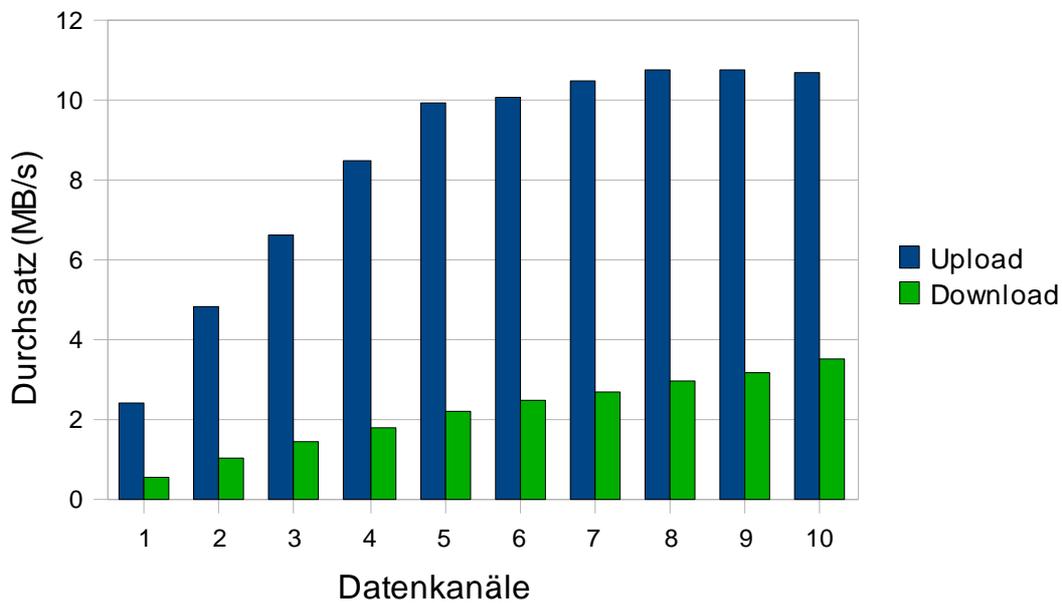


Abbildung 6: Vergleich Upload/Download 100 MBit Ethernet

OFFIS <-> DLR Gigabit Ethernet

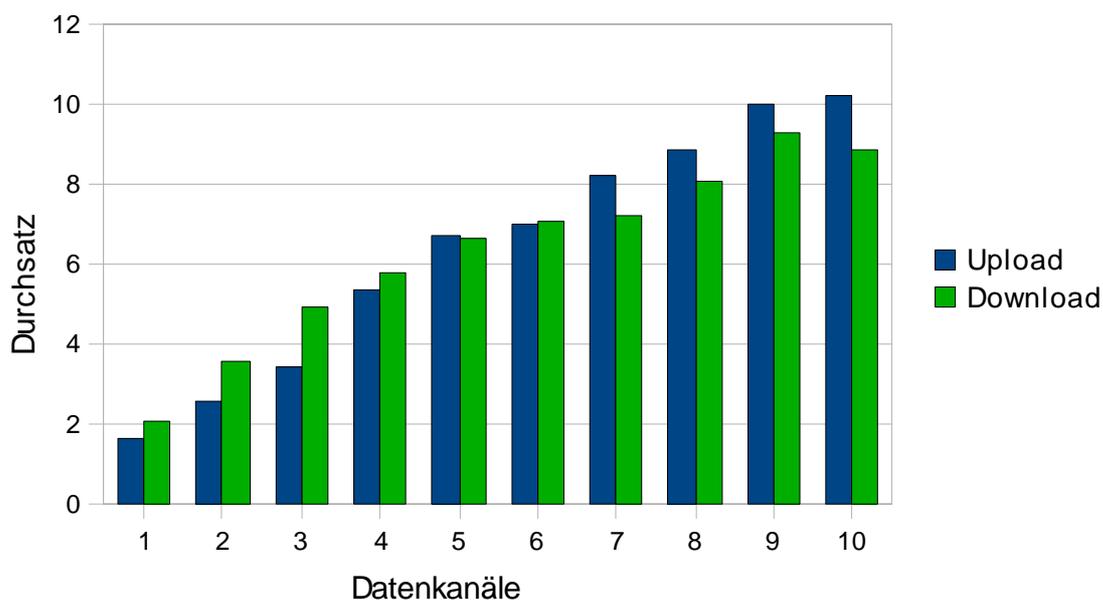


Abbildung 7: Vergleich Upload/Download Gigabit Ethernet

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	446,296	2,3	434,99	2,36	442,128	2,33	441,14	2,33	
2	226,008	4,56	228,887	4,5	373,66	2,76	276,19	3,94	1,69
3	158,425	6,52	156,937	6,58	160,031	6,45	158,46	6,52	2,8
4	124,734	8,29	123,341	8,39	127,227	8,15	125,1	8,28	3,55
5	Failed	Failed	108,677	9,55	112,029	9,25	110,35	9,4	4,03
6	111,791	9,72	108,445	9,56	109,455	9,47	109,9	9,58	4,11
7	106,664	9,72	109,725	9,46	108,977	9,52	108,46	9,57	4,11
8	106,752	9,72	108,137	9,59	110,85	9,37	108,58	9,56	4,1
9	106,757	9,72	*175,89	*5.87	110,985	9,35	108,87	9,54	4,09
10	106,571	9,76	*268,32	*3.87	108,667	9,55	107,62	9,66	4,15

Tabelle 7: Datenübertragungstest OFFIS->Jülich am 06.08.2007
(* Ausreisser, nicht gewertet)

Erfahrungen mit Grid-Technologien im Projekt WISENT (AP 1.2)

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	822,386	1,25	457,125	2,25	671,458	1,53	650,32	1,68	
2	697,589	1,47	237,427	4,35	366,055	2,81	433,69	2,88	1,71
3	364,469	2,85	163,045	6,36	166,192	6,22	231,24	5,14	3,06
4	129,088	8,02	129,763	7,98	132,473	7,82	130,44	7,94	4,73
5	104,57	9,92	105,042	9,88	107,107	9,81	105,57	9,87	5,88
6	94,246	11,02	89,973	11,56	91,867	11,31	92,03	11,3	6,73
7	89,869	11,57	79,711	13,08	82,529	12,63	84,04	12,43	7,4
8	*223,50	*4,67	71,094	14,69	77,124	13,54	74,11	14,11	8,4
9	*343,34	*3	66,811	15,68	70,496	14,84	68,65	15,26	9,08
10	63,27	16,57	62,762	16,7	62,144	16,87	62,73	16,71	9,95

Tabelle 8: Datenübertragungstest OFFIS->Jülich am 20.02.2008, mit Gigabit Ethernet (* Ausreisser, nicht gewertet)

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	463,33	2,22	445,143	2,31	467,254	2,2	458,58	2,24	
2	237,029	4,35	234,111	4,4	229,5	4,49	233,55	4,41	1,97
3	166,346	6,21	168,003	6,15	180,857	5,71	171,74	6,02	2,69
4	129,069	8,03	131,497	7,87	136,327	7,59	132,3	7,83	3,5
5	135,961	7,61	122,801	8,43	137,604	7,52	132,12	7,85	3,5
6	129,553	7,99	123,378	8,4	117,461	8,83	123,46	8,41	3,75
7	119,487	8,68	123,024	8,43	118,32	8,76	120,28	8,62	3,85
8	117,113	8,86	119,175	8,71	116,631	8,9	117,64	8,82	3,94
9	113,821	9,1	113,365	9,14	Failed	Failed	113,59	9,12	4,07
10	110,783	9,37	*149,19	*6,93	115,808	9,18	113,3	9,27	4,14

Tabelle 9: Datenübertragungstest Jülich->OFFIS am 06.08.2007 (* Ausreisser, nicht gewertet)

Erfahrungen mit Grid-Technologien im Projekt WISENT (AP 1.2)

Anzahl Datenkanäle	Durchgang 1 Zeit (s)	Durchgang 1 Durchsatz (MB/s)	Durchgang 2 Zeit (s)	Durchgang 2 Durchsatz (MB/s)	Durchgang 3 Zeit (s)	Durchgang 3 Durchsatz (MB/s)	Durchschnitt Zeit (s)	Durchschnitt Durchsatz (MB/s)	Speedup Durchsatz
1	446,836	2,34	434,612	2,37	449,39	2,29	443,61	2,33	
2	223,014	4,63	220,878	4,67	216,157	4,78	220,02	4,69	2,01
3	144,763	7,16	150,254	6,91	145,512	7,12	146,84	7,06	3,03
4	113,933	9,44	118,562	8,81	112,103	9,53	114,87	9,26	3,97
5	86,626	12,08	89,005	11,8	88,374	12,08	88	11,99	5,15
6	82,56	14,01	83,164	13,91	75,333	13,95	80,35	13,96	5,99
7	67,382	16,25	71,739	16,28	78,519	14,95	72,55	15,83	6,79
8	68,084	18,89	56,47	18,75	82,207	19,25	68,92	18,96	8,14
9	51,021	21,38	55,869	21,16	85,103	16,9	64	19,81	8,5
10	45,945	23,22	66,997	23,12	89,702	23,49	67,55	23,28	9,99

Tabelle 10: Datenübertragungstest Jülich->OFFIS am 20.02.2008, mit Gigabit Ethernet

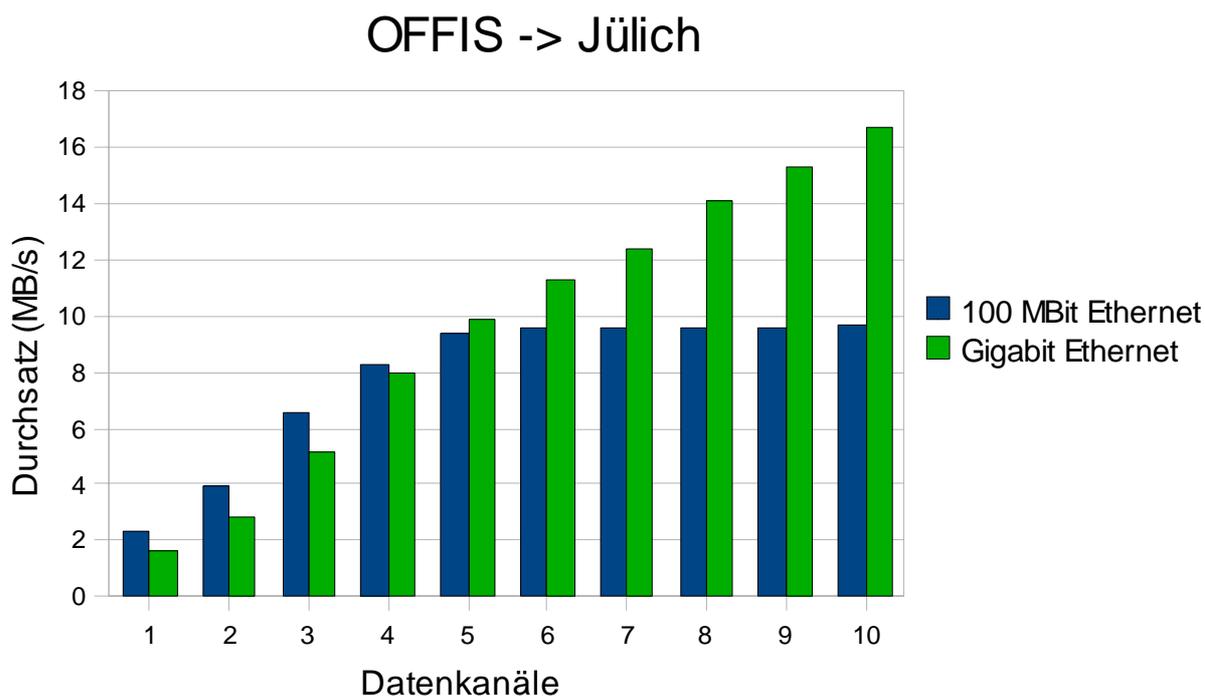


Abbildung 8: Datenübertragungstest OFFIS->Jülich

Jülich -> OFFIS

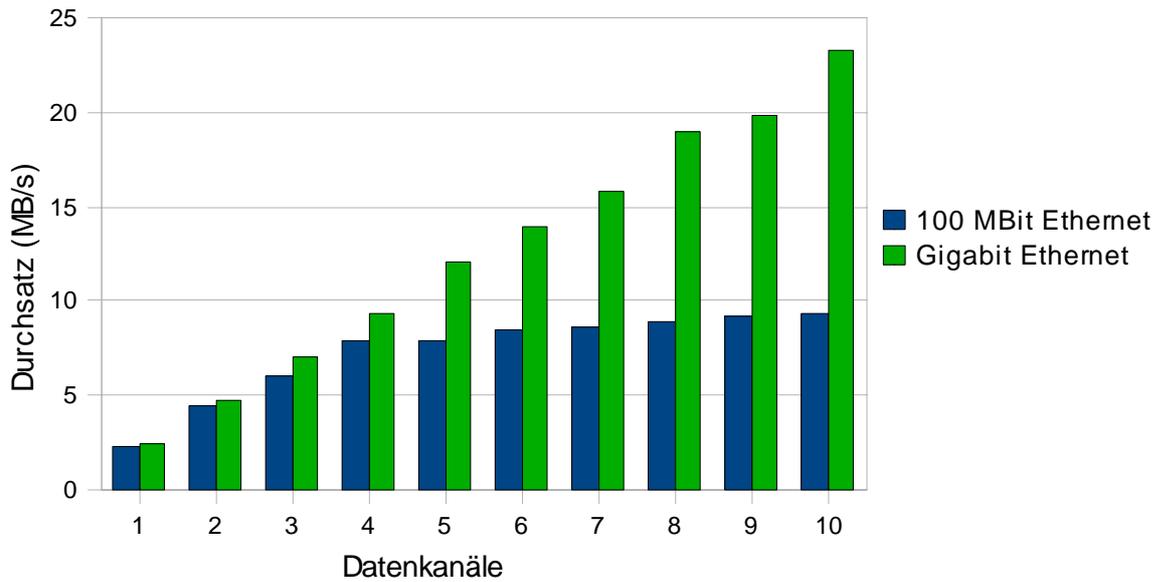


Abbildung 9: Datenübertragungstest Jülich->OFFIS

OFFIS <-> Jülich 100 MBit Ethernet

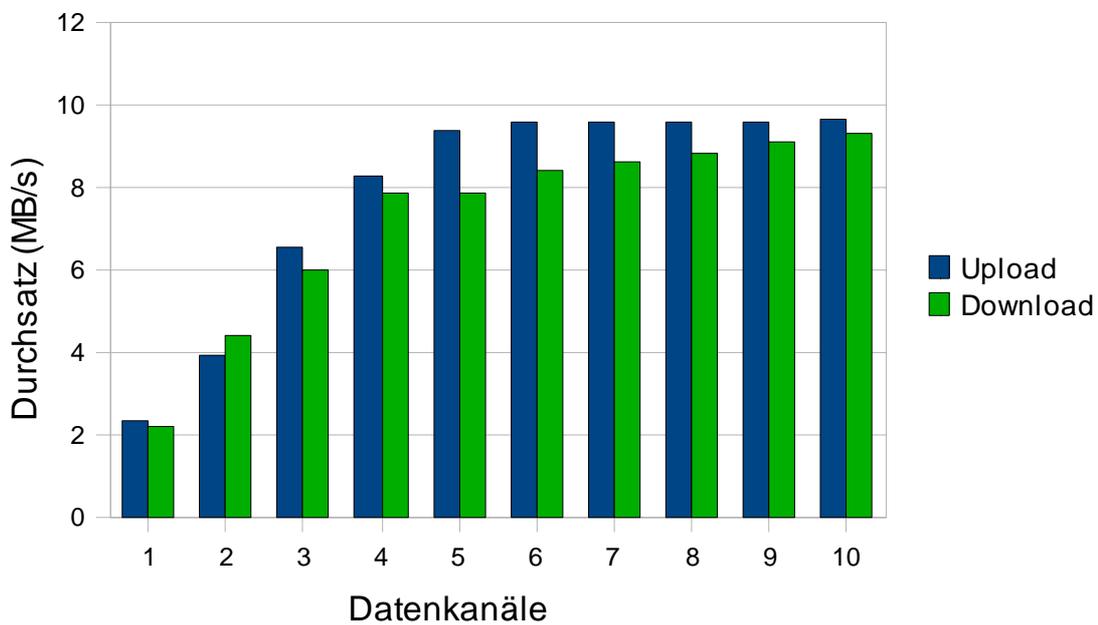


Abbildung 10: Vergleich Upload/Download 100 MBit Ethernet

OFFIS <-> Jülich Gigabit Ethernet

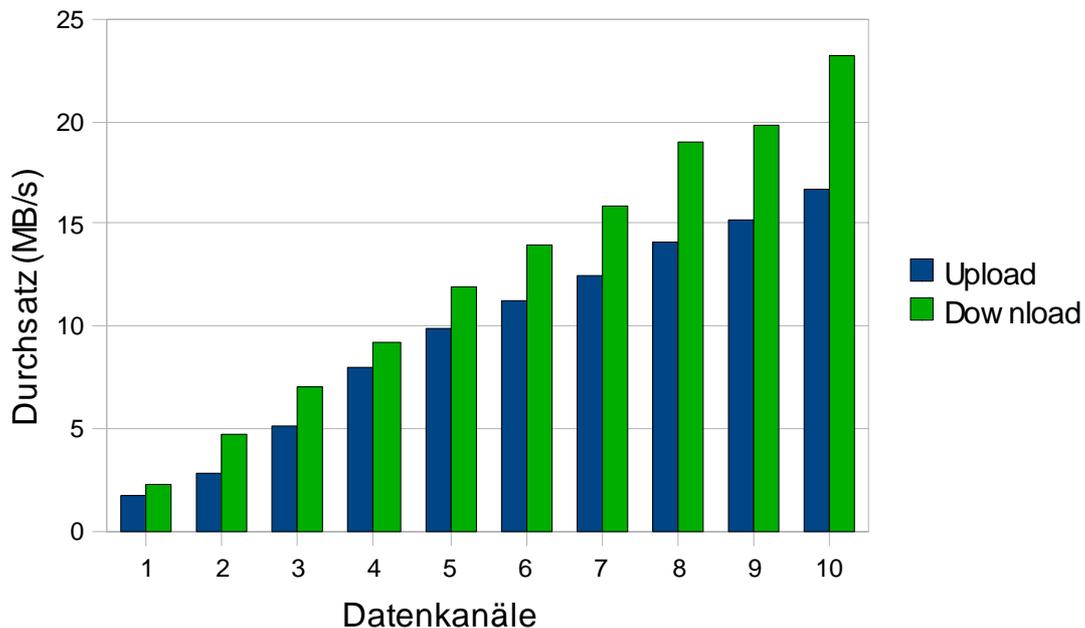


Abbildung 11: Vergleich Upload/Download Gigabit Ethernet

An den Tabellen und insbesondere an den Abbildungen ist zu erkennen, dass jeder Transferweg ein unterschiedliches, aber eindeutiges Bild zeigt.

- DLR<->Jülich: Bei beiden Richtungen ist zu erkennen, dass ein einzelner Datenkanal schon einen guten Durchsatz erreicht, ca. 10 MB/s bei der Richtung DLR->Jülich (siehe Abbildung 1) und ca. 14 MB/s bei der Richtung Jülich->DLR (siehe Abbildung 2). Bei weiteren Datenkanälen zeigt sich, dass der Gesamtdurchsatz nur mäßig steigt und ab ca. 5 parallelen Datenkanälen keine wesentliche Steigerung (zum Teil schwankend und rückläufig) mehr zu erkennen ist. In Abbildung 3 ist zu sehen, dass die Richtung Jülich->DLR deutlich schneller ist als die Richtung DLR->Jülich.
- OFFIS<->DLR: Diese Verbindung zeigt es etwas differenziertes Bild, was u.a. mit der unterschiedlichen Auslastung der Internet-Verbindung beispielsweise beim DLR zusammenhängen kann. Bei der Richtung OFFIS->DLR (siehe Abbildung 4) ist zu erkennen, dass bei der 100MBit-Leitung die Durchsatzrate schneller ansteigt als bei der Gigabit-Leitung, bei 10 parallelen Datenkanälen aber bei beiden Leitungen ungefähr derselbe Durchsatz von ca. 10 MB/s erreicht wird. Es ist zu vermuten, dass bei der Gigabit-Leitung noch mehr Durchsatz bei mehr als 10 Datenkanälen zu erreichen ist, bei der 100MBit-Leitung ist bei 5 parallelen Datenkanälen die physikalische Grenze (theoretisch 12,5 MB/s) der Leitung erreicht. Bei der anderen Richtung DLR->OFFIS (siehe Abbildung 5) scheint beim Test mit der 100MBit-Leitung ein Teil der Verbindungsstrecke (vermutlich beim DLR) ausgelastet gewesen zu sein oder es gab zu dem Zeitpunkt eine unterschiedliche Strategie bzgl. der erlaubten Durchsatzes beim Upload pro Verbindung. Die Gigabit-Leitung ist jedenfalls deutlich schneller und kann vermutlich bei mehr als 10 parallelen Datenkanälen noch zulegen. Die 100MBit-Leitung hat dann bei ca. 10 MB/s ihr physikalisches Maximum erreicht. In der Abbildung 6 ist nochmal zu sehen, dass bei der 100MBit-Leitung die Richtung

OFFIS->DLR deutlich schneller ist als die Richtung DLR->OFFIS. Bei der Gigabit-Leitung sind beide Richtungen ziemlich ausgeglichen, siehe Abbildung 7.

- OFFIS<->Jülich: Bei dieser Verbindung ist ein klares Bild zu erkennen. In beiden Richtungen (siehe Abbildungen 8 und 9) ist ab ca. 4-5 parallelen Datenkanälen die physikalische Grenze bei der 100MBit-Leitung erreicht. Bei der Gigabit-Leitung gibt es bis 10 parallelen Datenkanälen einen nahezu linearen Anstieg des Durchsatzes, der Durchsatz skaliert somit sehr gut. Bei weiteren kurzen Tests konnte mit mehr als 10 parallelen Datenkanälen ein Durchsatz von über 30MB/s erreicht werden, dort liegt auch die Grenze. In Abbildung 10 ist zu sehen, dass bei der 100MBit-Leitung die Richtung OFFIS->Jülich etwas schneller ist als die Richtung Jülich->OFFIS, aber im Prinzip befanden sich beide Richtungen an der physikalischen 100MBit-Grenze, ggf. war bei der Richtung Jülich->OFFIS die Leitung schon etwas ausgelastet. In Abbildung 11 ist zu hingegen zu erkennen, dass die Richtung Jülich->OFFIS schneller ist als die Richtung OFFIS->Jülich.

Abschließend kann festgehalten werden, dass mit fünf parallelen Datenkanälen für eigentliche alle Richtungen „optimale“ Durchsatzraten erreicht werden können, je nachdem, was die Leitung hergibt. Es hat sich gezeigt, dass lange Zeit die 100MBit-Anbindung des Clusters beim OFFIS ein Flaschenhals war und nach Aufrüstung eines Gigabit-Anschlusses zumindest mit Jülich deutliche bessere Ergebnisse erzielt werden konnten. Das OFFIS selbst ist über die Universität Oldenburg mit einem Gigabit-Anschluss mit dem DFN verbunden.

4.1.2 Erfahrungen bei DLR DFD

Auf Seiten des DLR-DFD bestand die größte Herausforderung bei der Einrichtung eines GridFTP-Server in dem Umgang mit den strengen Firewall-Richtlinien. Dies wurde bereits ausführlich in dem Dokument „AP 2.1 - Bericht zu den Auswirkungen der DLR-IT-Sicherheitsstrukturen beim Datentransfer mit Partnern außerhalb des DLR“ beschrieben. Zusammenfassend ist die Situation so, dass es internes DLR-Netz gibt und daneben eine DMZ (demilitarized zone – entmilitarisierte Zone). Verbindungen von außen in das interne DLR-Netz sind gänzlich verboten und vom DLR-Netz nach außen ebenfalls stark eingeschränkt. Die DLR-DMZ ist typischerweise etwas lockerer, d.h. es werden auch Verbindungen von außen zugelassen. Verbindungen von der DLR-DMZ in das interne DLR-Netzwerk sind wiederum sehr restriktiv bzw. eigentlich verboten. Im Endeffekt bedeutet dies, dass für die bidirektionale Kommunikation nach außen die DLR-DMZ zuständig ist und nach Möglichkeit Verbindungen zwischen DLR-DMZ und dem internen DLR-Netz vom internen DLR-Netz aus initiiert werden sollten, um die DLR-DMZ gut vom internen DLR-Netz zu isolieren. Prinzipiell gesehen sind auch Ausnahmen möglich, aber die Anforderungen von GridFTP an geöffneten Ports lässt nur ein Einsatz in der DLR-DMZ zu. Neben dem statischen Port für Kontrollverbindungen 2811 werden mehrere dynamische Ports für Datenverbindungen benötigt. Dieser Bereich kann zwar eingeschränkt werden, im D-Grid wird der Bereich 20000-25000 empfohlen, ist aber nur mit der DLR-DMZ verträglich. Um dennoch Daten mit dem internen DLR-Netz austauschen zu können, wurde zwischen dem Rechner *eridanus* in der DLR-DMZ und einer Speichereinheit (beides D-Grid Sonderinvestition) im internen DLR-Netz ein Synchronisationsmechanismus (*r_{sync}*) eingerichtet, der periodisch initiiert aus dem DLR-Netz heraus einen Bereich des Dateisystems von *eridanus* auf die Speichereinheit synchronisiert und dies für einen weiteren Bereich genau umgekehrt durchführt. So können zwischen DLR-DMZ und dem internen DLR-Netz bidirektional Daten ausgetauscht werden. Diese Technik wurde u.a. auch zur Umsetzung des MSG-Tailor-Web Services

verwendet, siehe Dokument „AP3.3 - Einsatz von und Erfahrungen mit Web Service-Technologien“.

Die Installation und Einrichtung von Globus Toolkit 4 bzw. GridFTP auf *eridanus* konnte nach der üblichen Anleitung durchgeführt werden, nachdem zuvor Betriebssystembedingte Problem mit Sun Solaris 9 behoben wurde, siehe Abschnitt 3.1. Lediglich bei der Konfiguration musste darauf geachtet werden, dass die beiden Netzwerkkarten richtig unterstützt werden. Die erste Netzwerkkarte hat eine interne IP-Adresse und ist nur von dem internen DLR-Netz erreichbar. Die zweite Netzwerkkarte hat eine interne IP-Adresse für die DLR-DMZ, die aber nach außen hin einer globalen IP-Adresse zugeordnet wird. Daher werden auf *eridanus* zwei GridFTP-Server gestartet, jeweils einer für jede Netzwerkkarte. Für die externe Netzwerkkarte wurde der GridFTP-Server so konfiguriert, dass die externe IP-Adresse verwendet wird. Für die interne Netzwerkkarte die entsprechende interne IP-Adresse. Danach war der „externe“ GridFTP-Server einsatzbereit und hat alle funktionalen Tests bestanden. Es wurden auch verschiedene Durchsatzraten getestet, siehe Abschnitt 4.1.1. Der „interne“ GridFTP-Server wurde nicht getestet, weil der Synchronisationsmechanismus zum Datenaustausch mit dem internen DLR-Netz zur Zeit ausreichend ist. Prinzipiell sollte die Richtung internes DLR-Netzwerk → DLR-DMZ funktionieren, die umgekehrte Richtung ist aus den bereits erwähnten Firewall-technischen Einschränkungen nicht möglich.

4.1.3 Erfahrungen an der Universität Oldenburg

In der Arbeitsgruppe Energiemeteorologie an der Universität Oldenburg wurde die Ausrichtung des AP 3.2 von einer eher passiven Beteiligung an GridFTP-Transfers zwischen dem Datenzentrum am DLR-DFD und der Arbeitsgruppe hin zu einer Evaluierung der Einsatzmöglichkeiten von GridFTP und dCache im Zusammenhang mit umfangreichen Grid-Rechnungen verschoben und erweitert.

Während der Beschäftigung mit GridFTP wurde insbesondere Wert auf einen sicheren Betrieb des eigenen GridFTP-Servers gelegt. Daher wurde ein Sicherheitskonzept erstellt und umgesetzt. Da die Verwendung von GridFTP im Rahmen dieses AP von Anfang an vorgesehen war, wurde bereits frühzeitig ein GridFTP-Server testweise auf einem relativ isolierten System installiert. Einfache Transfertests belegten die Funktionalität, allerdings waren jeweils zusätzliche Verschiebungen der Daten per `scp` notwendig, weil der Server keinen Zugriff auf die Netzlaufwerke hatte.

Die standardmäßig vorgeschlagene Installation des GridFTP-Servers sieht eine direkte Einbindung ins Dateisystem des Servers vor und regelt alle Zugriffsrechte über die entsprechenden UNIX-Mechanismen. Für dedizierte Grid-Maschinen ist dies eine sinnvolle und ausreichend sichere Einstellung, zumal der Nutzerkreis durch die Abhängigkeit von gültigen Zertifikaten sehr begrenzt ist. Alle diese Nutzer erhalten a priori vergleichbare Rechte, insbesondere der Lesezugriff ist normalerweise sehr umfassend.

Wird allerdings ein an ein internes Network File System (NFS) angebundenes System zum GridFTP-Server aufgerüstet, erhalten alle durch ein (Grid-)Benutzer-Zertifikat zugelassenen Nutzer ggf. Einblick in weitere Verzeichnisse der lokalen Nutzer, in Betriebssystem-spezifische Informationen oder in eingebundene Datenarchive. Denn zur Zeit werden vom GridFTP-Server des GT4 keine „Sandboxen“ unterstützt, wie es bei einigen FTP-Server der Fall ist. Die Nutzergruppen-basierte Verwaltung des Zugriffs über standardmäßige Verzeichnis- und Dateirechte oder auch über Access Control Lists (ACL) würde ein so hohes Maß an Disziplin und Kontrolle erfordern, so dass dieser Ansatz in der Arbeitsgruppe schnell verworfen wurde.

Die Anforderungen an eine sichere und leicht nutzbare Auslegung des GridFTP-Servers wurden zusammengetragen:

- Mitglieder der Gruppe Energiemeteorologie sollen direkt auf Verzeichnisse und Dateien im internen NFS zugreifen können.
- Mitglieder der Gruppe Energiemeteorologie sollen die Funktionalität von GridFTP nutzen können, ohne sich auf einem bestimmten System anmelden zu müssen.
- Nur Mitglieder der virtuellen Organisation WISENT sollen den GridFTP-Server nutzen können.
- Mitglieder von WISENT, die nicht zur Gruppe Energiemeteorologie gehören, sollen nur ein „public“-Verzeichnis sehen und für Dateitransfers nutzen können

Ausgehend von diesen Anforderungen wurde in Zusammenarbeit mit dem OFFIS ein Betriebskonzept entwickelt:

- Der GridFTP-Server wird auf einem operationellen, ins NFS eingebundenen Server installiert.
- Das Installationsverzeichnis liegt unterhalb einer `chroot`-Umgebung, in welche die notwendigen Systemverzeichnisse hineinkopiert oder per `mount` eingebunden werden. Der Rest des Dateisystems ist aus diesem Verzeichnisbaum heraus nicht erreichbar.
- Alle Mitglieder von WISENT, die über ein Benutzer-Zertifikat verfügen und somit GridFTP nutzen können, werden manuell in das `grid-mapfile` eingetragen und zur lokalen Gruppe `wisent` hinzugefügt.
- Die Gruppe `wisent` hat Lese- und Schreibrechte auf einem Verzeichnis „public“ innerhalb der `chroot`-Umgebung.
- Die Mitglieder der Gruppe Energiemeteorologie werden zusätzlich der Gruppe `condor` hinzugefügt. Die Gruppe wurde im Rahmen der Job-Ausführung mittels Condor bzw. Condor-G eingeführt.
- Die Gruppe `condor` hat Lese- und Schreibrechte auf einem Verzeichnis „restricted“ innerhalb der `chroot`-Umgebung.
- Alle relevanten Netzlaufwerke werden mit der Option „bind“ unterhalb von „restricted“ gemounted und sind so ausschließlich für die Mitglieder der Gruppe `condor` erreichbar.

Die Umsetzung dieser Eckpunkte gestaltete sich als nicht ganz einfach, insbesondere der Betrieb des GridFTP-Servers innerhalb der `chroot`-Umgebung ist eigentlich nicht vorgesehen und erfordert einige Vorarbeiten. Die dabei durchgeführten Arbeiten sind in Anhang A aufgeführt.

Die damit geschaffene Möglichkeit, per GridFTP auf das Dateisystem der Arbeitsgruppe Energiemeteorologie an der Universität Oldenburg zuzugreifen, wurde im Rahmen einer größeren Grid-Rechnung ausgenutzt. Dabei werden weitgehend automatisch Daten aus dem Satellitenbild-Archiv der Arbeitsgruppe auf verschiedene Rechencluster kopiert, dort bearbeitet und die Ergebnisse - Karten der solaren Einstrahlung unter verschiedenen Einfallswinkeln - nach einer ersten Zwischenspeicherung auf einem dCache-Server zwecks Visualisierung wieder ins Dateisystem des Benutzers kopiert.

Der Workflow ist in Abbildung 13 dargestellt und zeigt die notwendigen Schritte ausgehend vom Cloudindex (einer Darstellung des Bewölkungsgrades, aus Satellitenbildern extrahiert) bis hin zu einer Monats- oder Jahreskarte, die örtlich aufgelöst anzeigt, unter welchem Winkel die Einstrahlung am besten mit photovoltaischen Anlagen genutzt werden kann.

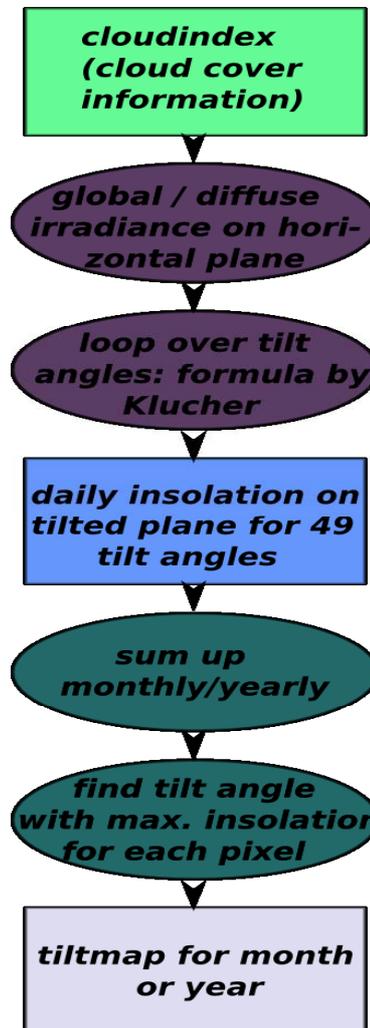


Abbildung 13: Workflow für die Berechnung von Tilffaktoren

Die rechteckigen Elemente in Abbildung 13 stellen jeweils ein Datenobjekt dar, die ovalen Elemente stehen für Berechnungen auf Rechenclustern. Das Zwischenprodukt (daily insolation on tilted planes ...) ist speicherplatzintensiv, da für jeden Tag (zunächst eines Jahres, später sogar von 10 Jahren) und jeden Winkel (bisher 49, später deutlich mehr) eine Karte gespeichert werden muß. Diese Daten wurden zentral im dCache-Archiv in Jülich abgespeichert.

Für die Aufsummierung von Tageskarten zu Monats- und Jahreskarten wurden dann gezielt die benötigten Dateien aus dem dCache-Archiv wieder heruntergeladen.

Sämtliche Transfers wurden mit GridFTP durchgeführt. Dabei wurde der File-Staging-Mechanismus von Condor und Globus nur für die verwendeten Programme benutzt, die Datentransfers wurden innerhalb eines Perl-Skripts ausgeführt, welches von einer zentralen Schleife mit dem zu bearbeitenden Datum versorgt wird.

Schließlich mussten die berechneten Monats- und Jahreskarten nach der Berechnung wieder auf den Arbeitsplatzrechner verschoben werden. Dafür wurden manuell ausgeführte GridFTP-Befehle verwendet.

In Abbildung 14 sind die Datenwege schematisch dargestellt. Die roten Pfeile entsprechen GridFTP-Transfers, mit den schwarzen Pfeilen sind die Grid-Jobs angedeutet.

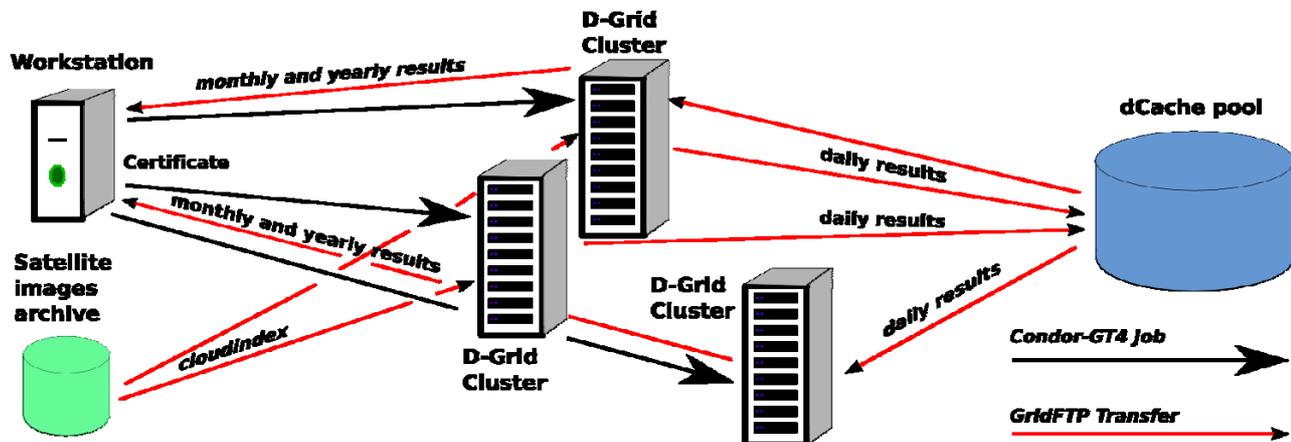


Abbildung 14: Schematische Darstellung der Datenwege für die Berechnung des optimalen Montagewinkels photovoltaischer Anlagen

Auch wenn die endgültige Auswertung der Berechnungen noch nicht abgeschlossen ist, hat sich deutlich abgezeichnet, dass die Durchführung dieser aufwendigen Berechnungen lokal kaum möglich gewesen wäre. Ohne die Möglichkeit der Zwischenspeicherung wäre auch die Bearbeitung im Grid sehr schwierig geworden. Der Einsatz von GridFTP als zuverlässiges und relativ leicht handhabbares Transfer-Werkzeug hat die Umsetzung stark vereinfacht.

Das GridFTP-Konzept hat sich in diesem Kontext sehr bewährt. So war bei gleichzeitiger maximaler Begrenzung des Zugriffs für AG-externe WISENT-Mitglieder möglich, die Ausgangsdaten direkt aus dem Gruppen-eigenen Archiv zu den Rechenclustern zu kopieren, ohne auf Grid-fremde Sonderlösungen wie beispielsweise `secure copy (scp)` ausweichen zu müssen. Damit war insbesondere der enorme Vorteil des `single sign on` erhalten, alle Aktivitäten konnten mit einem einzigen Benutzer-Zertifikat durchgeführt werden.

4.2 dCache

Ein Vorteil der Beteiligung des Projekts WISENT an der D-Grid-Infrastruktur besteht im Zugang zu externen Speicherkapazitäten in den Rechenzentren Jülich, Hannover und Karlsruhe, die über die Datenmanagementsoftware dCache genutzt werden können. Im Unterschied zum lokalen Festplattenspeicher z. B. im Oldenburger Rechencluster oder im Netzwerk der Forschungsgruppe handelt es sich um beträchtliche Kapazitäten (> 40 TB), die zum Teil über Bandspeicher realisiert werden und sich vorzüglich zur mittelfristigen Datenarchivierung eignen. Diese Speicher werden im D-Grid mit Hilfe der Middleware dCache verfügbar gemacht.

WISENT gehört zu den ersten D-Grid-Projekten, die Anwendungsszenarien für angebotene dCache-Installationen untersucht und effektive Lösungen diesbezüglich gefunden haben. Im Folgenden werden die gesammelten Erfahrungen ausführlich dokumentiert. Zusammenfassend wurde in WISENT ein Softwarewerkzeug und eine Vorgehensweise etabliert, die den wissenschaftlichen Anwendern die Auslagerung ihrer großvolumigen Daten (z. B. mehrere hunderte GB) zu externen D-Grid-Speichersystemen

und eine nachträgliche zuverlässige Datenwiederherstellung aus diesen Systemen ermöglicht. So belegte zum Zeitpunkt der Verfassung dieses Berichts WISENT über 3 TB Speicher im Jülicher dCache-Archiv.

dCache bietet mehrere Möglichkeiten (so genannte Doors) an, um auf Daten zuzugreifen. Dazu gehört unter anderem die GridFTP-Door, die prinzipiell mit jedem GridFTP-Client verwendet werden kann. Des Weiteren gibt es die Möglichkeit den Storage Resource Manager (SRM) zu nutzen. Dieser wird über das gleichnamige SRM-Protokoll angesprochen, das im Wesentlichen ein Vermittlungsprotokoll ist. Dabei wird nach einer Dateianfrage (mit dem Befehl `srmcp`) ein Übertragungsprotokoll ausgehandelt, z. B. GridFTP, und der SRM-Dienst gibt eine URL an, worüber die Datei zu beziehen ist. Letzteres dient unter anderem zum Lastausgleich, wenn eine Datei auf mehreren Servern vorhanden ist.

Trotz des erweiterten Funktionsumfangs von SRM gegenüber GridFTP bietet eine direkte Verwendung der GridFTP-Door aus Anwendersicht einige Vorteile. So kann vertraute GridFTP-Software, die konzeptionell einer herkömmlichen FTP-Software entspricht und sich lediglich in nicht-funktionalen Eigenschaften (positiv) unterscheidet, für Übertragungen von/zu dCache verwendet werden. Im Unterschied zu SRM-Clients kann dieselbe Software auch für Übertragungen zwischen einzelnen Rechenclustern eingesetzt werden, in denen weder dCache noch eine andere SRM-Implementierung vorhanden ist (die meisten zum D-Grid gehörenden Rechencluster fallen in diese Kategorie). Darüber hinaus können Datenübertragungen aus einem Globus Toolkit 4 Berechnungsjob im Rahmen des in WS-GRAM integrierten File Staging zurzeit ausschließlich mit GridFTP durchgeführt werden. Eine Übertragung mit SRM könnte zwar (falls die Client-Software am Ausführungsort verfügbar ist) aus einem bereits ausgeführten Job heraus erfolgen, würde aber zur ineffizienten Nutzung von CPU-Ressourcen und/oder zur Verkomplizierung der in WISENT typischerweise vom Anwender vorgenommenen Job-Programmierung führen. Die Vorteile von SRM gegenüber dem direkten Einsatz von GridFTP wären dagegen die Möglichkeit der Platzreservierung vor einer Datenübertragung und der Regelung von Gültigkeitsdauern der gespeicherten Daten. Diese Vorteile spielen jedoch im folgenden Anwendungsszenario eine untergeordnete Rolle.

4.2.1 Anwendungsszenario und Ausgangssituation

Die grundsätzliche Motivation für die Nutzung von dCache in WISENT ergibt sich aus der relativen Speicherplatzknappheit in lokalen Netzwerken der beteiligten Forschungsinstitute und aus der Notwendigkeit der mittelfristigen Datenarchivierung, um zum Beispiel auf Ergebnisse von vorherigen Berechnungen in der Zukunft zurückgreifen zu können, ohne diese nochmal ausführen zu müssen. Damit ergibt sich die Sicht auf dCache-Installationen als „virtuelle Festplatten“, die auf zuverlässige und performante Weise die üblichen Dateisystemoperationen unterstützen sollen. Beispielsweise soll eine Übertragung eines ausgewählten Verzeichnisbaums aus einem Rechencluster zum dCache-Archiv (und umgekehrt) möglich sein. Darüber hinaus soll es möglich sein, die Inhalte eines vorhandenen Archivs ähnlich wie die eines lokalen Dateisystems anzuschauen und den Platzbedarf für die archivierten Dateien vor einer Übertragung zu ermitteln.

Im Laufe des Arbeitspakets stellte sich heraus, dass die einfachen Kommandozeilenwerkzeuge wie `globus-url-copy` oder `srmcp`, deren Merkmale und Verwendungsmöglichkeiten in folgenden Abschnitten noch genauer erläutert werden, für den vorgestellten Zweck nicht ausreichen. Dies ergab die Motivation zur Entwicklung einer neuen Client-Software (siehe Dokument „AP 3.2 – GridFTP-Client“), die im Projektverlauf zum empfohlenen Werkzeug für WISENT-Anwender, die ihre großvolumigen Daten im D-Grid archivieren bzw. wiederherstellen möchten, wurde.

4.2.2 dCache-Performanz

Obwohl wie oben beschrieben aus Anwendersicht die Nutzung von dCache-Servern als „virtuelle Festplatten“ bzw. wie übliche File-Server im LAN wünschenswert ist, die sich von herkömmlichen Dateispeichern so wenig wie möglich unterscheiden sollten, ist in der Praxis dieses Ideal kaum erreichbar. Vielmehr müssen von Anwendern bestimmte aus fachlicher Sicht irrelevante Einschränkungen berücksichtigt werden, die zum einen aus der gegenwärtigen Implementierung von dCache, zum anderen aus inhärenten technischen Merkmalen eines verteilten, auf Bandmedien basierenden Speichersystems resultieren. Insbesondere die Performanz (im Sinne von maximal erreichbaren Datendurchsätzen) wird durch die vom Anwender abhängige Nutzungsweise und nicht nur durch die Netzwerkinfrastruktur stark beeinflusst.

Auf Grund von gesammelten Erfahrungen wurden für WISENT-Anwender folgende Empfehlungen zur Optimierung der Übertragungsperformanz ausformuliert:

1. Die Übertragung von/zu dCache ist schnell (Beispiel: 10-20 MB/s – vergleichbar mit Ergebnissen der allgemeinen GridFTP-Performanztests), wenn die Dateien sich im Pool (d. h. auf der Festplatte des dCache-Systems und nicht auf Band) befinden oder bei Übertragung zum dCache ausreichend Platz im Pool vorhanden ist, um sie aufzunehmen. Beispielsweise ist das Jülicher Archiv so konfiguriert, dass das kritische Datenvolumen 600 GB beträgt, wobei sich diese Datenmenge auf mehrere parallele Nutzer verteilen könnte. Nach der Überschreitung dieser Menge lässt der Datendurchsatz dramatisch nach.
2. Die Übertragung von/zu dCache ist langsam (Beispiel: 1 MB/s), wenn Dateien auf Band geschrieben oder von Band gelesen werden müssen. Man muss in diesem Fall entsprechend viel Geduld aufbringen.
3. Da das Bandsystem pro Datei eine Anlaufzeit von ca. 30 Sekunden hat, ist die Verwendung von Dateigrößen ≥ 1 GB empfohlen – es lohnt sich also, vor der Übertragung zum dCache mehrere Dateien in ein (komprimiertes) Archiv zu packen. Die Speicherung von großen Mengen kleiner Dateien führt dazu, dass ihre spätere Wiederherstellung sehr verlangsamt wird.
4. Dateien sollten nicht sofort nach der Übertragung zum dCache aus dem dortigen System wieder gelöscht werden. Dies verursacht eine „Verschmutzung“ des Pools – die Dateien werden dabei nicht wirklich entfernt – und senkt dessen freie Kapazität und damit auch die Performanz von späteren Übertragungen (vgl. Punkt 1). Die Dateien sollten erst dann gelöscht werden, nachdem sie erfolgreich auf Band übertragen worden sind (dies dauert in der Regel mindestens einige Stunden).

4.2.3 Grafischer GridFTP-Client

Speziell für die produktive Nutzung von dCache wurde in OFFIS ein grafischer GridFTP-Client auf der Basis von Eclipse entwickelt: <http://bi.offis.de/gridftp>

Die Nutzung dieses Clients, besonders für große Übertragungen von/zu dCache, ist empfohlen, da er speziell auf dCache ausgerichtete Fehlertoleranzmerkmale besitzt, die in den Kommandozeilenwerkzeugen fehlen. Es funktioniert nach dem Prinzip „fire and forget“, d. h. eine Übertragung wird gestartet und ggf. mehrere Tage lang überwacht, bis sie abgeschlossen ist. Ebenfalls kann eine Übertragung bei Bedarf angehalten und danach (auch mit einem neu erstellten Proxy-Zertifikat) fortgesetzt werden. Bei den primitiven Clients läuft die Übertragung dagegen meist nur bis zum ersten Fehler, wonach

sich der Benutzer selbst um die Feststellung und Rettung von halbübertragenen Dateien kümmern muss.

Damit der GridFTP-Client funktioniert, muss der Rechner, auf dem der Client läuft, mit den betroffenen GridFTP-Servern kommunizieren können. Dies bedeutet im D-Grid-Kontext, dass die ausgehende TCP-Kommunikation auf den Ports 20000-25000 und 2811 in der Firewall freigeschaltet sein muss. Im Zweifelsfall wenden sich die Anwender an ihren zuständigen Netzwerkadministrator, um diese Voraussetzung zu ermitteln bzw. zu schaffen.

4.2.4 dCache-Administrationsschnittstelle

Jede dCache-Installation bietet eine Webschnittstelle als Überblick des aktuellen Systemzustands. Diese Schnittstelle stellte sich als nützlich zur Feststellung von manchen Fehlertypen, zur Erklärung von beobachteten Performanzeigenschaften und als zusätzliche Überwachungsmöglichkeit heraus.

Beispielsweise kann man unter der Adresse <http://dcache.fz-juelich.de:2288/> genaueres über den aktuellen Zustand von dCache erfahren:

- Unter „Pool Usage“ sieht man, wie der Platz im Festplattencache aktuell genutzt wird. Die für WISENT vorgesehenen „Pools“ heißen dort „pool1_3“ und „pool2_3“. Dies entspricht zwei Dateisystemen auf der Festplatte, mit einer fixen Größe von jeweils 300 GB. Überträgt man eine Datei zum dCache, so wird sie zunächst in einen dieser Pools geschrieben. Möglichst bald nach der erfolgreichen Übertragung wird sie (asynchron und automatisch) von dCache auf Band gespeichert. „Precious Space“ in der Anzeige umfasst diese Dateien, die im Pool liegen, aber noch nicht auf Band gespeichert sind.
- In „Tape Transfer Queue“ sieht man, welche Transfers von Band zum Pool gerade aktiv sind. Die Transfers werden dann aktiviert, wenn man eine Datei herunterladen möchte, die sich nicht mehr auf der Festplatte befindet.

4.2.5 Kommandozeilentools

Die Basiswerkzeuge globus-url-copy und srmcp wurden in der frühen Phase der dCache-Nutzung in WISENT getestet und eignen sich vorrangig für kleine Übertragungen oder zur Feststellung von eventuell vorliegenden Fehlern. Sie werden wegen unkomfortabler Nutzung und unzureichender Zuverlässigkeit nicht für die produktive Nutzung empfohlen. Weitere Details zur Nutzung von globus-url-copy und srmcp befinden sich in Anhang B.

4.2.6 Erkannte Probleme mit dCache (GridFTP-Door)

Während der Entwicklung des GridFTP-Clients in OFFIS wurden einige technische Probleme in der produktiv eingesetzten Version 1.7 von dCache entdeckt, die größtenteils erfolgreich durch eine geschickte Client-Implementierung bewältigt werden konnten, um so ihre Auswirkungen auf Endanwender zu verhindern. Bei den meisten Problemen handelt es sich um Abweichungen von der GridFTP-Spezifikation oder von der Referenzimplementierung aus Globus Toolkit 4. Eine nachhaltige Korrektur dieser Probleme würde Anpassungen im Quellcode von dCache selbst erfordern. Aus diesem Grund wurden die dCache-Entwickler in DESY kontaktiert und die Liste von festgestellten Problemen an sie übermittelt. Alle aufgeführten Punkte wurden als tatsächliche, zum Teil aber bereits bekannte Unzulänglichkeiten von dCache anerkannt.

Die Korrekturen wurden größtenteils als durchführbar klassifiziert, allerdings seien alle Punkte bzgl. des erforderlichen Aufwands nur schlecht abschätzbar:

1. dCache unterstützt im Gegensatz zum GridFTP-Server von Globus Toolkit 4 nicht die Übertragung von mehreren Dateien über dieselbe Datenverbindung. Es werden in der Client-Software bei Versuchen, dieselbe Verbindung für mehrere Übertragungen wiederzuverwenden, Fehlermeldungen wie `->500-globus_xio: System error in writev: Broken pipe<-` berichtet. Praktisch zwingt das den Client, für jede Datei eine neue Verbindung aufzubauen, was nicht nur relativ langsam ist, sondern in einem Testfall mit 20000 Dateien vermutlich zu Folgefehlern führte (z. B. `500 Cannot enter passive mode: java.net.SocketException: Address already in use`).
2. dCache unterstützt nicht den 'RNFS'-Befehl. Damit ist die Umbenennung von einmal übertragenen Dateien und Verzeichnissen nicht möglich.
3. dCache berichtet nicht den Eigentümer von Verzeichnissen und Dateien, sondern nur die Zugriffsrechte, die Größe und den Namen. Dies führt dazu, dass, wenn z. B. „Permission denied“ mit Hinweis auf den Eigentümer berichtet wird, man den Eigentümer (per GridFTP) nicht erfahren kann. Es ist ein Direktzugriff auf das PNFS-Dateisystem von dCache erforderlich, den jedoch normale Anwender nicht haben.
4. Die Zeitstempel der letzten Dateiveränderung werden per GridFTP nicht angezeigt und vermutlich in dCache gar nicht gepflegt. Die aus dCache geholten Dateien bekommen neue Zeitstempel. Wenn man dCache als Datenarchiv verwenden möchte, möchte man aber im Prinzip die Dateien mit unveränderten Attributen einschließlich Zeitstempel herausholen. Ein Umweg besteht darin, die Dateien vor der Archivierung mit dem Programm `tar` zusammenzufügen.
5. dCache berichtet nur die Zugriffsrechte des Eigentümers, nicht aber die der Unix-Gruppe (die einer virtuellen Organisation im Grid-Kontext entspricht) oder der Welt. Die Gruppenzugriffsrechte sind dennoch wirksam (was positiv zu werten ist). So kann man z. B. `chmod 660` an eine Datei anwenden, aber es wird danach immer noch fälschlicherweise `-rw-----` berichtet (Lese- und Schreibzugriff nur für Eigentümer), obwohl jetzt die Gruppenmitglieder die Datei löschen/ändern dürfen.
6. dCache berichtet für Verzeichnisse die Zugriffsrechte `-rw` (ohne das Zugriffsrecht `x` im Gegensatz zum normalen (Grid)FTP-Server). Führt man `chmod 600 verzeichnis` aus (was gemäß der Anzeige nichts ändern sollte), so bleiben die berichteten Zugriffsrechte immer noch `-rw`, aber die Zugriffsrechte für die enthaltenen Dateien wechseln auf `--w`. Diese Dateien sind dann nicht mehr lesbar. Man kann den ursprünglichen Zustand mit `chmod 700 verzeichnis` wiederherstellen, wenn man weiß, was passiert ist. Das Problem besteht also darin, dass dCache die Verzeichnis- und Dateirechte unvollständig und anders als in Unix üblich berichtet.
7. Bei GridFTP-Transfers berichtet dCache den Fortschritt („performance markers“) aus Effizienzgründen nur jede 3 Minuten. Ein normaler GridFTP-Server liefert diese Informationen jedoch im 5-Sekunden-Takt. Die Transfers aus dCache erscheinen deswegen für Anwender oft so, als hätte sich die Übertragung aufgehängt. Das wirkt besonders verunsichernd, wenn man darauf mit interaktiven GridFTP-Clients zugreift. Ab Version 1.8 ist das Benachrichtigungsintervall konfigurierbar. Allerdings ist es laut dCache-Entwicklern nicht empfehlenswert, das Intervall so niedrig wie bei herkömmlichen GridFTP-Servern zu setzen, da es das System überlasten könnte.

8. Rekursives Kopieren mit globus-url-copy (Option -r) von dCache zu einem GridFTP-Server oder nach lokal (file://) ist nicht möglich.
9. dCache unterstützt nicht das „data channel authentication“-Feature, man muss also immer in globus-url-copy die Option `-nodcau` verwenden. RFT (von Globus 4.0.4) funktioniert nicht mit dCache (500 "dcau N" command not understood). Das fehlende DCAU-Feature stellt zudem eine potentielle Sicherheitslücke dar.
10. Das Anzeigen von Verzeichnisinhalten in dCache ist um Größenordnungen langsamer als bei einem normalen GridFTP-Server. Gleiches gilt auch für das Löschen von Dateien sowie Löschen/Anlegen von Verzeichnissen. Eine gewisse Beschleunigung wäre laut dCache-Entwicklern durch Tuning-Maßnahmen in zukünftigen Versionen erreichbar, insgesamt ist aber das Problem schwierig, da im Originalentwurf von dCache die Performanz von Metadatenzugriffen nicht ausreichend berücksichtigt wurde. Auch diesbezüglich ist es also empfehlenswert, über die Erhöhung von durchschnittlichen Dateigrößen den Overhead von Metadatenzugriffen zu beschränken. In der HEP-Community, die seit längerer Zeit dCache verwendet, betragen die typischen Dateigrößen ca. 1 GB, neuerdings auch ca. 10 GB. Auf der anderen Seite ist es auch empfehlenswert, dass einzelne Verzeichnisse keine große Anzahl von Dateien enthalten. Die in WISENT bisher üblichen 120 Dateien pro Verzeichnis entsprechen dieser Empfehlung.
11. Es passierte manchmal, dass Dateien in dCache angelegt wurden, die dem root-Benutzer gehörten und deswegen nicht mehr von ihrem tatsächlichen Eigentümer gelöscht werden konnten. Diese Dateien mussten vom dCache-Administrator manuell entfernt werden. Dieses Problem wurde laut Aussage von dCache-Entwicklern hoffentlich in Version 1.8 behoben. Es wurde auf jeden Fall bisher nicht von dCache 1.8 Benutzern berichtet, was aber eventuell auf bereits im Einsatz befindliche Workaround-Skripte in deren Installationen zurückzuführen ist.
12. Es wurde beobachtet, dass nachdem Dateien (zumindest laut PNFS) gelöscht worden sind, der angezeigte Wert von „Precious Space“ in „Disk Usage“ nicht auf 0 zurückging. Dieses Problem hängt vermutlich damit zusammen, dass die als „precious“ markierten Dateien per Default nicht gelöscht werden dürfen (als Schutz gegen einen potentiellen Datenverlust). Man könne aber das Löschen von solchen Dateien durch eine Konfigurationsoption für einen Pool administrativ freischalten.
13. Verbindungsversuche zu dCache scheiterten manchmal in unseren GridFTP-Client. Dieses Problem wurde durch Fehlertoleranz in Form von erneuten Verbindungsversuchen umgangen.
14. Beim Löschen von Dateien wurden von dCache Fehlermeldungen berichtet, obwohl die Dateien erfolgreich gelöscht wurden. Diese Fehlermeldungen wurden in der GridFTP-Client-Implementierung ignoriert, um den Anwender nicht über vermeintliche Fehler zu alarmieren:

```
2008-02-13 09:59:45,840 DEBUG [Worker-0]
org.globus.ftp.vanilla.FTPControlChannel:

Control channel sending: DELE gfs_3_20070405_0600_150.grb

2008-02-13 09:59:45,840 DEBUG [Worker-0]
org.globus.gsi.gssapi.GlobusGSSContextImpl: enter wrap

2008-02-13 09:59:45,841 DEBUG [Worker-0]
org.globus.gsi.gssapi.GlobusGSSContextImpl: exit wrap

2008-02-13 09:59:45,841 DEBUG [Worker-0] org.globus.ftp.vanilla.Reply: read 1st
line

2008-02-13 09:59:46,989 DEBUG [Worker-0]
org.globus.gsi.gssapi.GlobusGSSContextImpl: enter unwrap

2008-02-13 09:59:46,990 DEBUG [Worker-0]
org.globus.gsi.gssapi.GlobusGSSContextImpl: exit unwrap

2008-02-13 09:59:46,990 DEBUG [Worker-0] org.globus.ftp.vanilla.Reply: 1st
line: 553 Permission denied, reason:

CacheException(rc=5;msg=Pnfs error : java.lang.IllegalArgumentException:
Failed to remove entry 00010000000000000000061A40 : Unknown reason.)

2008-02-13 09:59:46,990 DEBUG [Worker-0]
org.globus.ftp.vanilla.FTPControlChannel:

Control channel received: 553 Permission denied, reason:
CacheException(rc=5;msg=Pnfs error :

java.lang.IllegalArgumentException: Failed to remove entry
00010000000000000000061A40 : Unknown reason.)
```

Zusätzlich verwiesen dCache-Entwickler darauf, dass die Version 1.8 von dCache das GETPUT-Feature aus der Version 2 des GridFTP-Protokolls umsetzt, was eine performantere parallele Speicherung von mehreren Dateien ermöglicht. Konkret muss bei der Verwendung von GETPUT die GridFTP-Door an der Datenkommunikation nicht als Vermittler teilnehmen, stattdessen kann sie die Adresse des für eine Datei zuständigen Pool-Rechners an den Client direkt weitergeben. Der für die Unterstützung dieses Feature erforderliche Client-seitige Patch für JGlobus wurde im GridFTP-Client angewendet.

5.Ausblick

Die in diesem Dokument geschilderten Erfahrungen mit Grid-Technologien spiegeln den aktuellen Stand am Ende Projektes WISENT wieder. Weil die Grid-Infrastruktur bis zum Projektende und darüber hinaus genutzt und auch erweitert wird, werden weitere Erfahrungen hinzukommen.

Insgesamt wurden innerhalb von WISENT Grid-Technologien erfolgreich eingesetzt, aber auch deren Grenzen aufgezeigt. Bestimmte Szenarien waren aus sicherheitstechnischen Gründen nur unter hohem Aufwand umsetzbar bzw. zum Teil gar nicht realisierbar. In vielen Fällen mussten „Workarounds“ geschaffen werden, u.a. um den Benutzern eine möglichst einfache Bedienung anbieten zu können, z.B. mit Hilfe des grafischen GridFTP-Clients. Die positiven Erfahrungen überwiegen aber deutlich den negativen, so dass der Einsatz von Grid-Technologien in der Community Energiemeteorologie nachhaltig gewünscht wird.

Anhang A: Einrichtung eines Grid-FTP-Server in einer chroot-Umgebung

Installiert wurde die Version 4.0.3 des Globus Toolkit auf `srvlx050.uni-oldenburg.de`, einem Debian 4.0-System mit X86_64-Architektur. Eine ausführliche englische Installations- und Konfigurationsanleitung ist auch unter <https://bi.offis.de/wisent/tiki-index.php?page=GridFTP-chroot-jail> zu finden.

Nach der Erfüllung der Abhängigkeiten (Java, PostgreSQL) und dem Anlegen des Benutzers `globus` sowie der Konfiguration der Datenbank gemäß dem Globus Toolkit-Manual folgte das Erstellen eines neuen Verzeichnisbaums mit einem Wurzelverzeichnis und dem notwendigen Baum an Systemverzeichnissen. Welche Verzeichnisse und Dateien benötigt werden, hängt stark von den Programmen ab, die innerhalb der `chroot-Umgebung` laufen sollen. Die Abhängigkeit von Systembibliotheken kann mittels `ldd <programm>` ermittelt werden. Detailliertere Informationen liefert der Aufruf `strace <programm>`.

```
mkdir /globus-root
mkdir -p /globus-root/usr/local/globus-4.0.3 /globus-root/etc/grid-security
mkdir -p /globus-root/home/wisent-users /globus-root/var \
/globus-root/usr/bin /globus-root/bin /globus-root/lib \
/globus-root/lib64 /globus-root/root /globus-root/tmp /globus-root/sys \
/globus-root/proc /globus-root/dev /globus-root/usr/lib \
/globus-root/usr/lib64 /globus-root/usr/share /globus-root/home/globus \
/globus-root/tmp /globus-root/public /globus-root/restricted/mntpnt
```

Das Verzeichnis `/globus-root/restricted/mntpnt` wird als mountpoint für das Einhängen eines Netzlaufwerks verwendet. Bei Bedarf müssen daher mehrere Verzeichnisse auf dieser Ebene erstellt werden.

Die meisten Systemdateien können aus dem Wirtssystem kopiert werden:

```
cp /lib/libncurses.so.5 /lib/libdl.so.2 /lib/libc.so.6 /lib/libm.so.6 \
/lib/libpthread.so.0 /lib/libselinux.so.1 /lib/libsepol.so.1 \
/lib/libnss_files.so.2 /lib/libnss_nis.so.2 /lib/libnsl.so.1 \
/lib/libnss_compat.so.2 /lib/libpam.so.0 /lib/libpam_misc.so.0 \
/lib/libproc-3.2.7.so /globus-root/lib/
cp /lib64/ld-linux-x86-64.so.2 /globus-root/lib64/
cp /usr/lib/libperl.so.5.8 /globus-root/usr/lib/
cp /usr/bin/cut /usr/bin/env /usr/bin/vmstat /globus-root/usr/bin/
cp -r /usr/lib/gconv /usr/lib/jvm /usr/lib/locale /usr/lib/perl5 \
/usr/lib/perl /globus-root/usr/lib/
cp -r /usr/lib64/jvm /globus-root/usr/lib64/
cp -r /etc/postgresql /etc/pam.d /etc/pam.conf /etc/java-1.5.0-sun \
/etc/shells /etc/login.defs \
/globus-root/etc/
cp -r /usr/share/locale /usr/share/perl5 /usr/share/perl \
/globus-root/usr/share/
```

Einige Systemdateien müssen editiert werden, um die geänderten Pfade innerhalb der `chroot-Umgebung` zu respektieren. Die folgenden Dateien sind beispielhaft und müssen individuell an das vorhandene System angepasst werden. Es folgt ein Beispiel für die Datei `/etc/fstab`.

```
# /etc/fstab: static file system information. - Dummy for chroot jail
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/cciss/c0d0p1 / ext3 defaults,errors=remount-ro 0 1
/dev/cciss/c0d0p2 none swap sw 0 0
/dev/hda /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto 0 0
```

Da innerhalb der chroot-Umgebung keine mounts erstellt oder aufgehoben werden, kann /etc/mtab als statische Datei erstellt werden.

```
/dev/cciss/c0d0p1 / ext3 rw,errors=remount-ro 0 0
tmpfs /lib/init/rw tmpfs rw,nosuid,mode=0755 0 0
proc /proc proc rw,noexec,nosuid,nodev 0 0
sysfs /sys sysfs rw,noexec,nosuid,nodev 0 0
procbususb /proc/bus/usb usbfs rw 0 0
udev /dev tmpfs rw,mode=0755 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
devpts /dev/pts devpts rw,noexec,nosuid,gid=5,mode=620 0 0
```

Auch in /etc/profile müssen Pfade angepasst werden.

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).
PATH="/bin:/usr/bin:/usr/local/bin:/usr/local/globus-4.0.3/bin:\
/usr/local/globus-4.0.3/sbin"
export PS1='# '
export PATH
umask 022
export LD_LIBRARY_PATH=/lib:/lib64:/usr/local/globus-4.0.3/lib:\
/usr/lib:/usr/lib64
export GLOBUS_LOCATION=/usr/local/globus-4.0.3/
. $GLOBUS_LOCATION/etc/globus-user-env.sh
. $GLOBUS_LOCATION/etc/globus-devel-env.sh
export JAVA_HOME=/usr/lib/jvm/java-1.5.0-sun/jre
```

Es folgt die eigentliche Installation des Globus Toolkit.

```
chown globus /globus-root/usr/local/globus-4.0.3 /globus-root/home/globus
su globus
cd /globus-root/usr/local/globus-4.0.3
wget http://www-unix.globus.org/ftppub/gt4/4.0/4.0.3/\
installers/bin/gt4.0.3-x86_64_rhas_4-installer.tar.gz
tar xfz gt4.0.3-x86_64_rhas_4-installer.tar.gz
cd gt4.0.3-x86_64_rhas_4-installer
export GLOBUS_LOCATION=/globus-root/usr/local/globus-4.0.3/
./configure --prefix=$GLOBUS_LOCATION --with-buildopts="--static"
make
make install
```

Als nächstes erfolgt die Initialisierung der Datenbank.

```
su globus

psql -d rftDatabase -f /globus-root/usr/local/globus-4.0.3/\
share/globus_wsrf_rft/rft_schema.sql
```

Die Änderungen in `/etc/grid-security` geschehen im Wirtssystem, das gesamte Verzeichnis wird in die `chroot`-Umgebung gemounted. Dadurch wird das Problem divergierender Parallelversionen vermieden. Die Existenz eines gültigen (Grid-)Server-Zertifikates wird vorausgesetzt.

```
mkdir -p /etc/grid-security/certificates /etc/grid-security/grid-mapfiles
cp hostcert.pem hostkey.pem /etc/grid-security/
cp /etc/grid-security/hostcert.pem /etc/grid-security/containercert.pem
cp /etc/grid-security/hostkey.pem /etc/grid-security/containerkey.pem
chown globus /etc/grid-security/containercert.pem \
  /etc/grid-security/containerkey.pem
chmod 400 /etc/grid-security/hostkey.pem

wget http://eugridpma.org/distribution/current/accredited/tgz/ca_GermanGrid-1.15.tar.gz

tar xzf ca_GermanGrid-1.15-tar.gz
mv ca_GermanGrid/* /etc/grid-security/certificates/
```

Wie bei einer normalen Globus-Installation wird jedes mit einem Benutzer-Zertifikat versehene Mitglied der VO WISENT einem eigenen UNIX-Nutzer zugewiesen, der zur Gruppe `wisent` (mit der Gruppen-ID 1002 in unserem System) gehört:

```
useradd -g 1002 -K UID_MIN=10000 -c "Johannes Hurka" \
  -d /home/wisent-users/dgws0008 dgws0008
```

Die Option `-K UID_MIN=10000` sorgt dafür, daß die Nutzer-IDs oberhalb von 10000 liegen, so daß es keine Kollisionen mit zentral verwalteten Nutzer-IDs gibt.

Zusätzlich wird auch der Nutzer `globus` in die Gruppe `wisent` aufgenommen.

Die VO-Mitglieder werden z.Z. manuell in `/etc/grid-security/grid-mapfile` eingetragen.

```
...
"/C=DE/O=GridGermany/OU=Carl von Ossietzky Universitaet
Oldenburg/OU=Physik/CN=Johannes Hurka" dgws0008
...
```

Die Home-Verzeichnisse für die angelegten User werden unter `/home/wisent-users/` angelegt, welches in die `chroot`-Umgebung gemounted wird.

Für den privilegierten Zugriff werden alle berechtigten Benutzer zur Gruppe `globus` hinzugefügt (in der Datei `/etc/group`, die nach jeder Änderung wieder nach `/globus-root/etc` kopiert werden muss).

```
...
globus:x:1001:dgws0008:dgws0009:dgws0010:
wisent:x:1002:globus
...
```

Aufgrund eines Bugs im GridFTP-Server können User auf eine Datei, für die sie keine Benutzerrechte haben, nur dann zugreifen, wenn die Gruppenrechte für die primäre Gruppe des Benutzers ausreichen. Entsprechend müssen die Einträge in `/etc/passwd` so geändert werden, dass die primäre Gruppe der privilegierten Nutzer nicht `wisent` (gid 1002) sondern `globus` (gid 1001) ist.

```
...
dgws0008:x:1125:1001:Johannes Hurka:/home/wisent-users/dgws0008:/bin/sh
...
```

Diese Datei muss ebenfalls nach jedem Update nach `/globus-root/etc/` kopiert werden.

Das Verzeichnis `/globus-root/restricted/` soll nur für Mitglieder der Gruppe `globus` zugänglich sein.

```
chown globus.globus /globus-root/restricted
chmod ug+rwx,o-rwx /globus-root/restricted
```

Stattdessen ist `/globus-root/public` für alle GridFTP-User offen. Da es auch aus dem Wirtssystem beschrieben und gelesen werden kann, können hier beispielsweise Daten temporär abgelegt werden, die von nicht-privilegierten GridFTP-Nutzern heruntergeladen werden dürfen.

```
chown globus.wisent /globus-root/public
chmod ugo+rwx /globus-root/public
```

Der komplette Prozess zum Starten und Stoppen des GridFTP-Servers wird von einem angepassten init-Skript `/etc/init.d/gridftp` verwaltet, welches sowohl die Vorbereitung der `chroot`-Umgebung übernimmt (Kopieren von Dateien, Mounten von Verzeichnissen), als auch den GridFTP-Server und den Globus-Container startet. Es kann von `root` manuell mit den Argumenten `start`, `stop` und `restart` aufgerufen werden, aber auch für den Start beim Boot-Vorgang in die entsprechenden Runlevel verlinkt werden. Im Wesentlichen besteht das Skript aus den Funktionen `do_start()` und `do_stop()`, die restliche Funktionalität entspricht einem Standard-Init-Skript.

```
# Function that starts the service
do_start()
{
    # Return 1 if service is already running
    if pgrep globus-gridftp > /dev/null; then
        return 1
    fi
    # Else start with mounting needed dirs:
    mount | grep /globus-root/etc/grid-security > /dev/null || mount -o bind
/etc/grid-security /globus-root/etc/grid-security
    mount | grep /globus-root/etc/default > /dev/null || mount -o bind
/etc/default /globus-root/etc/default
    mount | grep /globus-root/home/wisent-users > /dev/null || mount -o bind
/home/wisent-users /globus-root/home/wisent-users
    mount | grep /globus-root/dev > /dev/null || mount -o bind /dev /globus-
root/dev
    mount | grep /globus-root/sys > /dev/null || mount -o bind /sys /globus-
root/sys
    mount | grep /globus-root/etc/proc > /dev/null || mount -o bind /proc
/globus-root/proc
    mount | grep /globus-root/var > /dev/null || mount -o bind /var /globus-
root/var
    mount | grep /globus-root/tmp > /dev/null || mount -o bind /tmp /globus-
root/tmp
    mount | grep /globus-root/restricted/mntpnt > /dev/null || mount -o bind
/restricted-data /globus-root/restricted/mntpnt
```

```

# And more data mounts if needed
# Copy needed files:
cp /etc/hosts /etc/resolv.conf /globus-root/etc
cp /etc/localtime /etc/passwd /etc/group /globus-root/etc
cp /etc/shadow /etc/nsswitch.conf /globus-root/etc/
# Start the server from the chrooted system:
# Return 2 if service could not be started
cd /globus-root
chroot /globus-root /usr/local/bin/start_gridftp_server.sh || return 2
# Wait a moment - don't know whether it's really necessary...
sleep 1
}

# Function that stops the service
do_stop()
{
    if pgrep globus-gridftp > /dev/null; then
        # Kill the globus container
        # (using pkill to search for first part of command line):
        if pgrep -u globus -f "java -Dlog4j.configuration=container-log4j" >
/dev/null; then
            pkill -u globus -f "java -Dlog4j.configuration=container-log4j"
        fi
        # Stop the server:
        kill -9 `pidof globus-gridftp-server` || return 2
        # Wait a moment for resources to recover before unmount
        sleep 1
        # Clean up mounted dirs:
        umount /globus-root/etc/grid-security > /dev/null
        umount /globus-root/etc/default > /dev/null
        umount /globus-root/home/wisent-users > /dev/null
        umount /globus-root/sys > /dev/null
        umount /globus-root/proc > /dev/null
        umount /globus-root/var > /dev/null
        umount /globus-root/tmp > /dev/null
    fi
}

```

Wenn `/etc/init.d/gridftp start` aufgerufen wird, werden zunächst die Verzeichnisse und Dateien vorbereitet. Dann wechselt das Skript in das `chroot`-Verzeichnis `/globus-root` und ruft `chroot` in Verbindung mit dem Skript `/usr/local/bin/start_gridftp_server.sh` auf (im normalen System unter `/globus-root/usr/local/bin/start_gridftp_server.sh` zu finden), welches dann den GridFTP server sowie den Globus-Container startet.

```

#!/bin/bash
# Starts the GridFTP server inside the chroot
. /etc/profile
/usr/local/globus-4.0.3/sbin/globus-gridftp-server -S -p 2811
su globus "-c /usr/local/globus-4.0.3/bin/globus-start-container > /dev/null&"

```

Mit dieser Konfiguration konnten alle von der Arbeitsgruppe gewünschten Eigenschaften umgesetzt werden.

Anhang B: Einsatz von globus-url-copy und srmcp zum Zugriff auf dCache

Kopieren einzelner Dateien von lokal nach dCache oder umgekehrt

Dazu kann sowohl globus-url-copy als auch srmcp verwendet werden, z.B.:

```
globus-url-copy -vb gsiftp://dcache.fz-juelich.de/pnfs/fz-
juelich.de/data/wisent/file.img file:///tmp/file.img

globus-url-copy -vb file:///tmp/file.img gsiftp://dcache.fz-juelich.de/pnfs/fz-
juelich.de/data/wisent/file.img

srmcp -debug=false srm://dcache.fz-juelich.de:8443/pnfs/fz-
juelich.de/data/wisent/file.img file:///tmp/file.img

srmcp -debug=false file:///tmp/file.img srm://dcache.fz-
juelich.de:8443/pnfs/fz-juelich.de/data/wisent/file.img
```

Mit der Option `-vb` bei `globus-url-copy` wird zusätzlich die Datentransferrate angezeigt. Zu beachten ist die Anzahl der Schrägstriche (/) nach dem Protokoll `file:`. Bei `globus-url-copy` sind es 3 (`file:///tmp/file.img`) und bei `srmcp` 4 (`file:///tmp/file.img`). Der Befehl `srmcp` ist auch mit der Option `-debug=false` relativ geschwätzig. Folgender Output beim Kopieren einer Datei ist normal:

```
srmcp -debug=false srm://dcache.fz-juelich.de:8443/pnfs/fz-
juelich.de/data/wisent/file.img file:///tmp/file.img

set::trying to set unknown name "storagetype" to "permanent"
set::trying to set unknown name "server_mode" to "passive"
set::trying to set unknown name "long_ls_format" to "false"
set::trying to set unknown name "recursion_depth" to "0"
set::trying to set unknown name "srm_protocol_version" to "1"
set::trying to set unknown name "request_lifetime" to "86400"
user credentials are: /C=DE/O=GridGermany/OU=OFFIS e.V./OU=Grid-RA/CN=Guido
Scherp
SRMClientV1 : connecting to srm at httpg://dcache.fz-
juelich.de:8443/srm/managerv1
org.globus.ftp.exception.ServerException: Server refused performing the
request. ...
GridftpClient: transfer exception
org.globus.ftp.exception.ServerException: Server refused performing the
request. ...
copy failed with the error
org.globus.ftp.exception.ServerException: Server refused performing the
request. ...
try again
sleeping for 10000 before retrying
```

Warum diese Fehlermeldung einmal auftritt, konnte nicht geklärt werden. Danach wird der Transfer aber erfolgreich gestartet, was leider nicht berichtet wird. Mit der Option `-debug=true` erkennt man den Start eines Transfers, hat aber etwas mehr Output.

Kopieren ganzer Verzeichnisse von lokal nach dCache

Ganze Verzeichnisse können mit globus-url-copy kopiert werden, aber nur von lokal nach dCache und nicht umgekehrt. Der Befehl srmcp unterstützt derzeit nur das Kopieren von einzeln genannten Dateien. Zum Kopieren eines ganzen Verzeichnisses bietet globus-url-copy die Option `-r` an, z.B.

```
globus-url-copy -r -vb file:///tmp/verzeichnis/ gsiftp://dcache.fz-juelich.de/pnfs/fz-juelich.de/data/wisent/temp/
```

Die umgekehrte Richtung erzeugt folgende Fehlermeldung:

```
globus-url-copy -r -vb gsiftp://dcache.fz-juelich.de/pnfs/fz-juelich.de/data/wisent/temp/ file:///tmp/verzeichnis/
error: No files matched the source url.
```

Third-Party-Transfers

Third-Party-Transfers, d. h. Übertragungen von einem GridFTP-Server zum anderen (nicht von einem GridFTP-Server zum Client), sind mit globus-url-copy oder mit srmcp möglich. Im ersten Fall muss die Option `-nodcau` verwendet werden. dCache verhält sich dabei wie ein GridFTP-Client und spricht einen externen GridFTP-Server an. Auch in RFT und WS-GRAM (rftOptions), die bei der Ausführung von Globus Toolkit 4 Jobs zwecks File Staging verwendet werden, kann eine äquivalente Option zu `-nodcau` angegeben werden. RFT funktioniert allerdings auf Grund eines Bugs nicht, der von WISENT an die Globus-Entwickler berichtet wurde: http://bugzilla.mcs.anl.gov/globus/show_bug.cgi?id=5829

Ein File-Staging mit dCache (basierend auf WS-GRAM) wäre also vermutlich auch unmöglich, was aber noch nicht getestet wurde.

Ein Third-Party-Transfer mit globus-url-copy kann wie folgt aufgerufen werden:

```
globus-url-copy -vb -nodcau gsiftp://srvgrid01.offis.uni-oldenburg.de/tmp/file.img gsiftp://dcache.fz-juelich.de/pnfs/fz-juelich.de/data/wisent/file.img
globus-url-copy -vb -nodcau gsiftp://dcache.fz-juelich.de/pnfs/fz-juelich.de/data/wisent/file.img gsiftp://srvgrid01.offis.uni-
```

Da dCache nur alle 3 Minuten den Fortschritt eines Transfers berichtet, wird bei der Option „-vb“ entsprechend nur alle 3 Minuten eine Aktualisierung angezeigt.

Ein ThirdPartyTransfer mit srmcp kann wie folgt aufgerufen werden:

```
srmcp --debug=false srm://dcache.fz-juelich.de:8443//pnfs/fz-juelich.de/data/wisent/file.img gsiftp://srvgrid01.offis.uni-oldenburg.de//tmp/file.img
srmcp -debug=false gsiftp://eridanus.caf.dlr.de//tmp/file.img srm://dcache.fz-juelich.de:8443//pnfs/fz-juelich.de/data/wisent/file4.img
```

Zu beachten ist, dass bei dem GridFTP-Ziel nach dem Hostnamen zwei Schrägstriche (/) angegeben werden müssen (gsiftp://srvgrid01.offis.uni-oldenburg.de//tmp/file.img). Das Kopieren ganzer Verzeichnisse per ThirdPartyTransfer mit srmcp ist nicht möglich.

Wenn man als Zieladresse ein Verzeichnis angibt, kann man auch mehrere Quelldateien mit einem srmcp-Aufruf transferieren:

Erfahrungen mit Grid-Technologien im Projekt WISENT (AP 1.2)

```
srmcp --debug=false srm://dcache.fz-juelich.de:8443//pnfs/fz-
juelich.de/data/wisent/file1.img srm://dcache.fz-juelich.de:8443//pnfs/fz-
juelich.de/data/wisent/file2.img gsiftp://srvgrid01.offis.uni-
oldenburg.de//tmp/
```

srmcp ist auf jeden Fall auf Rechenknoten in Oldenburg, Jülich und Hannover verfügbar und wurde dort getestet. Vor der ersten Verwendung von srmcp musste eventuell noch `mkdir ~/.srmconfig` ausgeführt werden.



Bericht zu den Auswirkungen der DLR-IT-Sicherheitsstrukturen beim Datentransfer mit Partnern außerhalb des DLR

Arbeitspaket 2.1

Gerhard Gesell
Deutsches Zentrum für Luft- und Raumfahrt DLR, Deutsches
Fernerkundungsdatenzentrum DFD
Münchner Straße 20
82234 Weßling

Internet: <http://www.caf.dlr.de>

1. Anforderungen und Ausgangssituation

Im Projekt WISENT gibt es zwei Arbeitspakete, die sich mit dem Datenaustausch zwischen den WISENT-Partnern befassen, nämlich AP2.1 und AP3.2. Inhaltlich unterscheiden sich die beiden Arbeitspakete durch Art, Umfang und Ablauf der betrachteten Datentransfers. Während es im AP3.2 um unregelmäßige, explizit vom Datenempfänger beauftragte und inhaltlich nicht vorhersehbare Datentransfers geht, haben die im AP2.1 betrachteten Datenaustausch-Szenarien folgende Eigenschaften:

- Kleine Datenmengen – Einzelprodukte
- Regelmäßiger operationeller Transfer typisch
- Datentransfer direkt zum Abnehmer bzw. Pickup-Point (PP)
- Transferinitiative meist beim Provider
- Kürzere Transferzeiten, teils NRT
- Gute Datentransferraten mäßig relevant
- Transfermonitoring beschränkt auf Vollständigkeit
- Vereinbarte Formate der Einzelprodukte
- Vereinbarte Metadaten der Einzelprodukte
- Qualitätssicherung der Einzelprodukte
- Produkte sind fertig prozessiert
- Einzelprodukte sind prozessierungschronologisch sortiert
- Prozessierungsumfeld und –parameter liegen als Standard fest

Zu Beginn der Arbeit in AP2.1 wurde versucht, mit Hilfe von Globus Toolkit 4 (GT4) und GridFTP auf DFD-internen Systemen einen durch Grid-Technologie unterstützten Datenaustausch, der die genannten Eigenschaften unterstützt, zu konzipieren. Dazu wurde GT4 (und damit GridFTP als Bestandteil von GT4) auf einem DLR-DFD-internen System mit SUN SOLARIS 9 als Betriebssystem installiert. Der darauf folgende Datentransfer-Test zwischen diesem und einem System beim Projektpartner OFFIS ergab folgendes:

- Eine GT4-Verbindung zwischen DFD und OFFIS lässt sich nur vom DFD aus aufbauen.
- Die Verbindungsdaten, also die zum Aufbau der Verbindung notwendigen Informationen, können ausgetauscht werden.
- Sobald ein Transfer von Dateninhalten versucht wird, geschieht nichts.

Als Grund für dieses Verhalten ergab sich, dass die für den Austausch der Verbindungsdaten verwendeten Ports 8443 und 2811 vom DFD nach außen offen sind, der Portbereich 20000-25000, der für den eigentlichen Datentransfer frei geschaltet sein muss, jedoch geschlossen ist. Die Kommunikation zu einem Rechner mit Grid-Funktionalität erfolgt in Form von Web-Services über Port 8443/tcp. Der Datentransfer mittels „GridFTP“ ist an dem normalen FTP-Protokoll angelehnt, ist aber noch komplexer: Der Kontrollkanal ist beim GridFTP-Server auf den Port 2811/tcp fixiert. Die Datenkanäle werden dynamisch zwischen Server und Client in einem gewissen Port-Bereich ausgehandelt. Der Bereich ist zwar konfigurierbar, sollte aber in der Praxis einige hundert bis tausend Ports breit sein (z.B. TCP-Ports 20000 bis 25000). Dieser Bereich muss sowohl beim Client als auch beim Server offen sein. Das Ergebnis der Tests vom DFD-internen System aus war der Eindruck bzw. die Erkenntnis, dass diese Eigenheit des GridFTPs aus Sicht eines guten Firewall-Konzeptes sehr bedrückend ist:

Sollten sich die Grid-Technologien auf Basis von Globus Toolkit in der IT-Welt weiter verbreiten, werden einige Sicherheitskonflikte auftreten, wie etwa zum einen die Öffnung vieler Ports an den Firewalls, zum anderen die Bündelung von Diensten (Webservices) auf einem Netzwerkport. Damit vermindert sich der Nutzen des bisherigen Firewall-Prinzips, welches Dienste anhand von Netzwerkadressen und –ports gezielt erlaubt oder unterbindet. Nur die externen Webservices zu nutzen wäre noch mit der Firewall-Idee und dem DLR-IT-Sicherheitskonzept vereinbar. Aber der Datentransfer mit GridFTP und die Einbindung eigener Ressourcen im DLR-Netz in externe Prozesse lassen sich mit dem DLR-IT-Sicherheitskonzept nicht einfach realisieren und sind mit dem DFD-IT-Sicherheitskonzept gänzlich unvereinbar.

2. Lösungsansatz und Rolle des neuen Speicherelements

Bei DLR-DFD und beim WISENT- und C3Grid-Partner DLR-PA wurden fünf Ansätze für die Lösung des Firewall-Problems diskutiert:

1. Das Globus Toolkit wird doch auf einem unserer Institutsrechner installiert, die DLR-Firewall wird geöffnet und erlaubt die Kommunikation mit den Partnern.
2. Das Globus Toolkit wird auf einem isolierten Rechner in der DMZ installiert. Die DLR-Firewall wird für diesen Rechner geöffnet.
3. Das Globus Toolkit wird auf einem unserer Institutsrechner installiert und ein „ALG-Proxy“ (Application Level Gateway) in der DMZ vermittelt die Anfragen und Datentransfers von und nach außen.
4. Das Globus Toolkit wird in einem eigenen Netzwerk installiert, ähnlich einer DMZ. Die Rechner des Netzwerks sind virtuelle Maschinen (z.B. VMware, XEN), welche auf den Arbeitsplatzrechnern des Instituts laufen.
5. Das Globus Toolkit öffnet für Datentransfers dynamisch die DLR-Firewall.

Nachdem beim DFD schnell klar wurde, dass sich grid-technologisch basierte Datenaustausch-Szenarien nicht von innerhalb der DFD-Firewall aus umsetzen lassen, musste von einer Lösung mit Hilfe eines (isolierten) Systems in der DLR-DMZ ausgegangen werden. Es wurde für das DFD beschlossen, den Ansatz Nummer 2 zu verfolgen und die auszutauschenden Daten mit Hilfe von konventionellen Methoden zwischen DFD und DMZ zu transferieren.

Ende 2006 ergab sich dann die Möglichkeit, die ohnehin benötigten System-Ressourcen in der DMZ mit der Einrichtung einer Grid-Speicher-Ressource (Storage Element SE) aus BMBF-Sondermitteln für den Aufbau einer nationalen D-Grid-Infrastruktur zu verknüpfen. Ohne diese Gelegenheit wären Ressourcen von bereits in der DMZ verfügbaren Systemen, etwa der WDC-RSAT-Server, genutzt worden, um dort die für die Datenaustausch-Szenarien notwendigen Grid-Services zu installieren. Statt dessen wurde mit den Sondermitteln ein eigenes System, bestehend aus einem Rechner mit 1 TB SAN-Storage-Anteil in der DMZ vom Service-Provider T-Systems-SfR bis Ende 2010 angemietet und für zwei bereits existierende Systeme innerhalb der DFD-Firewall weitere Speicher-Ressourcen mit 10 TB Kapazität gekauft und installiert. Zwischen den Systemen innen und in der DMZ wird ein Synchronisations-Mechanismus auf der Basis von rsync eingerichtet. Dies entspricht nun einer größeren, komfortableren und grid-technologisch auch besseren Lösung als anfangs ohne Sonderinvestition geplant. Die Details der Datentransferstruktur und des Zusammenwirkens aller Komponenten des SE werden unter AP 3.2 beschrieben (Betriebskonzept SE).

Die Installation von Middleware (SRM/dCache) auf dem SE erwies sich im weiteren Verlauf als weiterhin problematisch. Es zeigten sich sowohl Probleme mit der Installation selbst als auch mit der DLR-FW nach außen. Innerhalb der DMZ ist jeweils eine bestimmte LINUX-Version als OS dringend empfohlen, da die Maschinen meist als virtuelle Plattformen größerer Blade-Center realisiert sind. Für das zur der Zeit genutzte LINUX gab es von DGI-Seite keine Referenzinstallation und auch unserer Bitte um eine Unterstützung vor Ort konnte nicht entsprochen werden. Zusammen mit den FW-Problemen führte das dann nach Rücksprache mit dem DGI (Frau Levendel, Desy) zum Verzicht auf SRM/dCache. Das SE wird nun letztlich mit einer Kombination von GT4-GridFTP als Middleware und gewöhnlichem FTP betrieben. Vom System in der DMZ aus wurden alle für GridFTP notwendigen Ports von und nach außen geöffnet. Tests zwischen DFD, OFFIS und einem dritten Partnersystem waren danach für alle relevanten Funktionlitäten erfolgreich.

3. Zusammenfassung

Die Erfahrungen mit den DLR-Firewall- und Sicherheitseinrichtungen im Zusammenhang mit Datentransfers und Grid-Konzepten lassen sich in folgender Tabelle chronologisch sortiert zusammenfassen:

Ereignis: Befund (chronologisch)	Ursache/Konsequenz/Ergebnis
GT4 und damit GridFTP werden auf einem DFD-internen Rechner installiert und getestet: GridFTP kann zwar eine Verbindung aufbauen, aber keine Daten übertragen.	Nicht alle notwendigen Ports sind an der DFD-FW frei geschaltet.
Die Freischaltung der notwendigen Ports wird beantragt.	Diese Freischaltung wird für die DFD-FW abgelehnt, alle Middleware soll nun auf dem WDC-RSAT-Rechner in der DMZ installiert werden.
Die Sondermittel für den Aufbau einer nationalen Grid-Infrastruktur werden ausgeschrieben.	DFD erhält Sondermittel zum Aufbau eines Storage Elements und kann damit auch eine eigene Plattform in der DMZ für Grid-Zwecke einrichten.
Für die DMZ-Plattform werden alle notwendigen Port-Freischaltungen durchgeführt, allerdings gelingt es nicht, dCache vernünftig zu installieren, da eine passende Referenzinstallation oder zusätzliche Unterstützung von DGI nicht angeboten wird (→ Frau Lewendel, Desy). Außerdem erscheint zweifelhaft, ob die DLR-FWs und –Sicherheitsrichtlinien das angestrebte dCache-Szenario ermöglichen.	Statt dCache wird nun (zunächst) GridFTP zum Betrieb des SE vorgesehen.
Der Versuch, DFD-Mitarbeiter mit Hilfe ihrer Zertifikate an einem Server am FZ Jülich in der VO WISENT anzumelden, scheitert. Der Server bzw. das Anmeldeportal ist von DFD-Arbeitsplatzrechnern aus nicht sichtbar.	Die Ursache des Problems ist bis zum Abschluss dieses Dokumentes nicht ermittelbar gewesen, wahrscheinlich steht es aber ebenfalls im Zusammenhang mit den DLR-Firewalls und deren Regularien.

Überwachung des Serverparks und der Datenströme mit Nagios

Arbeitspaket 2.1

Jochen Rehwinkel
meteocontrol GmbH
Spicherer Straße 48
86157 Augsburg

Internet: <http://www.meteocontrol.de>

1 Einführung

Meteocontrol verarbeitet große Datenmengen von weltweit verteilten Wetterstationen unterschiedlicher Betreiber. Die Daten werden über das Wettermessnetz von Meteomedia sowie über das Synop-Messnetz in regelmäßigen zeitlichen Abständen (z. B. stündlich) geliefert, bei meteocontrol dekodiert und in lokale Datenbestände importiert. Zusätzlich werden die täglichen Messwerte der Solarstromanlagen für das Webportal Safer'Sun importiert und weiterverarbeitet. Es gibt eine Vielzahl von automatisierten Down- und Uploads von Daten und internen Verarbeitungs- und Berechnungsaufgaben. Zu Beginn von WISENT existierte nur eine sehr einfache dezentrale Überwachung dieser Dienste, die aber nicht ausreichend war, um einen verlässlichen Überblick über das gesamte System zu geben. Im Hinblick auf die geplante größere Vernetzung von meteocontrol und eine Parallelisierung von Aufgaben innerhalb von WISENT wurde es unbedingt notwendig, die weitgehend manuelle Überwachung zu automatisieren, um den Überblick über das System nicht zu verlieren. Hierbei kommt das Open-Source-Überwachungssystem NAGIOS zum Einsatz.

1.1 NAGIOS

Bei NAGIOS handelt es sich um ein System-, Service- und Netzwerkmonitoringtool mit Plugin-basierter Architektur. Es besitzt ein leistungsfähiges und flexibles Benachrichtigungssystem sowie ein Webinterface für Zustandsinformationen, Logdateien und Reports. Zustände von überwachten Komponenten können mit Hilfe des SNMP (Simple Network Management Protocol) Protokolls sowie über selbst definierte Module abgefragt werden. Mithilfe des NRPE (NAGIOS Remote Plugin Executor) ist es auch möglich, Plugins auf entfernten Rechnern auszuführen, die die Ergebnisse ihrer Untersuchung an den NAGIOS-Server melden. NAGIOS konzentriert sich bei der Überwachung auf Ampelzustände, d. h. es werden keine numerischen Messwerte in Form von Zeitreihen gesammelt, sondern vielmehr die momentane Service-Verfügbarkeit geprüft und berichtet.

2 NAGIOS bei meteocontrol

2.1 Überblick

Um die Funktion des Serverparks von meteocontrol zu gewährleisten, ist es notwendig, alle Server und deren Dienste, die sich im Live-System befinden, zu überwachen. Die Verarbeitungsketten sind so komplex, dass durch den Ausfall eines einzelnen Servers das Gesamtsystem zum Erliegen kommt. Außerdem ist es notwendig, einen Server des WISENT-Partners Universität Oldenburg zu überwachen. Die Kunden von meteocontrol werden ab 6.00h mit Wetterdaten beliefert, u.a. auch mit Globalstrahlungsdaten aus Oldenburg. Hierbei ist es wichtig, dass eine mögliche Fehllieferung so früh wie möglich erkannt wird. Aus diesem Grund ist eine direkte Überwachung in Oldenburg notwendig. Abbildung 1 zeigt die überwachten Server.

Von NAGIOS überwachte Server

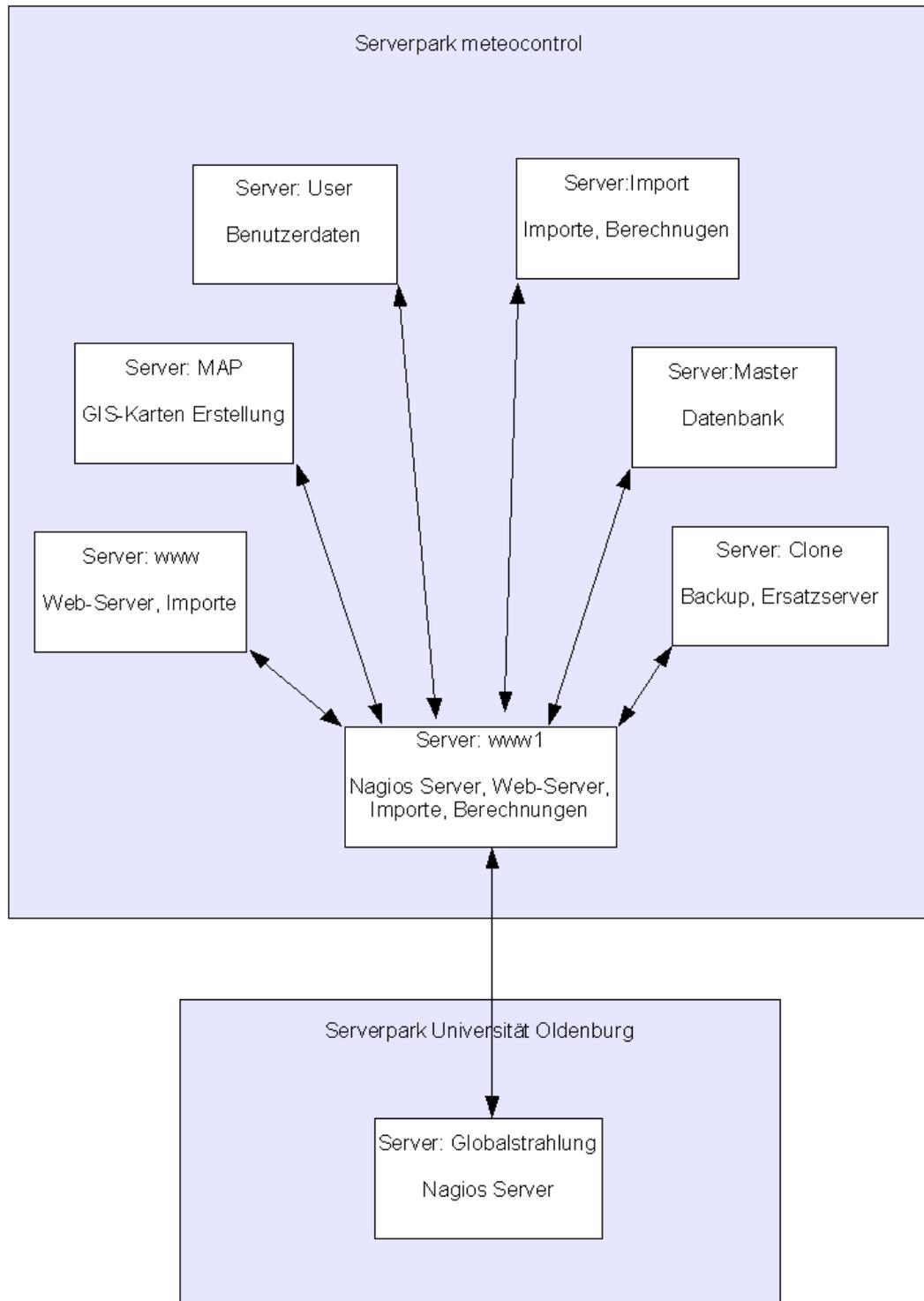


Abbildung 1: Mit NAGIOS überwachte Server

2.2 Ebenen der Überwachung

Die Überwachung mit NAGIOS bei meteocontrol lässt sich generell in 4 Ebenen aufteilen. Je niedriger die Ebene, desto höher ist die Gefahr, dass bei einem Alarm die Funktionalität

des Gesamtsystems bedroht ist. Die Einteilung in Ebenen soll auch als Hilfestellung für die spätere Festlegung einer Eskalationsstrategie dienen.

- Ebene 1: Überwachung der Grundfunktionalität des Serverparks. Hierzu zählen die „Alive“-Meldungen der einzelnen Server, sowie die Überprüfung der Hardware (z. B. Server-Load, Festplatten, Server-Raid).
- Ebene 2: Überwachung von Systemdiensten. Hierzu zählen Dienste wie z.B. HTTP, FTP, SSH, VNC.
- Ebene 3: Überwachung von einzelnen automatisierten Tasks. Es werden z.B. die Wetterdatenimporte der verschiedenen Wetterdatenformate überwacht.
- Ebene 4: Es werden Server und Dienste außerhalb des Serverparks von meteocontrol überwacht (z. B. Überwachung der Datenerzeugungskette für die Berechnung der Globalstrahlung bei der Universität Oldenburg)

2.3 Technologie

Die NAGIOS-Installation im meteocontrol-Serverpark nutzt verschiedene Arten der Kommunikation zwischen den einzelnen Servern (siehe Abbildung 2):

- NAGIOS-Plugins: NAGIOS selbst enthält keine Servicechecks. NAGIOS verwendet sogenannte Plugins, um die eigentlichen Checks durchzuführen. Plugins sind eigenständige Programme, die auch unabhängig von NAGIOS verwendet werden können. Ein NAGIOS-Plugin dient zur Kommunikation zwischen dem NAGIOS-Server und den überwachten Diensten.
- SNMP: SNMP ist ein Protokoll, mit dessen Hilfe Netzwerke verwaltet und gesteuert werden können. Es transportiert Management-Informationen, Status- und Statistikdaten bidirektional zwischen den einzelnen Netzwerkkomponenten und einem Managementsystem. NAGIOS richtet eine Anfrage an einen Agent. Der SNMP-Agent reagiert auf diese Frage mit einer Antwort, die mitteilt, ob die Anfrage erfolgreich beantwortet oder nicht bearbeitet werden konnte. Mit Hilfe von Management [Information](#) Bases (MIB) können über [SNMP](#) managbare Objekte überwacht und administriert werden. Ein [Objekt](#) ist ein abstrakter Begriff für eine Ressource, die einen bestimmten Teilbereich des Netzes darstellt. Diese Ressource kann über eine Vielzahl von Parametern verfügen. Kann dieses Objekt in ein Netzwerkmanagement-Konzept eingebunden werden, so spricht man von einem managbaren Objekt. Die Gesamtheit der managbaren Objekte wird als Management [Information](#) Base bezeichnet.. Bei meteocontrol wird SNMP u. a. für die Überwachung des Server-Loads, PROCS (verschiedene Dienste, z.B. ssh, http, ftp) und des freien Festplattenplatzes eingesetzt.
- NSCA: Sinn von NSCA (NAGIOS Service Check Acceptor) ist es, Systeme, die NAGIOS nicht direkt erreichen kann, von einem anderem System aus zu überwachen. Dies kann ein anderer NAGIOS-Server sein oder ein einfaches Skript. Die Besonderheit hierbei ist, dass Ereignisse vom Client selbst (passive Übertragung aus Sicht des NAGIOS-Servers) an den NAGIOS-Daemon übertragen werden.
- NRPE: Mit NRPE (NAGIOS Remote Plugin Executor) ist es möglich, NAGIOS-Plugins auf entfernten Rechnern auszuführen. Die Anfrage wird an den entfernten Rechner geschickt und der dort laufende NRPE-Dienst führt das lokale NAGIOS-Plugin aus. Das Ergebnis der Überprüfung wird an den NAGIOS-Server zurückgeschickt. Diese Übertragungsart wurde für die Kommunikation mit der

Universität Oldenburg gewählt, wobei der Status der Berechnung der Globalstrahlung übertragen wird.

NAGIOS - Kommunikation im Serverpark

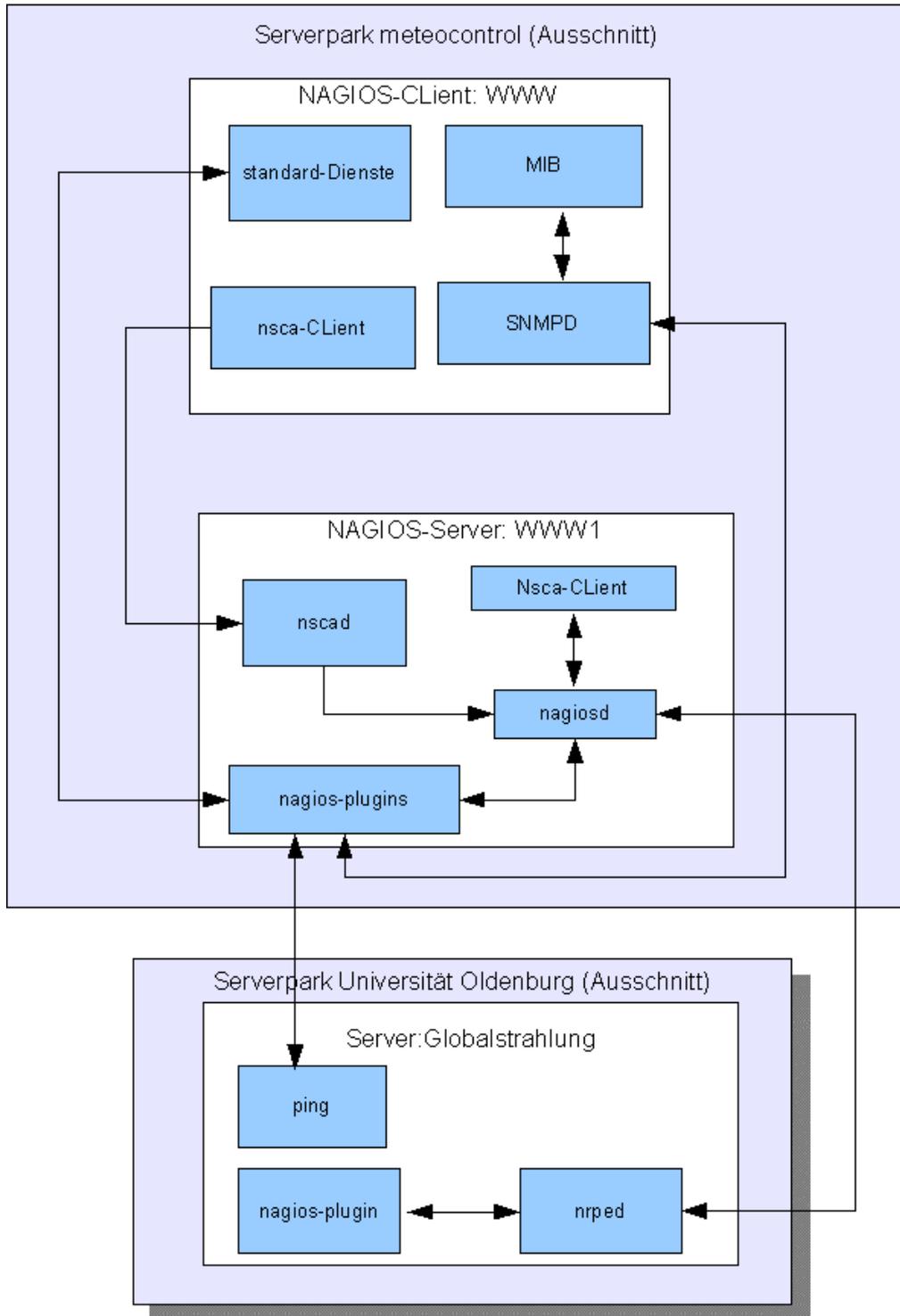


Abbildung 2: Verwendete Arten der Kommunikation

2.4 Alarminformation

Es wurden zwei Wege eingerichtet, sich über den Status des Systems einen Überblick zu verschaffen:

- Sämtliche Statusinformationen der Server und der einzelnen Dienste werden auf einer Webseite visualisiert. Hier ist es auch möglich, Alarme stumm zu schalten. Es wurden verschiedene Systemansichten eingerichtet. Am aussagekräftigsten sind der Überblick über die laufenden Server und deren Dienste (siehe Abbildung 3) sowie der detaillierte Überblick über alle Dienste (siehe Abbildung 4).
- Benachrichtigung per E-Mail. Sämtliche Alarmmeldungen werden an Programmierer und System-Administratoren versendet (siehe Abbildung 5). Eine Benachrichtigung über SMS wird noch hinzukommen.

Status Grid For All Host Groups

METEOCONTROL Server (meteocontrol-servers)

Host	Services
clone-1000.meteocontrol.intra	3ware Connectivity SSH diskspace load procs sync-tcp-873 sunrpc-tcp-111
dfue.meteocontrol.de	Connectivity SSH diskspace load procs
idrisi-1000	Connectivity SSH
image-1000	Connectivity SSH diskspace load procs
import-1000	Connectivity SSH diskspace load procs wetter-import-mos wetter-import-mos-mm
mapserver.meteocontrol.de	Connectivity SSH diskspace load postgres-tcp-5432 procs
master.meteocontrol.de	Connectivity SSH diskspace load procs sunrpc-tcp-111
solem-1000	Connectivity vnc
srvlx090.ehf.uni-oldenburg.de	Connectivity ftp safersun
user-1000	Connectivity HTTP
www.heise.de	Connectivity
www.meteocontrol.de	3ware Connectivity HTTP SSH diskspace global-cleanup-datencache https-tcp-443 load msrdp-tcp-3389 procs solon-webservice-availability
www1.meteocontrol.de	3ware Connectivity FTP SMTP SSH diskspace https-tcp-443 load procs sunrpc-tcp-111 time-tcp-37 wetter-import-blitz

Abbildung 3: NAGIOS-Webinterface: Übersicht über alle Server und deren Dienste

	sunrpc	OK	07-04-2007 15:22:45
	diskspace	OK	07-04-2007 15:23:33
	load	OK	07-04-2007 15:22:27
	procs	OK	07-04-2007 15:22:24
	sunrpc-tcp-111	OK	07-04-2007 15:22:05
solem-1000	Connectivity	OK	07-04-2007 15:23:01
	vnc	OK	07-04-2007 15:23:45
srvlx090.ehf.uni-oldenburg.de	Connectivity	OK	07-04-2007 15:21:48
	ftp	OK	07-04-2007 15:25:32
	safersun	OK	07-04-2007 15:22:13
user-1000	Connectivity	OK	07-04-2007 15:22:14
	HTTP	OK	07-04-2007 15:23:10
www.heise.de	Connectivity	OK	07-04-2007 15:22:22
www.meteocontrol.de	3ware	OK	07-04-2007 15:24:45
	Connectivity	OK	07-04-2007 15:25:34
	HTTP	OK	07-04-2007 15:21:33
	SSH	OK	07-04-2007 15:22:07
	diskspace	OK	07-04-2007 15:23:05
	global-cleanup-datencache	P OK	07-04-2007 02:15:01
	https-tcp-443	OK	07-04-2007 15:24:41
	load	OK	07-04-2007 15:20:51
	msrdp-tcp-3389	OK	07-04-2007 15:21:27
	procs	OK	07-04-2007 15:22:26
	solon-webservice-availability	P OK	07-04-2007 15:15:02
www1.meteocontrol.de	3ware	OK	07-04-2007 15:23:57
	Connectivity	OK	07-04-2007 15:24:46
	FTP	OK	07-04-2007 15:25:47
	SMTP	OK	07-04-2007 15:21:29
	SSH	OK	07-04-2007 15:22:16
	diskspace	OK	07-04-2007 15:23:13
	https-tcp-443	OK	07-04-2007 15:24:03
	load	OK	07-04-2007 15:24:44
	procs	OK	07-04-2007 15:25:47
	sunrpc-tcp-111	OK	07-04-2007 15:21:31
	time-tcp-37	OK	07-04-2007 15:22:23
	wetter-import-blitz	P  CRITICAL	07-03-2007 18:05:31

Abbildung 4: NAGIOS-Webinterface: Übersicht über alle Dienste im Detail (Auszug)

Ansicht: <input type="text" value="Alle"/>	
Betreff	Absender
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** RECOVERY alert - www1.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** RECOVERY alert - www1.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** RECOVERY alert - www1.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** RECOVERY alert - www1.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** RECOVERY alert - www1.meteocontrol.de/load is OK **	Daemon
** RECOVERY alert - www1.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** PROBLEM alert - srvlx090.ehf.uni-oldenburg.de/safersun is CRITICAL **	Daemon
** RECOVERY alert - srvlx090.ehf.uni-oldenburg.de/safersun is OK **	Daemon
** RECOVERY alert - www1.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** PROBLEM alert - srvlx090.ehf.uni-oldenburg.de/safersun is CRITICAL **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** RECOVERY alert - www1.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** RECOVERY alert - www1.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www.meteocontrol.de/load is CRITICAL **	Daemon
** RECOVERY alert - www.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www.meteocontrol.de/load is CRITICAL **	Daemon
** RECOVERY alert - www.meteocontrol.de/load is OK **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon
** PROBLEM alert - www1.meteocontrol.de/load is CRITICAL **	Daemon

Abbildung 5: NAGIOS-Alarmmeldungen per E-Mail

2.5 Konfiguration

Die Konfiguration der einzelnen Dienste findet über mehrere Konfigurationsdateien statt. Hier werden neue Dienste, neue Alarmschwellwerte und die Eskalationsstrategie eingetragen (siehe Abbildung 6).

```

define command{
    command_name    check_raid
    command_line    /usr/lib/nagios/plugins/check_raid $ARG1$
}

# 'ssh_procs' checkt laufende prozesse
define command{
    command_name    ssh_procs
    command_line    /usr/lib/nagios/plugins/check_by_ssh -H $HOSTADDRE
}

define command{
    command_name    check_nrpe
    command_line    /usr/lib/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -c $ARG
}

##### ab hier die snmp-abfragen #####

define command{
    command_name    netsnmp_disk
    command_line    /usr/lib/nagios/plugins/check_netsnmp -C METEOro -H
}

define command{
    command_name    netsnmp_load
    command_line    /usr/lib/nagios/plugins/check_netsnmp -C METEOro -H
}

define command{
    command_name    netsnmp_procs
    command_line    /usr/lib/nagios/plugins/check_netsnmp -C METEOro -H
}

# script muss so mit OID in snmp.conf eingetragen werden:
# exec .1.3.6.1.4.1.2021.8.42 check_3ware /root/check_3ware/check_3ware.pl
# fuer einen zweiten controller muessen dann parameter mit angegeben werde
define command{
    command_name    netsnmp_3ware
    command_line    /usr/lib/nagios/plugins/check_netsnmp -C METEOro -H
}

define command{
    command_name    check_dummy
    command_line    /usr/lib/nagios/plugins/check_dummy $ARG1$
}

```

Abbildung 6: Nagios Konfigurationsfile (Auszug)

3 Fazit

Es wurde eine lückenlose Verfügbarkeitsüberwachung des meteocontrol-Serverparks eingeführt. Server, Serverhardware, Dienste und eigene Skripte (Import) werden

überwacht. Zusätzlich wurde NAGIOS auch beim Projektpartner Universität Oldenburg eingeführt. Jegliche Überwachung von automatisierten Rechenaufgaben innerhalb des WISENT Projektes ist somit möglich.

4 Referenzen

[Barth 2005] Barth, W.: Nagios, System- und Netzwerk-Monitoring, 2005.

**Das MSG-Tailor System
zur Extraktion von Meteosat
Second Generation
value adding Datenprodukten**

WISENT AP 2.2/3.3

Version 1.1
Juli 2008

M. Schroedter-Homscheidt, DLR-DFD
K. Wisskirchen, DLR-DFD

Inhaltsverzeichnis

1. Einführung.....	3
2. Eigenschaften des MSG Tailor Systems	4
Anforderungen	4
Datenparameter	4
3. Standardzugriffe	5
4. Implementierung des MSG-Tailors	5
Beispiel für eine Extraktion eines Ausschnitts in Pixelkoordinaten	6
Beispiel für eine Extraktion mit einem Geopolygon.....	6
Beispiel für eine Extraktion mit der Angabe eines Radius	7
Reference Production Request states	7
5. Abkürzungen	10
Referenzen.....	10

1. Einführung

Im „Data Information and Management System“ DIMS des DLR werden heterogene digitale Datenbestände verschiedener Satellitenmissionen gespeichert. Diese haben i.A. einen für jeden Satelliten spezifischen Raum-Zeit-Bezug oder liegen in einer Geoprojektion vor. Für eine nutzerfreundliche Weiterverarbeitung dieser Satellitenmessungen werden häufig Ausschnitte in Raum und Zeit, sogenannte ‚Raum-Zeit-Kuben‘ benötigt.

Für die Erzeugung der Datenprodukte der Meteosat First Generation Satellitenserie wird derzeit ein „Tailor-System“ eingesetzt, der die in einer Kubus-Struktur abgelegten Rohdaten aus der Product Library des DIMS-Archivs extrahiert und für die weitere Berechnung von Wolkenindex- und Strahlungsprodukten bereitstellt. Eine genaue Beschreibung der Extraktionsmöglichkeiten ist in [1] gegeben.

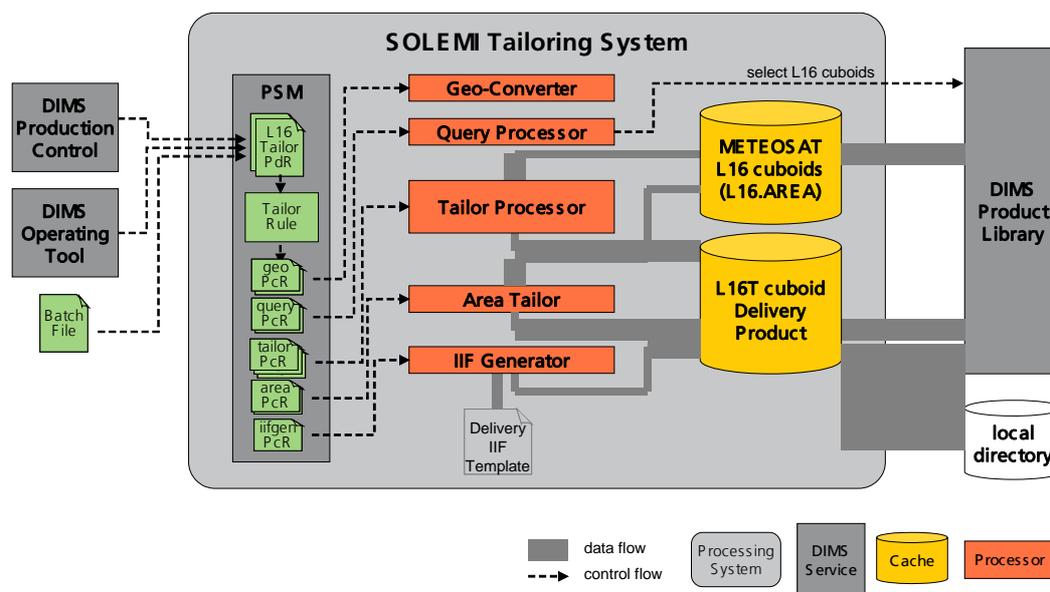


Abb. 1 : Solemi Tailor Processing System für Meteosat First Generation Rohdaten

Mit der neuen Satellitengeneration Meteosat Second Generation (MSG) verändern sich nun die Rahmenbedingungen. Die Rohdaten werden vom Satellitenbetreiber EUMETSAT nicht mehr vollscheibenweise sondern in 8 Segmente unterteilt geliefert. Für die Rohdatenextraktion existiert bereits das umfangreiche Programmpaket SCENES/APOLLO, das Daten aus dem Archiv bestellt, diese passend für die gewünschte Region ausschneidet und Wolken-, Schnee- und Wasserdampfprodukte erstellt. I.A. geschieht dies auf grossen Ausschnitten bzw. der vollen MSG-Scheibe.

Diese Produkte werden anschliessend in einer den Rohdaten der Meteosat First Generation vergleichbaren Cube-Struktur ins Archiv abgelegt. Dasselbe ist in Zukunft für die vom Kooperationspartner Ecole des Mines gelieferten Strahlungsprodukte geplant.

Die Herstellung eines einheitlichen Zugriffs auf die Produkte beider Satellitengenerationen ist ein Ziel dieses Arbeitspakets.

Das Tailor Prozessierungssystem muss nun statt Rohdaten bereits weiter verarbeitete Produktstufen extrahieren und nach Nutzerwunsch passend raum-zeitlich ausschneiden.

Ausserdem sollen für Studien auch Produkte verarbeitet werden können, die auf einer Festplatte ausserhalb des DIMS-Archivs vorliegen. Anwendungsbeispiele sind z.B. der Vergleich verschiedener wissenschaftlicher Algorithmusversionen bei der Vorbereitung einer Reprozessierung des gesamten Archivbestands. In diesem Fall würden die Testprodukte nicht im Archiv sondern auf einer Festplatte bereitliegen.

Daher werden die in [1] spezifizierten Anforderungen entsprechend angepasst.

2. Eigenschaften des MSG Tailor Systems

Anforderungen

Folgende Anforderungen sollen vom MSG Tailor System erfüllt werden:

- Aufbauend auf dem Meteosat First Generation Tailor so weit es geht
- Zeitl-räumliche Ausschnittbildung für MSG Value Adding Datenprodukte
- Nutzergetriebener Einsatz
- Interaktiver und batch-orientierte Mode
- Ansteuerung per DIMS Production Control oder Operating Tool
- Value adding Datenprodukte werden aus der DIMS Product Library oder von einem spezifizierten Festplattendirectory geholt und verarbeitet
- Das Delivery Produkt ist ein angepasster Cube für jeden gewünschten Level 2 Datenparameter
- Delivery Produkte werden entweder temporär in der Product Library für die Auslieferung gespeichert oder auf ein zu spezifizierendes Festplattenverzeichnis geschrieben.
- Eingabeparameter sind das raum-zeitliche Intervall
- Das räumliche Intervall kann in Geo- oder Pixelkoordinaten angegeben werden.
- Falls die Dateigröße des Delivery Produkt 2 GB überschreitet, wird eine zeitliche Unterteilung in Jahre, Monate oder Tage vorgenommen.
- Das Delivery Produkt kann in verschiedenen geographischen Projektionen konvertiert oder in der Satellitenprojektion erstellt werden.
- Das Outputformat ist HDF4.

Datenparameter

Mögliche Datenparameter sind:

- Wolken (Maske, Wolkenstockwerk, Wolkenoberkantentemperaturen, Wolkenoptische Dicke, Wolkenklassifikation, Schneebedeckung)
- Wasserdampfsäulenkonzentration
- In Zukunft gemeinsame Strahlungsprodukte in Kooperation mit der Ecole des Mines

3. Standardzugriffe

Prinzipiell können beliebige raum-zeitliche Ausschnitte gewählt werden. Typische Szenarien sind:

- 10 Jahre Zeitserie für MSG.SEVIRI.L2.CLOUDS.CUBE-PRIM für 1 – 4 cube_ids (Standortanalyse)
- 10 Jahre Zeitserie für MSG.SEVIRI.L2.CLOUDS.CUBE-PRIM für bis zu 10 cube_ids (Länderstudie)
- Vollscheiben-Extraktion eines einzelnen Monats von MSG.SEVIRI.L2.CLOUDS.CUBE-PRIM

- 10 Jahres-Zeitserie für MSG.SEVIRI.L3.TWC-PRIM
- Einzelprodukt MSG.SEVIRI.L2.TWC
- Einzelprodukt MSG.SEVIRI.L3.TWC

- 10 Jahre Zeitserie für zukünftiges MSG.LTTS.EMP.IRRAD.CUBE-PRIM für 1 – 4 cube_ids (Standortanalyse)
- 10 Jahre Zeitserie für zukünftiges MSG.LTTS.EMP.IRRAD.CUBE-PRIM für bis zu 10 cube_ids (Länderstudie)
- Vollscheiben-Extraktion eines einzelnen Monats des geplanten MSG.LTTS.EMP.IRRAD.CUBE-PRIM

4. Implementierung des MSG-Tailors

Folgende Parameter können als Eingabeparameter verwendet werden:

- Missionsname Meteosat Second Generation
- Datenparametername CLOUDS
- Startzeit, Endzeit
- Outputformat HDF4
- Information, ob räumliche Information als Radius, Pixel oder Geokoordinate angegeben wird
- Radius, Region oder Pixelkoordinaten

Es wurden folgende Änderungen implementiert:

- Anpassung an 3712 x 3712 Datenpunkte pro Vollscheibe
- Anpassung an 9 byte pro Pixel statt 1 byte pro Pixel Datenmenge
- Anpassung an 8x8 Cubes mit 1 Monat Länge
- Auswertelogik bzgl. täglicher/wöchentlicher Datenextraktion wurde an die größere Datenmenge angepaßt
- HDF4-Lese- und Schreibroutinen ersetzen das bisherige TAR-Format

Diese wurden in der DIMS-Testumgebung getestet. Es wurden z.B. Zeitserien an einzelnen Punkten; die Aufspaltung in Tages-, Wochen-, Monats- und Jahrespakete und Zeitserien über mehrere Cubes hinweg extrahiert.

Beispiel für eine Extraktion eines Ausschnitts in Pixelkoordinaten

```
<psm-batch>
  <ProductionRequest type="L16-Tailor">
    <ProcessingParameter key="mission">METEOSAT-08</ProcessingParameter>
    <ProcessingParameter key="productParameter">CLOUDS</ProcessingParameter>
    <ProcessingParameter key="startTime">2004-06-03T11:00:00</ProcessingParameter>
    <ProcessingParameter key="stopTime">2004-06-03T19:30:00</ProcessingParameter>
    <ProcessingParameter key="outputFormat">hdf4</ProcessingParameter>
    <ProcessingParameter key="regionType">pixel</ProcessingParameter>
    <ProcessingParameter key="region">
      <GeoPoint latitude="1600" longitude="560"/>
      <GeoPoint latitude="1850" longitude="850"/>
    </ProcessingParameter>
    <OutputProduct type="MSG.SEVIRI.L2T"
      id="((KP:sensor:SEVIRI),(KP:code:L2T))"
      path="/export/home/example" />
  </ProductionRequest>
</psm-batch>
```

Abb. 2 Beispiel-Parametersatz für Pixelkoordinaten

Beispiel für eine Extraktion mit einem Geopolygon

```
<psm-batch>
  <ProductionRequest type="L16-Tailor">
    <ProcessingParameter key="mission"> METEOSAT-08</ProcessingParameter>
    <ProcessingParameter key="productParameter">CLOUDS</ProcessingParameter>
    <ProcessingParameter key="startTime">2004-06-03T11:00:00</ProcessingParameter>
    <ProcessingParameter key="stopTime">2004-06-03T19:30:00</ProcessingParameter>
    <ProcessingParameter key="outputFormat">hdf4</ProcessingParameter>
    <ProcessingParameter key="regionType">geo </ProcessingParameter>
    <ProcessingParameter key="region">
      <GeoPoint latitude="10.546" longitude="25.36"/>
      <GeoPoint latitude="10.056" longitude="20.09"/>
      <GeoPoint latitude="15.189" longitude="20.37"/>
      <GeoPoint latitude="15.489" longitude="26.96"/>
    </ProcessingParameter>
    <OutputProduct type="MSG.SEVIRI.L2T"
      id="((KP:sensor:SEVIRI),(KP:code:L2T))"
      path="/export/home/example" />
  </ProductionRequest>
</psm-batch>
```

Abb. 3. Beispiel Parametersatz für Geopolygon

Beispiel für eine Extraktion mit der Angabe eines Radius

```
<psm-batch>
  <ProductionRequest type="L16-Tailor">
    <ProcessingParameter key="mission"> METEOSAT-08</ProcessingParameter>
    <ProcessingParameter key="productParameter">CLOUDS</ProcessingParameter>
    <ProcessingParameter key="startTime">2004-06-03T11:00:00</ProcessingParameter>
    <ProcessingParameter key="stopTime">2004-06-03T19:30:00</ProcessingParameter>
    <ProcessingParameter key="outputFormat">hdf4</ProcessingParameter>
    <ProcessingParameter key="regionType">radius </ProcessingParameter>
    <ProcessingParameter key="radius">200 </ProcessingParameter>
    <ProcessingParameter key="region">
      <GeoPoint latitude="10.546" longitude="25.36"/>
    </ProcessingParameter>
    <OutputProduct type="MSG.SEVIRI.L2T"
      id="((KP:sensor:SEVIRI),(KP:code:L2T))"
      path="/export/home/example" />
  </ProductionRequest>
</psm-batch>
```

Abb. 4: Beispiel Parametersatz für Radius-Auswahl

Je nach angeforderter Datenmenge werden Ausgabeprodukte in täglicher, wöchentlicher, monatlicher oder jährlicher Unterteilung erzeugt. Dabei wird eine maximale Dateigröße von 2 GB nicht überschritten.

Reference Production Request states

Auszug aus der METEOSAT Tailorbeschreibung [1].

The phase of a production request describes its progress, i.e. how far it is processed. What we call state further describes whether it is failed or successful.

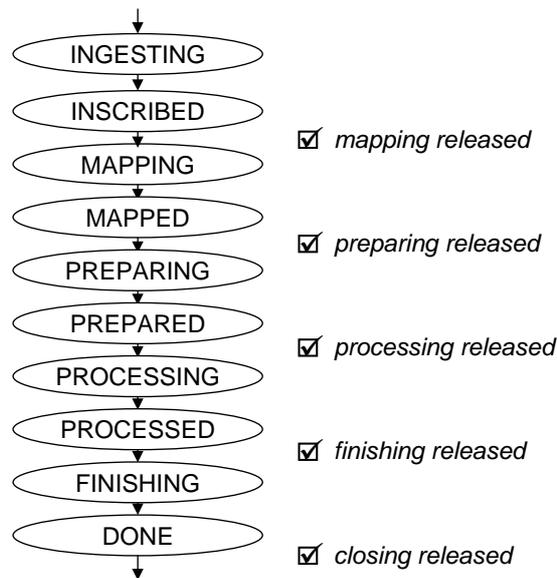


Abb. 6 Production Request Phases and Release Flags

PdR phases and states in the queue should be compared with values in the following tables.

Table 1: Production Request Phases and their Meaning

<i>Phase</i>	<i>Description</i>
INSCRIBED	ingestion processor has finished copying but request is not yet started
MAPPED	the workflow (PcRs) has been generated
PREPARING	space being allocated in cache for inputs and outputs
REALLOCATING	Inputs are retrieved from PL
PREPARED	space is allocated
PROCESSING	actual processing is carried out
PROCESSED	result product is completely generated
FINISHING	products being transferred from PSM cache to PL, allocated space being released
DONE	the request is done, all resources are released

Table 2: Production Request States and their Meaning

<i>State</i>	<i>Description</i>
INITIAL	the request has not yet been started
ACTIVE	the request is currently being handled
WAITING	the request is waiting for some resources or operator interaction (!) like flags to be set
FINISHED	the request is done, all resources are released
CANCELLED	the request has been cancelled by operator interaction
FAILED	the request is failed and requires operator interaction (!)
MAPPING FAILED	rule failed to generate workflow

<i>State</i>	<i>Description</i>
PREPARING FAILED	product allocation in cache failed
REALLOCATING FAILED	Input retrieval from PL failed
PROCESSING FAILED	processing on some processor failed
FINISHING FAILED	output product transfer to Product Library failed
DONE FAILED	request has released its resources and waits for close flag to be set. It can no longer be restarted

5. Abkürzungen

DFD	Deutsches Fernerkundungsdatenzentrum (German Remote Sensing Data Centre)
DIMS	Data Information and Management System
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DMS	Data Management System
EOWEB	Earth Observation on the WEB (Web Gateway of ISIS and MUIS)
FTP	File Transfer Protocol
HDF	Hierarchical Data Format
IIF	Item Information File
PcR	Processing Request
PdR	Production Request
PL	Product Library
PS	Processing System
PSM	Processing System Management
TAR	Tape Archive
TIFF	Tagged Image File Format
WWW	World Wide Web
XML	Extended Markup Language

Referenzen

[1] Processing System Solemi Tailor, Satellite User Manual, DIMS-DLR-SOLEMI-SUM-TAILOR, Iss. 1.0, Oct. 2004, DLR-DFD



**Deutsches Zentrum
für Luft- und Raumfahrt e.V.**
in der Helmholtz-Gemeinschaft



Integration energiemeteorologischer Daten in Geographische Informationssysteme

Arbeitspaket 2.3

Sönke Brummerloh
OFFIS – Institut für Informatik
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de>

1 Einleitung

Im Rahmen der energiemeteorologischen Arbeiten bei DLR-TT hat sich in den letzten Jahren eine Vielzahl heterogener geographischer Daten angesammelt. Dazu gehören klassische Datensätze zur Landnutzung und -beschaffenheit, energiespezifische Datensätze über die Strom- und Gasinfrastruktur sowie zahlreiche Datensätze zu verfügbaren erneuerbaren Ressourcen. Um diese Datensätze weiterhin effizient nutzen zu können, wird eine angemessene Dokumentation und Indizierung benötigt. In AP 2.3 (Integration energiemeteorologischer Daten in Geographische Informationssysteme) wurde daher ein Datenmanagementkonzept und ein entsprechendes Datenmanagementsystem erstellt, um relevante Datensätze effizient verwalten und rasch auffinden zu können.

Das zugrunde liegende Konzept bei der umgesetzten Lösung ist, dass beliebige Dateien zu Datensätzen zusammengefasst und diese mit Metadaten versehen werden können. Metadaten über die zu einem Datensatz gehörenden Dateien, werden in einer XML-Datei gespeichert. Wenn Datensätze auf Dateien aus Geoinformationssystemen (GIS-Dateien) basieren, werden bereits vorliegende Metadaten übernommen. Da die Daten oft in Microsoft Windows basierten GIS-Anwendungen verarbeitet werden, wurde eine Lösung umgesetzt, die es ermöglicht auch weiterhin über eine herkömmliche Verzeichnisstruktur auf einem Netzlaufwerk auf die Dateien zu zugreifen.

Die einzelnen durchgeführten Teilschritte in AP 2.3 waren:

1. Analyse der bisher genutzten GIS-Dateiformate und ihrer Metadaten. Im Einzelnen waren dies die Formate ArcGIS, IDRISI und SOLEMI.
2. Definition eines XML-basierten Datenformats zur Beschreibung der einzelnen Datensätze.
3. Für das XML-basierte Metadatenformat wurden Lese- und Schreiboperationen in C und Perl bereitgestellt. Bei diesen Bibliotheken ist der Zugriff sowohl auf lokale Dateien, als auch über Netzwerk-Protokolle auf entfernte Ressourcen möglich.
4. Auf Grundlage des XML-basierten Metadatenformats wurde eine Anwendung umgesetzt, die einen Suchindex und einen Katalog für die Datensätze bereitstellt. Die Suchkriterien stützen sich dabei auf die in den XML-Dateien gespeicherten Metainformationen. Über diese Anwendung ist es möglich Datensätze zu erstellen und zu durchsuchen. Soweit möglich, werden Metadaten aus der GIS-Datei, auf der ein Datensatz basiert, extrahiert und in den Datensatz automatisch übernommen.

2 Metadatenformat

Das GIS-Metadatenformat ist angelehnt an den Standard ISO 19115, der ebenfalls ein Metadatenformat für GIS-Daten beschreibt. Aus Kompatibilitätsgründen ist eine Verwendung oder zumindest Anlehnung an den ISO 19115 Standard sinnvoll. Allerdings ist das in der ISO 19115 spezifizierte Schema mit ca. 300 Elementen sehr umfangreich, enthält aber trotzdem nicht die Möglichkeit alle Metadaten der zu berücksichtigenden GIS-Formate (IDRISI, SOLEMI und ArcGIS) zu speichern. Daher wurde versucht das GIS-Metadatenformat dem Standard zwar möglichst weit anzunähern, aber ohne dabei das neue Schema unnötig zu verkomplizieren. Aus diesem Grund wurde der Standard an

vielen Stellen stark vereinfacht. Zusätzlich wurde das Element `dataInfo` eingeführt, um Daten aus den verwendeten Formaten unkompliziert übernehmen zu können.

Das Metadatenformat ist ein XML-basiertes Format, dass mit XML-Schema definiert wurde.

Auf der obersten Ebene finden sich folgende Elemente:

Element	Typ	Beschreibung
language	Zeichenkette	Dokumentationssprache in der die Texte verfasst sind. Wird gesetzt falls die Sprache nicht über <code>xml:lang</code> gesetzt wurde. (Format: ISO 639-1)
characterSet	Zeichenkette	Zeichenkodierung der Metadaten (z.B. UTF-8, UTF-16, ISO-8859-1, ISO-8859-2). Muss verwendet werden falls es nicht im Kopf der XML-Datei definiert wurde oder das Format ISO/IEC 10646-1 nicht verwendet wird.
contacts	Contacts	Kontaktinformationen zu den Verantwortlichen der Metadatenfile
dataStamp	Datum	Erstellungsdatum der Metadatenfile
metadataStandardName	Zeichenkette	Name des Metadatenstandards (hier: vIEM-GIS-Metadaten)
metadataStandardVersion	Zeichenkette	Versionsnummer des Metadatenstandards (hier: 1.0)
IdentificationInfos	IdentificationInfos	Basisinformationen zum Datensatz, wie z.B. ein Abstract, Schlagwörtern oder verantwortliche Personen.
distributionInfo	DistributionInfo	Informationen zum Datensatzformat und Bezugsquellen, wie z.B. das Format der GIS-Datei auf der dieser Datensatz basiert und die Dateien die zu diesem Datensatz gehören.
dataQualityInfo	DataQualityInfo	Textuelle Informationen zur Qualität der Daten, die aus den GIS-Dateien importiert wurden.
dataInfo	DataInfo	Metadaten, die größtenteils aus den GIS-Dateien extrahiert wurden.
additionalFiles	AdditionalFiles	Dateien die dem Datensatz zusätzliche zugeordnet sind.

Auf eine detailliertere Beschreibung des Metadatenschemas wird an dieser Stelle verzichtet. Eine detaillierte Beschreibung ist im GIS-Metadaten-Manager-Paket auf <http://sourceforge.net/projects/gis-manager> zu finden.

3 Parser-Bibliotheken

Wünschenswert war es, Bibliotheken zu entwickeln, um Dateien im Metadatenformat möglichst einfach lesen und schreiben zu können. Da beim DLR-TT vor allem die Programmiersprachen C und Perl eingesetzt werden, wurde die Parser-Bibliothek in diesen beiden Sprachen implementiert. Beide Bibliotheken sind in der Lage GIS-Metadaten Dateien vollständig einzulesen und in interne Speicherstrukturen zu überführen. Dabei ist es auch möglich, eine Datei über ein Netzwerkprotokoll wie HTTP oder FTP zu laden. Die internen Speicherstrukturen können modifiziert und wieder in eine GIS-Metadaten Datei gespeichert werden.

Einzelne Elemente können aus der GIS-Metadaten Datei über geeignete Pfadangaben im XPath-Format¹ ausgelesen werden. Auf diese Weise ist es möglich, gezielt einzelne Elemente abzuändern. Die interne Speicherstruktur kann jederzeit gegen das XML-Schema validiert werden.

Die Parser-Bibliothek in C basiert auf den frei verfügbaren Bibliotheken libxml2² und cURL³ und wird wie eine typische Linux-Bibliothek kompiliert und installiert mit:

```
./configure
make
make install
```

Nach der Installation sollte einmal der Befehl `/sbin/ldconfig` ausgeführt werden, um die neue Parser-Bibliothek verfügbar zu machen (wird normalerweise beim Starten von Linux automatisch durchgeführt). Danach kann die Parser-Bibliothek wie eine normale Bibliothek unter Linux in einem C-Programm verwendet werden.

Folgendes Beispiel demonstriert die Funktionsweise der Parser-Bibliothek:

```
#include <gisparser.h>

int main(int argc, char *argv[]) {
    // Before any function from the GISParser is used,
    // startGISParsing() has to be called!
    startGISParsing();

    // Note: The GISParser is working with the xmlDocPtr struct
    // from libxml2, so the functions from libxml2 can be used
    // on xmlDocPtr.
    xmlDocPtr doc;
```

1 <http://www.w3.org/TR/xpath>

2 <http://xmlsoft.org>

3 <http://curl.haxx.se>

```

// First load a GIS-Metadate file into a XML document
doc = loadGISMetadate("test.xml");

// Now the data in the XML document can be modified
setValue(doc, "//GIS_Metadate/metadateStandardVersion",
"1.23");

// After modifications, the XML data can be saved again
saveGISMetadate(doc, "test_out.xml");

// Some cleaning up:
xmlFreeDoc(doc);

// At the end stopGISParsing() has to be called!
stopGISParsing();

return 0;
}

```

Wie beim Parser in C werden in der Perl-Version XPath-Ausdrücke verwendet und im Hintergrund mit den gleichen Bibliotheken gearbeitet. Daher werden die beiden Perl-Bibliotheken XML::LibXML und WWW::Curl::Easy benötigt.

Die Verwendung der Parser-Bibliothek in Perl gleicht der in C, ist aber noch etwas prägnanter, wie folgendes Beispiel zeigt:

```

use strict;
use warnings;

use GISParser;

# First: load an existing GIS-Metadate file
my $doc = GISParser::loadFromFile("test.xml");

# Now the data can be modified
GISParser::setValue($doc,
"//GIS_Metadate/metadateStandardVersion", "12.3");

# The xml document can be checked for validity to an xml schema
if (GISParser::validateXML($doc, "schema/gis-metadate-
schema.xsd")){
    print "GIS-Metadate is valid\n"
}
else {
    print "GIS-Metadate is not valid\n";
}

```

```
# After the data was modified it can be saved into a file
GISParser::saveToFile($doc, "modified_gisdata.xml");
```

4 GIS-Metadatenmanagement

Zum Arbeiten und Verwalten der Datensätze mit ihren Metadaten wurde eine Java-basierte Desktop-Anwendung mit dem Namen GIS-Metadaten-Manager implementiert. Mit dieser ist es möglich mehrere Dateien zu einem Datensatz zusammenzufassen und diesen Datensatz mit Metadaten zu versehen. Dabei werden vorhandene Metainformationen aus bereits vorliegenden Daten extrahiert und in die XML-Metadaten-Datei übernommen. Die Metadaten aller Datensätze können in der Such-Ansicht nach Schlüsselwörtern durchsucht werden. Darüber hinaus kann eine Auswahl an Datensätzen auf Basis der Metadaten in der Katalog-Ansicht Katalog-artig immer weiter eingeschränkt werden.

Beim Einsatz der Anwendung bleibt die bestehende Verzeichnisstruktur des verwendeten Datenspeichers erhalten und wird nur um relativ kleinen XML-Dateien ergänzt. Da es hilfreich ist, Dateien auch manuell verwalten zu können, können Dateien und Datensätze mit dem GIS-Metadaten-Manager verschoben, kopiert und gelöscht werden, sowie aus der Verzeichnisstruktur heruntergeladen werden.

Externe Änderungen im Dateisystem können automatisch vom GIS-Metadaten-Manager erkannt werden. Dies ermöglicht es mehreren Nutzern, in gewissem Umfang parallel auf den selben Dateien zu arbeiten. Falls eine Datei bzw. ein Datensatz allerdings von mehreren Nutzern zur gleichen Zeit geändert wird, ist es wahrscheinlich, dass es zu einem Datenverlust kommt. Um dem entgegen zu wirken, können Verzeichnisse, Dateien und Datensätze für andere Nutzer innerhalb des GIS-Metadaten-Managers gesperrt werden.

Der GIS-Metadaten-Manager basiert auf Java⁴ 1.5, Groovy⁵ 1.5 und Qt Jambi⁶ 4.4. Auf Grund der Lizenzbedingungen von Qt Jambi steht der GIS-Metadaten-Manager unter der GNU General Public License (GPL⁷).

Die Programmiersprache Groovy⁸ wurde für Plugins verwendet mit denen Metadaten aus vorhandenen GIS-Dateien extrahiert werden. Groovy ist eine an Java angelehnte dynamische Skriptsprache. „Angelehnt“ heißt in diesem Fall, dass Java-Quellcode gültiger Groovy-Quellcode ist (aber in der Regel nicht andersherum) und alle Java-Bibliotheken nutzen kann. Wenn gewünscht, kann in Groovy auf weiterführende Konstrukte und einen gegenüber Java einfachere Syntax zurückgegriffen werden. Ein Grund aus dem Groovy zum Implementieren der Plugins im GIS-Metadaten-Manager verwendet wird, ist, dass Groovy-Dateien ausgeführt werden können, ohne dass sie vorher, wie in Java notwendig, explizit in class-Dateien kompiliert werden müssen. Da die Plugins als Quellcode-Dateien vorliegen können unkompliziert neue Plugins entwickelt und dazu bereits vorhandene Plugins als Grundlage verwendet werden.

Qt Jambi ist eine Bibliothek, die es ermöglicht rasch performante grafische Oberflächen (GUIs) zu erstellen. Entwickler werden durch die durchdachte API (Application

4 <http://java.sun.com>

5 <http://groovy.codehaus.org>

6 <http://troll.no/products/qt/jambi>

7 <http://www.gnu.org/copyleft/gpl.html>

8 <http://groovy.codehaus.org>

Programming Interface) und die mitgelieferten Werkzeuge, beispielsweise zum Erstellen von GUIs⁹ und zum Übersetzen der Oberflächenbeschriftungen¹⁰, sehr gut unterstützt.

Der GIS-Metadaten-Manager wurde über mehrere Prototypen hinweg entwickelt. Für jeden Prototypen wurde vor dessen Implementierungsphase die zu integrierende Funktionalität festgelegt und im WISENT-Wiki aufgelistet. Jeder Prototyp wurde im Wiki veröffentlicht und damit DLR-TT zur Begutachtung zur Verfügung gestellt. Durch kurze Release-Zyklen konnten schon früh in der Entwicklung Rückmeldungen gegeben werden und festgestellt werden, was nicht den Erwartungen entsprechend umgesetzt wurde. Auf diese Weise konnten auch fehlende Funktionalitäten identifiziert werden, die zu Beginn noch nicht geplant waren, aber das Arbeiten mit dem GIS-Metadaten-Manager verbesserten.

Zum Starten des GIS-Metadaten-Managers unter Microsoft Windows und Linux können die Dateien `start.bat` bzw. `start.sh` genutzt werden. Unter Microsoft Windows kann der GIS-Metadaten-Manager auch über eine `exe`-Datei gestartet werden. Prinzipiell kann der GIS-Metadaten-Manager auch unter MacOS X mit einem wie in `start.sh` verwendeten Aufruf gestartet und genutzt werden. Lediglich die für MacOS X notwendigen Qt Jambi Bibliotheken müssen beim Aufruf angegeben werden.

Der GIS-Metadaten-Manager gliedert sich in vier Hauptoberflächen und mehrere Dialoge. Im Folgenden werden diese Hauptoberflächen und der Metadaten-Dialog vorgestellt.

4.1 Datei-Ansicht

Die Datei-Ansicht zeigt alle Verzeichnisse und Dateien bis auf die XML-Metadaten-dateien an.

⁹ http://doc.trolltech.com/qtjambi-4.3.4_01/doc/html/com/trolltech/qt/qtjambi-designer.html

¹⁰ http://doc.trolltech.com/qtjambi-4.3.4_01/doc/html/com/trolltech/qt/linguist-manual.html

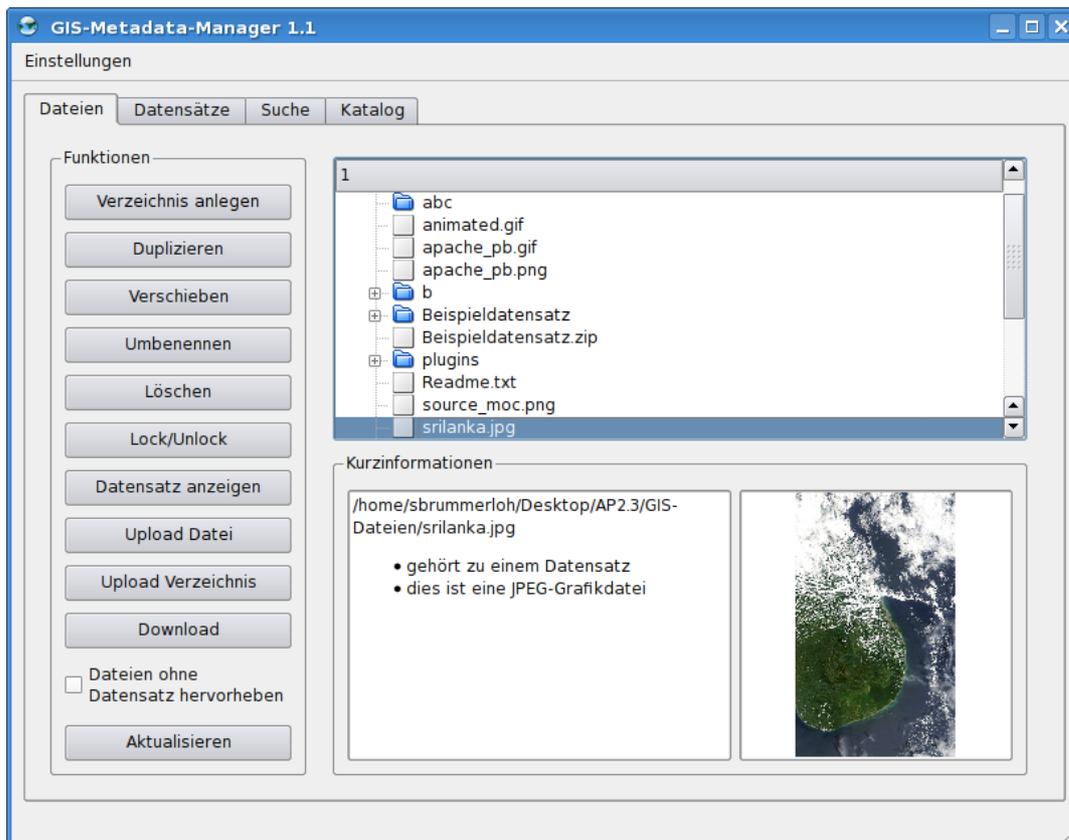


Abbildung 1: Die Datei-Ansicht

Rechts oben wird ein Dateibaum angezeigt. Rechts unten werden Informationen zu dem im Dateibaum selektierten Verzeichnis bzw. der selektierten Datei angezeigt. Falls es sich bei der selektierten Datei um eine Grafikdatei handelt, wird rechts unten eine Vorschau angezeigt. Ein Doppelklick auf die Vorschaugrafik öffnet einen Dialog in dem die Grafik in Originalgröße angezeigt wird. Auf der linken Seite sind verfügbare Befehle aufgelistet. Dort befindet sich auch eine Option mit der alle Dateien, die nicht einem Datensatz zugeordnet sind, hervorgehoben werden.

4.2 Datensatz-Ansicht

In der Datensatz-Ansicht werden nur Verzeichnisse und XML-Metadatendateien angezeigt.

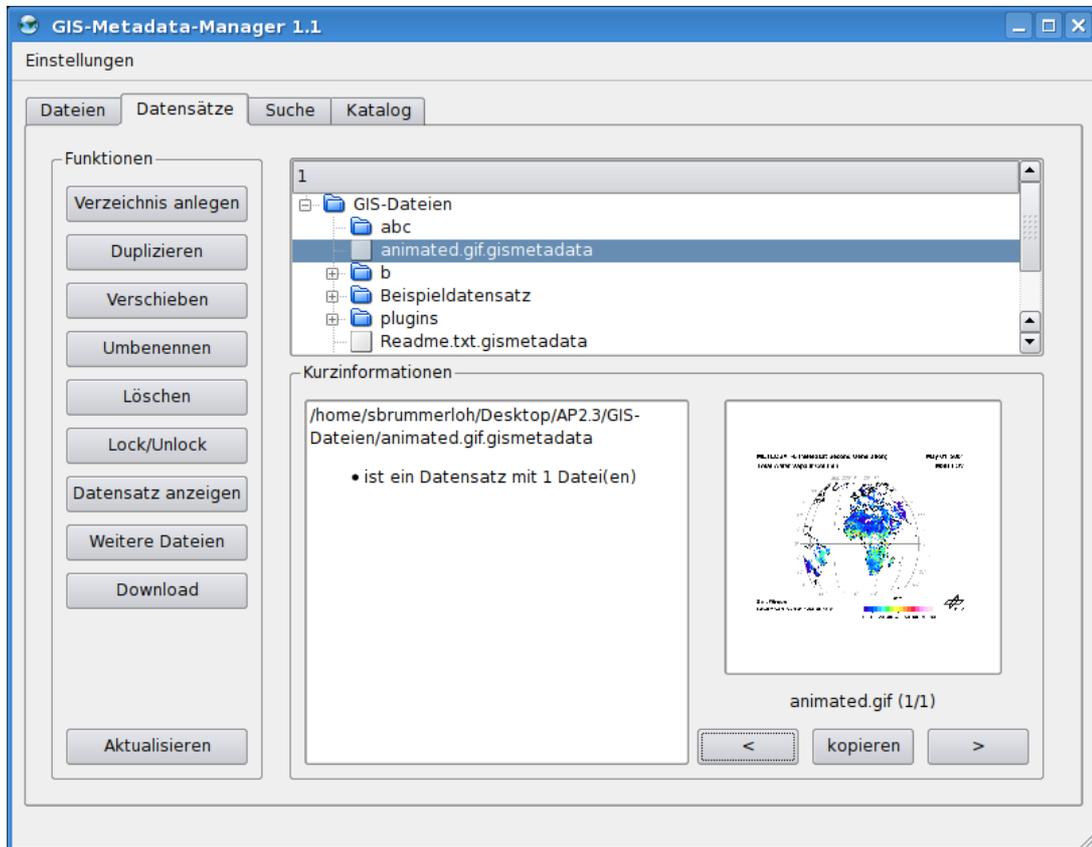


Abbildung 2: Die Datensatz-Ansicht

Rechts oben wird ein Dateibaum angezeigt. Rechts unten werden Informationen zu dem im Dateibaum selektierten Verzeichnis bzw. der selektierten Datei angezeigt. Falls ein Datensatz Grafikdateien enthält, wird eine Vorschau angezeigt. Die Bilder lassen sich über die Schaltflächen mit den Pfeilen durchschauen. Über die kopieren-Schaltfläche wird das aktuell gewählte Bild in die Zwischenablage kopiert und kann in einer beliebigen Anwendung (z.B. Microsoft Word) eingefügt werden. Ein Doppelklick auf die Vorschaugrafik öffnet einen Dialog in dem die Grafik in Originalgröße angezeigt wird. Auf der linken Seite sind verfügbare Befehle aufgelistet.

4.3 Such-Ansicht

In der Such-Ansicht können die Metadaten aller Datensätze nach bestimmten Zeichenfolgen durchsucht werden.

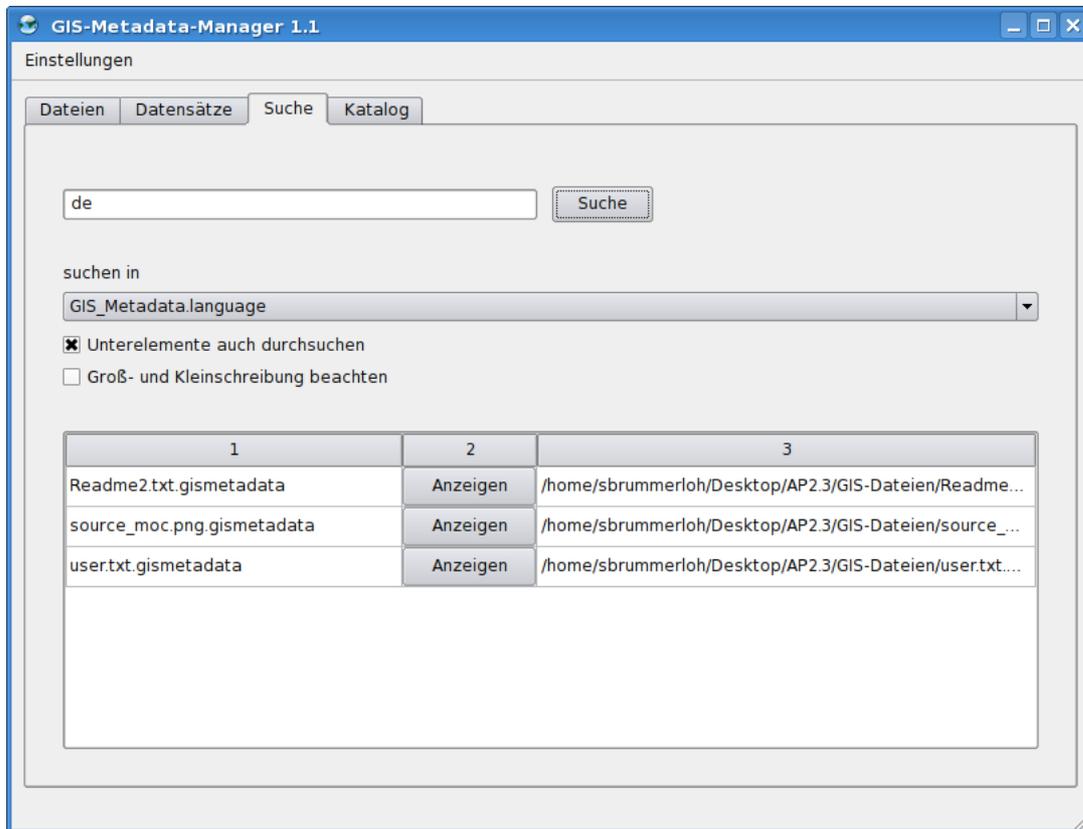


Abbildung 3: Die Such-Ansicht

Über *suchen in* können die zu durchsuchenden Metadatenelemente eingeschränkt werden. Wenn die Option *Unterelemente auch durchsuchen* aktiviert ist, werden alle Unterelemente des gewählten Metadatenelements mit durchsucht. Falls nur genau das gewählte Metadatenelement durchsucht werden soll, darf diese Option nicht markiert sein. Außerdem kann gewählt werden, ob die Groß- und Kleinschreibung des Suchbegriffs berücksichtigt werden soll. Suchergebnisse werden direkt in der Such-Ansicht angezeigt.

4.4 Katalog-Ansicht

Die Katalog-Ansicht ermöglicht es, schrittweise geeignete Datensätze zu ermitteln. Dazu können einzelne Ausprägungen von Metadatenelementen gewählt werden. Angezeigt werden dann jeweils nur die Datensatzdateien, die diese Ausprägungen besitzen.

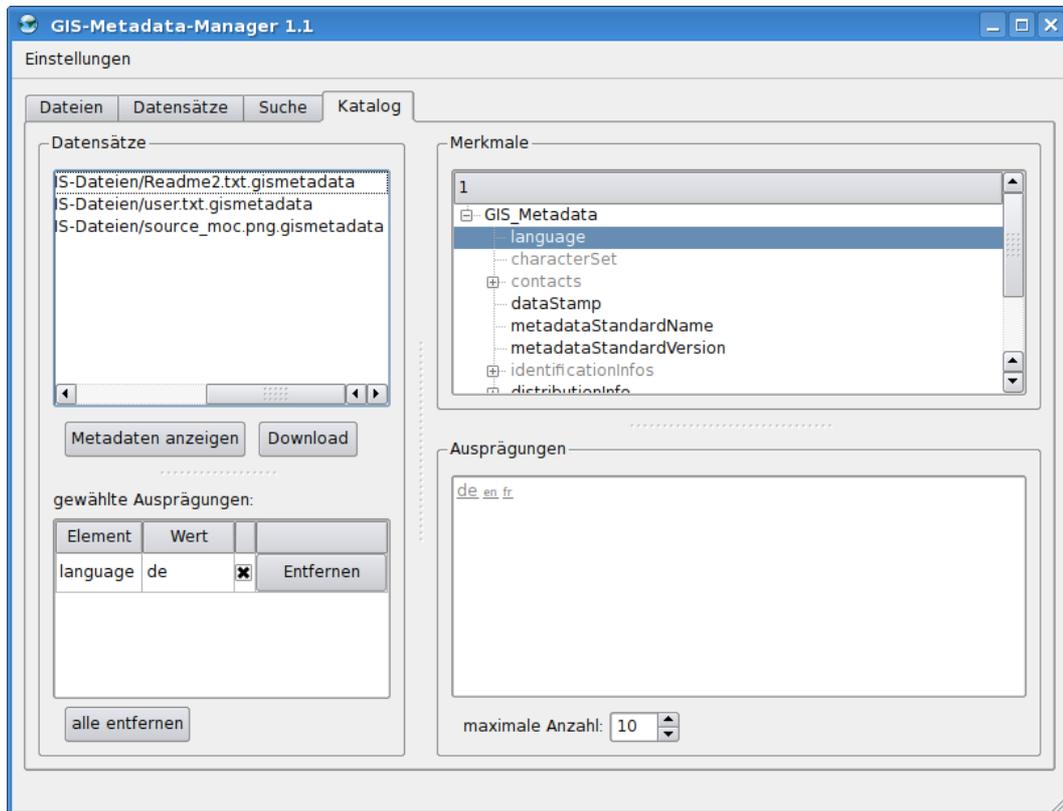


Abbildung 4: Die Katalog-Ansicht mit gewählter Ausprägung

Zum Erzeugen der Datenstruktur für die Katalog-Ansicht müssen zunächst alle Metadaten-dateien eingelesen und verarbeitet werden. Dies hat sich schon frühzeitig in der Entwicklung als sehr zeitaufwendig herausgestellt und den Startprozess des GIS-Metadaten-Managers um eine nicht akzeptable Zeit verzögert. Daher wurde eine Kommandozeilen-basierte Anwendung zum Analysieren des Verzeichnisbaums entwickelt. Diese prüft bei jeder Datei, ob es sich um eine Metadaten-datei handelt. Zusätzlich werden aus den Metadaten-dateien die Metadaten ausgelesen und so aufbereitet, dass sie rasch von GIS-Metadaten-Manager für die Katalog-Ansicht verarbeitet werden können. Die Ergebnisse werden in einer XML-Datei gespeichert, die in den Einstellungen des GIS-Metadaten-Managers angegeben werden kann. Die Funktionalität der Kommandozeilen-basierten Anwendung zum Erzeugen der XML-Datei kann alternativ auch direkt in den Einstellungen des GIS-Metadaten-Managers aufgerufen werden.

4.5 Metadatendialog

Im Metadatendialog wird die Struktur der Metadatenelemente angezeigt. Es ist möglich, die Werte der einzelnen Elemente zu ändern, zu löschen und neue Elemente hinzuzufügen. Der Metadatendialog kann aus allen vier Ansichten des Hauptfensters aufgerufen werden.

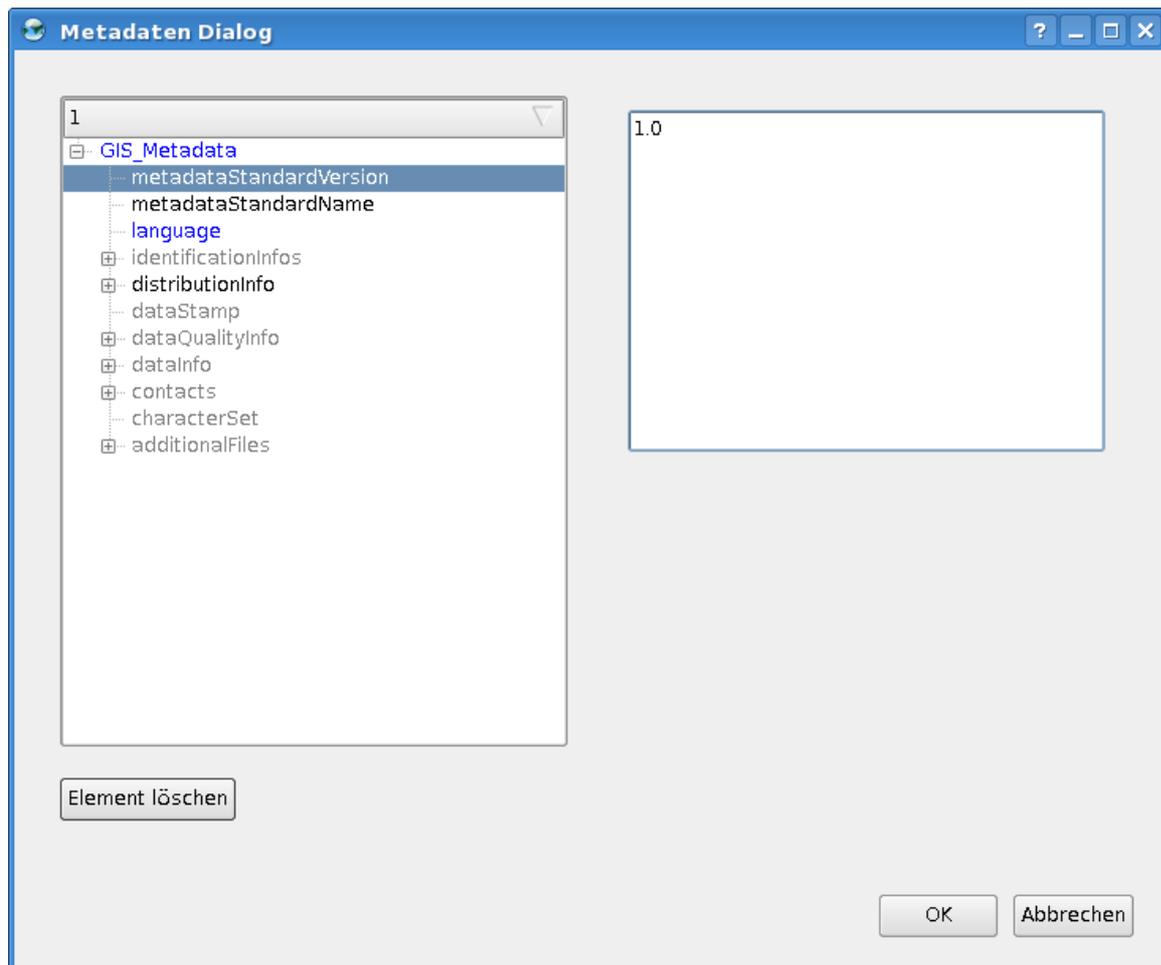


Abbildung 5: Der Metadatendialog zeigt die Metadaten zu einem Datensatz

Auf der linken Seite ist die Struktur der Metadatenelemente zu sehen. Auf der rechten Seite kann der Wert des Elements angegeben bzw. ein neues Unterelement zu dem selektierten Element hinzugefügt werden. Die Eingabefelder werden je nach Datenformat des Elements gewählt. Wenn ein Element ein Datum repräsentiert wird z. B. ein Kalender angezeigt.

5 Resultate und Nachhaltigkeit

Der GIS-Metadaten-Manager hat sich bei DLR-TT zum Verwalten von GIS-Dateien bewährt. Durch den Einsatz des GIS-Metadaten-Managers entstand der Wunsch nach weiterer Funktionalität, die bis zum Ende des WISENT-Projekts zu einem Teil integriert werden konnte. Der GIS-Metadaten-Manager wurde in dem Open-Source-Portal Sourceforge.net¹¹ als Open-Source-Projekt unter <http://sourceforge.net/projects/gis-manager> veröffentlicht und somit der Öffentlichkeit zur Nutzung und Weiterentwicklung zur Verfügung gestellt.

¹¹ <http://sourceforge.net>

Anforderungskatalog für Qualitätssicherungs- metadaten im Datenaustausch zwischen Universität Oldenburg und meteocontrol GmbH

Arbeitspaket 3.1

Thomas Scheidsteger
Carl von Ossietzky Universität Oldenburg
Ammerländer Heerstraße 114-118
26129 Oldenburg

Internet: <http://ehf.uni-oldenburg.de/>

1 Einleitung

Die meteocontrol GmbH hatte im Antrag den Wunsch geäußert, zusätzliche Informationen über die Datenherkunft zu erhalten (also Metadaten). Wir müssen im ersten Schritt ermitteln und festhalten:

- Welche vorhandenen Datenprodukte sind Gegenstand dieser Anforderung?
- Welche Zusatzinformationen sind erwünscht?
- Warum sind ausgerechnet diese Zusatzinformationen erwünscht, für welchen Zweck werden sie gebraucht?
- Liegen die gewünschten Informationen bereits vor und werden nur nicht übertragen?
- Können die gewünschten Informationen zusätzlich berechnet/erzeugt werden?

Diesen Fragen gehen wir in dem vorliegenden Lieferobjekt nach. Im zweiten Schritt soll die Methode der tatsächlichen Anlieferung der Informationen gewählt und umgesetzt werden:

- Reicht eine einfache Formatanpassung der bestehenden Datenprodukte zur Erfüllung der gegenwärtigen meteocontrol-Nachfrage aus?
- Welche Anpassungsarbeiten sind auf der Seite Universität Oldenburg, welche bei meteocontrol durchzuführen?

Diese Fragen werden im Lieferobjekt *Datenaustausch* zu AP 3.1 behandelt.

2 Datenprodukte

Die AG Energiemeteorologie der Universität Oldenburg empfängt die von EUMETSAT zur Verfügung gestellten hoch aufgelösten Bilder (HRV) des MSG-1 bzw. MSG-2 (erster bzw. zweiter Satellit der *Meteosat Second Generation*) und führt Berechnungen bzgl. Wolkenbedeckung und Solarstrahlung am Erdboden auf einem selbst definierten Ausschnitt aus, der den Großteil Europas umfasst.

Es gibt drei wiederkehrende Datenlieferungen der Uni Oldenburg, bei denen Metadaten für meteocontrol relevant sind:

- Tägliche *Near-Real-Time* (NRT)-Zeitreihen (bezogen auf PLZ-Gebiete in Deutschland bzw. auf die Koordinaten von ca. 7000 meteomedia-Wetterstationen weltweit)
- Historische Zeitreihen für **einzelne Standorte** im Europa-Ausschnitt
- Monatliche Strahlungskarten mit den Monatsmitteln für **alle Pixel** des Europa-Ausschnittes

Im Rahmen von WISENT liegt die Priorität eindeutig auf den NRT-Zeitreihen.

3 Zusätzlich gewünschte Metadaten

Folgende Metadaten sind grundsätzlich erwünscht.

3.1 Fehlermaße

Die Fehlermaße sind v.a. im EU-Projekt [PVSAT2](http://www.pvsat.de/) (<http://www.pvsat.de/>) entwickelt worden und dort dokumentiert. Sie werden von meteocontrol GmbH benötigt, um im Rahmen eines Alleinstellungsmerkmals ihren Kunden sog. *Konfidenzintervalle* angeben zu können,

die ihnen eine bessere Einschätzung der Qualität der Daten für ihren jeweiligen Zweck erlaubt.

Für die Berechnung dieser Fehlermaße mussten die operationellen Programme um entsprechende Funktionen erweitert werden.

SysF

Systematische Fehler nach PVSAT2 in W/m^2 (PVSAT Kürzel: *DELTA_G_M_sys*).

StdErr

Standardabweichung / Statistischer Fehler nach PVSAT2 in W/m^2 (PVSAT Kürzel: *DELTA_G_M*).

CSI

Der *Clear Sky Index* (CSI) ist definiert als die Einstrahlung im Verhältnis zur Einstrahlung im unbewölkten Fall. Der Fehler der Einstrahlung hängt vom CSI ab. Der CSI ist u.a. Input für die Fehler-Erkennungsroutinen im PVSAT2-Schema und erlaubt eine Einschätzung der aktuellen Einstrahlungssituation (PVSAT Kürzel: *CLEARSKY*) .

3.2 Prozess-bezogene Qualitätsmetadaten

Hier geht es um die Qualitätsmerkmale des Erzeugungsprozesses der Datenprodukte selbst.

DQ

Datenqualität

- von **vorhandenen** Werten, pro Wert als Integer (0-1)
 - 0 = Daten berechnet aufgrund vollständiger Datenbasis (geschieht aktuell nicht)
 - 1 = Zeitlich interpoliert (bis zu 2 h werden überbrückt)- (s. unter 1)
- von **nicht vorhandenen** Werten (NULL)
 - 0 = Fehlender Wert aufgrund von Prozessierungsfehlern (mit der Chance der Nachberechnung)
 - 1 = Fehlender Wert aufgrund fehlender Rohdaten (ohne Chance der Nachberechnung)
- pro Datei
 - Anzahl der gültigen Werte
 - Anzahl der ungültigen/fehlenden Werte

3.3 Datenlieferungen und zugehörige Metadaten

3.3.1 NRT-Zeitreihen

Aktuelle Globalstrahlungsdaten des Vortages für einzelne Standorte in Deutschland (Postleitzahl) und Europa (Koordinaten) werden in stündlicher Auflösung von der Universität Oldenburg an meteocontrol zur Verfügung gestellt. Die Übermittlung erfolgt in Form von täglich aktualisiert bereit gestellten Zeitreihen (NRT). Neben der Globalstrahlung werden auch Informationen zur Schneebedeckung übermittelt.

Globalstrahlung für alle deutschen PLZ-Gebiete

Messwert	Auflösung	Einheit	Metadaten
Globalstrahlung	Stündlich	W/m^2	StdErr, DQ
Globalsstrahlung Summe	Tagessumme	W/m^2	StdErr, DQ
Tagesenergie= Globalsstrahlung Summe/24	Tagessumme	W/m^2	StdErr, DQ
Anzahl Schneemeldungen bei Clearsky	Tag		-
Anzahl keine Schneemeldungen bei Clearsky	Tag		-
Anzahl Wolkenwerte	Tag		-
Zusätzlicher Wert: CSI	stündlich		-
Zusätzlicher Wert: CSI-Tagesmittelwert	Tagesmittelwert		-

Globalstrahlung für ca. 7000 meteomedia-Wetterstationen

Messwert	Auflösung	Einheit	Metadaten
Globalstrahlung	Stündlich	W/m^2	StdErr, DQ
Globalsstrahlung Summe	Tagessumme	W/m^2	StdErr, DQ
Tagesenergie= Globalsstrahlung Summe/24	Tagessumme	W/m^2	StdErr, DQ
Anzahl Schneemeldungen bei Clearsky	Tag	-	
Anzahl keine Schneemeldungen bei Clearsky	Tag	-	
Anzahl Wolkenwerte	Tag	-	
Zusätzlicher Wert: CSI	stündlich	-	
Zusätzlicher Wert: CSI-Tagesmittelwert	Tagesmittelwert	-	

3.3.2 Historische Zeitreihen

Auf Anfrage von meteocontrol übermittelt die Universität Oldenburg ferner für jede Stunde rückwirkend berechnete Globalstrahlungsdaten in Form von Zeitreihen über den Zeitraum von 1994 bis zum aktuellen Datum für jeweils einzeln definierte Standorte. Diese Daten können entweder die Globalstrahlung oder die Direkt- und Diffusstrahlung enthalten.

Messwert	Auflösung	Einheit	Metadaten
Globalstrahlung SatelLight-Verfahren	Stündlich	W/m^2	SysF, StdErr, DQ
Diffusstrahlung SatelLight-Verfahren	stündlich	W/m^2	StdErr, DQ
Cloudindex	stündlich	StdErr	
Clearskyindex	stündlich	-	
Sonnenelevation	stündlich	-	

3.3.3 Strahlungskarten

Europaweite monatliche Mittelwerte der Globalstrahlung werden in Form einer Karte in einem Binärformat ein Mal pro Monat von der Universität Oldenburg an meteocontrol übermittelt.

Messwert	Auflösung	Einheit	Metadaten
Globalstrahlung	Monat	W/m^2	DQ: ASCII-Liste der Zeit-Slots, für die ganze EUROPA-Bilder fehlen



Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten im Projekt WISENT

Arbeitspaket 3.1

Gerhard Gesell, Marion Schroedter-Homscheidt
Deutsches Zentrum für Luft- und Raumfahrt DLR, Deutsches
Fernerkundungsdatenzentrum DFD
Münchner Straße 20
82234 Weßling
Internet: <http://www.caf.dlr.de>

1. Die allgemeine Bedeutung von Metadaten

Datenprodukte können im Allgemeinen nur dann sinnvoll gespeichert und genutzt werden, wenn eine hinreichende Beschreibung ihres Inhaltes, ihrer Gültigkeit, ihrer räumlichen und zeitlichen Orientierung sowie gegebenenfalls ihrer Entstehung verfügbar ist. Eine solche Beschreibung wird mit Hilfe von Metadaten gegeben. Je nach dem, was mit den Datenprodukten geschehen soll bzw. wozu sie dienen sollen, werden die einen oder anderen Aspekte der Metadaten in den Vordergrund gestellt. Aus der Sicht des Archivators müssen die Metadaten es besonders leicht und eindeutig ermöglichen, ein Datenprodukt in eine Datenbank einzubinden und dort auch wieder zu finden. Aus der Sicht eines Nutzers ist es zunächst besonders wichtig, ein Produkt eindeutig über Metadaten identifizieren zu können oder beispielsweise auch gruppieren zu können. Bis hierhin geht es um so genannte äußere Metadaten, die separat bzw. in Zusatzdateien gespeichert über äußere Eigenschaften der Datenprodukte informieren. Darüber hinaus muss der Nutzer aber auch viele innere Eigenschaften der Datenprodukte kennen, um sie sinnvoll anwenden zu können. Solche inneren Eigenschaften sind zum Beispiel Angaben

- zu Struktur und Umfang von Datenobjekten und Variablen soweit sie nicht a priori festgelegt sind,
- zu physikalischen Größen und Einheiten,
- zu Prozessversionen sowie
- zur Vollständigkeit, Verlässlichkeit und Genauigkeit der Daten, also auch und vor allem zu deren Qualität.

Im Folgenden wird daher zwischen äußeren und inneren Metadaten unterschieden, wobei die inneren Metadaten sich stets auf die Eigenschaften von Einzelelementen oder Ebenen (Layer) eines Datenproduktes beziehen, die äußeren Metadaten dagegen auf das Datenprodukt als ganzes. Weiter ist zu beachten, dass die äußeren Metadaten so weit möglich nach einer ISO-Norm standardisiert sein sollten, um eine Produkt-Interoperabilität zu gewährleisten. Die inneren Metadaten sollen so weit wie möglich ebenfalls einer ISO-Norm entsprechen, jedoch ist es bei der Beschreibung von Einzelobjekten auf sehr tiefer Ebene unter Umständen nicht möglich, eine passende Norm zu finden. Zur formatierten Übermittlung von standardisierten Metadaten kommen Metasprachen wie XML in Frage. Ein im Jahr 2003 definierter Standard für den Inhalt von Metadaten in Geoinformationssystemen wird durch die ISO-Norm 19115 beschrieben. Diese Norm beinhaltet Angaben zur eindeutigen Identifikation, zur Ausdehnung, zur Qualität, zum räumlichen und zeitlichen Schema, zum Referenzsystem und zur Nutzbarmachung. Dazu gibt ISO-19139 die Struktur vor, jedoch ist ISO 19139 noch kein fertiger Standard. Allgemein finden sich unter ISO-191xx Normen zur Geoinformation. Standards für interoperable Metadaten haben die Aufgabe, Metadaten aus unterschiedlichen Quellen nutzbar zu machen. Diese Standards umfassen drei Aspekte. Die Semantik beschreibt die Bedeutung, die meist durch eine Normung festgelegt wird bzw. festgelegt werden sollte. Das Datenmodell legt fest, welche Struktur die Metadaten besitzen können, wobei die Daten die Beschreibung und das Modell die Struktur betreffen. Die Syntax dient schließlich dazu, die durch die Metadaten gemachten Aussagen darzustellen.

In diesem Dokument wird ausschließlich auf die Semantik der für die Qualitätssicherung notwendigen Metadaten eingegangen, d.h. es handelt sich um eine Analyse, welche Informationen zur Qualitätssicherung der Datenprodukte übermittelt werden müssen. Gleichzeitig werden diese Informationen inhaltlich strukturiert und es werden Namen und Bedeutungen der Metadatenobjekte festgelegt bzw. vorgeschlagen. Hierbei wird von einer Aspektematrix (Tabelle 1-1) für die Bedeutung von Metadaten ausgegangen, die die äußeren bzw. inneren Eigenschaften der Datenprodukte mit den Präferenzen der

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Produktlieferanten bzw. –nutzer verknüpft. Die in dieser Matrix aufgeführten Kategorien von Metadaten sollen für die Datenprodukte umfassend sein, d.h. kein notwendiger Aspekt zur Erzeugung, Archivierung und Nutzung der Produkte darf unbeschrieben bleiben. Allerdings ist dabei stets darauf zu achten, dass Metadaten wirklich nur Aussagen über die Eigenschaften und die Entstehung der Datenprodukte machen. Daten, die zur Erzeugung von Produkten über die Eingangsdaten hinaus notwendig sind, etwa Kalibrationskoeffizienten oder Kartenparameter etc. werden „Zusatzdaten“ genannt und zählen selbst nicht zu den Metadaten. Die Eigenschaften von Zusatzdaten haben in der Regel aber erheblichen Einfluss vor allem auf die Produktqualität, weshalb etwa Metadaten zur Qualitätssicherung stets die Rolle der Zusatzdaten und damit auch deren Qualität berücksichtigen sollten.

Metadaten Sichtweise	äußere	innere
Provider	<ul style="list-style-type: none">- Räumlich und zeitliche Orientierung- Orientierung in der Datenbank- Prozessor-ID(s)- Herkunft- etc.	<ul style="list-style-type: none">- Anzahl der Produktebenen- Datentyp und Feldgrößen- etc.
Nutzer	<ul style="list-style-type: none">- Räumliche und zeitliche Orientierung- Versionskontrolle- Produktbeschreibung, soweit sie für das Entpacken notwendig ist- QS: Vollständigkeit des Gesamtproduktes- QS: Gesamtbewertung evtl. aufgetretener Fehler bei der Prozessierung- etc.	<ul style="list-style-type: none">- Anzahl der Produktebenen- Datentyp und Feldgrößen- vorkommende physikalische Größen und Dimensionen- Beschreibung der Produktebenen- QS: Vollständigkeit, Verlässlichkeit und Genauigkeit sowohl ganzer Produktebenen als auch von Einzelwerten- etc.

Tabelle 1-1: Aspektematrix Metadaten

Neben allen *notwendigen* Aspekten zur Erzeugung, Archivierung und Nutzung der Produkte kann es auch Metadateninhalte geben, die zwar hilfreich sind, nicht aber notwendig, um eine ausreichend umfassende Produktbeschreibung zu gewährleisten. Der Begriff „notwendig“ ist hier im Sinne von „zur Beschreibung zwingend erforderlich“ (englisch mandatory) zu verstehen, im Gegensatz etwa zu „optional“. Daher werden im Folgenden alle Metadatenobjekte als „notwendig“ oder „optional“ gekennzeichnet.

2. Qualitätssicherung (QS) mit Hilfe von Metadaten

Ausgehend von bestehenden Produkt- und Prozessierungsstrukturen und den sie beschreibenden Metadatensätzen am DFD werden in diesem Abschnitt notwendige Qualitätssicherungs-Informationen identifiziert und die sie darstellenden Metadaten definiert.

2.1. Ausgangssituation und Zielsetzung

Als Ausgangssituation finden wir am DFD Produkt- und Metadatendefinitionen vor, die sich an den Anforderungen des Data- and Information Management System (DIMS) des DFD-Archivs für Fernerkundungs-Daten und -Produkte, dessen Nutzern und Betreibern orientiert. Die Abbildung 2-1 zeigt die typische Struktur der im DIMS gespeicherten Produkte. Ein solches Produkt besteht aus vier oder fünf Komponenten, wobei die Primärkomponente die eigentlichen (Mess-)Daten enthält, die übrigen Komponenten dagegen spezifische Daten zur Produktvisualisierung, -orientierung, -prozessierung und -

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

qualität. Für das Gesamtprodukt sowie jede einzelne Komponente sollen sowohl äußere als auch innere QS-Metadaten angegeben werden. Die QS-Metadaten können wie alle Metadaten als Attribute abgespeichert werden, jedoch muss für energiemeteorologische Produkte auch die Möglichkeit zur Speicherung von inneren Metadaten zur QS der Einzelwerte als Layer in der Primärkomponente vorgesehen werden.

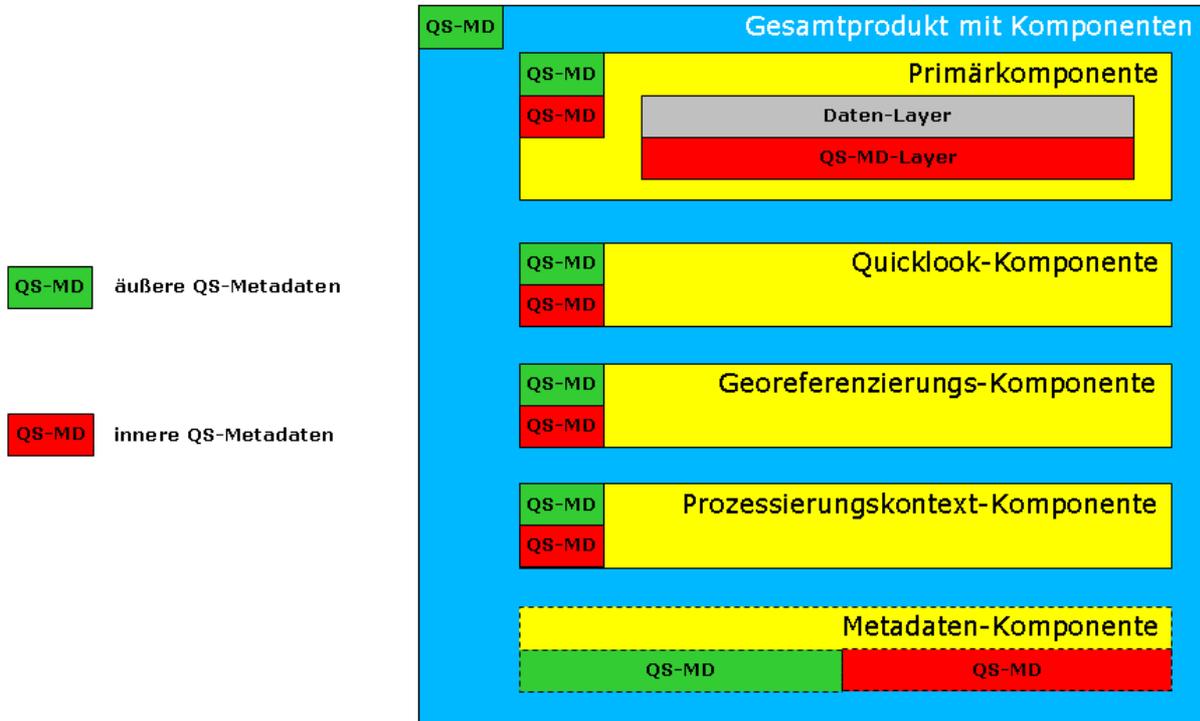


Abbildung 2-1: Typischer Aufbau eines Datenproduktes im DIMS-Archiv des DFD

Für die DIMS-Archivierung von Produkten des World Data Centers for Remote Sensing of the Atmosphere (WDC-RSAT) wurde in einem eigenen so genannten Product Definition Document (PPD) eine DIMS-konforme Produktstruktur und Metadatensammlung beschrieben. Allerdings existiert in dieser Beschreibung bisher nur eine Mindestanforderung an Metadaten zur Qualitätssicherung. Der Ausgangszustand dieser Metadatenbeschreibung für WDC-RSAT-Produkte ist in Tabelle 2-1 bis Tabelle 2-5 gegliedert nach den Produktkomponenten wiedergegeben. Gemäß den in Abschnitt 2.2 erhobenen Nutzeranforderungen werden die WDC-RSAT Metadatenspezifikationen als Ausgangspunkt genutzt, um QS-Metadatendefinitionen erweitert und mit einer eigenen Metadaten-Produktkomponente ergänzt. In Abschnitt 2.3 schließlich werden konkret für die HELIOSAT-3-Prozessierungskette diese QS-Metadaten hinsichtlich ihrer Bedeutung diskutiert und es werden Methoden und Algorithmen zu ihrer Gewinnung vorgeschlagen.

WDC-RSAT Gesamtprodukt

Parameter	Key Param.	Data Type	NOT NULL	Value Range	Description
availability		C	+		QA
reliability		C	+		QA
quality		C	+		QA
revision		C	+	<revision>	
sphere		C	+	Earth	
spatialCoverage			+		
spatialCoverage		C		isGlobal	

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Parameter	Key Param.	Data Type	NOT NULL	Value Range	Description
or					
boundingBox					
westBounding Coordinate				-180.0:180.0	
eastBounding Coordinate				-180.0:180.0	
southBounding Coordinate				-180.0:180.0	
northBounding Coordinate				-180.0:180.0	
temporalCoverage			+		
startTime		Date	+	YYYY-MM-DD THH:mm:ss	
stopTime		Date	+	YYYY-MM-DD THH:mm:ss	
creation			+		
time		Date	+	YYYY-MM-DD THH:mm:ss	date/time of creation
site		C	+	DLR-DFD	
creator		C	+	<processor>	\$LID mission proc system name version (Proc system name from parameter file) e.g. EURAD AQ model
predecessor		list			\$RIDs of input products
auxiliary			+		
doi		C		<DOI>	Digital Object Identifier
mission		C	+		e.g. ENVISAT1
sensor		C	+		e.g. SCIAMACHY, TOMS
code		C	+		e.g. L3
datasets		list			List elements denote datasets in PRIM component. If PRIM contains a single dataset only, the list has length 1.
species		C		<species>	e.g. O3, NO2
units		C		<units>	e.g. Dobson
altitudeCoverage					
top		Float			
bottom		Float			
verticalResolution		C	+	valids <resolution>	e.g. VCD, TCD, SCD
temporalResolution		C`		<nhours>	e.g. daily, monthly (in hours), applied only to lvl4 products
compositeType		C		valids<compositeTypes>	<i>daily, weekly, monthly, yearly</i>
method		C	+	<method>	e.g. Kalman Filtering, Composite, ROSE
methodVersion		C	+	<methodVersion>	Method version
processorName		C	+	<processor>	Processor software name

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Parameter	Key Param.	Data Type	NOT NULL	Value Range	Description
processorVersion		C	+	<version>	Processor software version
algorithm					algorithm used to produce each of the output datasets
algorithmName		C	+	<algorithm>	Algorithm name
algorithmVersion		C	+	<algorithmVersion>	Algorithm version
externalProcessorInput		list			list of tuples identifying processor input originating from external sources.
name		C			product name; should also identify the source
date		Date		YYYY-MM-DD THH:mm:ss	
version		C			

Tabelle 2-1: Metadatendefinition für WDC-RSAT-Produkte in DIMS (Gesamtprodukt)

WDC-RSAT Produkt - Primärkomponente

Parameter	Key Param.	Data Type	NOT NULL	Value Range	Description
availability		C	+		QA
reliability		C	+		QA
quality		C	+		QA
revision		C	+	<revision>	
sphere		C	+	Earth	
spatialCoverage			+		
spatialCoverage		C		isGlobal	
or					
boundingBox					
westBoundingCoordinate				-180.0:180.0	
eastBoundingCoordinate				-180.0:180.0	
southBoundingCoordinate				-180.0:180.0	
northBoundingCoordinate				-180.0:180.0	
temporalCoverage			+		
startTime		Date	+	YYYY-MM-DD THH:mm:ss	
stopTime		Date	+	YYYY-MM-DD THH:mm:ss	
creation			+		
time		Date	+	YYYY-MM-DD THH:mm:ss	date/time of creation
site		C	+	DLR-DFD	
creator		C	+	<processor>	\$LID mission proc system name version (Proc system name from parameter file) e.g. EURAD AQ model

Tabelle 2-2: Metadatendefinition für WDC-RSAT-Produkte in DIMS (Primärkomponente)

WDC-RSAT Produkt - Quicklook-Komponente

Parameter	Key Param.	Data Type	NOT NULL	Value Range	Description
availability		C	+		QA
reliability		C	+		QA
quality		C	+		QA
revision		C	+	<revision>	
sphere		C	+	Earth	
spatialCoverage			+		
spatialCoverage		C		isGlobal	
or					
boundingBox					
westBounding Coordinate				-180.0:180.0	
eastBounding Coordinate				-180.0:180.0	
southBounding Coordinate				-180.0:180.0	
northBounding Coordinate				-180.0:180.0	
temporalCoverage			+		
startTime		Date	+	YYY-MM-DD THH:mm:ss	
stopTime		Date	+	YYY-MM-DD THH:mm:ss	
creation			+		
time		Date	+	YYY-MM-DD THH:mm:ss	date/time of creation
site		C	+	DLR-DFD	
creator		C	+	<processor>	\$LID mission proc system name version (Proc system name from parameter file) e.g. EURAD AQ model
region		C	+	<region>	e.g. global, europe, south pole

Tabelle 2-3: Metadaten definition für WDC-RSAT-Produkte in DIMS (Quicklook-Komponente)

WDC-RSAT Produkt - Geografiekomponente

Parameter	Key Param.	Data Type	NOT NULL	Value Range	Description
availability		C	+		QA
reliability		C	+		QA
quality		C	+		QA
revision		C	+	<revision>	
sphere		C	+	Earth	
spatialCoverage			+		
spatialCoverage		C		isGlobal	
or					
boundingBox					
westBounding				-180.0:180.0	

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Parameter	Key Param.	Data Type	NOT NULL	Value Range	Description
Coordinate					
+ eastBounding Coordinate				-180.0:180.0	
southBounding Coordinate				-180.0:180.0	
northBounding Coordinate				-180.0:180.0	
temporalCoverage			+		
startTime		Date	+	YYY-MM-DD THH:mm:ss	
stopTime		Date	+	YYY-MM-DD THH:mm:ss	
creation			+		
time		Date	+	YYY-MM-DD THH:mm:ss	date/time of creation
site		C	+	DLR-DFD	
creator		C	+	<processor>	\$LID mission proc system name version (Proc system name from parameter file) e.g. EURAD AQ model
epsgCode		C	+	<EPSG Code>	EPSG code specifying the coordinate reference system used in the GeoTIFF file

Tabelle 2-4: Metadatendefinition für WDC-RSAT-Produkte in DIMS (Geografiekomponente)

WDC-RSAT Produkt – Prozessierungskontext

Parameter	Key Param.	Data Type	NOT NULL	Value Range	Description
availability		C	+		QA
reliability		C	+		QA
quality		C	+		QA
revision		C	+	<revision>	
creation			+		
time		Date	+	YYY-MM-DD THH:mm:ss	date/time of creation
site		C	+	DLR-DFD	

Tabelle 2-5: Metadatendefinition für WDC-RSAT-Produkte in DIMS (Prozessierungskomponente)

2.2. QS-Nutzeranforderungen

Die Erhebung der Anforderungen an QS-Metadaten gestaltete sich unerwartet schwierig aus folgenden Gründen:

- Provider und Nutzer haben zunächst ein völlig unterschiedliches Verständnis von Qualitätssicherung an sich sowie von Form und Inhalt der QS-Aussagen im Detail.
- Provider erzeugen QS-Metadaten in einem „Bottom-up-Prozess“. Den Nutzer dagegen interessieren ausschließlich integrale Aussagen und nur gegebenenfalls wird ins Detail gegangen, was einer Top-down-Sicht entspricht.

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

- QS-Metadaten können oft nur bei einer gewissen Grundkenntnis der physikalischen Grundlagen, der angewandten physikalischen, mathematischen und informatischen Methoden sowie der beteiligten Prozessoren und deren Implementation richtig interpretiert werden. Hier hat der Provider gegenüber dem Nutzer einen größeren Wissensvorsprung, der durch zusätzliche Beschreibung der Nutzung der QS-Metadaten durch den Provider gegenüber dem Nutzer ausgeglichen werden muss.

Auf Grund des Wissensvorsprungs des Produkt-Providers muss dieser der Initiator und Koordinator bei der Erhebung der Anforderungen sein. Es gilt, die Sichtweisen von Nutzer und Provider gegenüber zu stellen und sinnvolle Zusammenfassungen und inhaltliche Transformationen zu beschreiben. Dabei werden, ausgehend von Provider-Sicht, zuerst die wichtigsten Klassen von QS-Metadaten bestimmt. Dies geschieht auf Grund einer groben Gliederung dessen, was der Nutzer wünscht und was der Provider bei Kenntnis der Methoden und Prozessoren anbieten kann. Dabei ergeben sich sieben QS-Metadaten-Klassen, die jeweils unterschiedliche Aspekte zum Inhalt haben, nämlich

- Vollständigkeit (auch Verfügbarkeit),
- Konsistenz,
- Genauigkeit,
- Gültigkeit,
- Plausibilität,
- Überprüftheit und
- Bestimmtheit.

Die ersten drei Klassen finden sich auch in der ISO-Norm 19115 wieder, d.h. die Metadaten in diesen Klassen lassen sich gegebenenfalls an diese Norm anlehnen bzw. können der ISO-Beschreibung entnommen werden. Die Klasse „Gültigkeit“ (Validity) tritt in der ISO-19115 als Unterklasse auf, ebenso wie eine Klasse „Correctness“, die auf die energiemeteorologischen Produkte wegen der großteils fehlenden physikalisch-analytischen Überprüfungsmöglichkeiten nicht ohne weiteres abgebildet werden kann. Stattdessen kann die Korrektheit eines Ergebnisses hier mit Hilfe der Klassen Plausibilität und Bestimmtheit abgeschätzt werden.

In Tabelle 2-6 bis Tabelle 2-12 werden bei Angabe der QS-Metadaten-Inhalte Provider- und Nutzersicht gegenüber gestellt. Dabei wird von Nutzerwünschen ausgegangen, die sich bei der Erhebung als ziemlich allgemein herausgestellt haben und sich auch auf Nachfrage bei der Nutzerseite nicht weiter detaillieren ließen. Danach werden von den Nutzern im Projekt WISENT (im Wesentlichen meteocontrol) folgende QS-Metadaten-Angaben gewünscht:

- Beschreibung der systematischen Fehler (BIAS)
- Beschreibung statistischer Fehler (MEAN und RMSE = root mean square error, Wurzel aus dem mittleren Fehlerquadrat)
- Angabe von Konfidenzintervallen und Gütekennzahlen (die erst zu definieren sind)
- Angaben zur Gültigkeit sowohl physikalisch, interpretativ als auch evtl. juristisch
- Beschreibung der Datenqualität, bezogen sowohl auf einzelne Werte als auch auf ganze Datensätze

Als beschreibende Klassen für die Datenqualität werden vom Nutzer überwiegend die Vollständigkeit, Genauigkeit und Gültigkeit gesehen.

Folgende Abkürzungen werden in Tabelle 2-6 bis Tabelle 2-12 benutzt:

<QT = äußere QS-Metadaten für das Gesamtprodukt (Qualität total)

<QP = äußere QS-Metadaten der Primärkomponente

>QP = innere QS-Metadaten der Primärkomponente

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

- <QP** = QS-Metadaten-Layer der Primärkomponente (innere Metadaten)
- <QQ** = äußere QS-Metadaten der Quicklook-Komponente
- >QQ** = innere QS-Metadaten der Quicklook-Komponente
- <QG** = äußere QS-Metadaten der Georeferenzierungskomponente
- >QG** = innere QS-Metadaten der Georeferenzierungskomponente
- <QK** = äußere QS-Metadaten der Prozessierungs-Kontext-Komponente
- >QK** = innere QS-Metadaten der Prozessierungs-Kontext-Komponente
- <QM** = äußerer Metadatenatz der Metadatenkomponente
- >QM** = innerer Metadatenatz der Metadatenkomponente
- <QA** = betrifft äußere QS-Metadaten aller Produktkomponenten
- >QA** = betrifft innere QS-Metadaten aller Produktkomponenten

Vollständigkeit (auch Verfügbarkeit)	
Provider	Nutzer
<QT - Fehlende Produktkomponente oder Komponenten von Zeitreihen (Cubes)	Kombinierte Information zur Nutzbarkeit des Datenproduktes bei Ausfällen → <QT Die Kenntnis, ob fehlende Daten nachprozessierbar sind, ist hierbei wichtig.
<QA - Systematisch fehlende Teile von Komponenten (a priori bekannte zeitliche und räumliche Ausfälle), z.B. Zeilen, Kanäle etc.	
>QA ^QP - Sporadisch fehlende Teile von Komponenten (statistische Ausfälle, Rauschen), z.B. Zeilen, Kanäle, Pixel etc., Einzelwerte und Einzelwertgruppen	
<QK - Vollständigkeit der Methodik, alle Algorithmen angewandt, z.B. Schnee/Wolken-Unterscheidung	Die Nutzer entscheiden selbst aktuell über die Produkt-Nutzbarkeit bei unvollständiger Methodik oder geben dem Provider einen Katalog mit Mindestanforderungen hierzu vor
<QM >QM - Fehlen von Metadaten (QS- und andere Metadaten)	Die Nutzer legen zusammen mit den Providern für jedes Produkt fest, welche QS-Metadaten essentiell und welche optional sind

Tabelle 2-6: Vergleich Provider-/Nutzersicht – Qualitätsklasse „Vollständigkeit“

Konsistenz (auch Aktualität)	
Provider	Nutzer
<QG <QK >QP ^QP Übereinstimmung der Quellen von Ausgangs- und Zusatzdaten (z.B. aktuelle Daten oder Voreinstellungen oder Klimamittel etc.)	Erlaubt eine Einschätzung der Aktualität des Produktes bzgl. raumzeitlicher Start- und Rahmenbedingungen
<QP ^QP Ausreichende Grundgesamtheiten bei Statistiken (Mittelwerte, RMS etc.)	Erlaubt eine Einschätzung der Repräsentativität bzw. des Gewichtes von Statistiken
<QG >QG >QP ^QP Raumzeitliche Repräsentanz gleichmäßig (nicht 3 Pixel hier und 5000 dort)	Die räumliche und zeitliche Abdeckung mit Daten sollte möglichst gleichmäßig sein und Mindestpopulationen sollten erreicht werden. Die Anforderungen hierzu sind vorwiegend Nutzerangelegenheit.
<QM <QK >QM >QK Versionstreue entlang von Zeitreihen	Prozessierungsversionen sollten nicht gemischt werden und wenn doch, dann sollte das bekannt und einschätzbar sein.

Tabelle 2-7: Vergleich Provider-/Nutzersicht – Qualitätsklasse „Konsistenz“

Genauigkeit	
Provider	Nutzer
<QP Genauigkeitsangaben auf Grund von Validierungsergebnissen	Kombinierte Information zur Genauigkeit des Datenproduktes → <QT

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Genauigkeit	
Provider	Nutzer
<QP <QG Genauigkeitsangaben auf Grund von Vergleichen	Kann zur kombinierten Genauigkeitsinformation beitragen und/oder im Einzelfall zusätzliche Angaben erfordern, z.B. bei einer zeitlich befristeten Mess-Störung etc.
<QP <QG Genauigkeitsangaben auf Grund von Abschätzungen	
<QP >QP ^QP Genauigkeitsangaben auf Grund von Mess- und Kalibrationsfehlern	

Tabelle 2-8: Vergleich Provider-/Nutzersicht – Qualitätsklasse „Genauigkeit“

Gültigkeit	
Provider	Nutzer
<QQ <QG <QM >QQ >QG >QM Zeitliche und räumliche Einschränkungen	z.B. Tag/Nacht-Gültigkeit oder regionale Gültigkeit etc.
<QM >QM Zweckbindung und sonstige Aussagen zu Bedingungen der Nutzbarkeit	Nach Vereinbarung zwischen Nutzer und Provider
<QK >QK Zugehörigkeit zu einem bestimmten Prozess mit daraus resultierenden Einschränkungen	
<QM Bedingungen der Interpretierbarkeit	Nach physikalisch-technischer Dokumentation und/oder Erfahrung von Nutzer und/oder Provider
<QM Juristische Einschränkungen	Nach Auflagen durch Provider, Nutzer oder Dritter, z.B. Datenbesitz- und Nutzungsrechte

Tabelle 2-9: Vergleich Provider-/Nutzersicht – Qualitätsklasse „Gültigkeit“

Plausibilität	
Provider	Nutzer
>QP ^QP Ausreißermarkierung	Evtl. kombinierte Kennzeichnung nach Vereinbarung zwischen Provider und Nutzer
>QP ^QP Markierung typischer und untypischer Werte	
>QP ^QP Erkennung unsinniger Werte	
<QK <QM >QK >QM Kompatibilität	Betrifft Prozessorkonfiguration und – ablaufprotokoll sowie die Widerspruchsfreiheit verschiedener QS-Metadaten

Tabelle 2-10: Vergleich Provider-/Nutzersicht – Qualitätsklasse „Plausibilität“

Überprüftheit	
Provider	Nutzer
^QP Überprüftheitsstatus von Einzelwerten in Bezug auf definierte Klassen, Kategorien, Typisierungen etc.	Einfache Aussagen, ob und auf welche Klassen, Kategorien, Typisierungen etc. überprüft bzw. nicht überprüft wurde
<QM >QM Überprüftheitsstatus in Bezug auf die Weiterverwendung (durch Tests)	z.B. Test auf Erfüllung von Bedingungen wie Definitionsbereiche etc. für Eingabedaten
<QM Überprüftheitsstatus von Zeitreihen (durch Tests)	Gemäß vereinbarter Anforderungen
<QK Status in Bezug auf die Prozessierung (operationell, experimentell, prototypisch etc.)	Betrifft Glaubwürdigkeit, Robustheit und wissenschaftliche Akzeptanz des Verfahrens
<QM Existenz von Validierungen oder anderen Referenzen	Betrifft Glaubwürdigkeit und wissenschaftliche Akzeptanz des Verfahrens

Tabelle 2-11: Vergleich Provider-/Nutzersicht – Qualitätsklasse „Überprüftheit“

Bestimmtheit	
Provider	Nutzer
[^] QP (Grad der) Klassifikationssicherheit	Wahrscheinlichkeitsangaben für die Klasseneinordnung für den Nutzer
^{>} QM Wahrscheinlichkeit alternativer Interpretationen	
[^] QP Statistische Signifikanzen	Verschiedene statistische Tests
[^] QP Unbestimmtheit im Sinne von Ergebnislosigkeit	Informationen, falls kein Ergebnis trotz Anwendung physikalischer Methoden erzielt wurde

Tabelle 2-12: Vergleich Provider-/Nutzersicht – Qualitätsklasse „Bestimmtheit“

2.3. QS-Metadaten für energiemeteorologische Produkte

In diesem Abschnitt werden die QS-Metadaten, die für energiemeteorologische Produkte benötigt werden, spezifisch beschrieben. Dies beinhaltet erstens eine Inhaltsangabe, zweitens die Zugehörigkeit zu den Qualitätsklassen, drittens eine Prioritätsangabe und außerdem einen Hinweis zur Standardisierbarkeit hinsichtlich der Norm ISO-19115. Darüber hinaus werden Verfahren angegeben oder empfohlen, wie die QS-Metadaten abgeleitet werden können.

Als konkretes Anwendungsbeispiel für QS-Metadaten dient die HELIOSAT-Prozesskette (Abbildung 2-2) bis zu der Stufe, wo die Wolken- und Wasserdampfprodukte (MSG.SEVIRI.CLOUDS und MSG.SEVIRI.TWC) per FTP an T&M versandt werden. Die Kette beinhaltet bis dorthin die Prozessierung der SEVIRI-Daten von Level-0 zu Level-1.5, wofür der SCENES-Prozessor von EUMETSAT verwendet wird, die zusätzliche Einbeziehung von Ozon- und Aerosoldaten, letztere aus dem SYNAER-Prozessor, sowie den HELIOSAT-3-Prozessor für Atmosphärenparameter. Letzterer besteht aus dem APOLLO/SEVIRI-Prozessor, der die Wolkenerkennung und die benötigten Wolkenparameter liefert, sowie aus dem Prozessorteil, der die eigentlichen Atmosphärenprodukte fertig rechnet und zusammenstellt. Der APOLLO-Prozessor wird mit Eingabedaten aus SCENES gespeist und anschließend werden die APOLLO-Wolkenprodukte noch einer Nachprozessierung unterzogen und in die HDF-Produktdateien gepackt (Abbildung 2-3). Welche notwendigen und optionalen QS-Metadaten für die Produkte der HELIOSAT-Prozesskette erzeugt werden sollen, wird anhand der Betrachtung der Eingangsdaten, der prozeduralen Einzelschritte und Kontextgegebenheiten, der Qualität von Zwischenergebnissen und des a priori bekannten Wissens über die Verfahrensgüte erörtert und die beschreibenden QS-Metadaten werden definiert.

QS-Metadaten werden grundsätzlich auf drei Wegen, je nach Qualitätsklasse, erzeugt, nämlich

- nach physikalisch-analytischen Verfahren, z.B. durch Validierung und nach dem Fehlerfortpflanzungsgesetz,
- nach numerisch-statistischen Verfahren, z.B. durch Betrachtung von Abweichungen bei Vergleichen und Berechnung von Signifikanzen,
- sowie rein empirisch nach Erfahrungswerten, Vollständigkeitsanforderungen und Plausibilitätsbetrachtungen.

Bei der Erzeugung energiemeteorologischer Produkte ist der Provider tendenziell eher auf Statistik und Empirie angewiesen und wird seltener echte physikalische Qualitätsmaße oder Validierungen zur Verfügung haben. Die dabei erforderlichen Erfahrungswerte stammen naturgemäß meist vom Provider. Der Nutzer wird meist nicht von sich aus über einen solchen Erfahrungsschatz verfügen und muss sich daher auf die Providerangaben verlassen. Andererseits kann der Provider bei Qualitätsaussagen, die auf Erfahrung und Empirie beruhen, keine weiter reichenden Garantien gegenüber dem Nutzer übernehmen.

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Die QS-Metadaten geben also nur dann im Sinne einer Garantie belastbare Informationen, wenn sie ausschließlich mit physikalischen Verfahren erzeugt wurden oder auf Validierungen beruhen. In allen anderen Fällen geben QS-Metadaten nur Anhaltspunkte und (begründete) Einschätzungen der Güte energiemeteorologischer Produkte. Dieser Aspekt wird im Abschnitt 4 noch weiter diskutiert.

Bei der Definition der QS-Metadaten für die HELIOSAT-Prozesskette wird im Folgenden integrativ vorgegangen. Zunächst werden modular für einzelne Prozessierungsschritte und Zwischenergebnisse die QS-Informationen identifiziert, die verfügbar gemacht werden können. Danach werden die Einzelinformationen sinnvoll anhand der Qualitätsklassen und den Nutzeranforderungen Rechnung tragend zu QS-Metadaten verdichtet. Dabei darf diese Vorgehensweise als allgemeine Empfehlung für die Definition von QS-Metadaten energiemeteorologischer Produkte verstanden werden.

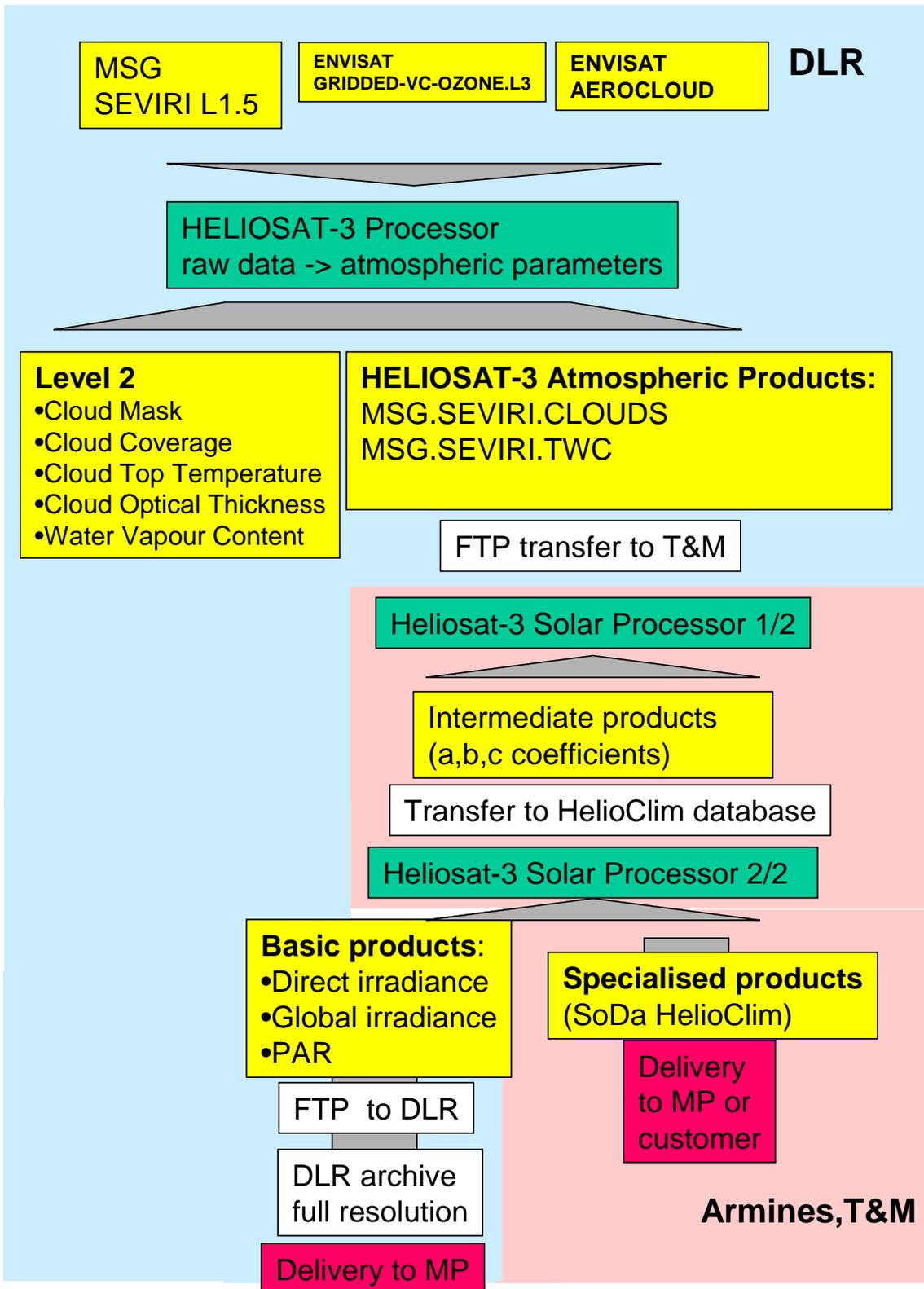


Abbildung 2-2: HELIOSAT-3 Prozesskette bei DFD und EdM

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

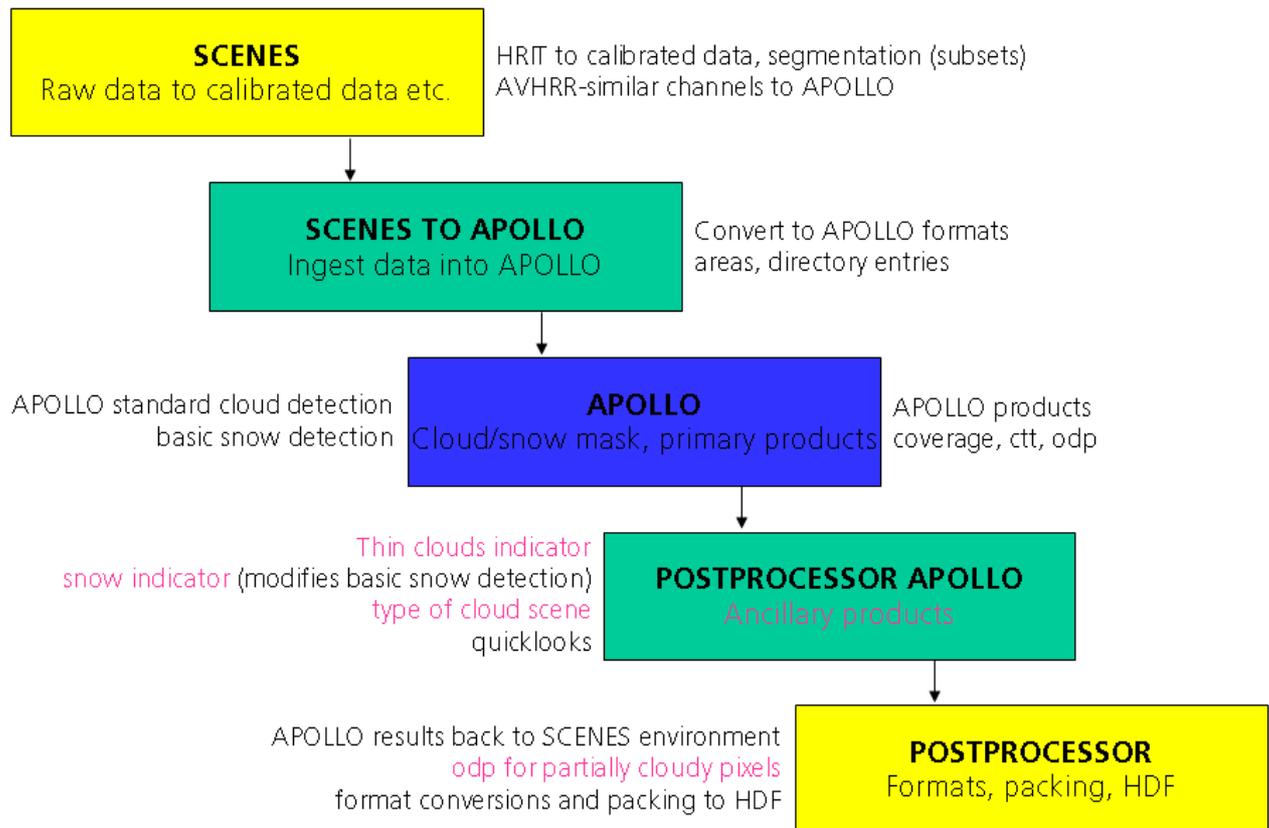


Abbildung 2-3: SCENES/APOLLO-Prozessierungskette für Wolkenparameter beim DFD

2.3.1. QS-Informationen der HELIOSAT-Prozesskette (DFD-Teil)

In Tabelle 2-13 werden alle im DFD-Teil der HELIOSAT-Prozesskette verfügbaren QS-Informationen aufgelistet und dabei nach Prozessierungsschritten und (Zwischen)Produkten gegliedert. Danach werden alle diese Informationen unter Angabe ihrer Referenznummer näher erläutert. QS-Informationen, die auf echten physikalischen Analysen oder Validierungen beruhen oder darauf prinzipiell begründet werden können, sind farblich hervorgehoben. Grün bedeutet, dass eine physikalische Herleitung oder Validierung der Genauigkeit tatsächlich existiert, blau, dass dies zwar möglich, jedoch (noch) nicht durchgeführt ist. Alle anderen Informationen beruhen auf rein statistischen Ableitungen oder empirischen Erfahrung. Der Begriff „Güte“ wird im Folgenden weitgehend gleichbedeutend mit dem Begriff „Genauigkeit“ verwendet, bezeichnet also eine absolute oder relative Fehlerangabe. Ein kleiner Unterschied besteht lediglich darin, dass „Güte“ sich mehr auf eine ganze Methodik, „Genauigkeit“ aber mehr auf einzelne Parameter bezieht.

Prozessierungsschritt	Zwischenprodukt	Verfügbare QS-Information	#
ENVISAT Ozondaten Input	Gridded vc ozone	Raumzeitliche Aktualität der vertical columns	1.1.1
		Klimawerte, Schätzwerte verwendet? (Vollständigkeit)	1.1.2
		Güte falls bekannt	1.1.3
SYNAER-Prozessor Aerosol Input	AOT	Raumzeitliche Aktualität (Verfügbarkeit)	2.1.1
		Klimawerte, Schätzwerte verwendet? (Vollständigkeit)	2.1.2
		Güte und Zuverlässigkeit falls bekannt	2.1.3
SCENES Level 1.5	Kalibrierte Kanäle – Reflexionsgrade und	Fehlzeilen und Badvalues (Vollständigkeit)	3.1.1

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Prozessierungsschritt	Zwischenprodukt	Verfügbare QS-Information	#
	Temperaturen	Kalibrationsgüte	3.1.2
	BRDF-Winkel	Verfügbarkeit	3.2.1
	Land/Wasser-Maske	Verfügbarkeit	3.3.1
	Georeferenzierung	Verfügbarkeit	3.4.1
	Subset-Information	Verfügbarkeit	3.5.1
SCENES to APOLLO	Georeferenzierung	Plausibilität und Genauigkeit	4.1.1
	BRDF-Winkel	Plausibilität und Genauigkeit	4.2.1
	APOLLO Areas und Directories	Vollständigkeit	4.3.1
		Plausibilität der Wertebereiche	4.3.2
APOLLO	BRDF-Winkel	(nicht oder eingeschränkt) nutzbare Bereiche	5.1.1
	Land/Wasser/Sunglint-Maske	Einschränkungen Sunglint	5.2.1
		Einschränkungen Küste	5.2.2
	Wolken/Schnee-Maske, Erkennungsalgorithmen	Vollständigkeit Algorithmen	5.3.1
		Aktualität der dynamischen Schwellen	5.3.2
	Bedeckungsgrad	nicht zum Wolkenstockwerk zugeordnete Pixel	5.4.1
		Verfahrensgüte	5.4.2
	Optische Dicke Wolken	Plausibilität	5.5.1
		Zuverlässigkeit in Abhängigkeit von den BRDF-Winkeln	5.5.2
		interpolierte Werte bei Teilbewölkung	5.5.3
		Verfahrensgüte	5.5.4
	Temperatur bzw. Höhe der Wolkenobergrenze	Aktualität des Temperaturprofils bei Bestimmung der Wolkenhöhe	5.6.1
	Postprozessor APOLLO	Indikator dünne Wolken	Plausibilität bei Vergleich mit wolkenoptischer Dicke
Wolkentyp		unklassifizierbare Pixel	6.2.1
		Grad der Klassifikationssicherheit	6.2.2
Schnee-Indikator		Bestimmtheit der benutzten dynamischen Schneeschwelle	6.3.1
		Schnee-Indikator selbst als Schnee-Wahrscheinlichkeitsmaß	6.3.2
Quicklooks	Verfügbarkeit	6.4.1	
Postprozessor Atmosphärenprodukte	Optische Dicke Wolken für teilbewölkte Pixel	Kennzeichnung der interpolierten Werte	7.1.1
	TWC	Plausibilität bei Vergleich mit Wolkentyp und evtl. mit LWP	7.2.1
	Metadatensammlung	Vollständigkeit der essentiellen (QS)-Metadaten	7.3.1
Kombinierte QS-Metadaten-Informationen als Vorbereitung für Transformation in QS-Nutzerinformationen		7.3.2	

Tabelle 2-13: Verfügbare QS-Informationen der HELIOSAT-Prozesskette (DFD-Teil)

Im nächsten Abschnitt wird im Detail auf die in Tabelle 2-13 aufgeführten verfügbaren QS-Informationen eingegangen:

1.1.1

Unter der Aktualität der verwendeten vertikalen Ozonsäulen ist die zeitliche und räumliche Nähe zu den übrigen für die Erstellung des Gesamtproduktes benötigten (Satelliten-)Daten zu verstehen. Die Qualität des Endproduktes wird davon beeinflusst, wie weit die Ozonsäule vom jeweiligen Messwert (Pixel) entfernt ist und wie viel Zeit zwischen der Erhebung der Ozoninformation und der anderen Daten liegt. Falls eine räumliche und/oder zeitliche Interpolation verwendet wurde, bedeutet ein Vermerk hierzu ebenfalls eine Qualitätsinformation.

1.1.2

Falls keine vertikalen Ozonsäulen verfügbar sind, können eventuell auch Klimamittelwerte oder Schätzwerte benutzt werden. Da die aktuellen Ozonsäulen von mittleren mehr oder weniger stark abweichen können, ist diese Information zur Beurteilung der Qualität des Endproduktes wichtig. Es können einzelne Werte fehlen oder ganze Ozon-Datensätze. Diese Angaben zur Vollständigkeit der Ozoninformationen hängen auch ab von deren Aktualität, d.h. ab einer bestimmten zeitlichen und räumlichen Distanz zwischen Produkt-Referenzzeit und -Ort und der nächsten Ozonsäule gilt die Ozoninformation als nicht aktuell und muss durch A-Priori-Wissen ersetzt werden. Welche Schwellen hierfür verwendet werden, kann allein im Ermessen des Providers liegen oder zwischen Provider und Nutzer vereinbart sein. 1.1.1 und 1.1.2 sollten stets gemeinsam und abgestimmt angegeben werden.

1.1.3

Ozonsäulen-Daten werden laufend mit Bodenmessungen des globalen Dobson und Brewer Netzwerkes verglichen. So können Effekte erkannt und eingeschätzt werden, die aus jahreszeitlichen Effekten, aus Sonnenstandeffekten oder aus der Sensordegradation resultieren. In der Regel sind die Vergleichsergebnisse geophysikalisch konsistent und es zeigen sich beim Vergleich mit den Bodenmessungen in Abhängigkeit vom Sonnenzenit Abweichungen zwischen 2% und 5%. Validierungs-Methoden und -Ergebnisse hierzu sind in Lambert et al, 2004, und von Bergen et al, 2004, publiziert.

2.1.1

Für die Aktualität der Aerosolinformation gilt gleiches wie unter 1.1.1 für die vertikalen Ozonsäulen beschrieben. Allerdings können nicht verfügbare aktuelle Aerosolinformationen nicht einfach durch Schätzwerte oder Klimamittel ersetzt werden. Das liegt daran, dass Klimamittel für den Aerosolgehalt der Atmosphäre einerseits auf einer schmalen Datengrundlage beruhen, andererseits Aerosol sehr viel inhomogener verteilt ist und auch zeitlich stärker variiert. Bei der Aerosolinformation wird also die Aktualität und Vollständigkeit zu einer QS-Information zusammengefasst.

2.1.2

Analoges wie für Ozonsäulen unter 1.1.2 gilt auch für aerosol-optische Dicken.

2.1.3

Die Aerosol-Ergebnisse aus SYNAER ENVISAT wurden validiert gegenüber multi-spektralen bodengebundenen Sonnen-Photometer-Messungen aus dem AERONET-Netzwerk. Die Resultate sind unter <http://www.gse-promote.org> → "Air Quality" → "Satellite-based Particulate Matter Demonstration Service" → "European Multi-annual Records" dargestellt.

3.1.1

Im Level-1.5-Prozessor der SCENES-Software werden die MSG-SEVIRI-Rohdaten zu Strahldichten kalibriert und daraus für die solaren Kanäle Reflexionsgrade und für die thermalen Kanäle Temperaturen erzeugt. Auf der Ebene der Level-1.5-Daten genügt es, fehlende Pixel und Zeilen sowie Fehler bei der Kalibration und der Ableitung der Reflexionsgrade und Temperaturen (Badvalues) zu kennzeichnen. Die Plausibilität, das heißt der physikalische Sinn der abgeleiteten Werte wird auf der Schnittstelle zu APOLLO (Gruppe 4.x.x) überprüft.

3.1.2

Die Genauigkeit der verwendeten Kalibrationskoeffizienten wird von EUMETSAT angegeben. Eine Angabe zur Aktualität der Kalibrationskoeffizienten wird empfohlen, da Sensorsysteme altern, die Empfindlichkeit sich dabei kontinuierlich verändert, aber die Koeffizienten nur epochenweise aktualisiert werden.

3.2.1, 3.3.1, 3.4.1, 3.5.1

Neben den SEVIRI-Level-1.5-Daten liefert SCENES auch noch für die weitere Prozessierung notwendige Zusatzdaten.

Die BRDF-Winkel (3.2.1), das sind Sonnen- und Satellitenzenit sowie relativer Azimut zwischen Sonne und Satelliten werden in APOLLO für die Korrektur der Reflexions-Anisotropie benötigt. Die Land/Wasser-Maske (3.3.1) ist nötig, weil die Wolkenerkennungsalgorithmen über Wasser und über Land verschieden sind. Eine Zuordnung von Erdkoordinaten zu jedem Pixel wird trivialerweise auch gebraucht, d.h. SCENES liefert diese Georeferenzierung (3.4.1) in Form von Dateien mit geografischen Breiten- und Längenangaben für jedes Pixel. Zusätzlich wird auch noch eine Angabe zu Lage und Größe des zu bearbeitenden Ausschnittes (Subset, 3.5.1)) von SCENES an APOLLO übergeben. Im Prozessierungsabschnitt SCENES werden diese Zusatzdaten noch nicht inhaltlich auf ihre Qualität geprüft, sondern es wird hier zunächst nur die Verfügbarkeit sicher gestellt. Dabei geht es weniger darum, ob die Daten überhaupt vorliegen, denn liegen sie nicht vor, kann nicht mehr weiter prozessiert werden. Vielmehr ist bezüglich der Qualität des Endproduktes wichtig, welche Zusatzdaten-Versionen verfügbar sind.

4.1.1

In dem Prozessierungsschritt „SCENES to APOLLO“ (Gruppe 4.x.x), der die Schnittstelle zur Datenübergabe von SCENES nach APOLLO darstellt, werden sowohl Sensordaten als auch Zusatzdaten überprüft und es können QS-Informationen abgeleitet werden. Die Plausibilität der Georeferenzierung wird einfach dadurch getestet, dass Breiten- und Längengrade außerhalb der MSG-Scheibe nicht vorkommen dürfen. Eine automatische Überprüfung der Genauigkeit in aktuell laufender Prozessierung ist mit den implementierten algorithmischen Verfahren nicht möglich. Hier kann nur eine interaktive vorherige Kontrolle per Augenschein erfolgen, die dann lediglich als durchgeführt vermerkt werden kann.

4.2.1

Für die Genauigkeit der BRDF-Winkel gilt das Gleiche wie unter 4.1.1. ausgeführt für die Georeferenzierung. Die Plausibilität der BRDF-Winkel wird unter Berücksichtigung der Tageszeit und damit des Sonnenstandes der SEVIRI-Szene geprüft. Beispielsweise dürfen keine Sonnenzenitwinkel kleiner als 90° minus geografische Breite des Pixels plus Sonnendeklination vorkommen. Außerdem erwartet APOLLO bestimmte Wertebereiche, nämlich den Sonnenzenit zwischen 0° und 180° , den Satellitenzenit innerhalb der MSG-Scheibe zwischen 0° und 90° und den relativen Azimut zwischen 0° und 180° .

4.3.1

Die SEVIRI-Sensordaten der 5 AVHRR-ähnlichen Kanälen und die Land/Wasser-Maske, die als Binärdaten von SCENES an APOLLO übergeben werden, werden in so genannte APOLLO-Areas transformiert. Bei dieser Gelegenheit kann auch wieder die Vollständigkeit der Daten, wie sie für die Ableitung der Wolkenparameter benötigt wird, geprüft und

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

angegeben werden. Da beim geostationären Satelliten die Land/Wasser-Maske zeitlich konstant ist, genügt für eine Überprüfung ihrer Vollständigkeit beispielsweise ein einfaches Zählen der Land- und Wasserpixel. Bei der Bestimmung der Vollständigkeit der SEVIRI-Sensordaten für die Prozessierung in APOLLO können die QS-Informationen aus 3.1.1 verwendet werden. Danach kann die unter 4.3.2 durchgeführte Plausibilitätsprüfung dazu führen, dass nicht plausible Pixelwerte ebenfalls als nicht verfügbar in der QS-Information angesehen werden.

4.3.2

Beim Anlegen der APOLLO-Areas mit den Sensordaten werden bestimmte Wertebereiche erwartet und Pixel, die aus diesem Bereich herausfallen, werden auf die untere bzw. obere Grenze des jeweiligen Bereiches begrenzt (Clip-Value). Für die Reflexionsgrade bei 0.6 μm und 0.9 μm wird auf das Intervall zwischen 0.005 und 2.000 begrenzt, bei 1.6 μm bzw. 3.7 μm auf das Intervall zwischen 0.005 und 1.000. Für die Temperaturen bei 11 μm und 12 μm wird das Intervall 170 K bis 330 K verwendet. Pixelwerte außerhalb dieser Intervall, die auf die jeweiligen Intervallgrenzen gesetzt werden, können als Ausreißer markiert werden.

5.1.1

Die BRDF-Winkel werden im APOLLO-Teil der HELIOSAT-Prozesskette gebraucht, um gegen die Anisotropie korrigierte Reflexionsgrade, vor allem von Wolken, zu berechnen. Dies ist eine notwendige Voraussetzung, um z. B. wolken-optische Dicken aus den Sensordaten ableiten zu können. Die Genauigkeit der abgeleiteten Wolkenparameter hängt dabei vor allem vom Sonnen- und vom Sensorzenitwinkel ab. Zu größeren Zenitwinkeln hin, also bei tief stehender Sonne und/oder tief stehendem Satelliten, nimmt die Genauigkeit nichtlinear und progressiv ab. Zwar gibt es keine quantitativen Untersuchungen zur Größe dieses Genauigkeitsverlustes mit zunehmenden Zenitwinkeln, jedoch lässt sich hierfür ein qualitatives relatives Qualitätsmaß im Sinne von Nutzungseinschränkungen bzw. Nutzungshinweisen generieren.

5.2.1, 5.2.2

Die Information über die Land- und Wasserverteilung in der SEVIRI-Szene wird in APOLLO zusammen mit einem Vermerk über die eventuelle Lage eines Pixels im Bereich des direkten Sonnenreflexes (Sunlint) abgespeichert. Diese Datei wird Land/Wasser/Sunlint-Maske genannt und enthält die genannte Information für jedes einzelne Pixel. Für alle Wasserpixel wird zunächst mit Hilfe der zugehörigen BRDF-Winkel geprüft, ob geometrisch ein Sonnenreflex auftreten kann. Die Pixel, die in diesem Bereich potentiellen Sonnenreflexes liegen, werden danach auf Grund ihrer spektralen Signatur in den Sensorkanälen als reflexkontaminiert oder reflexfrei eingestuft. Zusätzlich wird in der Datei entlang der Übergangslinie Land-Wasser ein Küstenbereich von zwei Pixeln Breite definiert. Die Information, ob ein Pixel über Land, Küste oder Wasser liegt, sowie die Information, ob ein Wasserpixel im Sonnenreflex liegt oder nicht, entscheidet zum einen, welche Wolkenerkennungs-Algorithmen mit welchen Parametern laufen. Andererseits sind die bei der Anisotropiekorrektur verwendeten Koeffizienten für Land und Wasser unterschiedlich. Bei Pixeln, die als reflexkontaminiert gekennzeichnet sind, werden die Wolkenerkennungs-Algorithmen, die solare Kanäle verwenden, abgeschaltet. Einige Wolkenerkennungs-Algorithmen müssen wegen der Gefahr von Fehlklassifikationen im Küstenbereich abgeschaltet werden. Diese Einschränkungen im Bereich des Sonnenreflexes und über Küstenpixeln bedeuten eine reduzierte Effizienz bei der Wolkenerkennung und sollten in Form von QS-Metadaten vermerkt sein.

5.3.1, 5.3.2

Das Endergebnis nach dem Lauf aller Wolkenerkennungs-Algorithmen in APOLLO ist eine Datei, die die kombinierte Information über Wolken- und Schneekontamination der Pixel enthält. Diese Wolken/Schnee-Maske ist zusammen mit der Land/Wasser-Maske und den Sensordaten Ausgangsinformation für die nun folgende Berechnung der Wolkenparameter, z.B. der optischen Dicke. Wichtig für die Qualität des Endproduktes ist die vollständige Anwendung aller möglichen Wolkenerkennungs-Algorithmen sowie des Wolken/Schnee-Unterscheidungsalgorithmus. Normalerweise werden immer alle Algorithmen angewandt und im Fehlerfall die gesamte Prozessierung verworfen. Wie unter 5.2.1 und 5.2.2 erläutert, sind aber bei Küstenpixeln und im Sonnenreflex verfahrensbedingt Unvollständigkeiten gegeben. Die Gruppe 5.2.x und 5.3.1 sind also teilweise redundant, jedoch kann unter 5.2.x zwischen Küste und Sonnenreflex unterschieden werden. Die meisten der Wolkenerkennungs-Algorithmen sowie der Wolken/Schnee-Unterscheidungsalgorithmus bestimmen sich ihre Abfrageschwellen dynamisch aus den aktuellen Daten heraus. Falls das scheitert, d.h. falls gemäß bestimmter Kriterien solche dynamischen Schwellen nicht abgeleitet werden können, werden Voreinstellungen benutzt. Dies beeinflusst die Qualität der Wolken/Schnee-Maske und soll in Form von QS-Metadaten vermerkt werden.

5.4.1, 5.4.2

Für den APOLLO-Wolkenbedeckungsgrad aus AVHRR-Daten gibt es eine publizierte Validierung (Kriebel et al, 2003) gegen synoptische Beobachtungsdaten vom Deutschen Wetterdienst. Das Ergebnis zeigt eine Genauigkeit des totalen Bedeckungsgrades im Vergleich mit Synopdaten von weniger als $\pm 1/8$ Bedeckung für 70 % der Pixel, d.h. mehr als zwei Drittel der zugehörigen Bodenbeobachtungen für alle Pixel weichen vom mit APOLLO bestimmten Bedeckungsgrad um weniger als 12,5 % ab. Eine solche Validierung stellt eine echte physikalische Grundlage dar für Aussagen zur Genauigkeit (5.4.2). Die QS-Information 5.4.1 ist dagegen eine reine innere Vollständigkeitsinformation, die für so genannte teilbewölkte Pixel anfällt. Da nur die vollständig bewölkten Pixel über ihre Temperatur einem Wolkenstockwerk zugeordnet werden können, die teilbewölkten wegen ihrer Mischtemperatur dagegen nicht, muss eine statistische Methode angewandt werden, um auch teilbewölkten Pixeln ein Wolkenstockwerk zuzuordnen zu können. Dies geschieht in kleinen lokalen Ausschnitten nach dem Mehrheitsprinzip, d.h. die teilbewölkten Pixel werden dem Wolkenstockwerk zugeordnet, das in der lokalen Umgebung am häufigsten den vollständig bewölkten zugeordnet worden ist. Dies gelingt manchmal nicht, etwa wenn in der lokalen Umgebung keine vollständig bewölkten Pixel oder zwei oder mehr Klassen gleich häufig vorkommen. Diese keinem Wolkenstockwerk zugeordneten Pixel zu kennzeichnen ist beispielsweise wichtig für eine spätere Berechnung der Schattenwirkung von Wolken aus deren angenommenen Höhe.

5.5.1, 5.5.2, 5.5.3, 5.5.4

Zusammen mit dem Wolkenbedeckungsgrad ist die optische Dicke von Wolken eine zentrale Ausgangsgröße für die Abschätzung der solaren Einstrahlung am Boden. Für die optische Dicke lassen sich vier wesentliche QS-Informationen angeben. Die Plausibilität der abgeleiteten Werte muss zunächst rein numerisch in einem vorgegebenen Rahmen zwischen 0 und 500 liegen. Physikalisch sind aber auch schon Werte oberhalb von 200 meist unrealistisch. Außerdem kann die Plausibilität der wolken-optischen Dicke vor allem beim Vergleich mit dem Indikator für dünne Wolken (6.1.1) und dem Wolkentyp (6.2.x) des APOLLO-Postprozessors überprüft werden. Beispielsweise passen hohe optische Dicken

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

nicht zu dünnen Wolken, wenn nicht gleichzeitig als Wolkentyp die Klasse „multi-layer“ angegeben ist. Wie bereits unter 5.1.1 erwähnt hängt die Zuverlässigkeit vor allem von der optischen Dicke von Wolken von der BRDF-Geometrie des Pixels ab. Hierzu sind relative, qualitative QS-Angaben möglich. Die optische Dicke wird in APOLLO nur für vollständig bewölkte Pixel gerechnet, da wiederum im teilbewölkten Fall nicht zwischen Strahlung vom Boden und von der Wolke unterschieden werden kann. Dennoch möchte man auch für teilbewölkte Pixel wenigstens eine Abschätzung der optischen Dicke der in diesem Pixel vorhandenen Wolken haben. Dies wird ebenfalls durch Bezugnahme zur lokalen Pixelumgebung durchgeführt. Zur Zeit wird eine solche Extrapolation der optischen Dicken auf die teilbewölkten Pixel im Postprozessor für Atmosphärenprodukte erstellt (7.1.1), könnte aber prinzipiell auch schon auf der APOLLO-Stufe der HELIOSAT-Kette implementiert werden. Die QS-Information 5.5.3 hat also derzeit nur eine Schalterfunktion, die besagt, dass in APOLLO noch keine teilbewölkten optischen Dicken erzeugt wurden. Die Kennzeichnung der extrapolierten Werte wird also unter 7.1.1 gemacht. Die Güte des in APOLLO verwendeten Verfahrens zur Ableitung wolken-optischer Dicken ist nicht physikalisch validiert, jedoch existiert eine theoretische Abschätzung der Genauigkeit (Kriebel et al, 1989) auf der Grundlage vernünftiger physikalischer Annahmen zu den Genauigkeiten aller Ausgangsgrößen und Algorithmen.

5.6.1

Der für die Bestimmung der Höhe der Wolkenobergrenze notwendige Parameter ist die Wolkenoberflächen-Temperatur. Mit Hilfe eines Temperaturprofils kann einer bestimmten Temperatur eine Höhe zugeordnet werden. Die besten Ergebnisse erzielt man dabei natürlich, wenn das jeweils aktuelle Temperaturprofil bekannt ist. Häufig ist dies aber nicht möglich und man muss sich mit Annahmen behelfen, etwa mit Modellatmosphären oder mit in numerischen Modellen vorhergesagten Profilen. Eine Angabe zu dem aktuell verwendeten Temperaturprofil stellt daher eine bedeutende QS-Information dar, da beispielsweise eine Temperaturabweichung von fünf Grad schon einer Höhenabweichung von etwa 500 m entspricht.

6.1.1, 6.2.1, 6.2.2, 6.3.1, 6.3.2, 6.4.1

Aus den im APOLLO-Kern erzeugten Masken und Wolkenparameter werden im APOLLO-Postprozessor weitere für die Interpretation und Weiterverarbeitung hilfreiche Informationen zur Art der Bewölkung und zur Schneekontamination abgeleitet. Hierzu gehört auch die Erzeugung von Quicklooks zur Augenscheinprüfung der Datenqualität. Ein Indikator-Layer gibt für jedes Pixel an, ob die Anwesenheit einer dünnen Wolkenschicht festgestellt wurde. Die Plausibilität dieser Information kann gegengeprüft werden, indem man mit der korrespondierenden optischen Dicke vergleicht. Ist die optische Dicke der Wolken unerwartet hoch, muss angenommen werden, dass unter der dünnen Wolkenschicht eine weitere, optisch dickere Schicht liegt. In diesem Fall muss bei der Wolkentypangabe der Vermerk „multi-layer“ gesetzt werden. Die Angabe des Wolkentyps ist ebenfalls für jedes Pixel vorgesehen, ist aber stellenweise wegen unzureichender Informationen nicht möglich. Solche unklassifizierbaren Pixel sollen gekennzeichnet werden. Zusätzlich kann zur Wolkentyp-Information der Grad der Klassifikationssicherheit angegeben werden. Sehr wichtig für die weitere Prozessierung ist die Kenntnis der Schneekontamination der Pixel. Hierzu wird ein Indikator berechnet, der im Bereich zwischen den Zahlenwerten 0 und 100 die Wahrscheinlichkeit für Schneekontamination des Pixels angibt. Die eindeutige Kennzeichnung der Pixel als schneekontaminiert oder schneefrei erfordert aber die Festlegung auf einen bestimmten Schwellenwert für den ermittelten Schneeindikator. Dieser Schwellenwert wird aus dem

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Erwartungswert des Histogramms ermittelt und dem Produkt als Metadateneintrag zusätzlich zum Indikator-Layer mitgegeben. Der Schwellenwert kann je nach Aussehen des Histogramms mehr oder weniger bestimmt sein, was als QS-Information angegeben werden soll. Der Schnee-Indikator selbst ist als innerer QS-Informations-Layer anzusehen. Die Generierung von Quicklooks schließlich stellt eine visualisierende QS-Information dar, deren Verfügbarkeit ebenfalls in den Metadaten vermerkt sein sollte.

7.1.1

Beim Postprozessor für die Atmosphärenprodukte werden, wie bereits unter 5.5.3 erwähnt, wolken-optische Dicken für teilbewölkte Pixel extrapoliert. Diese extrapolierten Werte sollen für die Qualitätskontrolle bei der weiteren Prozessierung gekennzeichnet werden.

7.2.1

Die Qualität des Säulen-Wasserdampfgehalts (TWC) kann durch Vergleich mit dem in APOLLO gerechneten Flüssigwasserweg und mit dem Wolkentyp-Indikator auf Plausibilität geprüft werden.

7.3.1, 7.3.2

Die Aufgabe des Postprozessors für die Atmosphärenprodukte ist es, die Endprodukte der HELIOSAT-Prozesskette auf der DFD-Seite zusammenzustellen. Dazu gehört auch die Überprüfung der QS-Metadaten-Information. Zumindest die essentiellen QS-Metadaten sollten vollständig sein, aber auch die Vollständigkeit der optionalen sollte vermerkt werden. Für die Transformation in QS-Nutzerinformationen ist es notwendig, Zusammenfassungen aller QS-Metadaten-Informationen zu erstellen. Wie genau dies geschehen soll, wird in Abschnitt 3 beschrieben.

2.3.2. QS-Metadaten-Definition für die HELIOSAT-Prozesskette (DFD-Teil)

In Tabelle 2-14 werden nun auf der Basis der verfügbaren QS-Metadaten-Informationen die QS-Metadaten selbst definiert, d.h. sie werden in der Spalte „Parameter“ namentlich aufgeführt. Metadaten können gemäß der Norm ISO-19115 definiert werden, wenn in ISO19115 eine Rolle (Bedeutung) definiert ist, nach der aus der QS-Information die QS-Metadatenätze konstruiert werden können. In diesem Fall ist der Name der ISO-19115-Rolle oder -Klasse in grüner Farbe angegeben. Darüber hinaus werden in diesem Dokument keine weiteren Schritte zur Erstellung der Metadatenstruktur gemäß der Norm angegeben, da dies den Rahmen sprengen würde und Gegenstand jeweiliger prozessor-spezifischer Dokumentationen ist. In der Spalte „QS-Information“ wird als Referenz die Nummerierung der Tabelle 2-13 angegeben. In der Spalte „Typ“ wird gemäß der in Abschnitt 2.2 eingeführten abkürzenden Kennzeichnung angegeben, zu welcher Produktkomponente die Metadaten gehören und ob es sich um innere oder äußere Metadaten handelt. Unter der „Priorität“ wird verstanden, ob die Angabe der Metadaten notwendig, also unverzichtbar für die Prozessierung und die QS-Nutzerinformation, oder optional, also ergänzend ist. Falls Metadaten mit einer Menge von konkreten (alpha)numerischen Werten belegt werden, ist deren Wertebereich angegeben. In der Spalte „Klasse“ wird angegeben, zu welcher der sieben unter Abschnitt 2.2 eingeführten QS-Metadaten-Klassen der Metadaten-Parameter gehört. Im Anschluss an Tabelle 2-14 werden noch Details zur Zuordnung der Metadaten-Entitäten zu den Produktkomponenten erläutert.

Die Tabelle 2-14 gibt die QS-Metadaten-Sammlung für die HELIOSAT-Prozessierungskette wieder. Für die anderen Prozessorteile bis zum Endprodukt „solar irradiance“ müssen

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

weitere solche Sammlungen definiert werden. Dabei kann und soll aber in analoger Weise vorgegangen werden. Die hier für energiemeteorologische Produkte vorgeschlagene Systematik der QS-Metadatendefinition wird in Abschnitt 5 in ihren wesentlichen Aspekten zusammengefasst.

Parameter	QS-Inf.	Typ	Priorität	Wertebereich	Klasse
DQ_TemporalValidityOzone	1.1.1	[^] QP <QP	notwendig	Zeit: 0 - outdated	Gültigkeit
DQ_OzoneDefault	1.1.2	[^] QP <QP	optional	estimated, climatological	Konsistenz
DQ_ThematicAccuracyOzone	1.1.3	<QP	notwendig	Bias, RMSE	Genauigkeit
DQ_TemporalValidityAerosol	2.1.1	[^] QP <QP	notwendig	Zeit: 0 - outdated	Gültigkeit
DQ_AerosolDefault	2.1.2	[^] QP <QP	optional	estimated, climatological	Konsistenz
DQ_ThematicAccuracyAerosol	2.1.3	<QP	notwendig	Bias, RMSE	Genauigkeit
DQ_CompletenessScenesLv15	3.1.1	[^] QP	notwendig	any badvalues	Vollständigkeit
DQ_ThematicAccuracyScenesCal	3.1.2	<QP	notwendig	Bias, RMSE	Genauigkeit
DQ_CompletenessScenesBRDF	3.2.1	<QG	notwendig	(un)available	Vollständigkeit
DQ_CompletenessScenesLSM	3.3.1	<QG	notwendig	(un)available	Vollständigkeit
DQ_CompletenessScenesGeo	3.4.1	<QG	notwendig	(un)available	Vollständigkeit
DQ_CompletenessScenesSubset	3.5.1	<QG	notwendig	(un)available	Vollständigkeit
DQ_PlausibilityApolloGeo	4.1.1	>QG	notwendig	(un)reasonable	Plausibilität
DQ_ThematicAccuracyApolloGeo	4.1.1	<QG	optional	mean deviation	Genauigkeit
DQ_PlausibilityApolloBRDF	4.2.1	>QG	notwendig	(un)reasonable	Plausibilität
DQ_ThematicAccuracyApolloBRDF	4.2.1	<QG	optional	mean deviation	Genauigkeit
DQ_CompletenessApolloAreas	4.3.1	[^] QP >QP >QQ	notwendig	any badvalues	Vollständigkeit
DQ_PlausibilityApolloAreas	4.3.2	[^] QP	notwendig	(un)reasonable	Plausibilität
DQ_PositionalAccuracyApolloBRDF	5.1.1	[^] QP	optional	0% - 100%	Genauigkeit
DQ_PositionalValidityApolloGlint	5.2.1	[^] QP	optional	(no) glint	Gültigkeit
DQ_PositionalValidityApolloCoast	5.2.2	[^] QP	optional	(no) coast	Gültigkeit
DQ_CompletenessApolloAlgorithms	5.3.1	>QK	notwendig	name missing Algorithm	Vollständigkeit
DQ_ApolloDefaultAlgorithms	5.3.2	[^] QP >QK	optional	none, which default	Konsistenz
DQ_CertaintyApolloCloudLevel	5.4.1	[^] QP	optional	(un)classified	Bestimmtheit
DQ_ThematicAccuracyApolloCov	5.4.2	<QP	notwendig	mean deviation	Genauigkeit
DQ_PlausibilityApolloOdp	5.5.1	[^] QP	notwendig	(un)reasonable	Plausibilität
DQ_PositionalAccuracyApolloOdp	5.5.2	[^] QP	optional	0% - 100%	Genauigkeit
DQ_CompletenessApolloOdp	5.5.3	<QK	optional	totals only, all	Vollständigkeit
DQ_ThematicAccuracyApolloOdp	5.5.4	<QP	notwendig	mean deviation	Genauigkeit
DQ_ApolloDefaultCth	5.6.1	[^] QP >QK	optional	actual, model, predicted	Konsistenz
DQ_PlausibilityApolloThinIndex	6.1.1	[^] QP	optional	(un)reasonable	Plausibilität
DQ_CertaintyApolloCloudType	6.2.1	[^] QP	notwendig	(un)classified	Bestimmtheit
DQ_CertaintyApolloCloudClass	6.2.2	[^] QP	optional	0% - 100%	Bestimmtheit
DQ_CertaintyApolloSnowIndexThr	6.3.1	>QK	notwendig	threshold, RMS	Bestimmtheit
DQ_CertaintyApolloSnowIndex	6.3.2	[^] QP	notwendig	0% - 100%	Bestimmtheit

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Parameter	QS-Inf.	Typ	Priorität	Wertebereich	Klasse
DQ_CompletenessApolloQL	6.4.1	>QQ	optional	(un)available	Vollständigkeit
DQ_CompletenessAtmoProdOdp	7.1.1	<QK	optional	totals only, all	Vollständigkeit
DQ_PlausibilityAtmoProdTWC	7.2.1	^QP	optional	(un)reasonable	Plausibilität
DQ_CompletenessAtmoProdMDSset	7.3.1	>QM	notwendig	(un)complete	Vollständigkeit
DQ_CorrectnessAtmoProdMDSset	7.3.2	>QM	optional	which quality proofs checked	Überprüftheit

Tabelle 2-14: QS-Metadaten-Definition für energiemeteorologische Produkte (DFD-Teil)

Die Einordnung und Darstellung der Metadaten-Parameter aus Tabelle 2-14 wird im Folgenden kurz begründet und erläutert:

1.1.1

Hier wird der zeitliche Abstand zwischen den verwendeten vertikalen Ozonsäulen und der aktuellen Szene angegeben. Falls dieser zeitliche Abstand für alle Pixel, d.h. für alle Säulenwerte derselbe ist, kann ein äußerer MD-Parameter für die Produkt-Primärkomponente angegeben werden, falls nicht, wird für jedes Pixel ein innerer MD-Parameter im Metadaten-Layer der Primärkomponente angegeben. Wird eine bestimmt, evtl. zwischen Nutzer und Provider vereinbarte Zeitschwelle überschritten, wird ein Wert eingetragen, der die Bedeutung „veraltet (outdated)“ hat.

1.1.2

Falls 1.1.1 den Wert „veraltet“ hat, kann hier angegeben werden, ob als Ersatz geschätzte oder klimatologische Ozonsäulen verwendet wurden. Dies kann wiederum pauschal durch eine äußere Metadaten-Angabe oder durch pixelweise innere Metadaten-Angabe in der Primärkomponente erfolgen.

1.1.3

Die Genauigkeit der Ozonsäulen wird durch Angabe der RMSE aus Vergleichen angegeben. Da dies a priori bekannt ist, handelt es sich um einen äußeren MD-Parameter der Primärkomponente.

2.1.1

Angaben und Einordnung hierzu werden analog zu 1.1.1 verstanden.

2.1.2

Angaben und Einordnung hierzu werden analog zu 1.1.1 verstanden.

2.1.3

Angaben und Einordnung hierzu werden analog zu 1.1.1 verstanden.

3.1.1

Hier wird pixelweise das Fehlen von Pixeln oder ganzen Zeilen mit Hilfe von „Badvalues“ im inneren QS-Metadaten-Layer der Primärkomponente vermerkt.

3.1.2

Die Genauigkeit der in SCENES angewandten Kalibration sollte a priori bekannt sein und wird durch Bias und RMSE angegeben.

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

3.2.1

Im äußeren QS-Metadatenatz der Georeferenzierungs-Komponente wird die Verfügbarkeit der BRDF-Winkel vermerkt. In der normalen Prozessorversion führt das Fehlen dieser Information zum Abbruch der Prozessierung. Es ist aber für spätere Versionen denkbar, die BRDF-Winkel aus Erdkoordinaten und Sensorgeometrie hinreichend genau zu berechnen. Voraussetzung hierfür ist aber, dass die Sensordaten jedenfalls in Erdkoordinaten orientierbar sind.

3.3.1

Im äußeren QS-Metadatenatz der Georeferenzierungs-Komponente wird die Verfügbarkeit der Land/Wasser-Maske vermerkt. In der normalen Prozessorversion führt das Fehlen dieser Information zum Abbruch der Prozessierung. Die Land/Wasser-Maske könnte aber auch aus den Erdkoordinaten erzeugt werden.

3.4.1

Im äußeren QS-Metadatenatz der Georeferenzierungs-Komponente wird die Verfügbarkeit der Georeferenzierung, also der Zuordnung von Erdkoordinaten zu den Pixeln, vermerkt. Ist diese Zuordnung nicht möglich, muss die Prozessierung entweder abbrechen oder das Endergebnis ist unsinnig.

3.5.1

Im äußeren QS-Metadatenatz der Georeferenzierungs-Komponente wird die Verfügbarkeit der Lage und Größe des zu prozessierenden Ausschnittes (Subset) vermerkt. Ist eine Subset-Angabe nicht möglich, bricht die Prozessierung entweder ab oder es wird die komplette METEOSAT-Scheibe bzw. der gesamte Eingabebereich prozessiert.

4.1.1

Im inneren QS-Metadatenatz der Georeferenzierungs-Komponente wird das Ergebnis der Plausibilitätsprüfung der Erdkoordinatenzuordnung abgelegt. Falls die Präzision dieser Zuordnung auf der Ebene der Schnittstelle zwischen SCENES und APOLLO bekannt ist, wird eine mittlere Abweichung im äußeren QS-Metadatenatz der Georeferenzierungs-Komponente vermerkt.

4.2.1

Im inneren QS-Metadatenatz der Georeferenzierungs-Komponente wird das Ergebnis der Plausibilitätsprüfung der BRDF-Winkel abgelegt. Falls deren Genauigkeit auf der Ebene der Schnittstelle zwischen SCENES und APOLLO bekannt ist, wird eine mittlere Abweichung im äußeren QS-Metadatenatz der Georeferenzierungs-Komponente vermerkt. Ungenauigkeiten in der Georeferenzierung selbst wirken sich nur in Horizontnähe (von Sonne und/oder Satellit) gravierend auf die BRDF-Winkel aus.

4.3.1

Hier wird die unter 3.1.1 gegebene Information auf die APOLLO-Areas abgebildet, d.h. die so genannten „Badvalues“ werden in die Areas selbst eingetragen. Zusätzlich wird im inneren QS-Metadatenatz der Primärkomponente und der Quicklook-Komponente eine Zusammenfassung eingetragen. Der Eintrag in der Quicklook-Komponente ist sinnvoll, da sich das Fehlen von Pixeln in den Sensordaten auf die Quicklooks selbst auswirkt.

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

4.3.2

Pixelwerte, die aus dem als sinnvoll angesehenen Wertebereich herausfallen, werden im QS-Metadaten-Layer der Primärkomponente markiert.

5.1.1

Die BRDF-Winkel beeinflussen die Genauigkeit der abgeleiteten optischen Wolkenparameter in der Weise, dass etwa mit zunehmenden Zenitwinkeln deren Genauigkeit abnimmt. Dafür kann ein relatives Maß angegeben werden, ausgedrückt in Werten zwischen 0% und 100%. 100% bedeutet, dass an diesem Pixel die größtmögliche Genauigkeit erreicht werden kann, bei Werten darunter aber nur ein entsprechender Anteil.

5.2.1

Da im Bereich des direkten Sonnenreflexes die solaren Wolkenerkennungs-Algorithmen abgeschaltet werden, ist dort damit zu rechnen, dass die Wolkenerkennung weniger zuverlässig funktioniert. Pixel, an denen ein Sonnenreflex erkannt wurde, können daher gekennzeichnet werden, was im Prinzip der Übertragung eines Informationsauszuges aus der APOLLO-Land/Wasser-Maske in den QS-Metadaten-Layer der Primärkomponente entspricht.

5.2.2

Einige Wolkenerkennungs-Algorithmen funktionieren nicht im unmittelbaren Küstenbereich und müssen daher dort abgeschaltet werden. Daher können solche Küstenpixel analog zu den Sonnenreflex-Pixeln in 5.2.1 (die Küstenpixel sind ebenfalls in der Land/Wasser-Maske enthalten) gekennzeichnet werden.

5.3.1

Wurde ein als wesentlich angesehenes Wolkenerkennungs-Algorithmus oder die Schnee/Wolken-Unterscheidung nicht angewandt, muss dies im internen QS-Metadatensatz der Prozessierungs-Kontext-Komponente vermerkt werden.

5.3.2

Falls Wolkenerkennungs-Algorithmen nicht in der Lage waren, dynamische Schwellenwerte zu ermitteln, müssen Voreinstellungen (Defaults) verwendet werden. Diese Defaults können im QS-Metadatenlayer der Primärkomponente angegeben werden. Im QS-Metadatensatz der Prozessierungs-Kontext-Komponente kann dazu eine Zusammenfassung gegeben werden.

5.4.1

Können teilbewölkte Pixel keinem Wolkenstockwerk zugeordnet werden, kann dies pixelweise vermerkt werden.

5.4.2

Die aus einer Validation mit Hilfe von Synop-Daten gewonnenen Genauigkeitsaussagen für den APOLLO-Wolkenbedeckungsgrad (Kriebel et al, 2003) werden im äußeren QS-Metadatensatz der Primärkomponente vermerkt.

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

5.5.1

Die Plausibilität der Werte für die optische Dicke von Wolken sollte pixelweise festgestellt werden. Werte, die aus dem als sinnvoll angesehenen Wertebereich herausfallen oder mit der Typisierung von Wolken konfliktieren, werden im QS-Metadaten-Layer der Primärkomponente markiert.

5.5.2

Das unter 5.1.1 beschriebene relative Genauigkeitsmaß für die unter Nutzung der BRDF-Winkel abgeleiteten optischen Wolkenparameter wird hier analog angewandt.

5.5.3

Im äußeren QS-Metadatensatz der Prozessierungs-Kontext-Komponente kann vermerkt werden, ob während der APOLLO-Prozessierung eine Extrapolation der wolken-optischen Dicke für teilbewölkte Pixel durchgeführt wurde.

5.5.4

Für die mit dem in APOLLO implementierten Verfahren abgeleiteten wolken-optischen Dicken gibt es eine Genauigkeitsabschätzung auf der Basis von plausiblen physikalischen Annahmen zu den möglichen Fehlerquellen (Kriebel et al, 1989, enthält diese Angaben für den Wolkenparameter „Flüssigwasserweg“, der eine aus der wolkenoptischen Dicke abgeleitete Größe ist). Diese Genauigkeitsangaben werden im äußeren QS-Metadatensatz der Primärkomponente vermerkt.

5.6.1

Für die Umrechnung von Wolkenoberflächen-Temperaturen in eine Höhe der Wolkenobergrenze werden Temperaturprofile benötigt. Diese Profile können aktuell oder vorhergesagt sein, oder aber einer Modell-Atmosphäre entsprechen. Was genutzt wurde, kann im QS-Metadatenlayer der Primärkomponente angegeben werden. Im QS-Metadatensatz der Prozessierungs-Kontext-Komponente kann dazu eine Zusammenfassung gegeben werden.

6.1.1

Die Plausibilität der Indikatoren für dünne Wolken kann in Verbindung mit der wolken-optischen Dicke und der Wolkentypisierung geprüft werden. Der Vermerk dazu wird pixelweise im QS-Metadatenlayer der Primärkomponente abgelegt.

6.2.1

Pixel, die in Bezug auf den Wolkentyp nicht klassifizierbar sind, sollen im QS-Metadatenlayer der Primärkomponente gekennzeichnet werden.

6.2.2

Zusätzlich zu 6.2.1 kann für die hinsichtlich des Wolkentyps klassifizierbaren Pixel die Klassifikationssicherheit angegeben werden.

6.3.1

Im inneren QS-Metadatensatz der Prozessierungs-Kontext-Komponente wird der Schwellenwert vermerkt, der für die Separation schneekontaminierter und schneefreier Pixel mit Hilfe des Schnee-Indikators verwendet wurde. Dazu kann eine Angabe zum Aussehen der Verteilung der Schnee-Indikator-Werte gemacht werden.

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

6.3.2

Der pixelweise vorliegende Schnee-Indikator selbst stellt ein Bestimmtheitsmaß für die Schneekontamination dar, deshalb ist er Teil des QS-Metadatenlayers der Primärkomponente.

6.4.1

Quicklooks sind z.B. für die Vorab-Auswahl von Datenprodukten nutzbar, etwa für das Auffinden von Szenen, in denen bestimmte Gebiete wolkenfrei sind. Daher ist es hilfreich, im inneren QS-Metadatenlayer der Quicklook-Komponente zu vermerken, ob Quicklooks überhaupt (erfolgreich) generiert wurden.

7.1.1

Im äußeren QS-Metadatenlayer der Prozessierungs-Kontext-Komponente kann vermerkt werden, ob während der Nachprozessierung für die Atmosphärenprodukte eine Extrapolation der wolken-optischen Dicke für teilbewölkte Pixel durchgeführt wurde. Dies macht natürlich nur Sinn, wenn dies nicht schon während der APOLLO-Prozessierung vorgenommen wurde (vgl. 5.5.3).

7.2.1

Die Plausibilität des Säulen-Wasserdampfgehalts (TWC) kann vom Postprozessor für die Atmosphärenprodukte im QS-Metadatenlayer der Primärkomponente gekennzeichnet werden.

7.3.1

Das vollständige Vorliegen aller (notwendigen) QS-Metadaten wird vom Postprozessor für die Atmosphärenprodukte festgestellt und im inneren Datensatz der QS-Metadaten-Komponente vermerkt.

7.3.2

Ebenfalls im inneren Datensatz der QS-Metadaten-Komponente kann eine Zusammenfassung aller erhobenen und in Metadaten transformierten QS-Information gegeben werden.

Die Abbildung 2-4 zeigt, dem Farbkomposit eines SEVIRI-Datenausschnittes gegenüber gestellt, einen Auszug aus dem QS-Metadaten-Layer der Primär-Komponente mit Aussagen zur Vollständigkeit der Daten und Bestimmtheit der Wolkentyp-Klassifikation. Rot bedeutet keine bzw. fehlerhafte Daten, Weiß steht für Wolken, die typisiert werden konnten und Blau zeigt die als wolkenkontaminiert erkannten Pixel an, für die keine Wolkentyp-Klassifikation möglich war.

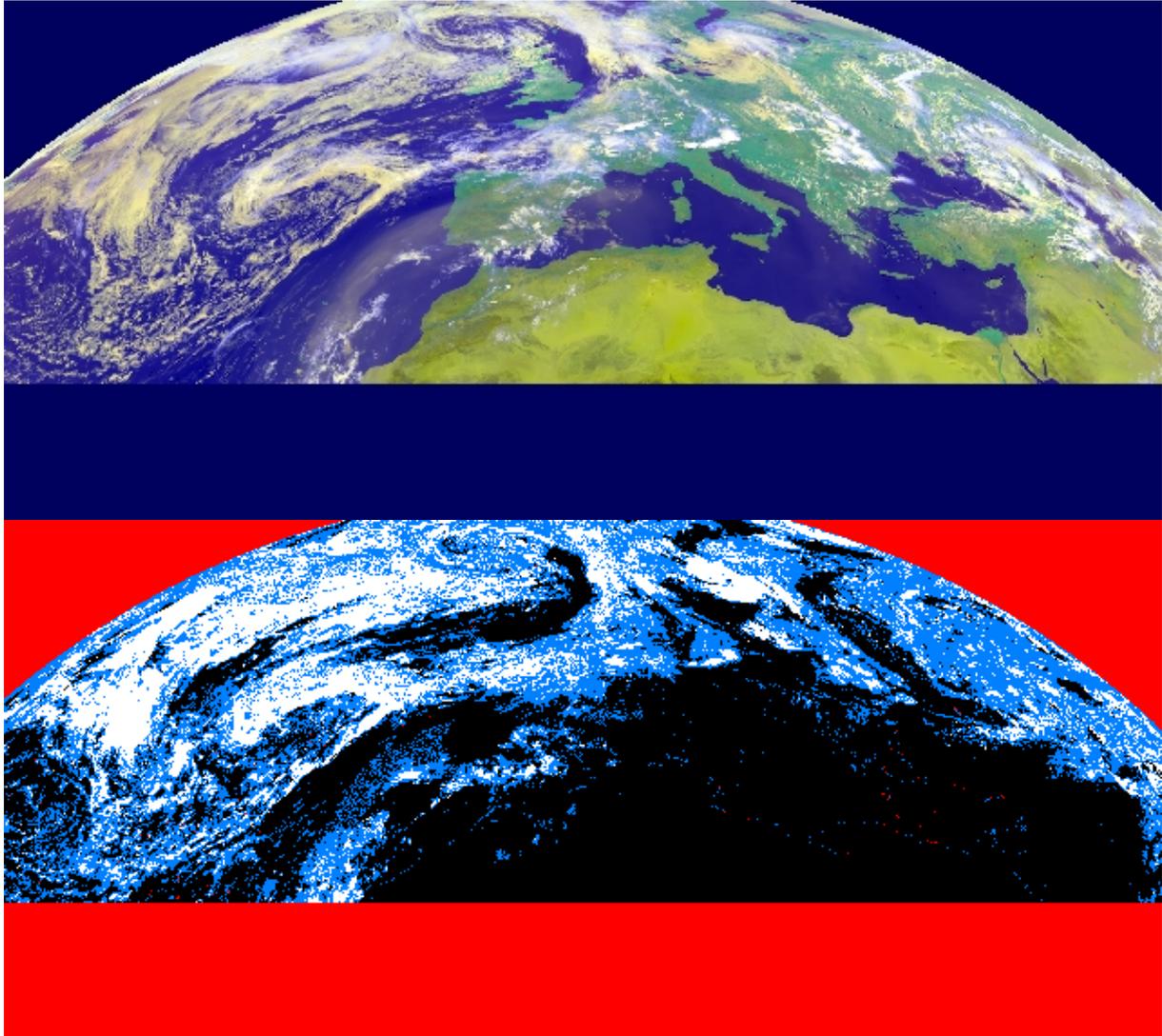


Abbildung 2-4: Beispiel für einen Auszug aus [^]Q_P zur Vollständigkeit und Bestimmtheit

3. Transformation der QS-Metadaten in Nutzerinformation

Es ist offensichtlich, dass der Nutzer der endgültigen Datenprodukte die QS-Informationen und QS-Metadaten in der Vielfalt und Form, so wie sie der Provider während der Produkt-Prozessierung und Produkt-Zusammenstellung generiert, nicht unmittelbar anwenden kann. Der Nutzer hat, zumal wenn es sich um ein kommerzielles Unternehmen handelt, den Wunsch, seinen Kunden übersichtliche und leicht verständliche Informationen über die Qualität der Datenprodukte zu geben. Deshalb ist es notwendig, dass der Provider, teils in Absprache mit dem Nutzer, die QS-Metadateninhalte bündelt (dieser Abschnitt) und die Belastbarkeit der daraus resultierenden Aussagen klar stellt (siehe Abschnitt 4).

Die Bündelung der QS-Metadateninhalte umfasst die Berücksichtigung mindestens der notwendigen (plus evtl. einiger optionalen) QS-Metadaten und möglichst aller Gruppen von vorliegenden Metadaten-Informationen und -Klassen. Dabei möchte der Nutzer stets drei grundsätzliche Dinge wissen, nämlich

- wie genau sind die gelieferten Datenprodukte,
- sind die Datenprodukte vollständig, so dass sie weiter prozessiert werden können und
- welche Grenzen haben die Qualitätsaussagen.

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Die nutzergerechte Transformation der QS-Metadaten-Information sollte also so erfolgen, dass eine Übersicht über die Genauigkeit und die Vollständigkeit des Datenproduktes entsteht. Dabei muss noch berücksichtigt werden, dass sich die Genauigkeit eines Datenproduktes in der Regel nicht komplett analytisch-physikalisch angeben lässt. Meist ist ein Verfahren nur in Teilen, also in Bezug auf einige der in der Methode genutzten physikalischen Größen über eine Validierung oder validierungs-ähnliche Genauigkeitsabschätzung abgesichert. Über weite Teile eines Verfahrens und viele der verwendeten Zwischengrößen lassen sich dagegen nur plausible Genauigkeits-Annahmen machen. Es ist daher notwendig, gegenüber dem Nutzer die Aussagen über die Produktgenauigkeit aufzuteilen in einen analytisch-physikalisch gesicherten Teil und in einen empirischen, evtl. sogar spekulativen Teil. Es ergeben sich also drei Zielbereiche bei der Transformation der QS-Metadaten in Nutzerinformationen, nämlich die

- Produkt-Vollständigkeit,
- die analytisch-physikalische Produkt-Genauigkeit und
- die empirische Produkt-Genauigkeit.

Aussagen über Risiken und Grenzen der Nutzung der Qualitätsinformation sind genereller Natur und beziehen sich jeweils auf Produktgruppen mit kompatibler QS-Metadatenstruktur, die mit vergleichbaren Methoden generiert wurden. Sie werden daher in Abschnitt 4 dargelegt.

Mit Hilfe der im Abschnitt 2 beispielhaft entwickelten QS-Metadaten für den DFD-Teil der HELIOSAT-Prozesskette wird in **Tabelle 3-1** gezeigt, wie daraus die QS-Informationen für den Nutzer in den drei Zielbereichen zusammengestellt werden. Im grün unterlegten Teil der Tabelle sind die notwendigen QS-Metadaten aufgelistet, d.h. diejenigen, die in die Informations-Zusammenfassung einfließen müssen. Im blau unterlegten Tabellenteil sind dann noch die optionalen QS-Metadaten genannt. Schließlich werden im gelb unterlegten Tabellenteil noch die QS-Metadaten eingeordnet, die sich nicht direkt den (vom Nutzer erwarteten) Klassen „Vollständigkeit“ und „Genauigkeit“ zuordnen lassen, aber in der jeweiligen Spalte eine Ergänzung oder Präzisierung darstellen. Im gelben Teil wird zwischen notwendigen und optionalen QS-Metadaten unterschieden durch Kennzeichnung der optionalen mit einem „(o)“.

Vollständigkeit	analytische Genauigkeit	empirische Genauigkeit
DQ_CompletenessScenesLv15	DQ_ThematicAccuracyOzone	DQ_ThematicAccuracyScenesCal
DQ_CompletenessScenesBRDF	DQ_ThematicAccuracyAerosol	DQ_ThematicAccuracyApolloOdp
DQ_CompletenessScenesLSM		
DQ_CompletenessScenesGeo		
DQ_CompletenessScenesSubset		
DQ_CompletenessApolloAreas		
DQ_CompletenessApolloAlgorithms		
DQ_CompletenessAtmoProdMDSet		
DQ_CompletenessApolloOdp		DQ_ThematicAccuracyApolloGeo
DQ_CompletenessApolloQL		DQ_ThematicAccuracyApolloBRDF
DQ_CompletenessAtmoProdOdp		DQ_PositionalAccuracyApolloBRDF
		DQ_PositionalAccuracyApolloOdp
	DQ_TemporalValidityOzone	DQ_PlausibilityApolloGeo
	DQ_OzoneDefault (o)	DQ_PlausibilityApolloBRDF
	DQ_TemporalValidityAerosol	DQ_PlausibilityApolloAreas
	DQ_AerosolDefault (o)	DQ_PositionalValidityApolloGlint (o)
		DQ_PositionalValidityApolloCoast (o)
		DQ_ApolloDefaultAlgorithms (o)
		DQ_CertaintyApolloCloudLevel (o)
		DQ_PlausibilityApolloOdp

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Vollständigkeit	analytische Genauigkeit	empirische Genauigkeit
		DQ_ApolloDefaultCth (o)
		DQ_PlausibilityApolloThinIndex (o)
		DQ_CertaintyApolloCloudType
		DQ_CertaintyApolloCloudClass (o)
		DQ_CertaintyApolloSnowIndexThr
		DQ_CertaintyApolloSnowIndex
		DQ_PlausibilityAtmoProdTWC (o)
		DQ_CorrectnessAtmoProdMDSet

Tabelle 3-1: Transformationstabelle QS-Metadaten zu Nutzerinformation

Um ein für den Nutzer brauchbares Datenprodukt zur Verfügung zu stellen, werden also für den DFD-Teil der HELIOSAT-Prozesskette mindestens die im grünen Tabellenbereich genannten QS-Metadaten benötigt, ergänzt durch die im gelben Tabellenbereich genannten notwendigen QS-Metadaten (ohne „(o)“). Dabei sichern die QS-Metadaten der Klassen Gültigkeit (validity), Plausibilität (plausibility) und Bestimmtheit (certainty), dass die Genauigkeitsangaben ohne zusätzliche, nachträgliche Einschränkungen übernommen werden können. Die Einzelgenauigkeiten der Ausgangsdaten, physikalischen Zwischengrößen und Zusatzdaten müssen nach den Regeln der Fehlerfortpflanzung zusammengefasst werden, wobei in der Regel angenommen werden kann, dass sie untereinander unabhängig sind.

Die im blauen und gelben Tabellenteil genannten optionalen QS-Metadaten bieten eine Ergänzung hinsichtlich inhaltlicher Details oder Details einiger Zusatzdaten.

4. Risiken und Grenzen der QS mit Metadaten

Die Annahme, vom Provider mitgelieferte QS-Metadaten könnten ein Datenprodukt dem Nutzer gegenüber hinsichtlich dessen Qualitätskategorien „Vollständigkeit“ und „Güte“ ohne zusätzliche Kommunikation sofort und hinreichend beschreiben, ist falsch. Grundsätzlich sind zwischen Provider und Nutzer Absprachen notwendig, die am Anfang, wenn Produkte definiert werden, intensiver sein müssen als später in der Routinephase. Die Inhalte dieser Absprachen sind die Folgenden:

- Der Provider definiert den Produktinhalt nach den Anforderungen des Nutzers.
- Der Provider definiert eine sinnvolle Produktstruktur einschließlich (QS-)Metadaten, die sich an den Erfordernissen der Inhaltsstruktur, der Archivierung und des Transfers orientiert und dokumentiert dies dem Nutzer gegenüber.
- Der Nutzer informiert den Provider darüber, mit welchen Verfahren die Datenprodukte bei ihm weiterprozessiert werden und welche Produkteigenschaften vor allem hinsichtlich der Qualität hierfür notwendig sind.
- Dabei genügt es nicht, wenn sich Provider und Nutzer über den Inhalt und die Struktur der Datenprodukte absprechen. Vielmehr müssen auch konkrete Schwellenkriterien vereinbart werden, nach denen bei der Produkt-Generierung die zu prüfenden Qualitätsparameter abgefragt und beurteilt werden.
- Der Provider erläutert dem Nutzer, ob Qualitätsaussagen auf gesicherten physikalischen Analysen beruhen oder ob es sich um Abschätzungen und daraus resultierend bloße Empfehlungen handelt.
- Der Nutzer informiert den Provider darüber, welche Qualitätserwartungen seine Kunden haben und ob unerfüllte Qualitätserwartungen der Kunden oder nicht genügend belastbare Qualitätsaussagen juristische Konsequenzen haben können und wenn ja, welche.

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Ein großes Risiko, dass QS-Metadaten ihren Nutzen verfehlen, besteht in einem Ausbleiben oder Vernachlässigen dieser Kommunikation.

Aus Tabelle 3-1 folgt nun sofort, dass zumindest für den DFD-Teil der HELIOSAT-Prozesskette die QS-Informationen, die echte physikalische Genauigkeitsangaben erlauben, gegenüber rein empirischen QS-Abschätzungen in der Minderzahl sind. Das bedeutet, dass sich für die Produkte MSG.SEVIRI.CLOUDS und MSG.SEVIRI.TWC nur wenig quantitative Genauigkeitsangaben machen lassen. Dagegen gibt es viele plausible empirische Qualitätskriterien, die der vom Provider in langer Zeit erworbenen Erfahrung bei der Generierung energiemeteorologischer Datenprodukte entstammen. Dieses Unverhältnis lässt sich ein wenig dadurch ausgleichen, dass möglichst alle QS-Metadaten, auch die optionalen, vom Provider zum Nutzer transferiert werden und dem Nutzer Werkzeuge in die Hand gegeben werden, wie alle Details dieser Informationen gegenüber dem Kunden zu brauchbaren QS-Aussagen weiterverdichtet werden können. Hierfür lassen sich für die QS-Metadaten-Informationen der sieben relevanten Qualitätsklassen folgende einfache Regeln stichwortartig aufstellen, um das Risiko der Fehlinterpretation der Metadaten und damit des ganzen Produktes zu reduzieren (vgl. auch mit Tabelle 2-6 bis Tabelle 2-12):

Vollständigkeit/Verfügbarkeit:

- Einordnung in einschränkende und kritische Vollständigkeitsdefizite durchführen
- Durchziehen der einschränkenden Vollständigkeitsdefizite bei der weiteren Prozessierung und in Zeitreihen
- Das Fehlen von notwendigen QS-Metadaten sollte definitionsgemäß immer kritisch sein

Konsistenz:

- Niemals Prozessor-Versionen mischen (Versionstreue)
- Quellentreue bei Eingangs- und Zusatzdaten
- Regelmäßige, ausreichend häufige Aktualisierung von Eingangs- und Zusatzdaten
- Stets die Repräsentativität von Interpretationen abfragen, d.h. möglichst breite Datengrundlage anstreben

Genauigkeit:

- Stets klare Trennung zwischen analytisch-physikalischen und empirischen Genauigkeitsangaben
- Unterscheidung zwischen Gesamt- und Einzelwertgenauigkeit

Gültigkeit:

- Vermeiden von pauschaler Nutzung aller Daten zugunsten konsequenter Beachtung von Geltungsbedingungen
- Erzeugung von Gültigkeitstabellen zum Einordnen von Einzelwerten in einen passenden Geltungsraum (wann, wo und unter welchen Voraus- und Randbedingungen)

Plausibilität:

- Verfolgen einer Bottom-up-Strategie, d.h. ein aus einem unplausiblen Zwischenergebnis erzeugtes Ergebnis ist ebenfalls als unplausibel anzusehen.
- Anwendung raum-zeitlich arbeitender Filter, um unplausible Werte durch raum-zeitlich kompatible zu ersetzen (Ausreißerfilter)

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Überprüftheit:

- Sicherstellung der Überprüftheit auch der Eingangs- und Zusatzdaten auf jeder Zwischenstufe der Prozessierung
- Sicherstellung der Abprüfung aller notwendigen QS-Kriterien durch Protokollierung
- Lückenlose Protokollierung des Prozessierungsverlaufs inkl. Fehlerdiagnostik

Bestimmtheit:

- Diskussion alternativer Interpretationen von Ergebnissen
- Durchreichen von Unbestimmtheiten durch den Rest der Prozessierungskette

Diese Auflistung kann in anderen multilateralen Prozessierungs-Szenarien weitere Empfehlungen zur Risiko-Minimierung enthalten. In jedem Fall sollten Postprozessoren an verschiedenen Stellen des Prozessierungsverlaufes eingebaut werden, die durch Kombination von verfügbaren QS-Metadaten-Inhalten das Risiko von Fehlinterpretationen oder falschen Qualitätserwartungen reduzieren.

5. Zusammenfassung

Im Kontext der Erstellung und Nutzung energiemeteorologischer Datenprodukte bezweckt dieses Dokument

- die Klarstellung eines Begriffs von Metadaten zur Qualitätssicherung,
- die Abgrenzung solcher Metadaten gegenüber anderen Daten,
- strukturelle und inhaltliche Definitionen,
- die Identifikation methodischer Notwendigkeiten und methodischer Empfehlungen und
- die Verdeutlichung von Risiken und Grenzen im Umgang mit solchen Metadaten.

Obwohl dieses Dokument speziell anhand der und aufbauend auf die Produktstruktur des DFD die Erzeugung der QS-Metadaten beschreibt, haben die im Folgenden kurz dargestellten Aussagen generelle Gültigkeit für energiemeteorologische Datenprodukte.

Erstellte Definitionen:

Zweck und Abgrenzung gegenüber anderen Metadaten:

Außer zur Qualitätssicherung gibt es natürlich andere Metadaten, die der Beschreibung sowie räumlichen und zeitlichen Orientierbarkeit sowohl aus Provider- als auch aus Nutzersicht dienen.

Abgrenzung gegen Zusatzdaten:

Zusatzdaten sind alle Daten, die zusätzlich zur Initiierung, Erzeugung oder Verteilung von Datenprodukten benötigt werden, ohne diese weiter zu beschreiben. Die Qualität von Zusatzdaten kann dennoch Auswirkungen auf die Qualität des Datenproduktes haben. Beispiel hierfür sind etwa die Kalibrationskoeffizienten.

Einteilung in innere und äußere QS-Metadaten, je aus Nutzer- und Providersicht:

Sind QS-Informationen a priori bekannt oder lassen sie sich aus Rand- und Startbedingungen bei der Produkterzeugung ableiten, werden die daraus generierten QS-Metadaten als äußere QS-Metadaten bezeichnet. Dem gegenüber führen QS-Informationen, die erst im Verlauf der Prozessierung bekannt werden bzw. ermittelbar sind, zu inneren QS-Metadaten.

Erstellung einer Produktstruktur auf Basis der DFD-DIMS-Produktstruktur:

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

Die bereits im DFD-DIMS-Archiv- und Prozessierungsumfeld existierende Struktur von Datenprodukten wird um die in diesem Dokument aufgeführten QS-Metadaten-Strukturen erweitert.

Identifizierung von sieben erforderlichen Qualitätsklassen mit inhaltlicher Beschreibung, davon 4 Qualitätsklassen als Standard beschrieben in ISO19115 (Completeness, Accuracy, Validity, Correctness):

Die beim Provider anfallenden und vom Nutzer angeforderten QS-Informationen lassen sich in sieben Qualitätsklassen einteilen. Es stellt sich heraus, dass vier dieser Klassen sich in der Norm ISO19115 wiederfinden.

Konkrete Definition aller für den DFD-Teil der HELIOSAT-Prozesskette möglichen QS-Metadaten (d.h. Namens- und Inhaltsdefinition, wo möglich angelehnt an ISO19115):

Bei der hier durchgeführten Definition von QS-Metadaten beschränken wir uns auf den DFD-Teil der HELIOSAT-Prozessierungskette, da wir hierfür der Provider und damit der Experte sind. Die anderen Teile können analog von den zuständigen Providern bearbeitet werden.

Vorgehensweise zur QS-Metadatengewinnung:

Erhebung der gesamten verfügbaren QS-Informationen unter Berücksichtigung der Nutzeranforderungen (konkret gezeigt am DFD-Teil der HELIOSAT-Prozesskette):

Der erste Schritt ist, festzustellen, welche QS-Informationen im Voraus, im Umfeld und während der Prozessierung ableitbar sind.

Beschreibung der Quantifizierung der QS-Informationen allgemein und konkret für den DFD-Teil der HELIOSAT-Prozesskette:

Die QS-Informationen sollten quantifizierbar sein oder es muss ein Verfahren zur Quantifizierung beschrieben werden.

Abbildung der verfügbaren QS-Informationen auf definierte QS-Metadaten-Einheiten, wo möglich angelehnt an ISO19115:

Die QS-Metadaten-Einheiten, die die QS-Informationen beinhalten und transportieren sollen, werden mit möglichst weitgehender Anlehnung an ISO19115 definiert. Die QS-Informationen werden zuvor den sieben Qualitätsklassen zugeordnet.

Festlegung der Priorität (notwendig oder optional):

Die Festlegung der Priorität der QS-Metadaten geschieht nach dem Prinzip der Dominanz des Notwendigen. D.h. QS-Metadaten sind nur dann als optional einzustufen, wenn sie sowohl für den Provider als auch für den Nutzer optional sind, d.h. für beide Parteien nicht unbedingt notwendig sind zur ausreichenden Beurteilung der Produktqualität.

Ableitung von Nutzerinformationen, Behandlung von Risiken:

Festlegung von nur drei QS-Informationsklassen für den Nutzer: Vollständigkeit, analytische Genauigkeit, empirische Genauigkeit:

Da der Nutzer eher eine komprimierte, übersichtlichere QS-Beurteilung des Datenproduktes wünscht, muss die Vielfalt der QS-Metadaten reduziert werden. Dies geschieht durch eine Abbildung aller QS-Metadaten der sieben Qualitätsklassen in die zwei für den Nutzer bestimmten QS-Aussageklassen „Vollständigkeit“ und „Genauigkeit“. Die Genauigkeitsaussagen müssen noch differenziert werden in Abschätzungen auf Grund empirischer Annahmen und in

Sicherung der Qualität energiemeteorologischer Datenprodukte durch Metadaten (AP 3.1)

analytisch-physikalisch ermittelte Genauigkeiten, so dass dann effektiv drei Aussageklassen dem Nutzer vermittelt werden.

Verdichtung der QS-Metadaten-Informationen zu Aussagen in diesen drei Klassen auf der Basis von Vereinbarungen zwischen Provider und Nutzer (Bewertung der Gesamt-Vollständigkeit, quantitative Angaben zu Gesamtgenauigkeiten in Form von MEAN, RMSE, Konfidenzintervallen):

Für diese Verdichtung der QS-Metadateninformation sind Bedeutungs- und Bewertungs-Vereinbarungen zwischen Nutzer und Provider notwendig.

Inhaltliche Beschreibung der notwendigen Kommunikation zwischen Provider und Nutzer zur Risiko-Minimierung und Aufstellung von Regeln zur Vermeidung von Fehlinterpretationen und falschen Güte-Erwartungen bzgl. der QS-Nutzerinformationen:

Um das Risiko einer unzureichenden Nutzbarkeit der QS-Aussagen durch den Nutzer zu minimieren, sowie Fehl-, Falsch- und Überbewertungen der Aussagen zu vermeiden, bedarf es einiger weiterer Kommunikationsregeln zwischen Provider und Nutzer.

Referenzen

A. von Barga, T. Schröder, A. Doicu, K. Kretschel, C. Lerot, M. Van Roozendaal, A. Kokhanovsky, M. Vountas, H. Bovensmann, M. Hess, B. Aberle & F. Schreier:

“SCIAMACHY level 1b-2 data processing: Update of off-line data processor to version 3.0”, in: Proceedings of the Second Workshop on the Atmospheric Chemistry Validation of ENVISAT (ACVE-2), 3-7 May 2004, ESA ESRIN, Frascati, Italy, edited by: Danesy, D., ESA Publications Division, Noordwijk, Niederlande, SP-562, 2004

Kriebel, K. T., Saunders, R. W. and Gesell, G., 1989, „Optical properties of clouds derived from fully cloudy AVHRR pixels“. Beiträge zur Physik der Atmosphäre, Vol. 62, No. 3, 165-171

Kriebel, K. T., Gesell, G., Kästner, M. and Mannstein, H., „The cloud analysis tool APOLLO: improvements and validations“, Int. J. Remote Sensing, 2003, Vol. 24, No. 12, 2389-2408

Lambert, J.-C., Alaart, M., Andersen, S. B., Blumenstock, T., Bodeker, G., Brinksma, E., Cambridge, C., de Maziere, M., Demoulin, P., Gerard, P., Gil, M., Goutail, F., Granville, J., Ionov, D. V., KyrÅ“o, E., Navarro-Comas, M., Piters, A., Pommereau, J.-P., Richter, A., Roscoe, H. K., Schets, H., Shanklin, J. D., Suortti, T., Sussmann, R., Van Roozendaal, M., Varostos, C., Wagner, T., Wood, S., and Yela, M.: “First ground-based validation of SCIAMACHY v5.01 ozone column”, in: Proceedings of the Second Workshop on the Atmospheric Chemistry Validation of ENVISAT (ACVE-2), 3-7 May 2004, ESA ESRIN, Frascati, Italy, edited by: Danesy, D., ESA Publications Division, Noordwijk, Niederlande, SP-562, 2004.



Qualitätssicherungsmetadaten und Validierungen im Datenaustausch zwischen Universität Oldenburg und meteocontrol GmbH

Arbeitspaket 3.1

Thomas Scheidsteger
Carl von Ossietzky Universität Oldenburg
Ammerländer Heerstraße 114-118
26129 Oldenburg

Internet: <http://ehf.uni-oldenburg.de/>

1 Einleitung

Im Lieferobjekt *Anforderungskatalog* in AP 3.1 wurden die von der meteocontrol GmbH zusätzlich geforderten Metadaten bzgl. ihres semantischen Gehaltes spezifiziert. Im zweiten Schritt soll die Methode der tatsächlichen Anlieferung der Informationen gewählt und umgesetzt werden:

- Reicht eine einfache Formatanpassung der bestehenden Datenprodukte zur Erfüllung der gegenwärtigen Nachfrage der meteocontrol GmbH?
- Welche Anpassungsarbeiten sind auf der Seite Uni OL, welche bei der meteocontrol GmbH durchzuführen?

2 Formate

2.1 Fortschreibung bisher verwendeter Formate

Die Metadaten könnten in zusätzlichen Dateien bereitgestellt werden, deren Format dem der bisher erzeugten CSV-Dateien mit den Globalstrahlungswerten entspricht, mit dem Strichpunkt als Trenner. Die zeitliche Auflösung sowie das Datumsformat würden entsprechend übernommen. Die neuen Dateinamen würden um die Kennzeichnung der zusätzlich berechneten Größe erweitert werden.

2.2 Gründe für die Einführung eines generischen Formates:

Die oben beschriebene Weiterverwendung des CSV-Formates für die neuen Datenprodukte scheint zunächst einen geringeren Implementationsaufwand zu bedeuten, sie erschwert aber langfristig die konsistente Datenhaltung bei beiden Partnern. Besser wäre ein Format, das alle Daten und ihre zugehörigen Metadaten in **einer** Datei vereinigt, sich leicht auf Konsistenz prüfen und leicht archivieren lässt. Für diesen Zweck bietet sich XML an, für das es verbreitete Standard-Software-Werkzeuge zum Schreiben, Lesen, Validieren, etc. gibt und das sich leicht archivieren lässt (Komprimieren von ASCII-Dateien).

2.3 Anpassungen in den Berechnungsroutinen

Wie schon im Anforderungskatalog erwähnt, mussten die operationellen numerischen Programme an der Universität Oldenburg um entsprechende Funktionen erweitert werden, die vor allem die Fehlermaße berechnen. Außerdem mussten die prozessbezogenen Metadaten intern erhoben und Variablen zugewiesen werden.

2.4 Anpassungen in den Formaten

2.4.1 Ausgabe in zusätzlichen CSV-Dateien

Wenn die neuen Größen (Fehlermaße) erst einmal berechnet sind, ist die Ausgabe in analog aufgebaute CSV-Dateien nicht mehr schwierig. Sie konnte bald umgesetzt werden, so dass die meteocontrol GmbH im täglichen operationellen Betrieb auf diese Daten zugreifen kann, ohne große Umstellungen in den Einleseroutinen vornehmen zu müssen.

- mmYYYYMMDD.dat
- mm_stderrYYYYMMDD.dat
- mm_clearYYYYMMDD.dat

bzw.

- ssplzYYYYMMDD.dat
- ssplz_stderrYYYYMMDD.dat
- ssplz_clearYYYYMMDD.dat

2.4.2 Ausgabe in eine XML-Datei

Definition eines XML-Schemas

Dazu musste zunächst ein XML-Schema (<http://www.w3.org/XML/Schema>) *safersun.xsd* definiert werden, das alle Daten und Metadaten für den Dienst *SaferSun* der meteocontrol GmbH in konsistenter Weise erfasst:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Einführung zum XML-Schema:
    http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/primer.html
-->

<xsd:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:xse="http://www.codesynthesis.com/xmlns/xml-schema-extension"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ns="http://www.w3.org/1999/xhtml"
            xmlns:hfp="http://www.w3.org/2001/XMLSchema-hasFacetAndProperty">
  <xsd:annotation>
    <xsd:documentation xml:lang="de">XML-Schema für das Datenprodukt
      Globalstrahlung PLZ</xsd:documentation>
  </xsd:annotation>

  <xsd:element name="safersun" type="t_safersun"/>

  <xsd:complexType name="t_safersun">
    <xsd:annotation>
      <xsd:documentation xml:lang="de">
        Dieses Element enthält stündliche Globalstrahlungsdatensätze für
        einen oder mehrere Tage, an einem oder mehreren Standorten.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="1" name="daily_set"
        type="t_daily_set"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="t_daily_set">
    <xsd:annotation>
      <xsd:documentation xml:lang="de">
        TODO:
        Für jede Stunde muss genau EIN hour -element existieren.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element minOccurs="1" maxOccurs="25" name="hour" type="t_hour"/>
      <xsd:element maxOccurs="1" minOccurs="1" name="summary" type="t_summary">
        <xsd:annotation>
          <xsd:documentation xml:lang="de">
            Zusammenfassende Informationen, die sich auf den gesamten Tag beziehen.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>

  <xsd:attribute name="date" type="t_date" use="required">
    <xsd:annotation>
```

```

    <xsd:documentation xml:lang="de">
      Datum für alle enthaltenen Stundenwerte
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>

<xsd:attribute name="location_id" type="t_location_id" use="required">
  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      Entweder deutsche PLZ oder meteomedia-Standort-ID (jeweils 6-stellig).
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>

<xsd:attribute name="lat" type="xsd:decimal" use="required">
  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      Geografische Breite des Standorts
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>

<xsd:attribute name="lon" type="xsd:decimal" use="required">
  <xsd:annotation>
    <xsd:documentation xml:lang="de">
      Geografische Länge des Standorts
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<xsd:complexType name="t_summary">
  <xsd:sequence>
    <xsd:element name="clearsky_pixel" type="t_pixel_count">
      <xsd:annotation>
        <xsd:documentation xml:lang="de">
          TODO
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>

    <xsd:element name="cloudy_pixel" type="t_pixel_count">
      <xsd:annotation>
        <xsd:documentation xml:lang="de">
          TODO
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>

    <xsd:element name="snowy_pixel" type="t_pixel_count">
      <xsd:annotation>
        <xsd:documentation xml:lang="de">
          TODO
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>

    <xsd:element name="daily_sum_global_irrad" type="t_irrad">
      <xsd:annotation>
        <xsd:documentation xml:lang="de">
          TODO
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>

    <xsd:element name="stderr_daily_sum_global_irrad" type="t_irrad">
      <xsd:annotation>

```

```

        <xsd:documentation xml:lang="de">
            TODO
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:element name="daily_mean_global_irrad" type="t_irrad">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">
            TODO
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>

<xsd:element name="stderr_daily_mean_global_irrad" type="t_irrad">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">
            TODO
        </xsd:documentation>
    </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="t_pixel_count">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">Pixelanzahl im Bild (TODO?)</xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="number" type="xsd:nonNegativeInteger" use="required"/>
</xsd:complexType>

<xsd:complexType name="t_irrad">
    <xsd:attribute name="value" type="t_non_negative_decimal" use="required">
        <xsd:annotation>
            <xsd:documentation xml:lang="de">Globalstrahlung, Einheit
W/m*m</xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="t_CSI">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">Clear Sky Index CSI (Einstrahlung
im Verhältnis zu Einstrahlung im unbewölkten Fall); Fehler der Einstrahlung
hängt vom CSI ab; CSI u.a. Input für Error Detection und erlaubt Einschätzung
der aktuellen Einstrahlungssituation (PVSAT-Kürzel: CLEARSKY)</xsd:documentation>
    </xsd:annotation>
    <xsd:attribute name="value" type="t_non_negative_decimal" use="required"/>
</xsd:complexType>

<xsd:complexType name="t_hour">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">Stundenwerte</xsd:documentation>
    </xsd:annotation>
    <xsd:choice>
        <xsd:element name="values" type="t_hour_values">
            <xsd:annotation>
                <xsd:documentation xml:lang="de">
                    Falls Stundenwerte vorliegen, befinden sie sich innerhalb dieses
Elements
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="missing" type="t_missing">
            <xsd:annotation>
                <xsd:documentation xml:lang="de">
                    Falls Stundenwerte fehlen, enthält dieses Element genauere Angaben
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:choice>
</xsd:complexType>

```

```

        </xsd:annotation>
    </xsd:element>
</xsd:choice>
<xsd:attribute name="start" type="t_hour_start" use="required">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">
            Startzeitpunkt für die Stunde
        </xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<xsd:complexType name="t_missing">
    <xsd:attribute name="reason" type="t_missing_reason" use="required">
        <xsd:annotation>
            <xsd:documentation xml:lang="de">
                Beschreibung, warum die Stundenwerte fehlen
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="t_hour_values">
    <xsd:sequence>
        <xsd:element name="global_irrad" type="t_irrad">
            <xsd:annotation>
                <xsd:documentation xml:lang="de">Globalstrahlung</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="stderr_global_irrad" type="t_irrad">
            <xsd:annotation>
                <xsd:documentation xml:lang="de">RMSE (?) für die Globalstrahlung -
                    Verweise auf GLossare möglich
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="clearsky_index" type="t_CSI">
            <xsd:annotation>
                <xsd:documentation xml:lang="de">Clear Sky Index</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="availability" type="t_availability">
        <xsd:annotation>
            <xsd:documentation xml:lang="de">
                Nähere Angaben zur Verfügbarkeit und Vollständigkeit der Eingabedaten
                für die Berechnung
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>

<xsd:simpleType name="t_date">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">
            Datum im Format yyyy-mm-dd, gregorianischer Kalender
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:date">
        <xsd:pattern value="\d\d\d\d-\d\d-\d\d" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="t_hour_start">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">

```

```

        Stundenstartzeitpunkt in der Zeitzone UTC (= keine Probleme mit
Winter/Sommerzeit)
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:dateTime">
        <xsd:pattern value="\d\d\d\d-\d\d-
\d\dT(00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|16|17|18|19|20|21|22|23):00:00Z" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="t_non_negative_decimal">
    <xsd:restriction base="xsd:decimal">
        <xsd:minInclusive value="0" />
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="t_missing_reason">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">
            Angaben zur Ursache für fehlende Stundenwerte
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="0">
            <xsd:annotation>
                <xsd:documentation xml:lang="de">
                    Fehlender Wert aufgrund Prozessierungsfehler (Chance der Nachberechnung)
                </xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="1">
            <xsd:annotation>
                <xsd:documentation xml:lang="de">
                    Fehlender Wert aufgrund fehlender Rohdaten
                </xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="t_availability">
    <xsd:annotation>
        <xsd:documentation xml:lang="de">
            Angaben zur Datenverfügbarkeit/Vollständigkeit für vorhandene Stundenwerte
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="0">
            <xsd:annotation>
                <xsd:documentation xml:lang="de">
                    Daten berechnet aufgrund vollständiger Datenbasis
                </xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
        <xsd:enumeration value="1">
            <xsd:annotation>
                <xsd:documentation xml:lang="de">
                    Zeitlich interpoliert (bis zu 2h werden überbrückt)
                </xsd:documentation>
            </xsd:annotation>
        </xsd:enumeration>
    </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="t_location_id">
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="\d\d\d\d\d\d"/>
    </xsd:restriction>

```

```
</xsd:simpleType>
</xsd:schema>
```

XML-Instanzen

Die auf diesem Schema basierenden Dateien *ssplzYYYYMMDD.xml* und *mmYYYYMMDD.xml* müssen nun von den numerischen Programmen ausgegeben werden. Zur Erleichterung dieser Aufgabe wird z.B. die Open-Source-Bibliothek *Genx* (<http://www.tbray.org/ongoing/When/200x/2004/02/20/GenxStatus>) für die Programmierung in C verwendet, die auf einfach zu bedienende Weise die Erzeugung von kanonischen und validen XML-Dokumenten unterstützt.

Synthetisches Beispiel einer *ssplz20060529.xml*:

```
<?xml version="1.0"?>
<safersun>
  <!--xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="safersun.xsd" -->

  <daily_set date="2006-05-29" location_id="001067" lat="51.0626" lon="13.7194">
    <hour start="2006-05-29T00:00:00Z">
      <values availability="0">
        <global_irrad value="55" />
        <stderr_global_irrad value="123" />
        <clearsky_index value="234" />
      </values>
    </hour>
    <hour start="2006-05-29T01:00:00Z">
      <missing reason="0"/>
    </hour>
    ...
    <hour start="2006-05-29T23:00:00Z">
      <missing reason="0"/>
    </hour>
    <summary>
      <clearsky_pixel number="3" />
      <cloudy_pixel number="55" />
      <snowy_pixel number="0" />
      <daily_sum_global_irrad value="4331" />
      <stderr_daily_sum_global_irrad value="123" />
      <daily_mean_global_irrad value="180.4" />
      <stderr_daily_mean_global_irrad value="15" />
    </summary>
  </daily_set>
  <daily_set date="2006-05-29" location_id="002068" lat="50.0766" lon="12.1967">
    <hour start="2006-05-29T00:00:00Z">
      <values availability="0">
        ...
      </values>
    </hour>
  </daily_set>
</safersun>
```

Umsetzung des XML-Schemas im Verarbeitungsprozess und Ausgabe

Für die Erzeugung von exemplarischen validen XML-Dateien sowie das Lesen und Validieren von XML-Dateien wurden Perl-Skripte geschrieben, die die weitere Entwicklungsarbeit sehr erleichtert haben. Dadurch können die Ausgabedaten noch vor der Auslieferung an die meteocontrol GmbH überprüft und validiert werden. Zur Sicherheit könnte bei der meteocontrol GmbH eine zusätzliche Validierung mit denselben Routinen durchgeführt werden, zumindest mit Skripten oder Programmen, die auf dem selben XML-Schema basieren, bevor die Daten in den bestehenden Prozessketten bei der meteocontrol GmbH weiterverarbeitet werden.

2.5 Weiterführende Arbeiten

Für die anderen ASCII-basierenden Datenprodukte, also insbesondere die *Historischen Zeitreihen für Standorte in Europa* lassen sich, auf *safersun.xsd* aufbauend, entsprechende Schemata entwickeln und die XML-Instanzen mit den o.g. Softwarewerkzeugen ebenso leicht und vollständig validieren, wiederum vor der Einspeisung in die bestehenden Verarbeitungsketten.

Für das im Binärformat erzeugte Datenprodukt *Strahlungskarte der europaweiten monatliche Mittelwerte der Globalstrahlung* bietet sich eine XML-Formulierung nicht an, aber evtl. für die Qualitätsmetadaten, die bisher in einer einfachen Liste der bei der Berechnung fehlenden Satellitenbilder besteht. Diese wird zurzeit noch nicht automatisch weiterverarbeitet. Zu diesem Zweck wäre aber eine starke Formalisierung wie mit einem XML-Schema und eine größere Detaillierung der Meta-Aussagen notwendig.

Auf der anderen Seite besteht schon vor der Prozessierung von Daten an der Universität Oldenburg der Bedarf, ASCII-Daten mit den Bodenmesswerten zu überprüfen und gegen ein formales Schema zu validieren. Dadurch würden die numerischen Rechenprogramme, die diese Bodendaten mit den Satellitendaten verknüpfen, von langwierigen und fehleranfälligen internen Validierungs- und Auswahlverfahren entlastet werden. Die dadurch geförderte strikte Trennung von Daten und Verarbeitung würde die Wiederverwendbarkeit der Daten in anderen Zusammenhängen sowie die Zuverlässigkeit der operationellen Prozessketten spürbar erhöhen.



GridFTP-Client

Arbeitspaket 3.2

Jan Ploski
OFFIS
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de>

1 Einleitung

Das GridFTP-Protokoll unterstützt die effiziente Datenübertragung großer Datenmengen zwischen kompatiblen Servern. GridFTP-Server können entweder in Clustern installiert sein, die für die Ausführung von wissenschaftlichen Berechnungen zuständig sind, oder sie können den Zugriff auf dedizierte Speichereinheiten anbieten. Die letzteren können insbesondere für die mittelfristige Datensicherung gebraucht werden, wodurch es auch für kleinere wissenschaftlichen Einrichtungen möglich wird, große Datenmengen zu behalten, die für spätere Vergleiche oder Weiterverarbeitung benötigt werden.

Im D-Grid-Kontext wird vor allem die GridFTP-Implementierung von Globus Toolkit 4 als Datenübermittlungsdienst in einzelnen Clustern eingesetzt. Vergleichsweise weniger bekannt, aber ebenso wichtig ist die GridFTP-Implementierung von dCache, der Middleware, die die dedizierten D-Grid-Speicherelemente in Jülich und Karlsruhe mit mehreren Terabytes an Bandkapazität zur Verfügung stellt.

Am Anfang von Arbeitspaket 3.2 (Standardisierte Datenspeicherung und Datenaustausch) stellten wir fest, dass überraschenderweise noch keine Client-Software existierte, die einen bequemen, interaktiven und zuverlässigen Zugang zu den beiden Arten von GridFTP-Servern bot. Angetrieben durch Anwenderanforderungen insb. seitens der Uni Oldenburg wurde in OFFIS ein grafischer GridFTP-Client entwickelt, der in seiner Benutzungsschnittstelle zwar einer gewöhnlichen FTP-Software ähnelt, aber zugleich die für große Datentransfers (z. B. mit >20000 Dateien, >500 GB Volumen) notwendigen Fehlertoleranzeigenschaften besitzt. Die Software wurde mit der dCache-Installation in Jülich getestet. Erfahrungen aus der Entwicklung wurden an dCache-Entwickler in Hamburg weitergereicht. Darüber hinaus wurde der GridFTP-Client an der Uni Oldenburg operativ (d. h. durch eigentliche Anwender) erfolgreich eingesetzt.

Die Weiterentwicklung des GridFTP-Clients wird als Open-Source-Projekt unter der Eclipse Public License erfolgen. Die erste Version, die in die populäre Entwicklungsumgebung Eclipse integriert ist, ist installationsbereit; eine von Eclipse unabhängige Version wird bis zum Arbeitspaketende veröffentlicht.

Das Softwareprodukt selbst ist unter der Adresse <http://bi.offis.de/gridftp/> veröffentlicht und steht damit ebenfalls Interessenten außerhalb des WISENT-Projekts zur Verfügung. Seit der Veröffentlichung Ende März 2008 verzeichnete diese Webseite über 300 verschiedene Besucher aus 33 Ländern, überwiegend aus Deutschland, USA und Großbritannien. Die interaktive Demonstration wurde dabei von über 70 Besuchern angeschaut, und die Software selbst über 40 mal heruntergeladen.

Der Rest dieses Dokuments, in englischer Sprache, besteht aus dem Benutzerhandbuch für den GridFTP-Client und schildert im Detail den Funktionsumfang der Software. Die aktuellste Version dieser Dokumentation wird bei der Installation geliefert bzw. kann separat heruntergeladen werden.

2 Introduction

The GridFTP protocol supports the efficient transfer of large amounts of data between conforming servers deployed as part of a (scientific) Grid infrastructure. These servers may be either a part of execution sites to which user jobs are submitted, or they may provide access dedicated storage elements with substantial disk or tape capacity for archiving Grid users' data. For example, the GridFTP server shipped in the popular Globus Toolkit middleware belongs to the former, and the GridFTP Door of the dCache middleware to the latter category.

During the course of the research project [WISENT](#), we found that, surprisingly, no user-friendly software existed that would provide convenient, reliable access to both types of GridFTP servers in order to browse, maintain and move around large (TeraByte scale) amounts of data accumulating in our project's research community. The tools provided by other projects were uncomfortable and restricted in functionality, not compatible with the German Grid's storage solution dCache, no longer maintained by their authors, and not at all integrated with the development environment (Eclipse) we endorsed for the participants of our project. For all these reasons, and taking advantage of the libraries provided by the Globus project, we embarked on the development of our own, general-purpose graphical GridFTP client software that would resemble in its user interface and features the familiar, off-the-shelf FTP clients, while still supporting the large (hence, failure-prone) transfers and the security infrastructure typical of a Grid computing environment. Ultimately, we wanted a GridFTP client that would make the conceptually simple task of moving data from A to B as free from technical considerations as possible for our users. Having developed the first useful working prototype, we released it under the open-source Eclipse Public License for everyone to enjoy and to contribute their feedback.

The remaining parts of this guide give an overview of features, describe the required installation steps, configuration, and functionality of the GridFTP Client. If you prefer a more lively introduction and learning-by-doing, we also provide an [online screencast](#) which walks you through a typical session with the program. All users are recommended to read to gain a deeper conceptual understanding of the core functionality.

3 Features Overview

Here is a short overview of GridFTP Client's features to help you determine whether the GridFTP Client fulfills your needs:

- Create Grid proxy and authenticate using a Globus-compliant X509 certificate
- Transfer files in third-party mode between two GridFTP servers
- Transfer directories recursively, possibly overwriting contents at the target location
- Recursively compute disk space usage on a GridFTP server
- Create and remove directories and files on a GridFTP server
- Initiate multiple concurrent transfers between different servers
- Browse GridFTP servers' contents while background transfers are underway
- Pause and resume transfers
- Report progress and errors for transfers
- Automatically persist transfers between application restarts
- Lightweight and self-contained package with few dependencies on other software
- Transfer using multiple parallel data channels per file

Here is a short overview of quite important features that are not yet implemented:

- Non-third-party file transfers (from client's local file system to a GridFTP server or the other way around)
- Renaming of files and directories
- Client-server architecture for managing transfers from multiple clients
- Firewall-friendliness (it's just as bad as any GridFTP client)

In addition to these features, the performance and fault tolerance of the GridFTP Client still have some room for improvement.

4 Installation

This chapter describes the required installation steps for the GridFTP Client.

4.1 Prerequisites

In the longer term, we plan to deliver the GridFTP Client in two variants:

- A plug-in variant integrated with the Eclipse development environment
- A standalone, easier to install "rich client" variant which doesn't need Eclipse

However, at present only the first variant is released, to which the following description applies.

4.1.1 Eclipse

Before installing the GridFTP Client plug-in, a recent version of Eclipse has to be installed. The minimum requirement is version 3.2 of Eclipse. Eclipse comes in several flavors. For new users, we recommend the "classic" version, which includes the Java and plug-in development environment (the latter provides an Error Log view, which can be convenient for diagnosing problems).

Eclipse does not include a Java Runtime Environment (JRE). You will need a 1.4.2 level or higher Java Runtime or Java Development Kit (JDK) installed on your machine in order to run Eclipse. Many systems already include Java. When in doubt, you can always quickly and easily install the latest version through the www.java.com web site.

Eclipse can be downloaded from www.eclipse.org.

4.1.2 Firewall Settings

One of the biggest hindrances for the use of GridFTP are (too) restrictive firewall settings that prevent the required communication between the GridFTP client and servers. In order for the software to work properly, consult your network administrator to make sure that the firewall allows the following types of connections:

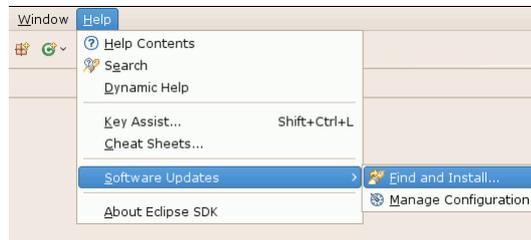
- Every computer on which the GridFTP Client is executed must be able to connect to the TCP port 2811 on the GridFTP servers that you wish to access. This port is used for the "control connection" (sending commands to the server and receiving command acknowledgements).
- Every computer on which the GridFTP Client is executed must be able to connect to a certain range of TCP ports on the GridFTP servers. This range is designated for "data connections" (used for transferring files, but also for reporting directory listings to the client). The exact range is likely prescribed by the Grid infrastructure project to which the servers belong. For example, in the German Grid the port range 20000-25000 must be reachable from the client.

4.2 Installing the GridFTP Client

The installation is properly performed by using the Eclipse Update Manager. The Update Manager connects to the GridFTP Client Update Site [<http://bi.offis.de/gridftp/updates>].

The Update Site can also be stored locally if no Internet connection is available and can be downloaded from the [GridFTP Client project page](#).

After starting Eclipse, select **Help > Software and Updates > Find and Install...** from the menu.



Find and Install...

Select **Search for new features to install** and press **Next**. Press the **Add Update Site...** button for a remote installation via HTTP or the **Add Local Site...** button if the Update Site is available locally. In a remote installation (recommended), enter any desired **Name** and the **URL** `http://bi.offis.de/gridftp/updates`.



Adding a new update site

Tick the newly created site and press the **Next** button. Follow the instructions of the install wizard. The warning about the installation of an unsigned feature can be ignored. Eclipse has to be restarted after the installation.

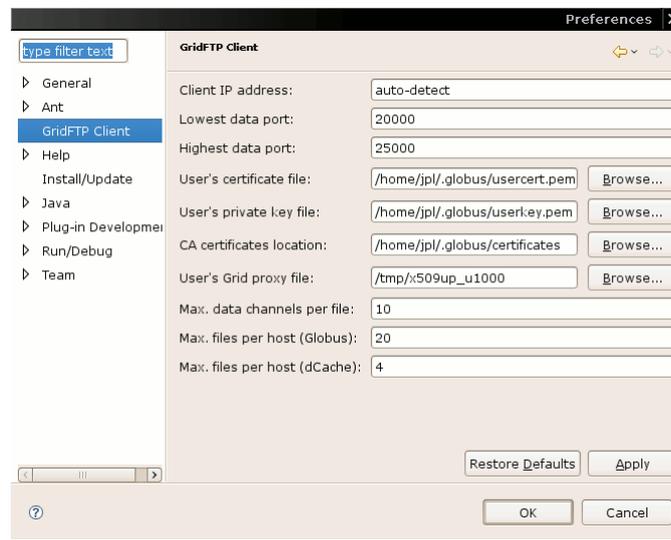
Now the GridFTP Client installation should be complete.

4.3 Uninstalling the GridFTP Client

To temporarily disable or uninstall the currently active version of the GridFTP Client, select **Help > Software and Updates > Manage Configuration** from the menu. Expand the tree in the dialog window which appears and select the GridFTP Client feature. Click on the option **Disable** in the panel on the right side and restart Eclipse when asked. At this point the feature is disabled, but still present on disk. To remove it completely, enter the same dialog again and select the previously disabled version of the GridFTP Client (you may need to toggle showing disabled features in the dialog's toolbar). The option **Uninstall** will now remove the chosen version completely.

5 Initial Configuration

Before setting out to transfer files, to avoid configuration problems, it may be a good idea to have a look at the GridFTP Client's configuration preferences. They can be accessed by selecting **Window > Preferences...** from the Eclipse menu.



Preferences dialog

The meanings of the preferences are quite technical. Fortunately, you can forget about them once they are correctly set, and the automatically guessed values should work without changing on many systems. This is especially true for machines that have been already configured to access Grid resources in the past.

- Client IP address - the IP address of your computer, which is transmitted in communication with GridFTP servers. This is particularly relevant if you wish to use the "active" connection mode where the servers make a connection instead of the other way around. You should leave the setting at its default value of auto-detect unless you experience connectivity problems, in which case you should enter the public (externally accessible) IP address of your machine.
- Lowest data port and highest data port - bounds for the TCP port range used for file transfers and directory listings. This must match whatever port range is unblocked in your local firewall as well as in the accessed GridFTP servers' firewalls. This setting is used to select source ports on the client side. When in doubt, consult your administrator.
- User's certificate file - path to the PEM encoded user certificate, the same certificate file which can be also used by the command-line tool grid-proxy-init. The default value is guessed depending on your operating system and generally shouldn't need adjusting. This path is only relevant if you wish to use the GridFTP Client to generate the proxy certificate (which is recommended, see below).
- User's private key file - path to the PEM encoded and encrypted private key, the same key file which can be also used by grid-proxy-init. This goes along with the user certificate, and likewise no changes should be needed.

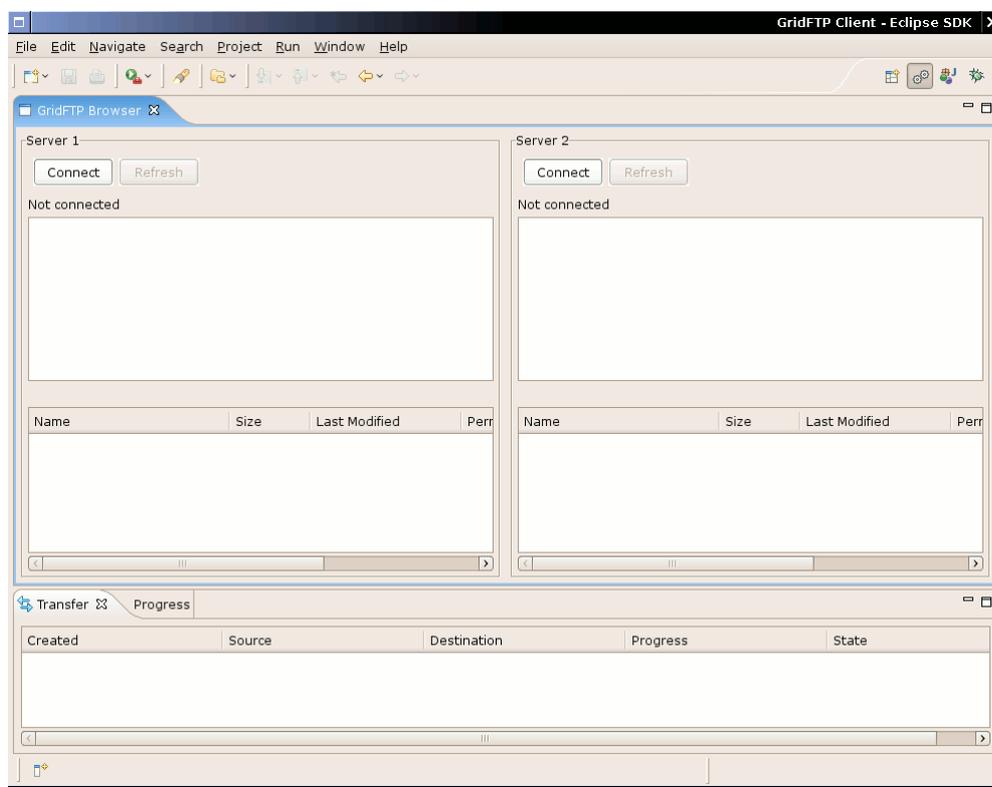
- CA certificates location - path to a directory which contains the certificates of Certification Authorities used to verify certificates of the GridFTP servers you wish to access. This path is also automatically guessed. When wrong, you will get error messages saying "Unknown CA" on connection attempts. Consult your administrator or the operators of the Grid infrastructure to find out where you can download the required CA certificate files.
- User's Grid proxy file - unlike the above paths, which are for read-only files, this one is read/write: the location of the proxy certificate, which can, but does not have to, be generated by the GridFTP Client from your user certificate, private key and password.
- Max. data channels per file - the maximum number of data channels that should be used within a single file transfer. This setting (like the option `-p` of `globus-url-copy`) affects the transfer rate, especially for large files. The GridFTP Client determines the actual number of data channels to use based on the file's size, up to and including the entered setting, with fewer data channels used for small files. Based on our experience, settings as high as 32 can still improve transfer rates. However, your target server might not be configured to accept that many data channels per transfer, in which case transfers would fail with an appropriate error message. For this reason, the default number is 10, which seems to be acceptable out-of-the-box for both Globus and dCache servers.
- Max. files per host - the maximum number of files that may be transferred at the same time from or to a GridFTP host. This can be set for Globus and dCache server types separately. The parameter is intended to keep the servers from being overloaded with requests and reporting connection errors. This has been in particular found out to be a problem for dCache servers. You may try raising the values up to 20 (which is the GridFTP Client's global limit on parallel file transfers) to see whether it improves or hurts the performance in your particular setup.

6 Working with the GridFTP Client

This chapter describes the GridFTP Client's user interface in terms of the supported user activities and explains the concepts underlying the software's functionality.

6.1 The GridFTP Client Perspective

All functionality is most conveniently accessed from a custom Eclipse perspective implemented by the GridFTP Client. You can open this perspective by selecting **Window > Open Perspective > Other...** and selecting the item **GridFTP Client** in the following dialog. When the perspective is active, the screen is by default divided into two areas: the upper area contains a view titled **GridFTP Browser**, while the lower area contains two views, titled **Transfer** and **Progress**:



The **GridFTP Client** perspective

TIP: Should you inadvertently change the layout or content of the GridFTP Client perspective, the easiest way to restore defaults is to select Window Reset Perspective from the Eclipse menu.

The **GridFTP Browser** view is where you will see the contents of a GridFTP server after connecting. The view is split vertically and contains two functionally equivalent panes (labeled **Server 1** and **Server 2**), so that you can connect to two servers at once to arrange file transfers between them. It does not matter which server is in which pane or in which order you connect. It is also possible to connect to the same server in both panes.

The **Transfer** view is where you will see all currently active (or paused) transfers. The purpose of this view is to provide monitoring information (about the progress and

encountered errors) and to allow user-level control of long-running transfers. Concepts associated with transfers are explained in more detail in .

Finally, the **Progress** view is the generic view familiar from other Eclipse perspectives, used to monitor the progress of long-running operations (such as recursive file deletions or disk usage calculations). Active transfers also appear in this view; however, it provides less detailed information and less sophisticated control than the more specialized **Transfer** view.

6.2 Connecting to a GridFTP Server

Click on one of the **Connect** buttons in the **GridFTP Browser** view to open the connection dialog:



The connection dialog

In the **URL** text field, enter a `gsiftp://` URL with the hostname of the server, an optional port number, and the initial directory (only subdirectories deeper in the directory tree, but not parent directories will be accessible after connecting). An example of a valid URL would be `gsiftp://srvgrid01.offis.uni-oldenburg.de:2811/data/nwp`. The `gsiftp://` prefix is optional; if not entered, it will be inserted automatically.

The **URL** text field remembers servers to which successful connections were made previously and is filled with the last used server address by default. To make use of this history feature (like in a web browser), simply delete the entire contents of the field by pressing backspace. A list with past server addresses will pop up.

The **Connection Mode** setting controls in which direction data connections (used to obtain directory listings) are established between the GridFTP Client and the server. The **Passive** mode means that the client will connect to the server. The **Active** mode means that the server will connect to the client. Depending on your firewall settings, one or both modes will work, with the passive mode more likely to work (and thus default). However, we have observed that the passive mode often results in connection errors in combination with the dCache GridFTP server.

When you click **OK**, the GridFTP Client first checks whether a proxy certificate file needed to authenticate on the server already exists (the location of this file can be set in preferences). If the file is not found, a message will be shown and next the authentication dialog will appear:



The authentication dialog

You should enter the password which protects your private key (the "grid-proxy-init" password) in this dialog. The other fields are mostly for information, but you might wish to change them to assume a different (non-default) identity for this particular connection.

After the proxy certificate is successfully created (or found), the connection process continues. The chosen GridFTP server is contacted and authentication performed, after which the **GridFTP Browser** view becomes populated with the contents of the initial directory from the previously entered URL. The **Connect** button's label changes to **Disconnect**, with the obvious purpose.

6.3 Browsing a GridFTP Server

After a successful connection, the pane of the **GridFTP Browser** view in which you clicked the **Connect** button first displays the initial directory's contents. You can click on entries in the upper part to expand them and to navigate to a desired directory. Whenever you select some directory in the upper part, the lower part is synchronized to display the files and subdirectories contained in that directory. An alternative method of navigation is to double-click a subdirectory entry in the lower part.

Name	Size	Last Modified	Permissions	Owner	Group
globus	12288	2008-03-13 16:39:19	drwxr--r--	dgws0006	users
.ssh	4096	2007-09-11 14:00:27	drwx-----	dgws0006	users
data	92	2008-03-26 12:51:28	drwx-----	dgws0006	dgrid
.bash_history	15756	2008-03-14 10:04:40	-rw-----	dgws0006	users
.rnd	1024	2008-03-13 16:37:14	-rw-----	dgws0006	ws

The lower part of the GridFTP Browser

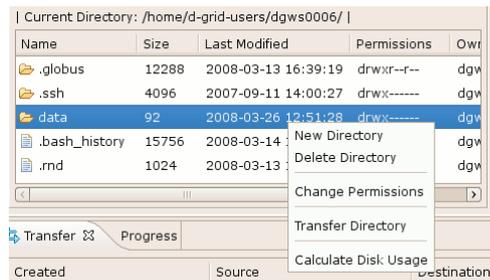
The meanings of the columns in the lower part are as follows:

- Name - name of a file or directory
- Size - size of a file or directory entry in bytes (for directories, this is not the size of contained files or subdirectories; however, the Calculate Disk Usage action provides that size, see below)
- Last Modified - timestamp of the last modification
- Permissions - Unix-style access permissions, formatted as in the output of the Unix command `ls -l`
- Owner - Unix username of the file or directory owner

- Group - Unix group of the file or directory owner

NOTE: The GridFTP implementation of dCache 1.7.0 has the following known limitations which affect the displayed data: Last modified timestamps are neither recorded nor displayed Permissions are shown for the owner only, but not for 'group' or 'world' Neither owner nor group are displayed; however, the invisible owner does in fact affect file permissions

Unlike the upper part, the lower part of the **GridFTP Browser** view supports multiple selections (hold CTRL or SHIFT while clicking on entries). Moreover, it contains a context menu (right-click) which lists several actions:



GridFTP Browser's context menu

The context menu's actions are explained below:

- New Directory - creates a new directory in the currently displayed one (selection in the lower part doesn't matter). You will be prompted for the name of the new directory.
- Delete - deletes the selected items. If any directories are selected, their contents will be deleted recursively (equivalent to the Unix command `rm -rf`). This is a long-running operation, which is tracked by the **Progress** view. The **GridFTP Browser** view is automatically refreshed after the operation finishes.
- Change Permissions - opens a dialog in which you can change the file access permissions for the selected items
- Transfer - initiates a transfer that copies the selected items to the directory whose contents are currently displayed in the opposite (left/right) pane of the **GridFTP Browser** view. Selected directories are copied recursively (equivalent to the Unix command `cp -r`). Because copying is in general a long-running operation which may be affected by transient or permanent faults of the servers and client, transfer management is a sophisticated feature in GridFTP Client, which is explained in more detail in .
- Calculate Disk Usage - recursively checks the amount of disk space occupied by the selected directories (equivalent to the Unix command `du`). Like deletion, this is a long-running operation. The calculated disk space appears in the upper part of the **GridFTP Browser** view next to the directory names. Some subsequent operations may invalidate the calculation and cause the displayed sizes to disappear.

The **GridFTP Browser** view caches displayed contents to improve responsiveness of the user interface. The **Refresh** button in the top part of the view can be used to discard all cached information displayed in the respective pane and to force obtaining fresh directory listings. This is necessary to see how the GridFTP Server contents were updated by some

external application (e.g., another user) and can be also employed to get an updated view of the target server during an active file transfer.

6.4 Transferring Files and Directories

The GridFTP Client is designed to keep track of and to control large transfers consisting of many potentially large-sizes files and directories. Such transfers may span hours or even days, and there is significant risk that certain types of failures will occur during the transfer, which need to be addressed (if possible, automatically) in order to ensure the eventual successful completion of the transfer. While the GridFTP Client must be executing in order to keep a transfer alive, it is possible to perform a large transfer in multiple sessions (with breaks in between). When servers go away or software crashes (as it inevitably does), the GridFTP Client does its best to protect your transfers from being aborted in a half-completed, difficult to understand state.

For all of the above reasons transfers, big or small, are a first-class concept in the GridFTP Client - you can create them, see them running, and interact with them during their entire lifetime. A transfer is defined as an operation which copies a set of files and/or subdirectory trees from some source directory on one GridFTP server to some target directory on another GridFTP server. Unlike Unix copy, which overwrites or skips pre-existing target files, the GridFTP Client's transfer works in "append" mode. That means that any existing files in the target location(s) will be treated as incomplete (cut off) versions of their corresponding source files. Only the still missing pieces of the files will be actually transferred, which improves performance when errors are encountered (especially for large files) and makes retrying transfers easy (even after they are completely aborted). It is your responsibility to ensure that the target location is "clear" from files which have a different identity, but the same name as the source files. Note that concurrent overlapping transfers into the same target location are not supported and will likely lead to errors or data corruption. It is your responsibility that your own and someone else's transfers don't collide.

When you start a transfer using the **GridFTP Browser's** context menu, an entry for it appears in the **Transfer** view:



Created	Source	Destination	Progress	State
Wed Mar 26 21:28:04 CET 200	gsiftp://srvgrid01.offis.uni-olde	gsiftp://juggle-glob.fz-juelich.d	18.69 GiB prepar	PREPARING

The **Transfer** view

6.4.1 Transfer States

A transfer is performed in two distinct, consecutive stages: the preparation stage, which gathers information about source files to be copied and creates all target directories, and the copy stage, which actually copies files from their source directories to the appropriate target directories created during the previous stage. The transfer's stage can be usually deduced from its **State** shown in the **Transfer** view:

- State **PREPARING** means that the preparation stage is currently underway. The **Progress** column of the **Transfer** view will reflect the total size of source files that have been found so far.
- State **ACTIVE** means that the preparation stage has been successfully completed and the copy stage is underway. The **Progress** column of the **Transfer** view will reflect the completion percent of the transfer. To account for failures, two percentual progress meters are displayed: the first number reflects the proportion of source files that were either successfully transferred or were classified as having encountered non-transient failures. The second number reflects only the proportion of source files that were successfully transferred. In other words, the first number tracks the progress of the transfer's execution - when it reaches 100%, it means that no more files are being transferred, while the second number tracks the progress of the transfer's goal - when it reaches 100%, there are no more files to transfer.

A transfer can also enter three other states:

- State **FINISHED** is equivalent to the second progress meter reaching 100%. Once the transfer reaches this state, there is little more to do other than remove it from the **Transfer** view.
- State **HELD** is equivalent to the first progress meter reaching 100%, with the second progress meter finishing below 100%. In other words, some of the files have not been successfully transferred (yet), and the transfer is suspended. There are only two scenarios that can lead to this situation:
 - You have explicitly paused the transfer.
 - All transferable files have been either successfully transferred or encountered a non-transient failure. Non-transient failures are such that could not be simply worked around by retrying the file's transfer; they require your intervention. For example, a missing target directory is a non-transient failure (you have to check manually why the expected target directory is not there and create it), while temporary network connectivity problems are not. The GridFTP Client contains some built-in expert rules used to classify failures in this manner. In the current version, unrecognized failures are generally classified as non-transient to provide early warnings to the user. In the future, depending on user experiences, this policy may be reversed.
- State **WILL HOLD** occurs when the first progress meter exceeds the second progress meter. In other words, some files have encountered non-transient failures, but there are still other files that can be transferred. This state is meant as an early indicator to make the user aware that a manual intervention will be eventually needed to repair the transfer.

6.4.2 Transfer Details

You can obtain more extensive information about a transfer, which is particularly useful if the transfer unexpectedly reaches the **WILL HOLD** or **HELD** state. Double-click the transfer's entry in the **Transfer** view to open the transfer details dialog:



The transfer details dialog

Apart from the obvious information, the dialog contains a **Progress** text area which reports the following statistics:

- Both progress meters, as explained above.
- Total - the total size and number of all files within the transfer
- Scheduled - the total size and number of files that are not being actively transferred yet, but are awaiting their turn in the queue.
- Rescheduled - the total size and number of files that have reentered the waiting queue (because of errors or user actions).
- Active - the total size and number of files that are being actively transferred. The GridFTP Client transfers up to 20 files concurrently.
- In Progress - the size of the already transferred part of the active files.
- Held - the total size and number of files that encountered non-transient failures or were held when the transfer was suspended on user's request.
- Finished - the total size and number of successfully transferred files.

You can click on the **Refresh** button to update the displayed statistics. The button **View Errors** opens a new dialog, which allows you to browse the errors encountered by the transfer as a whole (i.e., during the preparation stage) or during attempts to transfer the individual source files. The current version of the GridFTP Client simply displays unadorned exception stack traces. These messages are certainly easier to understand to administrators and software developers than to end users. However, by inspecting them, you might find hints about the actual cause of the problem (e.g., "Permission denied" somewhere in the message would suggest that you should examine the source file's and target directory's permissions and ownership). Future versions will hopefully improve the quality of the reported error messages, especially as we accumulate insights into possible failure scenarios.

6.4.3 Controlling Transfers

The **Transfer** view contains a context menu (right-click) which allows you one of the actions that affect the selected transfer's state:

- Hold - (only enabled for active transfers) temporarily suspends the transfer and causes it to enter the **HELD** state. This action is also executed if you "cancel" the transfer operation displayed in the **Progress** view. Note that if you hold a transfer in the **PREPARING** state, it will have to repeat the entire preparation stage, which may take a while depending on the number of source files and directories.
- Resume - reschedules all files that were held: another attempt will be made to copy these files. This effectively resumes all (non-finished) files from the transfer, regardless of whether they got hold on your request or because of some encountered failures. In the latter case, the failures might reoccur, and the transfer might become held again after a while.
- Remove - removes the transfer from the **Transfer** view. This is the last action in every transfer's lifetime and not reversible. If the transfer has not finished yet, it will be interrupted and removed. Note that the files that have been already transferred will not be automatically deleted in any case - it is up to you to clean up if necessary.

6.4.4 Transfer Persistence

All transfers recorded in the **Transfer** view survive the termination of the GridFTP Client - regardless of whether you normally close the application or whether it crashes for some reason. After you restart Eclipse, the GridFTP Client will attempt to reconnect to the GridFTP servers of any recorded transfers and after a short while the **Transfer** view will become repopulated. Transfers that were active beforehand will automatically start running (you may need to minimize their progress dialogs), other transfers will remain in their respective states (**HELD** or **FINISHED**).

7 Obtaining Help

The [Bugs](#) and [Feature Requests](#) trackers on SourceForge contain descriptions of the current open issues for the GridFTP Client. If you encounter a problem, you can also search the [Help forum](#) for previous reports and solutions and ask your question there.

8 Acknowledgements

The GridFTP Client was originally developed at the [OFFIS](#) Institute for Information Technology within the research project [WISENT](#), sponsored by the German Federal Ministry of Education and Research (BMBF) as part of the D-Grid initiative under grant No. 01C5968.

We would like to thank the [Jülich Supercomputing Centre](#) for supporting our tests with different versions and configurations of dCache.

The GridFTP Client relies heavily on [CoG JGlobus](#) library provided by the Globus project. Thanks to the JGlobus team for providing such a stable foundation.



Betriebskonzept des Grid-Speicherelements des DFD

Arbeitspaket 3.2

Gerhard Gesell
Deutsches Zentrum für Luft- und Raumfahrt DLR, Deutsches
Fernerkundungsdatenzentrum DFD
Münchner Straße 20
82234 Weßling

Internet: <http://www.caf.dlr.de>

1 Hintergrund und Ausgangssituation

Durch die Bereitstellung von Sondermitteln fördert das BMBF den Aufbau einer D-Grid-Infrastruktur in Deutschland. Das DFD des DLR beteiligt sich daran in kleinerem Rahmen durch den Aufbau eines Speicherelements (Storage Element, SE) im Wert von 50000 €. Die Deutsche Grid-Initiative (DGI) unterstützt den Aufbau von Compute Elements (CEs) und SEs durch die Herausgabe von Referenz-Installationen und allgemeinen Betriebskonzepten. Für die SEs ist vorgesehen, dass als Middleware entweder OGSA-DAI oder SRM-dCache installiert wird. Weiterhin gibt es die Empfehlung, SEs mit einer Kapazität kleiner als 25 TB der jeweiligen beantragenden Community resp. eng kooperierenden Communities ausschließlich zur Verfügung zu stellen. Das bedeutet, dass das SE des DFD von den Projekten WISENT (beantragend) und C3Grid (eng kooperierend) genutzt wird.

Die Hardware, aus der das SE des DFD besteht, verteilt sich, wie unter AP 2.1 beschrieben, aus IT-Sicherheitsgründen auf zwei logisch (und auch räumlich) verschiedene Bereiche im DLR (DMZ und DFD-intern). Sie besteht aus:

einer virtuellen Maschine in einem Blade-Center (1/8 Blade) (DMZ),

einem 1 TB (zunächst in der Startphase 0.5 TB) großen SAN-Speicheranteil an der Achtel-Blade (DMZ),

zwei große Plattensystemen an zwei SUN SOLARIS Servern mit einer Nettokapazität von etwas mehr als 10 TB (DFD-intern) und

einem LTO-3 Bandgerät für die Speicherung offline auf Band (DFD-intern).

Die HW-Ressourcen in der DMZ wurden nicht gekauft, sondern vom IT-Dienstleister für das DLR, T-Systems-SfR, bis Ende 2010 angemietet. Die Miete wurde im Voraus aus etwa der Hälfte der erhaltenen BMBF-Sondermittel bezahlt (vgl. 3.3 Betrieb nach 2010). Von der anderen Hälfte wurden die Plattensysteme und das LTO-Gerät zum Anschluss an die DFD-internen SUN SOLARIS-Server gekauft.

2 Struktur der Hardware und Installations-Charakteristika der Software

Unter AP 2.1 befindet sich im Bericht zu den Auswirkungen der DLR-IT-Sicherheitsstrukturen eine Übersicht zu den für die Datenkommunikation im Projekt WISENT relevanten System-Komponenten und ihre Zugehörigkeit zu den jeweiligen Netzwerkbereichen zwischen den beiden Firewalls. In diesem Abschnitt werden HW und SW der zum Grid-Speicherelement gehörenden Komponenten genauer beschrieben.

2.1 Hardware – Übersicht, Konfiguration und Dienststruktur

Der Steuerrechner des Speicherelements in der DMZ ist in der Abbildung 1 dargestellt als gelb schraffierte Box an dem das grün schraffiert dargestellte Plattensystem des Pickup-Point angeschlossen ist. Dieser Rechner hat den DNS-Eintrag „eridanus.caf.dlr.de“ und fungiert als Schnittstellenrechner für das SE nach außen. Das System hat folgende Eigenschaften:

- Virtuelle Maschine mit 1/8 der Ressourcen eines Blade-Centers unter Suse LINUX 9.3
- Zwei Netzwerk-Schnittstellen mit unterschiedlichen IP-Adressen nach innen und außen
- 500 GB (ab 1. Juli 2008 1000 GB) Speicherplatz zugeteilt aus einem SAN-Pool mit reiserfs Dateisystem

Innerhalb der DFD-FW sind Plattensysteme an zwei SUN SOLARIS Server mit folgenden Eigenschaften angeschlossen:

- 1 SUN FIRE V440 (kurz V440) mit 4 CPUs @ 1,275 GHz (Operationsrechner) und 16 GB RAM
- 1 SUN FIRE V880 (kurz V880) mit 8 CPUs @ 1,200 GHz (Operations- und Backuprechner) und 32 GB RAM

Die beiden Plattensysteme an den SUN-Maschinen (an jedem Server je eines) haben folgende Eigenschaften:

- Die Plattensysteme von TRANSTEC bestehen je aus 16 Platten @ 400 GB Bruttokapazität
- Die Plattensysteme sind mit Fibre-Channel-Technik an die Server angeschlossen
- Die konfigurierten Dateisysteme sind auf RAID-Level 6 realisiert, d. h. es können pro System zwei Platten gleichzeitig ausfallen ohne Datenverlust.

An der V880 ist noch ein LTO-3 Bandgerät angeschlossen, mit dessen Hilfe Daten offline gesichert werden können. Die LTO-3 Bänder haben eine Kapazität von 400 GB unkomprimiert und bis zu 800 GB mit Datenkompression.

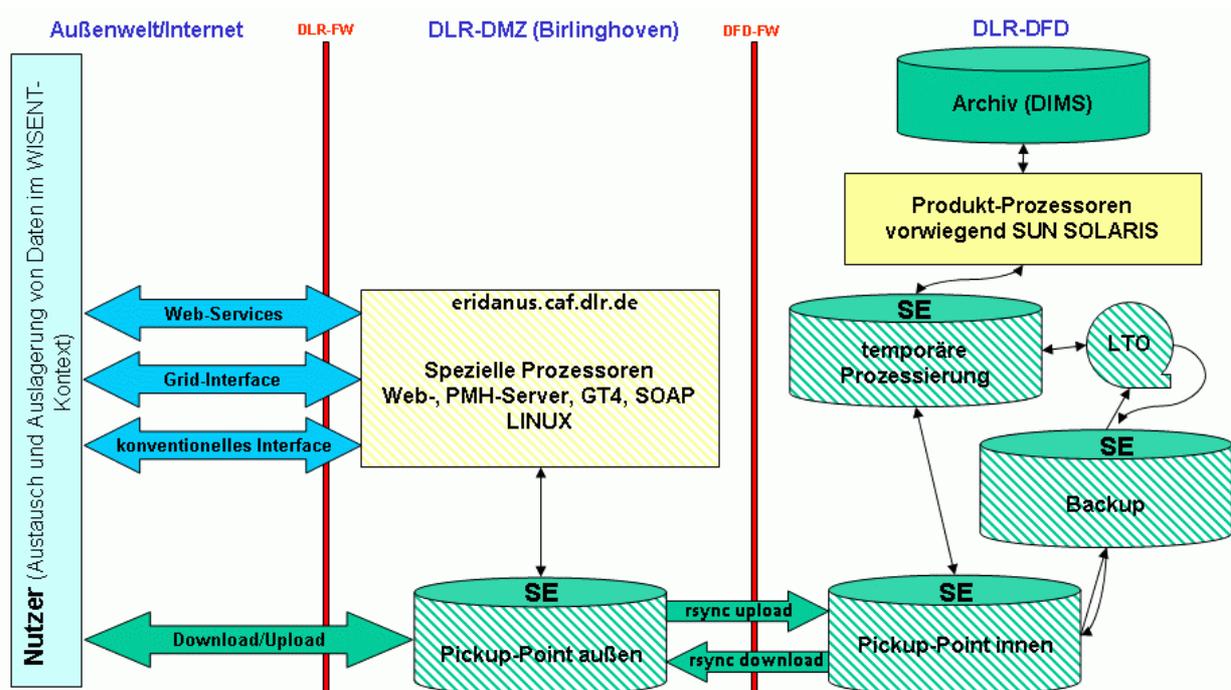


Abbildung 1: Beim Datentransfer beteiligte HW-Komponenten und Schnittstellen des Speicherelements

Die in Abbildung 1 gegebene Darstellung aller am Datentransfer über das Speicherelement exklusiv beteiligten HW-Komponenten und Schnittstellen entspricht dem Zustand des SE nach weitgehender Klarstellung des Betriebskonzeptes. Die für den Ablauf des Datentransfers notwendigen Dienste sind in Abbildung 2 dargestellt, d.h. an entsprechender Stelle der Abbildung 1 überlagert. Es hat sich im Verlauf des Aufbaus des SE herausgestellt, dass neben dem Grid-Interface zur informativen Kommunikation und zum Steuern von Prozessen auch Webservices genutzt werden müssen. Zum Datentransfer selbst wird auch ein konventionelles FTP weiterhin benötigt. Die Gründe hierfür werden im Kapitel zu den Erfahrungen mit GridFTP erläutert.

Mit Hilfe der in Abbildung 2 in den orangefarbenen Boxen dargestellten Dienste, Formate und Strukturen stehen über das Gesamtsystem des SE folgende Funktionalitäten zur Verfügung:

- Transfer von größeren Datenmengen über die synchronen Pickup-Points
- Bestellung von Produkten und Steuerung von (Post)prozessoren von außen
- Austausch von Statusinformationen
- Nutzung über zertifizierte und nicht-zertifizierte Anmeldung
- Automatischer Datenaustauschbetrieb bei regelmäßigem Produkt-Service

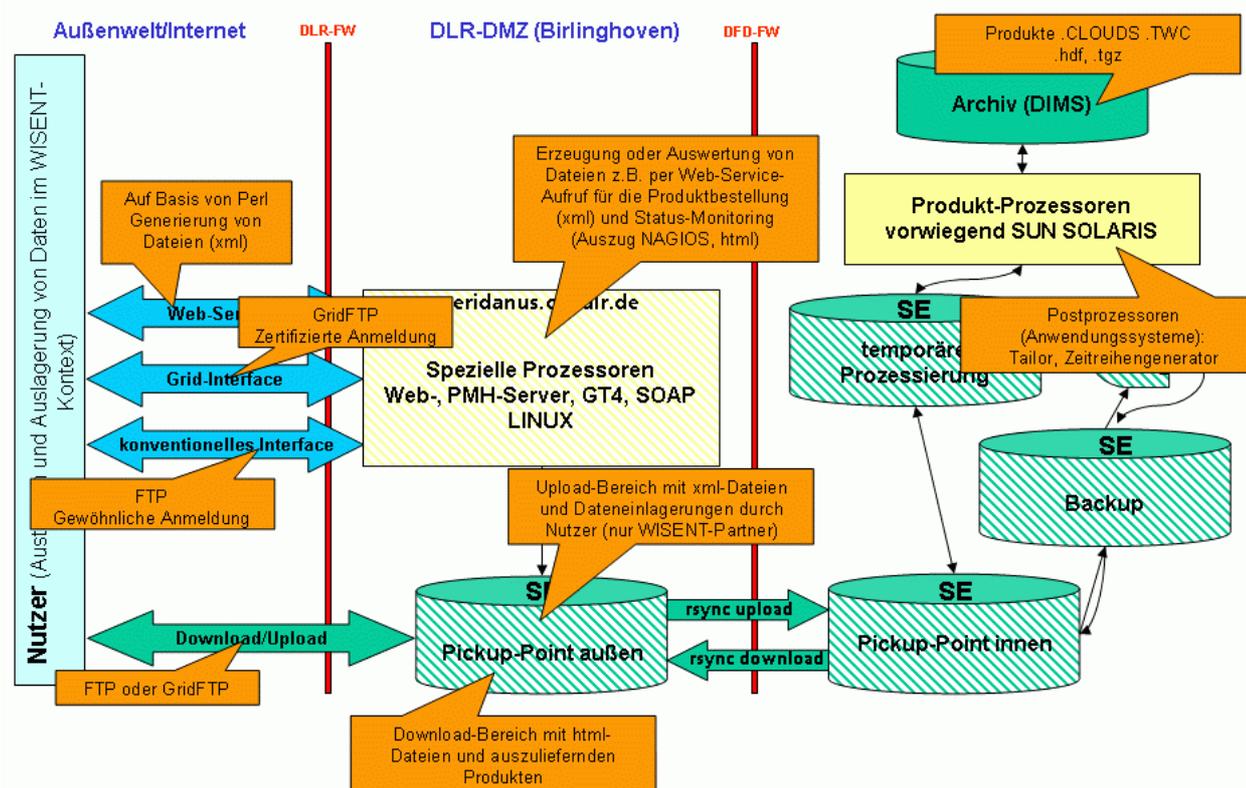


Abbildung 2: Organisation des Datentransfers und beteiligte Dienste, Formate und Strukturen

2.2 Software - Übersicht

Auf dem LINUX-System eridanus.caf.dlr.de ist folgendes installiert:

Software	Version	Einsatzgebiet
Apache Tomcat	5.5.20	Webserver und Servlet-Container
DLESE jOAI	3.0.6	Bereitstellung der Metadaten über OAI-PMH
Apache Axis	1.4	Webservice-Container
Globus GridFTP-Server	2.3 (GT4.0.3)	GridFTP-Server

Alle Pakete, die nicht zur Standard-Installation gehören, sind unter `/home/globus/pkg` abgelegt.

2.3 Sicherheitsaspekte

2.3.1 Benutzer

Für die Installation wurden zwei spezielle Nutzer angelegt. Zum einen der Nutzer `globus`, der Besitzer der Globus Toolkit Installation ist und unter dessen ID auch das Frontend des GridFTP-Servers läuft. Außerdem wurde der Nutzer `c3` angelegt, der Zugriff auf die Daten und Verzeichnisse für das C3Grid hat. Unter diesem Nutzer läuft auch der Apache Tomcat mit sämtlichen bereitgestellten Diensten.

2.3.2 Netzwerk

Für einen erfolgreichen Betrieb werden die folgenden Ports benötigt:

Dienste	benötigte Ports
Apache Tomcat, OAI, Webservice	8080
Globus GridFTP-Server (Management)	8443, 2811
Globus GridFTP-Server (Transfer)	20000-25000

2.3.3 Public-Key-Infrastruktur

Für die Authentifizierung und Autorisierung kommt die Grid-PKI des DFN zum Einsatz.

2.4 Installierte Software-Komponenten im Detail

2.4.4 Apache Tomcat

- <http://eridanus.caf.dlr.de:8080>

Installation

Zur Installation muss der Apache Tomcat einfach nur in ein beliebiges Verzeichnis entpackt werden. In der aktuellen Installation liegt er unter `/home/c3/prog/tomcat` (`$CATALINA_HOME`).

Konfiguration

Im Wesentlichen kann die Grundkonfiguration von Tomcat so benutzt werden. Um eventuelle Schwachstellen zu vermeiden, wurde jedoch der AJP-Connector, der normalerweise auf Port `8009` lauscht, deaktiviert. Dazu muss einfach der entsprechende Eintrag in der Datei `$CATALINA_HOME/conf/server.xml` auskommentiert werden.

Da Tomcat standardmäßig keine voreingestellten Benutzer mitbringt, öffnet es auch keine Lücken und es ist in Bezug auf die Konfiguration der Nutzer(-rechte) nichts zu tun.

2.4.5 DLESE jOAI-Server

- <http://eridanus.caf.dlr.de:8080/oai>
- <http://eridanus.caf.dlr.de:8080/oai/provider>

Installation

Der DLESE jOAI-Server ist als vorbereitetes Binary erhältlich. In dem Paket ist unter anderem eine Datei mit dem Namen `oai.war` enthalten. Diese muss einfach in das Verzeichnis `$CATALINA_HOME/webapps` kopiert werden. Nach einem Neustart des Tomcat-Servers ist der OAI-Server dann erreichbar.

Konfiguration

Die Konfiguration des OAI-Servers funktioniert komplett über das Web-Interface. Die Metadaten liegen im Verzeichnis `/home/c3/metadata` und können einfach in den Server importiert werden.

2.4.6 Apache Axis

- <http://eridanus.caf.dlr.de:8080/axis>

Installation

Die Installation von Axis läuft ähnlich der von jOAI. Aus dem vorbereiteten Binär-Paket muss der Ordner `$CATALINA_HOME/webapps/axis` in den `webapps`-Ordner von Tomcat kopiert werden. Außerdem sind ggf. auch noch die Java Mail API (`mail.jar`) sowie die XMLSecurity API (`xmlsec.jar`) in das Verzeichnis `$CATALINA_HOME/webapps/axis/WEB-INF/lib` zu kopieren, damit alle (optionalen) Abhängigkeiten erfüllt sind. Danach ist der Tomcat neu zu starten. Diese Bibliotheken können von <http://xml.apache.org/security/dist/java-library/> bzw. <http://java.sun.com/products/javamail/> heruntergeladen werden.

Eine Validierung der Axis-Installation kann direkt auf der Startseite von Axis (siehe oben) durchgeführt werden.

Konfiguration

Prinzipiell ist nichts weiter einzurichten, es ist jedoch empfehlenswert, den SOAPMonitor-Service zu deaktivieren, da dieser sonst noch den Port 5001 öffnet. Dazu sind die beiden entsprechenden Einträge in der Datei `$CATALINA_HOME/webapps/axis/WEB-INF/web.xml` auszukommentieren.

2.4.7 Stager-Webservice

- <http://eridanus.caf.dlr.de:8080/axis/services/C3ProviderSOAP>

Installation

Als Stager-Webservice (gemäß Schnittstelle D) kam die vom ZIB entwickelte Referenz-Implementierung hinzu. Außerdem wurde noch eine Stager-Klasse entwickelt, die es erlaubt den Staging-Vorgang per `ssh` auf einem entfernten Rechner vorzunehmen.

Zum Installieren wurden einfach die benötigten `jar`-Files in das Verzeichnis `$CATALINA_HOME/webapps/axis/WEB-INF/lib` kopiert. Mittels des AdminServices von Axis und der generierten `deploy.wsdd` kann der Webservice dann aktiviert werden. Der Befehl hierzu ist `$ java -cp $CP org.apache.axis.client.AdminClient deploy.wsdd`, wobei `$CP` eine Liste aller `jar`-Dateien aus der Axis-Installation enthalten sollte.

Konfiguration

Die Konfiguration des Staging-Webservice erfolgt über die Datei `/home/c3/c3grid.properties`. Diese beinhaltet die folgenden Parameter:

Property	Beschreibung
<code>c3grid.workspace</code>	Das lokale Verzeichnis, das als <code>workspace</code> verwendet wird. (Also etwa <code>/home/c3/workspace</code> .)
<code>c3grid.workspaceURI</code>	Die Basis-URI des <code>workspaces</code> , so wie er von außen angesprochen wird. (Also etwa <code>gsiftp://eridanus.caf.dlr.de/home/c3/workspace</code> .)
<code>c3grid.statusdir</code>	Das lokale Verzeichnis, in dem

	Statusinformationen des Staging-Webservices gespeichert werden. (Also etwa /home/c3/status.)
c3grid.statusdir.forceUnlock	Diese Einstellung kann genutzt werden, wenn der Webservice nach einem Absturz neugestartet werden soll. Wenn das Property auf yes gesetzt wird, dann wird die Dateisperre für die Statusdaten ignoriert.

2.4.8 Globus Toolkit und GridFTP-Server

- gsiftp://eridanus.caf.dlr.de/

Installation

Die Installation des Globus Toolkits aus den Quellen ist nicht trivial. Eine ausführliche Beschreibung kann unter ... gefunden werden. Soll nur der GridFTP-Server installiert werden, so ist die Entwickler-Version der zlib (zlib-devel) einzige Voraussetzung.

Die Installation des Globus Toolkits ist unter /usr/local/globus bzw. /usr/local/globus-4.0.1 zu finden. Eigentlich sollte alles klar laufen, wenn man die Zeilen

```
$ ./configure --prefix=/usr/local/globus
$ make gridftp
$ make install
```

ausführt. Nach dem Übersetzen sollte der GridFTP-Server und Client (samt notwendiger Infrastruktur) lauffähig und unter /usr/local/grid zu finden sein.

Konfiguration

Globus braucht auf jeden Fall eine ordentlich aufgesetzte PKI. Wesentlich dabei ist, dass ein gültiges CA-Zertifikat unter /etc/grid-security eingerichtet ist. Außerdem muss der Server auch sein eigenes Zertifikat kennen. Das Root-Zertifikat für die PKI ist beim DFN als Paket erhältlich. Dieses muss nach /etc/grid-security entpackt werden und dann mittels

```
$ grid-default-ca
```

Eingerichtet werden. In dieses Verzeichnis müssen auch die beiden Dateien hostcert.pem und hostkey.pem, welche die beiden Teile (privater und öffentlicher Schlüssel) des Host-Zertifikats darstellen. Um die Sicherheit zu erhöhen, wird der GridFTP-Server in zwei Teile geteilt. Eine unprivilegierte Instanz kümmert sich um die Kommunikation nach außen. Eine zweite Instanz behandelt dann den Datenzugriff und die Benutzerrechte und ist nur von der anderen Instanz erreichbar.

3 Nutzungskonzept, Datenverwaltung und Nachhaltigkeit

Die Abbildung 3 zeigt, wie die Dateisysteme des SE auf die Bereiche innerhalb und außerhalb der DFD-Firewall aufgeteilt werden. Die 1 TB große Partition am System in der DMZ dient als Pickup-Point für Datenprodukte. Dieser Pickup-Bereich wird nach innerhalb der DFD-Firewall gespiegelt, und zwar so, dass ein Teilbereich für Uploads, ein anderer für Downloads genutzt wird. Auf diese Weise lassen sich die Dateisysteme in beide Richtungen konsistent halten. Der verbleibende, bei weitem größere Speicherbereich innerhalb der DFD-Firewall wird für verschiedene Zwecke aufgeteilt. Diese Zwecke sind

- die Arbeitsablage während temporärer Prozessierungen, also eine Art Daten-Puffer,

- ein Puffer, um auszutauschende Datenprodukte in Art einer Warteschlange vorhalten zu können und
- ein Backup-Bereich zum kumulativen Sichern des Pickup-Points und Sichern auf LTO-Bänder.

Komponente	Zweck	Größe	Bemerkungen Beispiele
DFD-Archiv	FE-Daten- und Produktarchiv	bis 2.2 PB	je nach eingesetzten Datenträgern
WDC-RSAT	Online-Produkt-Archiv und FTP-Daten-Speicher	1 TB Portal plus 350 GB FTP	Bisheriger Bereich für den Datenaustausch
temporäre Prozessierung	Arbeitsablage Prozessoren	1 TB	Pufferung auf Platte
temporäre Prozessierung Warteschlange	Puffer für abzuliefernde Produkte/Produktpakete	5 TB	1 Produktpaket HELIOSAT für 1 Monat bis 200 GB
Backup	kumulatives Backup zur Pickup-Point-Sicherung	3 TB	LTO-Offline-Komponente verfügbar
Pickup-Point DFD (Spiegel Pickup-Point DMZ)	Pickup-Point-Zugang innerhalb DFD	1 TB	rsync-mirror
Pickup-Point DMZ	Pickup-Point-Zugang außerhalb DFD	1 TB (bis 30.06.08 0.5 TB)	rsync-mirror

Abbildung 3: Zweck der Komponenten des Speicherelements

Die innerhalb der DFD-FW befindlichen Speicher-Ressourcen werden an Rechnern betrieben, die als Ziel für Transfers von Datenprodukten aus dem DFD-Archiv eingerichtet sind. D.h. Produkte aus dem Archiv können, nach evtl. Nachprozessierung, direkt in die Speicherbereiche des Grid-SE abgelegt werden. Von dort werden sie mit Hilfe einer Synchronisierung auf eridanus in der DMZ transferiert. Umgekehrt können Daten, die aus dem Internet nach eridanus hochgeladen wurden, mit dem gleichen Synchronisationsmechanismus auf die Speicherbereiche innerhalb der DFD-FW gebracht werden, wo sie entweder warten, bis sie wieder abgerufen werden oder von wo aus sie auch ins DFD-Archiv geladen werden können.

Im Folgenden werden, ausgehend von den Speicherbereichen, deren Nutzung, die Datenaustausch-Mechanismen zwischen ihnen und die weitere Betriebsplanung des DFD-Grid-SE beschrieben.

3.1 Nutzungsbereiche des SE innerhalb und außerhalb der DFD-FW

In diesem Abschnitt wird beschrieben, welche Daten in welchen Speicherbereichen abgelegt werden, wie die Speicherbereiche aufgebaut sind und an welchen Servern sie angeschlossen sind.

Bereich der temporären Prozessierung (1 TB, innerhalb der DFD-FW):

Dieser Bereich verteilt sich auf die beiden innerhalb der DFD-FW befindlichen SUN Server. Je ein Anteil von 0,5 TB befindet sich auf SUN FIRE V440 in einer Partition mit 2 TB Größe und auf SUN FIRE V880 in einer Partition mit 1,08 TB Größe. Durch Beschränkung der anderen Bereiche mittels Disk-Quota können diese Anteile frei gehalten

werden. Diese Bereiche werden von Prozessoren für WISENT- und C3Grid-relevante Produkte genutzt, um temporär Eingangsdaten, Zusatzdaten und Zwischenergebnisse abzulegen.

Bereich der temporären Prozessierung und Warteschlange (5 TB, innerhalb der DFD-FW):

Dieser Bereich besteht aus je einer 2 TB großen Partition auf V440 und V880. Zusätzlich werden noch Teile anderer Partitionen auf beiden SUN Servern als Zwischenablage benutzt, so dass insgesamt etwa 5 TB zur Verfügung stehen. In diesem Bereich werden vor allem Daten und Daten-Produkte gespeichert, die entweder aus DFD-Archiv-Input zur Verteilung an Nutzer generiert wurden oder von Nutzern zur längerfristigen Speicherung auf eridanus hochgeladen wurden.

Backup-Bereich (bis 3 TB, innerhalb der DFD-FW):

Der Backup-Bereich befindet sich exklusiv auf V880, da hierhin die Sicherung des auf V440 befindlichen Spiegels des Pickup-Points auf eine unabhängige Maschine durchgeführt wird. Dieser Bereich besteht aus einer 2 TB großen Partition und Teilen anderer Partitionen. An V880 angeschlossen ist ein LTO-3-Bandlaufwerk, das zur Offline-Sicherung von der 2-TB-Partition benutzt wird. Über den Backup-Bereich laufen auch evtl. vom Nutzer in Auftrag gegebene sehr langfristige Datenspeicherungen auf Band.

Pickup-Point DFD (1 TB, innerhalb der DFD-FW):

Dieser Bereich ist ein Spiegel des Pickup-Points in der DMZ und liegt lokal an V440. Allerdings wird nicht die gesamte Partition 1:1 gespiegelt, sondern es werden dynamisch über diesen Bereich Daten von und zur DMZ kopiert und es werden Daten von diesem Bereich in andere interne Bereiche verschoben oder in diesem Bereich gelöscht. Anschließend wird der neue Zustand in die DMZ gespiegelt. Umgekehrt wird ein auf dem Pickup-Point in der DMZ von außen durch Nutzer veränderter Zustand nach innen gespiegelt. Dies erfordert eine ausgeklügelte Ablaufplanung und -kontrolle und vom Nutzer die Beachtung einiger zusätzlicher Regeln (siehe 3.2).

Pickup-Point DMZ (1 TB, außerhalb der DFD-FW in der DMZ an eridanus):

In diesem Bereich auf eridanus in der DMZ finden im Prinzip nur zwei Arten von Veränderungen statt. Zum einen können Nutzer Daten hierhin hochladen. Zum anderen werden über die Synchronisation von innerhalb der DFD-FW Daten hierher hochgeladen oder Daten werden implizit gelöscht, d.h. Daten werden auf dem Gegenstück dieses Bereiches innen gelöscht und der neue Zustand wird hierher synchronisiert. Der Pickup-Point, der aus einer einzigen Partition besteht, enthält daher zwei Basisverzeichnisse, nämlich „upload“ und „download“ aus Nutzersicht.

3.2 Datenaustausch-Mechanismen zwischen den Nutzungsbereichen

Die Synchronisation zwischen den Pickup-Points außerhalb und innerhalb der DFD-FW wird mit Hilfe von „rsync“, einer Public-Domain-Software, durchgeführt. „rsync“ benutzt den ssh-Dienst, d.h. den Port 22, der deshalb zwischen eridanus und V440 in beide Richtungen offen ist. „rsync“ kann in einem kumulativen und einem spiegelnden Modus betrieben werden, d.h. mit oder ohne Löschung von Dateien auf dem Zielrechner, die auf dem Quellrechner nicht mehr existieren. Innerhalb der DFD-FW wird der Transfer von Daten zwischen den beschriebenen Speicherbereichen teils mit „rsync“, teils mit gewöhnlichem Kopieren durchgeführt. Alle Datentransfer-Aktionen werden mit Hilfe von Skripten und FIFO-Dateien zeitlich koordiniert. Da dies allein aber nicht genügt, um zu

verhindern, dass z.B. gerade während ein Nutzer Daten hochlädt eine Löschoption ausgeführt wird, müssen zusätzlich Regeln eingeführt werden, die es erlauben, „fertige“ von „unfertigen“ Dateien zu unterscheiden. Hierfür ist die Mitarbeit des Nutzers notwendig, d.h. er muss beim Hochladen von Dateien bestimmte Namenskonventionen beachten. Beispielsweise können Dateien während des Hochladens mit bestimmten Endungen versehen werden, die nach Abschluss des Vorganges wieder zurückgenommen werden. „rsync“ kann dann auch unter Angabe von entsprechenden Ausschluß-Kriterien gestartet werden, so dass während eines Synchronisations-Zyklus „unfertige“ Dateien nicht angefasst werden und statt dessen bei einem der nächsten Zyklen an die Reihe kommen.

3.3 Weitere Betriebsplanung

Bei der Beantragung der Sondermittel für das DFD-SE wurde als Vorbedingung akzeptiert, dass das Grid-SE nachhaltig über das Ende des Projektes WISENT hinaus betrieben wird für mindestens die typische Funktions- und Nutzungsdauer der Hardware. Für die Komponenten innerhalb der DFD-FW an V440 und V880 ist dies ohne zusätzlichen Aufwand einzuhalten, da sie von den Sondermitteln gekauft wurden. Die Komponenten in der DLR-DMZ dagegen sind für den Zeitraum bis Ende 2010 von T-Systems-SfR gemietet. Das DFD hat daher zugesichert, dass nach Ablauf des Mietzeitraumes die DMZ-Komponenten mit DFD-eigenen Mitteln weiter betrieben werden. Dies wird aus heutiger Sicht mit einer Zusammenlegung von eridanus und dem WDC-RSAT-Steuerrechner verbunden sein.

Organisatorisch wird der Betrieb des SE wahrscheinlich ausschließlich mit Hilfe von GridFTP durchgeführt werden, da für die Middleware SRM/dCache nicht sichergestellt werden konnte, dass deren Funktionalität über die Firewalls hinweg genutzt werden kann und zudem für keine Referenzinstallation für LINUX 9.3 und keine weitere Unterstützung durch DGI zu bekommen war.

Ob sonstige vom DGI empfohlene oder im Zusammenhang mit Grid-Technologie übliche Administrationskonzepte angewandt werden, ist noch unklar. Die Nutzung von VOs etwa macht nur Sinn, wenn eine gewisse Grundmenge von Nutzern aus verschiedenen Communities sich die Ressource teilen will. Da das DFD-SE aber eine recht bescheidene Kapazität hat, zeichnet sich eher ein ebenfalls sehr kleiner Nutzerkreis ausschließlich aus WISENT und C3Grid ab. Auch von Funktionsträgern des DGI wurde bereits vorgeschlagen, dass SEs mit Kapazitäten unter 25 TB ausschließlich von den „beantragenden und eng kooperierenden Communities“ genutzt werden sollen.



Einsatz von und Erfahrungen mit Web Service-Technologien

Arbeitspaket 3.3

Guido Scherp, Ludger Bischofs
OFFIS – Bereich Energie
Escherweg 2
26121 Oldenburg
Internet: <http://www.offis.de>

Jochen Rehwinkel
meteocontrol GmbH
Spicherer Straße 48
86157 Augsburg
Internet: www.meteocontrol.de

1 Einleitung

Web Service-Technologien werden eingesetzt, um Dienste insbesondere für externe Nutzer anzubieten, die über einheitliche, standardisierte Schnittstellen und Protokolle genutzt werden können. Häufig wird dabei lediglich der Zugriff auf eine Altanwendung über einen Web Service „gewrapped“. Web Services ermöglichen zudem eine einfache Integration mehrerer Dienste, z.B. auf Basis von Workflows.

In dem Projekt WISENT gibt es zwei Anwendungen, für die entsprechende Web Service-Wrapper implementiert wurden. Dazu gehören ein Dienst zur Simulation von Photovoltaikanlagen (PV-Sim) sowie ein Dienst zur Kapselung des in Arbeitspaket 2.2 entwickelten MSG-Tailor-Systems (MSG-Tailor), die von meteocontrol (PV-Sim) sowie von DLR DFD (MSG-Tailor) angeboten werden.

Parallel dazu realisierte DLR-TT im EU-Projekt MESOR einen Webservice zur Extraktion von SOLEMI-Strahlungszeitserien mit Hilfe von JBoss und einer mit Eclipse erzeugten WSDL-Beschreibung ([http://www.webservice-energy.org /services/solemi.html](http://www.webservice-energy.org/services/solemi.html)). Diese Technologie scheint sich im Rahmen des Global Earth Observation System of Systems (GEOSS) international als Element der Datenverteilerarchitektur durchzusetzen.

Im Folgenden wird näher auf Umsetzung der PV-SIM und MSG-Tailor Web Services eingegangen, sowie von den Erfahrungen im Umgang mit diversen Web Service-Technologien, die dabei gemacht wurden.

2 Anbindung der Solarsimulation (PV-Sim) an einen Web-Service

Es handelt sich hierbei um die Anbindung der Simulation von Solaranlagen (AP 5.3) an andere Systeme. Es wurde eine Web-Service-Schnittstelle geschaffen, die es möglich macht, vom Internet aus Simulationen im Serverpark von meteocontrol anzustoßen und das Resultat abzufragen. Im Zuge dieser Tätigkeiten war es auch notwendig, die Simulation zu erweitern und an die neuen Anforderungen anzupassen. Aus diesem Grund wird dieses Dokument in zwei Teile gegliedert. Im ersten Teil wird die speziell für den Web-Service erstellte Simulation beschrieben. Der zweite Teil beleuchtet den Web-Service selbst und das dazugehörige Logging.

Durch den in AP 3.3 realisierten Solar-Web-Service wurde für meteocontrol die Voraussetzung für eine neue kommerzielle Dienstleistung geschaffen. Kunden können jetzt live erwartete Solaranlagenenerträge abfragen und diese Abfrage in eigene Softwareprodukte einbinden. Ein konkret realisierter Fall ist die Einbindung in eine Angebotserstellung zum Solaranlagenbau. Der Vertrieb des Kunden kann jetzt vor Ort die erwartete erzeugte Energie für die gewünschte Anlagenkonfiguration berechnen.

2.1 Die Ertragssimulation

2.1.1 Projektplan zur Simulationsprogrammierung

Die Haupttätigkeit von meteocontrol im AP 5.3 ist die Erstellung und Parallelisierung einer eigenen Ertragssimulation von Solaranlagen. Die Entwicklung wird in drei Versionen vollzogen:

- Version 1: Erstellung einer Simulation, die den jährlichen erwarteten Ertrag einer noch nicht gebauten Solaranlage errechnet. Es wird ein Programm entwickelt, das auf Grund einer vorgegebenen Solaranlagenkonfiguration (Wechselrichter, Module, Verschaltung, etc.) und der meteocontrol-Wetterdaten (Einstrahlung, Temperatur) des geplanten Standorts die erwartete erzeugte Energie simuliert. Version 1 wird

vor allem von Kunden von meteocontrol benötigt, es ist also notwendig eine Live-Schnittstelle zur Simulation zu bieten. Dies geschieht in Form eines Web-Service. Die Programmertätigkeiten für Version 1 sind nicht genau auf AP 3.3 und AP 5.3 zu trennen. Aus diesem Grund wird in diesem Dokument ein Überblick über die gesamte Version 1 gegeben.

- Version 2: Die Simulation in Version 2 greift direkt auf Solaranlagenkonfigurationen im System von meteocontrol zu. Es handelt sich hierbei um die ca. 5000 Solaranlagen, die momentan überwacht werden. Ziel ist hier eine tägliche Simulation des erwarteten Ertrags des Vortages jeder Anlage im System. Im meteocontrol Internetportal SaferSun kann dann die Simulation mit den tatsächlich gemessenen Erträgen verglichen werden. Weicht die Simulation stark vom tatsächlichen Ertrag ab, so ist ein Defekt bei der Solaranlage zu vermuten, und es wird ein Alarm ausgelöst. Die Tätigkeiten innerhalb Version 1 und 2 überschneiden sich, da teilweise der gleiche Programmcode benutzt werden kann. Die Programmierung der Version 2 findet in AP 5.3 statt.
- Version 3: Die Programmierung von Version 3 zielt auf die Parallelität des Simulationsvorgangs ab und soll auf dem Rechencluster des WISENT-Partners OFFIS realisiert werden. Version 2 wird durch Erweiterung zu Version 3. Es wird aber auch Programmcode von Version 1 weiterverwendet werden. Die Tätigkeiten für Version 3 zählen auch zu AP 5.3.

2.1.2 Version 1 der Ertragssimulation

Überblick

Bei Version 1 ist die Schnittstelle zum Web-Service unbedingt notwendig, ansonsten gibt es keine Verwendung für diese Simulationsart. In der Regel sind Verkäufer von Solaranlagen direkt vor Ort beim Kunden und erstellen live im Internet ein spezielles auf die Kundenbedürfnisse angepasstes Angebot. Mit einer lauffähigen Version 1 ist es nun möglich, das Angebot um einen erwarteten Jahresertrag der geplanten Solaranlage zu erweitern.

Über den Web-Service-Client wird eine Anfrage über das Internet an den Web-Service im Serverpark von meteocontrol verschickt. Die notwendigen Konfigurationsdaten der Solaranlage werden übermittelt. Die Simulation wird gestartet und greift dabei auf die Datenbank von meteocontrol zu. Es werden Wetterdaten sowie technische Informationen über verwendete Wechselrichter und Solarmodule ausgelesen. Der erwartete Jahresertrag wird abschließend dem Web-Service-Client übermittelt.

Im Zuge der Erstellung von Version 1 wurde auch eine Logging-Einheit programmiert, die es möglich macht, jede Funktion jedes Web-Services im meteocontrol-System zu überwachen und den Aufruf eines Kunden zu protokollieren. Die Kosten für den Kunden werden in die Datenbank geschrieben und können dann in Rechnung gestellt werden. Dieser Logging-Mechanismus kann auch für weitere Web-Services, die im Rahmen von AP 3.3. erstellt werden, genutzt werden.

Die Web-Service-Schnittstelle kann vom Kunden in beliebigen Programmcode eingebunden werden. Es ist z.B. möglich, den Web-Service-Client direkt in Programme zur Angebotserstellung einzubetten.

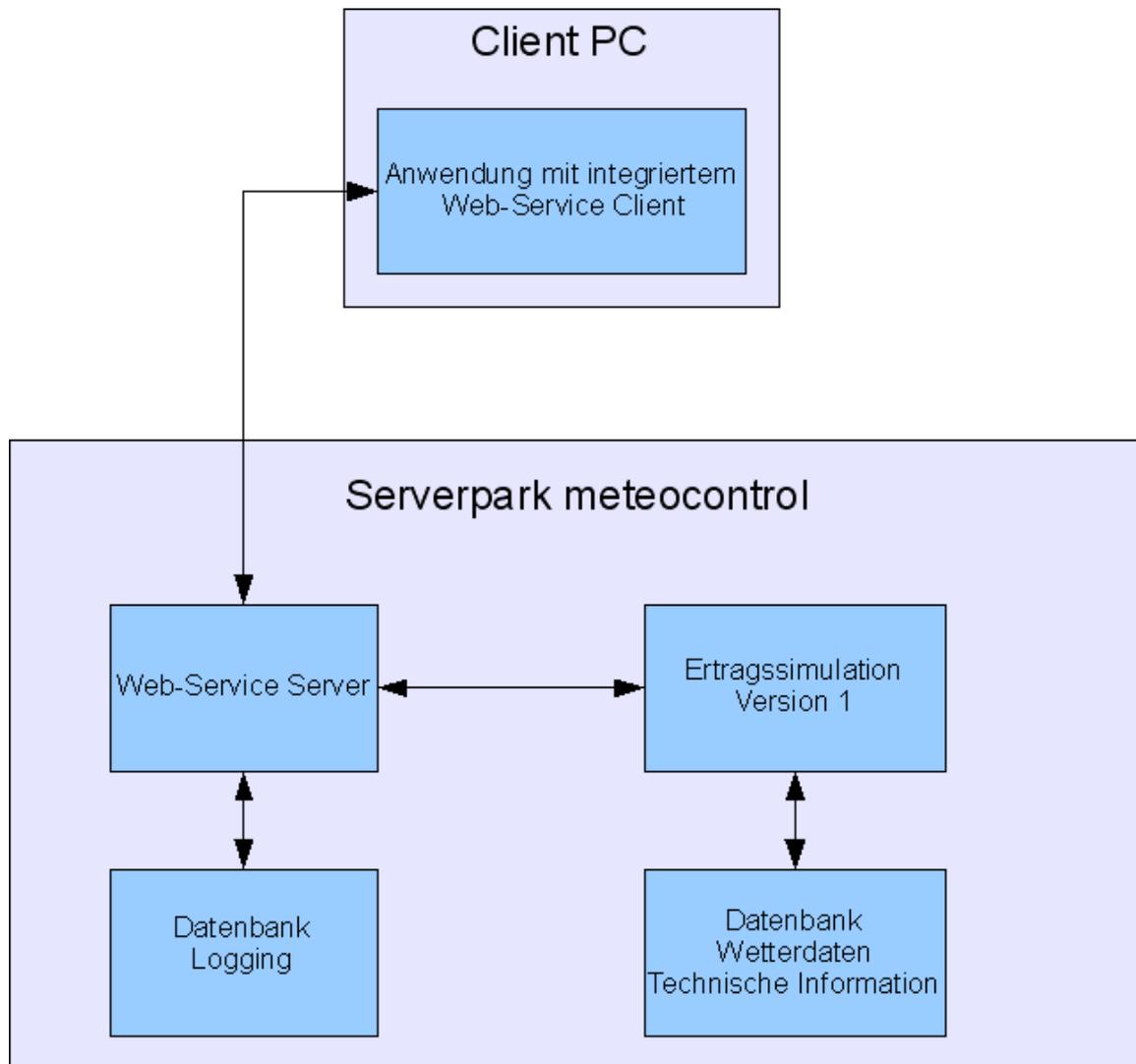


Abbildung 1 zeigt den schematischen Verlauf der Ertragssimulation. Endergebnis ist der zu erwartende Jahresertrag einer vom Kunden geplanten Solaranlage.

Verlauf der Ertragssimulation Version 1

Im Folgenden wird die innere Programmlogik der Ertragssimulation Version 1 dargestellt. Es handelt sich hierbei um den in PHP programmierten Rechenkern, der dann vom Web Service-Server über entsprechende Programmschnittstellen gesteuert werden kann.

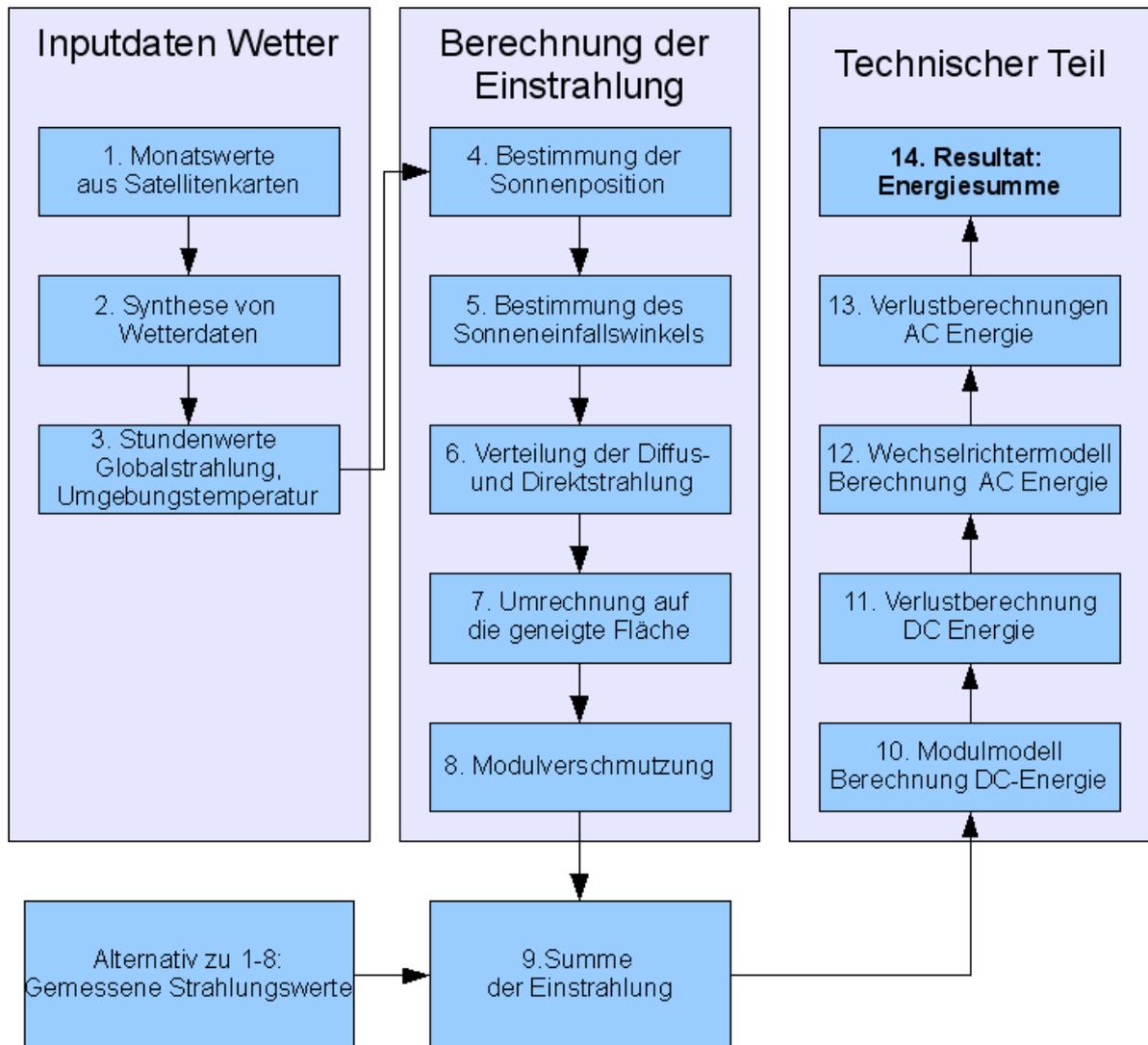


Abbildung 2: Schematische Darstellung der Programmlogik von Version 1

Die in Abbildung 2 dargestellten notwendigen Schritte zur Ertragssimulation werden im Folgenden näher erläutert. Ein Durchlauf von Schritt 1 bis 14 wird immer für einen Kalendertag durchgeführt. Soll ein ganzes Jahr berechnet werden, so wird die Simulation 365 mal durchgeführt. Es ist also generell möglich, den erwarteten Ertrag für einen beliebig großen Zeitraum zu berechnen

1. Schritt 1: Monatswerte aus Satellitenkarten: Es werden aus vorhandenen digitalen Satellitenkarten Wetterdaten für den Standort der geplanten Solaranlage ausgelesen. Es handelt sich hierbei um Globalstrahlung und Umgebungstemperatur. Es gibt nur einen Wert pro Monat.
2. Schritt 2: Berechnung von synthetischen Intervallwerten. Für die Simulation werden Intervallwerte im Stundenrhythmus benötigt. Aus diesem Grund wird eine Tagesverteilung der Satellitendaten auf Basis von statistischen Verfahren und physikalischen Zusammenhängen errechnet.
3. Schritt 3: Das Ergebnis sind Stundenwerte für die Globalstrahlung und die Umgebungstemperatur.
4. Schritt 4: Bestimmung der Sonnenposition für den zu berechnenden Tag
5. Schritt 5: Bestimmung des Sonneneinfallswinkels auf Grund des Standorts der

Solaranlage

6. Schritt 6: Die gewürfelte Einstrahlung wird aufgeteilt in Diffus- und Direktstrahlung.
7. Schritt 7: Die berechnete Einstrahlung wird umgerechnet auf die Neigung der Solaranlagenmodule. Die Neigung ist ein Parameter der dem Web-Service übergeben wird.
8. Schritt 8: Eine simulierte Modulverschmutzung verringert die Einstrahlung auf die Solarmodule.
9. Schritt 1-8 alternativ: Bei vor Ort gemessener Einstrahlung kann sofort im Schritt 9 begonnen werden. Diese Alternative ist vor allem in Version 2 und 3 möglich. Es können hier die an bereits gebauten Solaranlagen befestigten Einstrahlungssensoren verwendet werden. Ist eine Anlage noch in Planung, besteht diese Möglichkeit nicht.
10. Schritt 9: Die erwartete Summe der Einstrahlung, die auf die Solarmodule trifft, wird auf Stundenbasis berechnet.
11. Schritt 10: Es wird die erwartete Energie auf der Gleichstromseite der Solaranlage berechnet. Hierzu werden die technischen Daten der verwendeten Solarmodule benötigt. Außerdem ist es erforderlich zu wissen, wie die Module miteinander verbunden sind (Strings). Es wird der so genannte modulabhängige Faktor pro Wechselrichtereingang bestimmt. In diesem Schritt fließen auch die ermittelten Temperaturwerte mit ein.
12. Schritt 11: Es werden Verluste (z. B. Kabelverluste) auf der Gleichstromseite berechnet.
13. Schritt 12: Es wird die erwartete Energie auf der Wechselstromseite berechnet. Hierzu wird das Wechselrichtermodell verwendet. Es werden technische Daten zum ausgewählten Wechselrichterhersteller aus der Datenbank ausgelesen.
14. Schritt 13: Es werden Verluste (z. B. Kabelverluste) auf der Wechselstromseite berechnet.
15. Schritt 14: Es wird die simulierte Energiesumme des Tages und letztendlich des Gesamtzeitraumes berechnet.

2.2 Der Solar-Web Service Version 1

2.2.1 Die Funktionen des Web-Service

Bei der Programmierung des Web-Service wurde die PHP-Bibliothek NUSOAP verwendet. NUSOAP bietet alle technischen Möglichkeiten, die eine standardisierte Web-Service-Kommunikation erfordert. Der Web-Service bietet folgende Funktionen, die dann vom Web-Service-Client aufgerufen werden:

1. function doLogin
2. Übergabeparameter: Benutzerkennung, Passwort
3. Rückgabewert: Session-ID
4. Es wird die Benutzerkennung und das Passwort des Kunden übergeben. Username und Passwort sind in der generellen Web-Service Konfigurationsdatenbank abgespeichert. Es wird eine Session-ID zurückgegeben. Bei jedem Aufruf einer anderen Web-Service-Funktion wird die Session-ID mit übergeben. So kann jeder Aufruf einer Funktion dem Kunden zugeordnet werden und dementsprechend verrechnet werden.
5. function getAllModulTyp
6. Übergabeparameter: Session-ID
7. Rückgabewert: Alle für die Simulation verfügbaren Solarmodule und deren ID's
8. Der Web-Service-Client kann hier alle dem Kunden verfügbaren Solarmodule

abfragen. Es werden der Hersteller und der Modelltyp übermittelt. Diese Information kann beim Client z. B. dafür benutzt werden, im Programm zur Angebotserstellung eine Auswahlliste von allen konfigurierbaren Solarmodultypen bereitzustellen.

9. function getAllWrTyp

10. Übergabeparameter: Session-ID

11. Rückgabewert: Alle für die Simulation verfügbaren Wechselrichtermodelle und deren ID's

12. Der Web-Service Client kann hier alle dem Kunden verfügbaren Wechselrichtermodelle abfragen. Es werden der Hersteller und der Modelltyp übermittelt. Diese Information kann beim Client z.B. dafür benutzt werden, im Programm zur Angebotserstellung eine Auswahlliste von den konfigurierbaren Wechselrichtern bereitzustellen.

13. function getSimErtrag

14. Übergabeparameter: Session-ID, Postleitzahl, Neigung, Ausrichtung, Wechselrichter-ID, Wechselrichteranzahl, Modul-ID, Modulanzahl

15. Rückgabewert: Erwarteter Jahresertrag in kWh

16. Der Simulation werden alle benötigten Eingangsparameter übergeben. Die Postleitzahl dient zur Ermittlung des Standorts mit Längen- und Breitengrad. Mit Neigung ist die Neigung der Solarmodule in Grad gemeint. Die Ausrichtung bezieht sich auf die Himmelsrichtung in die die Module zeigen, sie wird ebenfalls in Grad angegeben. Die Wechselrichter-ID sowie die Modul-ID dienen zur Identifikation der verwendeten Hardware der Solaranlage. Die ID's wurden in der Regel vorher vom Client über die Funktionen getAllWrTyp und getAllModulTyp beschafft. Zusätzlich wird vom Client noch übergeben, wie viele Module und Wechselrichter verwendet werden. Der Rückgabewert ist das eigentliche Resultat der Simulation. Es kann folgende Aussage getroffen werden: „Für die Solaranlage mit der übermittelten Konfiguration wurde ein erwarteter Jahresertrag von XXX,XX kWh simuliert.“

2.2.2 Die Konfigurationsdatenbank und das Logging

Die Konfigurationsdatenbank von meteocontrol für die Web-Services ist so aufgebaut, dass jeder weitere Web-Service ebenfalls verwaltet werden kann. Die Web-Service Konfigurationsdatenbank enthält folgende Tabellen:

- T_KUNDE: Hier wird generelle Information zum Aufrufer des Web-Service gespeichert, sowie Passwort und Benutzerkennung.
- T_WEBSERVICE: Hier werden die Web-Services eingetragen. Die gesamte Solarsimulation Version 1 ist durch einen Eintrag in dieser Tabelle dargestellt.
- T_FUNKTION: Hier wird jede Funktion eines Web-Services hinterlegt
- T_KOSTEN: Hier wird jede Funktion eines Web-Services für jeden Kunden mit Kosten belegt. Es ist also möglich, eine genaue Preisdifferenzierung pro Kunde vorzunehmen.
- T_REI_KUNDEN_MODUL: Hier wird hinterlegt, welcher Kunde auf welche Solarmodule zugreifen darf. Die Funktion getAllModulTyp liefert alle Einträge eines Kunden.
- T_REL_KUNDEN_WR: Hier wird hinterlegt, welcher Kunde auf welche Wechselrichtertypen zugreifen darf. Die Funktion getAllWRTyp liefert alle Einträge eines Kunden.

- T_LOGGING: Hier wird hinterlegt, welcher Kunde zu welchem Zeitpunkt eine bestimmte Funktion eines Web-Service aufgerufen hat. Zusätzlich werden die dadurch entstandenen Kosten mitprotokolliert (siehe Abbildung 3).

	ID_LOGGING	TIMESTAMP ▲	ID_KUNDE	ID_FUNKTION	SESSIONID	KOSTEN
<input type="checkbox"/>  	4	2004-06-22 14:53:44	1	4	1087908816_1	144
<input type="checkbox"/>  	8	2004-06-22 14:55:14	1	4	1087908910_1	144
<input type="checkbox"/>  	12	2004-06-22 14:55:30	1	4	1087908918_1	144
<input type="checkbox"/>  	16	2004-06-22 14:56:37	1	4	1087908991_1	144
<input type="checkbox"/>  	20	2004-06-22 14:59:38	1	4	1087909172_1	144
<input type="checkbox"/>  	24	2004-06-22 14:59:47	1	4	1087909181_1	144
<input type="checkbox"/>  	28	2004-06-22 15:23:56	1	4	1087910631_1	144
<input type="checkbox"/>  	32	2004-06-24 12:02:00	1	4	1088071305_1	144
<input type="checkbox"/>  	36	2004-06-24 12:05:43	1	4	1088071538_1	144
<input type="checkbox"/>  	40	2004-06-24 12:07:25	1	4	1088071641_1	144
<input type="checkbox"/>  	59	2004-06-24 15:34:08	1	4	1088084038_1	144
<input type="checkbox"/>  	63	2004-06-24 15:34:20	1	4	1088084056_1	144

Abbildung 3: Auszug aus der Web-Service Tabelle T_LOGGING

```

//-----
$server->register('getAllModulTyp',           // method name
    array('sSessionId' => 'xsd:string'),    // input parameters
    array('sModulTyp' => 'xsd:string'),     // output parameters
    'urn:anlagenertrag',                  // namespace
    'urn:anlagenertrag#getModulTyp',      // soapaction
    'rpc',                                 // style
    'encoded',                             // use
    'Returns available Modules'           // documentation
);

//-----
$server->register('getAllWrTyp',             // method name
    array('sSessionId' => 'xsd:string'),    // input parameters
    array('sWrTyp' => 'xsd:string'),        // output parameters
    'urn:anlagenertrag',                  // namespace
    'urn:anlagenertrag#getWrTyp',         // soapaction
    'rpc',                                 // style
    'encoded',                             // use
    'Returns available Converters'        // documentation
);

//-----
$server->register('getSimulation',          // method name
    array('sSessionId' => 'xsd:string',
        'iPlz' => 'xsd:int',
        'iNeigung' => 'xsd:int',
        'iAusrichtung' => 'xsd:int',
        'iIdModul' => 'xsd:int',
        'iIdWr' => 'xsd:int',
        'iModulAnzahl' => 'xsd:int',
        'iWRAnzahl' => 'xsd:int'),        // input parameters
    array('iSimErtrag' => 'xsd:double'),   // output parameters
    'urn:anlagenertrag',                  // namespace
    'urn:anlagenertrag#getWrTyp',         // soapaction
    'rpc',                                 // style
    'encoded',                             // use
    'Returns simualated energy yield'     // documentation
);

```

Abbildung 4: Programmcode (Auszug) Web-Service

2.3 Ausblick

Meteocontrol möchte unterschiedliche Zugriffsmöglichkeiten auf bereitgestellte Web-Services anbieten können, so dass ein einfacher Zugriff über bekannte Programmiersprachen unter Zuhilfenahme vorgefertigter Bibliotheken möglich ist. Ein vielversprechender Lösungsansatz dazu ist die automatische Generierung von Web-Service-Client-Stubs auf Basis von WSDL-Dateien (WSDL = Web Service Description Language), um Bibliotheken für den Zugriff auf Web-Services in unterschiedlichen Programmiersprachen mit geringem Aufwand anbieten zu können. Eine automatische Generierung von Web Service-Clients auf Grund der Web Service-Struktur würde eine erhebliche Arbeitserleichterung darstellen. Zusätzlich ist es sehr interessant für meteocontrol, künftig Kunden auf Knopfdruck Clients in unterschiedlichen Programmiersprachen (z.B. Java, C++) bereitzustellen. Erfahrungen zur automatischen

Generierung von Web-Service-Clients werden im Kapitel zu „Erfahrungen mit Web-Service-Technologien“ beschrieben. Eine weitere Tätigkeit im Arbeitspaket 3.3 ist die Anbindung der Wetterdaten an einen Web-Service. Es soll für den Client möglich sein, standortbezogene Wetterdaten direkt vom Server von meteocontrol abzufragen. Dieser Web-Service kann dann auch als Schnittstelle für die in AP 5.3. entstehende Solaranlagen simulation dienen.

3 MSG-Tailor-Web-Service

3.1 Einleitung

Für die Erzeugung der Datenprodukte der Meteosat First Generation wird derzeit ein Tailor (siehe Abbildung 5) eingesetzt, der die in einer Cube-Struktur abgelegten Rohdaten aus dem Archiv extrahiert und für die weitere Berechnung von Wolkenindex- und Strahlungsprodukten bereitstellt.

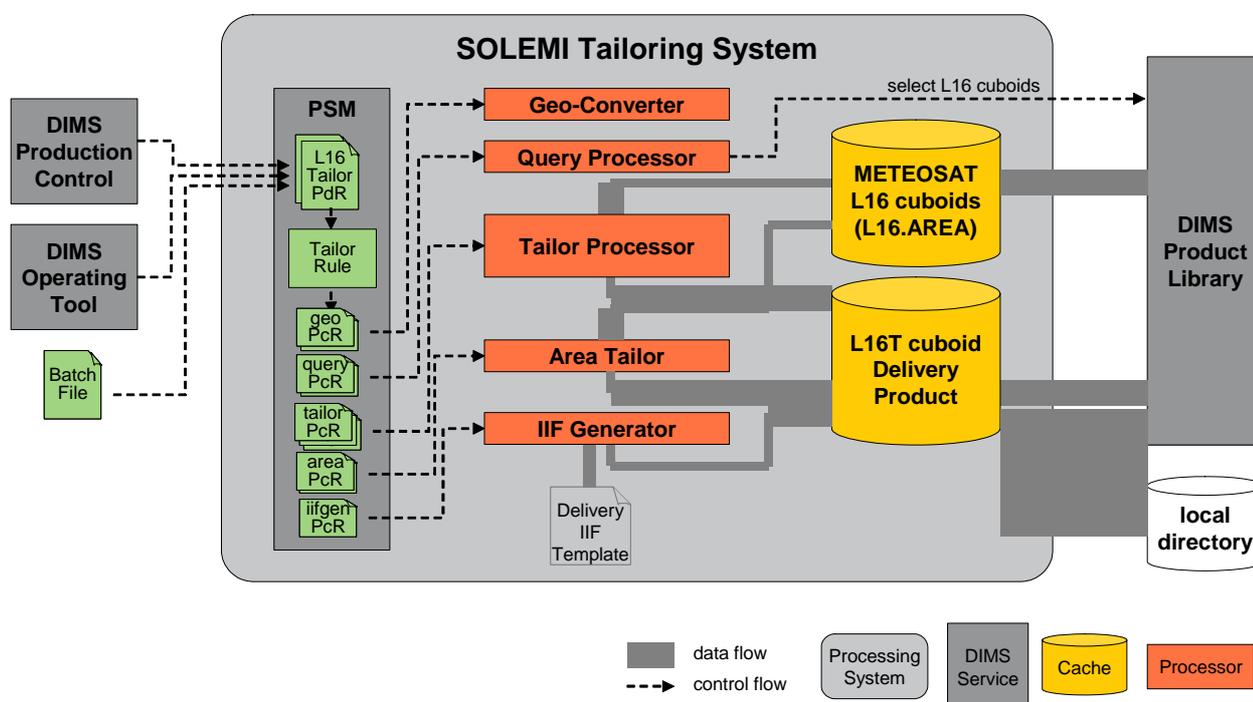


Abbildung 5: Solemi Tailor Processing System für Meteosat First Generation Rohdaten

Mit der neuen Satellitengeneration Meteosat Second Generation (MSG) verändern sich nun die Rahmenbedingungen. Die Rohdaten werden vom Satellitenbetreiber EUMETSAT nicht mehr Vollscheiben-weise sondern in 8 Segmente unterteilt geliefert. Für die Rohdatenextraktion existiert bereits das umfangreiche Programmpaket SCENES/APOLLO, das Daten aus dem Archiv bestellt, diese passend für die gewünschte Region ausschneidet und Wolken-, Schnee- und Wasserdampfprodukte erstellt. I.A. geschieht dies auf großen Ausschnitten bzw. der vollen MSG-Scheibe.

Diese Produkte werden anschließend in einer den Rohdaten der Meteosat First Generation vergleichbaren Cube-Struktur ins Archiv abgelegt. Dasselbe ist in Zukunft für die vom Kooperationspartner Ecole des Mines gelieferten Strahlungsprodukte geplant.

Daher muss der Tailor nun statt Rohdaten bereits weiter verarbeitete Produktstufen extrahieren und nach Nutzerwunsch passend raum-zeitlich ausschneiden. Eine

entsprechende Software-Adaption wurde parallel in WISENT AP 3.2 vorgenommen.

Für diese Tailor-Prozessoren soll in WISENT AP3.3 nun ein webbasierter Zugang für Nutzer außerhalb des DLR geschaffen werden. Diese sollen zukünftig auch ins WDC integriert werden.

3.2 Realisierung

Aufgrund der strengen IT-Sicherheitsregeln beim DLR ist es Firewall-technisch (siehe Abbildung 6) nicht möglich, den MSG-Tailor-Service direkt an das eigentliche MSG-Tailor-System anzukoppeln. Das MSG-Tailor-System befindet sich innerhalb des internen DLR-Netzwerks und ein darauf eingerichteter Web Service wäre somit von außen nicht erreichbar. Daher wurde der Web Service auf einem Rechner (eridanus.caf.dlr.de) in der DMZ eingerichtet, der bereits zur Einrichtung der im Rahmen der D-Grid-Sonderinvestition angeschafften Speichereinheit verwendet wird.

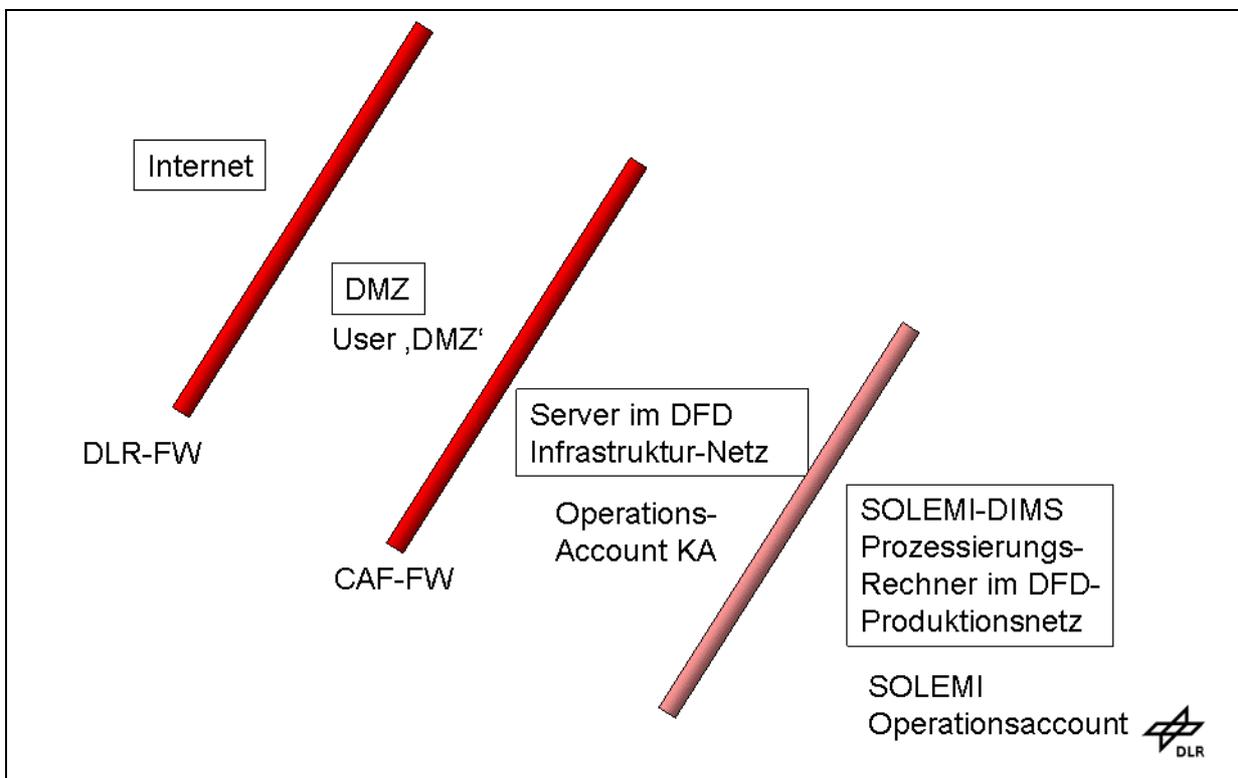


Abbildung 6: Firewall-Struktur zwischen DMZ, Infrastruktur- und Produktionsnetz im DFD

Auf dem Dateisystem von eridanus gibt es zwei Bereiche, die über „rsync“ mit der eigentlichen Speichereinheit, die sich im internen DLR-Netzwerk befindet, periodisch synchronisiert werden. Der erste Bereich wird dabei von eridanus auf die Speichereinheit synchronisiert und der zweite Bereich in die umgekehrte Richtung. D.h. Daten können aus dem internen DLR-Netzwerk in die DMZ übertragen werden, um sie beispielsweise externen Nutzern zur Verfügung zu stellen („Pick-up point“) und umgekehrt. Detailliertere Informationen zur Firewallproblematik und dem Betrieb der Speichereinheit befinden sich in dem Dokument „Bericht zu den Auswirkungen der DLR-IT-Sicherheitsstrukturen beim Datentransfer mit Partnern außerhalb des DLR“. Darauf aufbauend sind mit dem MSG-Tailor-Web Service folgende Prozessschritte verbunden, die auch in den Abbildungen 8 und 9 grafisch dargestellt werden.

1. Der Nutzer schickt die Anfrage an der MSG-Tailor-Web Service.

2. Der MSG-Tailor-Web Service generiert eine XML-Datei (siehe Abbildung 7), die in dem Speicherbereich abgelegt wird, der auf die interne Speichereinheit synchronisiert wird. Der Client bekommt ein Verzeichnis (wird temporär angelegt) zurück, auf das per GridFTP zugegriffen werden kann und in dem sich später die gewünschten Daten befinden.
3. Nach der Synchronisation wird intern (das Verzeichnis wird periodisch abgefragt) ein Prozess basierend auf den Informationen in der XML-Datei im MSG-Tailor-System gestartet, der die gewünschten Daten extrahiert.
4. Die extrahierten Daten werden in den Speicherbereich der internen Speichereinheit kopiert, der auf eridanus synchronisiert wird. Das Verzeichnis entspricht dem Verzeichnis, das der Nutzer nach seiner Anfrage erhalten hat.
5. Nach der Synchronisation kann der Nutzer die Daten per GridFTP oder FTP abholen.

```
<psm-batch>
  <ProductionRequest type="MSG-Tailor">
    <ProcessingParameter key="mission">MSG</ProcessingParameter>
    <ProcessingParameter key="productParameter">mask</ProcessingParameter>
    <ProcessingParameter key="startTime">2004-06-03T11:00:00</ProcessingParameter>
    <ProcessingParameter key="stopTime">2004-06-03T19:30:00</ProcessingParameter>
    <ProcessingParameter key="outputFormat">hdf4</ProcessingParameter>
    <ProcessingParameter key="regionType">pixel</ProcessingParameter>
    <ProcessingParameter key="region">
      <GeoPoint latitude="1600" longitude="560"/>
      <GeoPoint latitude="1850" longitude="850"/>
    </ProcessingParameter>
    <OutputProduct type="MSG.SEVIRI.L2T"
      id="((KP:sensor:SEVIRI),(KP:code:L2T))"
      path="/export/home/example" />
  </ProductionRequest>
</psm-batch>
```

Abbildung 7: Beispiel-Parametersatz für eine Zeitreihenextraktion mit Pixelkoordinaten

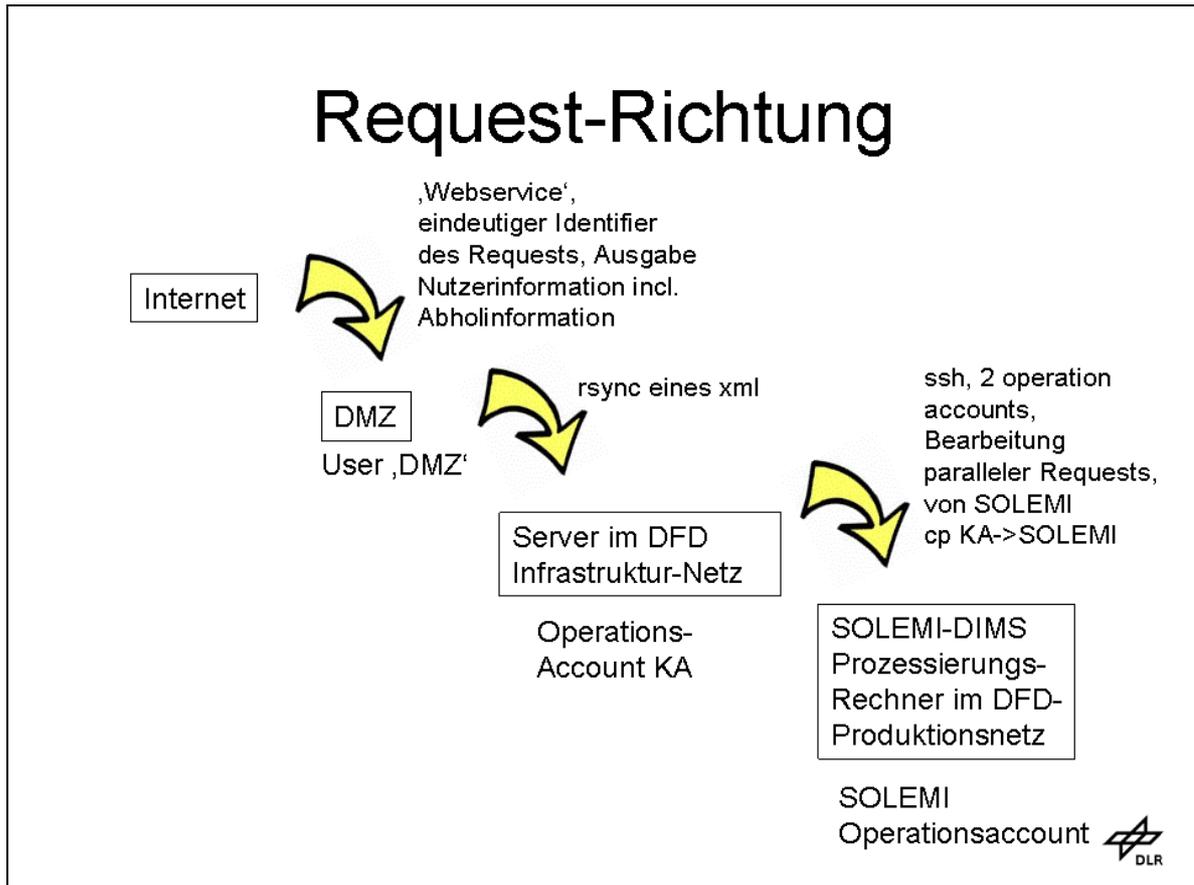


Abbildung 8: Struktur eines Tailor-Auftrags bis ins Produktionsnetz hinein

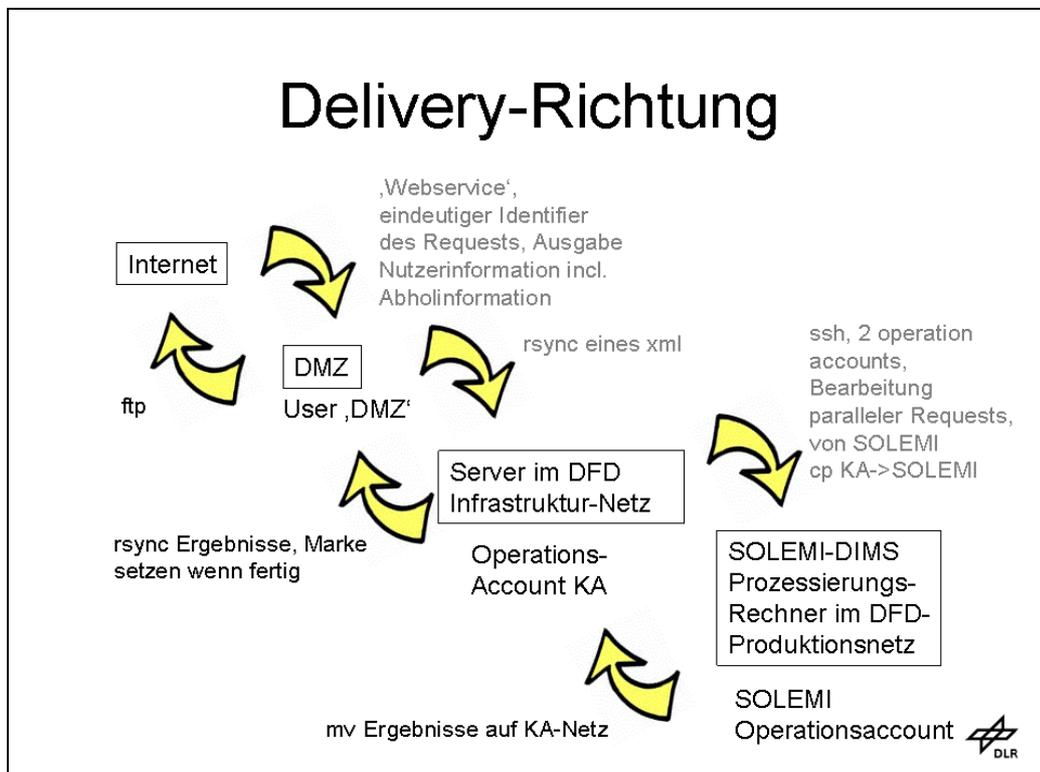


Abbildung 9: Struktur eines Tailor-Delivery-Prozesses bis zum Endnutzer

Demnach bietet der MSG-Tailor-Web Service nur eine Funktion an, um das MSG-Tailor-System im Hintergrund anzusprechen. Der Nutzer erhält ein Verzeichnis, aus dem die Daten später abgeholt werden können. Eine Überwachung des Fortschritts auf Web Service-Ebene wird zurzeit nicht umgesetzt. Der Nutzer muss durch wiederholtes Abfragen (Polling) des Verzeichnisses per GridFTP schauen, ob alle Daten vorhanden sind, was durch eine entsprechende Datei markiert wird, die bei der Synchronisation grundsätzlich zuletzt geschrieben wird.

Technisch gesehen wurde der MSG-Tailor-Web Service auf 2 Arten umgesetzt, mit Perl und mit Axis2C, wobei jeweils Apache2 als HTTP-Server verwendet wurde. Version 1 des MSG-Tailor-Web Service ist auf Basis von Perl entstanden, weil damit sehr schnell und einfach ein funktionierender Web Service implementiert werden kann, lediglich ein paar Perl-Pakete installiert werden müssen und nichts zu konfigurieren ist und die SOAP-Engine von Perl auf CGI basiert und somit einfach mit jedem Web Server wie Apache zusammenarbeitet. Leider gab es einige Probleme mit der Client-Generierung, was dazu geführt hat, dass Version 2 in Axis2C implementiert wurde. Eine detaillierte Beschreibung dazu gibt es in Abschnitt 3.

4 Erfahrungen mit Web Service-Technologien

Dieser Abschnitt beinhaltet die Erfahrungen mit diversen Web Service-Technologien, die im Laufe des Projektes WISENT gemacht wurden. Sowohl zur Implementierung von Web Services selbst, als auch zur Generierung von so genannten Client-Stubs. Zum Teil gibt es eine kurze Anleitung, wie mit der verwendeten Technologie ein Web Service implementiert werden kann.

4.1 Axis2CPP

4.1.1 Installation

Mit AxisCPP (<http://ws.apache.org/axis/cpp/>) können Web Services bzw. entsprechende Clients auf Basis von C++ implementiert werden. Allerdings wird AxisCPP seit März 2006 nicht weiterentwickelt. Für die Installation von AxisCPP wird ein entsprechendes Apache-Modul verwendet, somit ist es an diesen Web Server gebunden. Des Weiteren ist AxisCPP nicht so einfach lauffähig zu bekommen. Zur Installation wurden sowohl die Binary- als auch die Source-Version verwendet. Im Folgenden wird aufgeführt, welche Probleme aufgetreten sind bzw. was für manche Schritte wichtig ist, damit AxisCPP läuft. Auf dem Testsystem war Debian-Linux (sid) installiert.

- Laut Webseite werden für AxisCPP Apache 2.0.* (oder Apache 1.3.*) und Xerces-C 2.20 benötigt. Tatsächlich ist das Apache2-Modul auf eine entsprechende Version angewiesen (aktuell ist zurzeit allerdings Apache 2.2.*). In diesem Test wurde Apache bzw. httpd 2.0.63 verwendet (kann über die offizielle Webseite <http://archive.apache.org/dist/httpd/> bezogen werden). Die Xerces-C-Version kann ruhig neuer sein, im Test wurde 2.8.0 verwendet.
- Für Xerces-C müssen noch folgende Links in `/usr/lib` eingetragen werden
 - `libxerces-c.so` -> `libxerces-c.so.28` (wird von der Source-Version benötigt)
 - `libxerces-c.so.20` -> `libxerces-c.so.28` (wird von der Binary-Version benötigt)

- Zur Sicherheit sollten folgende Umgebungsvariablen gesetzt sein. Bis auf `XERCES_HOME` entsprechen aber alle Umgebungsvariablen den eigentlichen Standardwerten).
 - `APACHE2_HOME=/usr/local/apache2/`: Verzeichnis, in dem die entsprechende Apache-Version installiert ist.
 - `AXISCPP_DEPLOY=/usr/local/axiscpp_deploy`: Verzeichnis, in das AxisCPP „deployed“ warten soll. `AXISCPP_HOME=/work/AxisCPP/axis-c-1.6b-Linux-trace-(src|bin)`: Verzeichnis, in das das Archiv (Binary|Source) entpackt wurde.
 - `XERCES_HOME=/usr/`: Eigentlich das Installationsverzeichnis von XERCES-C, aber es wird lediglich die Bibliothek `lib/libxerces-c.so` (Source) bzw. `lib/libxerces-c.so.20` (Binary) benötigt. Daher muss XERCES-C nicht zwingend installiert werden, es reicht daher der Pfad auf `/usr` aus.
 - `LD_LIBRARY_PATH=$XERCES_HOME/lib:$AXISCPP_DEPLOY/lib`
- Das Deploy-Verzeichnis (`$AXISCPP_DEPLOY`) muss immer manuell gefüllt werden, d.h. sowohl in der Binary als auch in der Source-Version muss der Inhalt des in dem jeweiligen Archivs enthaltenen, vorgefertigten Deploy-Verzeichnisses (`deploy/`) nach `$AXISCPP_DEPLOY` kopiert werden. Alternativ kann natürlich auch ein entsprechender Link gesetzt, bzw. die Variable `$AXISCPP_DEPLOY` passend gesetzt werden.
- In der Source-Version gibt es in der Konfigurationsdatei `$AXISCPP_HOME/build/build.common.properties` die Property `dir.obj`, die auf das Verzeichnis gesetzt ist, in dem die Ergebnisse des Build-Prozesses kopiert werden sollen. Per Default ist diese Property seltsamerweise `${basedir}/../../../../obj`, wobei `${basedir}` in dem dazugehörigen ANT-Build-Skript für das aktuelle Verzeichnis steht, in dem der Build-Prozess gestartet wird. In dem hier beschriebenen Testfall ergab sich daraus das Verzeichnis `/obj/`.

4.1.2 Beispiel-Service

Ein Beispiel-Service (Calculator) ist unter <http://ws.apache.org/axis/cpp/arch/End-2-End-Sample.html> aufgeführt. Dieses und weitere Beispiele befinden in dem Verzeichnis `$AXISCPP_HOME/samples`. An dieser Stelle wird nur anhand des Calculator-Service auf die größten Fehler des Beispiel-Services eingegangen.

- Als WSDL wird empfohlen `$AXISCPP_DEPLOY/wsdl/calculator.wsdl` zu nehmen und gleich anzupassen. Und zwar muss am Ende der Adresse das `calculator` klein geschrieben werden. Die Services in AxisCPP werden immer in der Kleinschreibung der Namen angeboten.

```
<wsdlsoap:address location="http://localhost/axis/calculator"/>
```

- Der Befehl zur Kompilierung der Quelldateien ist falsch. Der richtige Befehl lautet.

Binary:

```
g++ *.cpp -shared -I$AXISCPP_HOME/include -L$AXISCPP_HOME/lib/axis/ -laxis_server -o Calculator.so
```

Source:

```
g++ *.cpp -shared -I$AXISCPP_HOME/include -L/obj/bin -laxis_server -o Calculator.so
```

- Das gleiche gilt für den Befehl, um den Client zu kompilieren. Der richtige Befehl lautet.

Binary:

```
g++ *.cpp -I$AXISCPP_HOME/include -L$AXISCPP_HOME/lib/axis/ -laxis_client -o Calculator
```

Source:

```
g++ *.cpp -I$AXISCPP_HOME/include -L/obj/bin/ -laxis_client -o Calculator
```

- Beim „Deployment“ (Einfügen in \$AXISCPP_DEPLOY/etc/server.wsdd) muss der Name der Services immer klein geschrieben werden, s.o. Sonst sind die URLs für die WSDL-Publizierung und den Service unterschiedlich, d.h. Calculator?WSDL für die WSDL und calculator für den eigentlichen Service am Ende der URL.

```
...  
<service name="calculator" ...  
  ...  
</service>  
...
```

4.1.3 Aufgetretene Probleme

Im Folgenden sind die wesentlichen Nachteile von AxisCPP aufgeführt.

- Wird seit März 2006 nicht mehr weiterentwickelt.
- Die Dokumentation ist miserabel und es hat sehr lange gedauert, um einen lauffähigen Service umzusetzen.
- Ausgangspunkt für die Web-Service-Entwicklung ist eine manuell erstellte WSDL-Datei. Etwas Analoges zu Java2WSDL auf Basis einer Java-Schnittstelle, z.B. CPP2WSDL, gibt es nicht.
- Nach außen wird lediglich die selbst erstellte WSDL-Datei publiziert (?WSDL am Ende der URL) und nicht automatisch erzeugt (wie bei Axis2/Java).

4.2 Axis2/C

4.2.1 Installation

Axis2/C (<http://ws.apache.org/axis2/c/>) ist der C-Ableger von Axis2/Java und basiert analog zu AxisCPP auf einem Apache-Modul. Die Installation und Konfiguration mit einem Apache2-Web Server ist mit der vorgegebenen Installationsanleitung (<http://ws.apache.org/axis2/c/docs/installationguide.html>) recht einfach. Im Folgenden wird angenommen, dass die Umgebungsvariable \$AXIS2C_HOME richtig gesetzt ist, z.B. auf /usr/local/axis2c, sowie das Axis2C über die URL <http://localhost/axis2> erreichbar ist.

4.2.2 Beispiel-Service

Als Beispiel wird der auf der Axis2-Webseite aufgeführte Hello-Service (http://ws.apache.org/axis2/c/docs/axis2c_manual.html#quick_start) verwendet. Der Code der Web-Service-Implementierung sieht allerdings sehr unübersichtlich aus, weil alles in einer Datei gespeichert ist. Insbesondere bestimmte Axis-Funktionen werden vom Anwender nie angefasst. Mit dem Skript WSDL2C.sh (aus \$AXIS2_HOME/bin/) kann aus einer WSDL-Datei sowohl Client- als auch Server-Code generiert werden, der besser strukturiert ist. WSDL2C.sh ruft allerdings Java-Bibliotheken auf, die nur in Axis2/Java vorhanden sind. Daher muss Axis2/Java installiert werden, sowie dessen Installationsverzeichnis in das Skript WSDL2C.sh eingetragen werden. Des Weiteren ist das Skript im MS-DOS-Encoding gespeichert und muss unter Linux noch im entsprechenden UNIX-Encoding abgespeichert werden.

Es folgt die Implementierung des Hello-Service (`hello_svc.c`).

```
#include <axis2_svc_skeleton.h>
#include <axutil_log_default.h>
#include <axutil_error_default.h>
#include <axutil_array_list.h>
#include <axiom_text.h>
#include <axiom_node.h>
#include <axiom_element.h>
#include <stdio.h>

axiom_node_t *axis2_hello_greet(const axutil_env_t *env,
                                axiom_node_t *node);

int AXIS2_CALL
hello_free(axis2_svc_skeleton_t *svc_skeleton,
           const axutil_env_t *env);

axiom_node_t* AXIS2_CALL
hello_invoke(axis2_svc_skeleton_t *svc_skeleton,
             const axutil_env_t *env,
             axiom_node_t *node,
             axis2_msg_ctx_t *msg_ctx);

int AXIS2_CALL
hello_init(axis2_svc_skeleton_t *svc_skeleton,
           const axutil_env_t *env);

axiom_node_t* AXIS2_CALL
```

```

hello_on_fault(axis2_svc_skeleton_t *svc_skeli,
               const axutil_env_t *env, axiom_node_t *node);

axiom_node_t *
build_greeting_response(const axutil_env_t *env,
                       axis2_char_t *greeting);

axiom_node_t *
build_greeting_response(const axutil_env_t *env, axis2_char_t *greeting)
{
    axiom_node_t* greeting_om_node = NULL;
    axiom_element_t * greeting_om_ele = NULL;

    greeting_om_ele = axiom_element_create(env, NULL, "greetResponse", NULL,
&greeting_om_node);

    axiom_element_set_text(greeting_om_ele, env, greeting, greeting_om_node);

    return greeting_om_node;
}

static const axis2_svc_skeleton_ops_t hello_svc_skeleton_ops_var = {
    hello_init,
    hello_invoke,
    hello_on_fault,
    hello_free
};

axis2_svc_skeleton_t *
axis2_hello_create(const axutil_env_t *env)
{
    axis2_svc_skeleton_t *svc_skeleton = NULL;
    svc_skeleton = AXIS2_MALLOC(env->allocator,
                               sizeof(axis2_svc_skeleton_t));

    svc_skeleton->ops = &hello_svc_skeleton_ops_var;

    svc_skeleton->func_array = NULL;

    return svc_skeleton;
}

int AXIS2_CALL
hello_init(axis2_svc_skeleton_t *svc_skeleton,
           const axutil_env_t *env)
{
    svc_skeleton->func_array = axutil_array_list_create(env, 0);
    axutil_array_list_add(svc_skeleton->func_array, env, "helloString");
    return AXIS2_SUCCESS;
}

axiom_node_t* AXIS2_CALL
hello_invoke(axis2_svc_skeleton_t *svc_skeleton,
             const axutil_env_t *env,
             axiom_node_t *node,
             axis2_msg_ctx_t *msg_ctx)
{
    return axis2_hello_greet(env, node);
}

```

```

}

axiom_node_t* AXIS2_CALL
hello_on_fault(axis2_svc_skeleton_t *svc_skeli,
               const axutil_env_t *env, axiom_node_t *node)
{
    axiom_node_t *error_node = NULL;
    axiom_node_t* text_node = NULL;
    axiom_element_t *error_ele = NULL;
    error_ele = axiom_element_create(env, node, "EchoServiceError", NULL,
                                    &error_node);
    axiom_element_set_text(error_ele, env, "Echo service failed ",
                           text_node);
    return error_node;
}

int AXIS2_CALL
hello_free(axis2_svc_skeleton_t *svc_skeleton,
           const axutil_env_t *env)
{
    if (svc_skeleton->func_array)
    {
        axutil_array_list_free(svc_skeleton->func_array, env);
        svc_skeleton->func_array = NULL;
    }

    if (svc_skeleton)
    {
        AXIS2_FREE(env->allocator, svc_skeleton);
        svc_skeleton = NULL;
    }

    return AXIS2_SUCCESS;
}

AXIS2_EXPORT int
axis2_get_instance(axis2_svc_skeleton_t **inst,
                  const axutil_env_t *env)
{
    *inst = axis2_hello_create(env);
    if (!(*inst))
    {
        return AXIS2_FAILURE;
    }

    return AXIS2_SUCCESS;
}

AXIS2_EXPORT int
axis2_remove_instance(axis2_svc_skeleton_t *inst,
                     const axutil_env_t *env)
{
    axis2_status_t status = AXIS2_FAILURE;
    if (inst)
    {
        status = AXIS2_SVC_SKELETON_FREE(inst, env);
    }
    return status;
}

```

```

axiom_node_t *
axis2_hello_greet(const axutil_env_t *env, axiom_node_t *node)
{
    axiom_node_t *client_greeting_node = NULL;
    axiom_node_t *return_node = NULL;

    AXIS2_ENV_CHECK(env, NULL);

    if (node)
    {
        client_greeting_node = axiom_node_get_first_child(node, env);
        if (client_greeting_node &&
            axiom_node_get_node_type(client_greeting_node, env) ==
AXIOM_TEXT)
        {
            axiom_text_t *greeting = (axiom_text_t
*)axiom_node_get_data_element(client_greeting_node, env);
            if (greeting && axiom_text_get_value(greeting , env))
            {
                const axis2_char_t *greeting_str =
axiom_text_get_value(greeting, env);
                printf("Client greeted saying \"%s\" \n", greeting_str);
                return_node = build_greeting_response(env, "Hello Client!");
            }
        }
    }
    else
    {
        AXIS2_ERROR_SET(env->error,
AXIS2_ERROR_SVC_SKEL_INVALID_XML_FORMAT_IN_REQUEST, AXIS2_FAILURE);
        printf("ERROR: invalid XML in request\n");
        return_node = build_greeting_response(env, "Client! Who are you?");
    }

    return return_node;
}

```

Die zentrale Methode ist `axis2_hello_greet`, in der die Antwortnachricht zusammengesetzt wird. Als nächstes wird die Datei `hello_svc.c` kompiliert.

```

gcc -shared -olibhello.so -I$AXIS2C_HOME/include/axis2-1.2/ \ -
L$AXIS2C_HOME/lib -laxutil -laxis2_axiom \
-laxis2_parser -laxis2_engine -lpthread -laxis2_http_sender \ -
laxis2_http_receiver hello_svc.c

```

Die dabei erzeugte Bibliothek `libhello.so` wird zusammen mit dem Folgenden Service Descriptor `services.xml` in das Verzeichnis `$AXIS2C_HOME/services/hello` kopiert (das Verzeichnis `hello` muss vorher angelegt werden). Danach ist noch ein Neustart des Apache-Web Servers notwendig.

```

<service name="hello">
  <parameter name="ServiceClass" locked="xsd:false">hello</parameter>
  <description>
    Quick start guide hello service sample.
  </description>

```

```
<operation name="greet"/>
</service>
```

Jetzt steht der Hello-Service zur Verfügung und sollte unter `http://localhost/axis2` angezeigt werden. Ein entsprechender Client `hello.c` sieht wie folgt aus.

```
#include <stdio.h>
#include <axiom.h>
#include <axis2_util.h>
#include <axiom_soap.h>
#include <axis2_client.h>

axiom_node_t *
build_om_request(const axutil_env_t *env);

const axis2_char_t *
process_om_response(const axutil_env_t *env,
                   axiom_node_t *node);

int main(int argc, char** argv)
{
    const axutil_env_t *env = NULL;
    const axis2_char_t *address = NULL;
    axis2_endpoint_ref_t* endpoint_ref = NULL;
    axis2_options_t *options = NULL;
    const axis2_char_t *client_home = NULL;
    axis2_svc_client_t* svc_client = NULL;
    axiom_node_t *payload = NULL;
    axiom_node_t *ret_node = NULL;

    env = axutil_env_create_all("hello_client.log",
                               AXIS2_LOG_LEVEL_TRACE);

    options = axis2_options_create(env);

    address = "http://localhost:9090/axis2/services/hello";
    if (argc > 1)
        address = argv[1];
    if (axutil_strcmp(address, "-h") == 0)
    {
        printf("Usage : %s [endpoint_url]\n", argv[0]);
        printf("use -h for help\n");
        return 0;
    }
    printf("Using endpoint : %s\n", address);
    endpoint_ref = axis2_endpoint_ref_create(env, address);
    axis2_options_set_to(options, env, endpoint_ref);

    client_home = AXIS2_GETENV("AXIS2C_HOME");
    if (!client_home && !strcmp(client_home, ""))
        client_home = "../..";

    svc_client = axis2_svc_client_create(env, client_home);
    if (!svc_client)
```

```

{
    printf("Error creating service client\n");
    AXIS2_LOG_ERROR(env->log, AXIS2_LOG_SI, "Stub invoke FAILED:
Error code:"
                " %d :: %s", env->error->error_number,
                AXIS2_ERROR_GET_MESSAGE(env->error));
    return -1;
}

axis2_svc_client_set_options(svc_client, env, options);

payload = build_om_request(env);

ret_node = axis2_svc_client_send_receive(svc_client, env, payload);

if (ret_node)
{
    const axis2_char_t *greeting = process_om_response(env,
ret_node);
    if (greeting)
        printf("\nReceived greeting: \"%s\" from service\n",
greeting);

    axiom_node_free_tree(ret_node, env);
    ret_node = NULL;
}
else
{
    AXIS2_LOG_ERROR(env->log, AXIS2_LOG_SI, "Stub invoke FAILED:
Error code:"
                " %d :: %s", env->error->error_number,
                AXIS2_ERROR_GET_MESSAGE(env->error));
    printf("hello client invoke FAILED!\n");
}

if (payload)
{
    axiom_node_free_tree(payload, env);
    payload = NULL;
}

if (svc_client)
{
    axis2_svc_client_free(svc_client, env);
    svc_client = NULL;
}

if (env)
{
    axutil_env_free((axutil_env_t *) env);
    env = NULL;
}

return 0;
}

```

```

axiom_node_t *
build_om_request(const axutil_env_t *env)
{
    axiom_node_t* greet_om_node = NULL;
    axiom_element_t * greet_om_ele = NULL;

    greet_om_ele = axiom_element_create(env, NULL, "greet", NULL,
&greet_om_node);
    axiom_element_set_text(greet_om_ele, env, "Hello Server!",
greet_om_node);

    return greet_om_node;
}

const axis2_char_t *
process_om_response(const axutil_env_t *env,
                    axiom_node_t *node)
{
    axiom_node_t *service_greeting_node = NULL;
    axiom_node_t *return_node = NULL;

    if (node)
    {
        service_greeting_node = axiom_node_get_first_child(node, env);
        if (service_greeting_node &&
            axiom_node_get_node_type(service_greeting_node, env) ==
AXIOM_TEXT)
        {
            axiom_text_t *greeting = (axiom_text_t
*)axiom_node_get_data_element(service_greeting_node, env);
            if (greeting && axiom_text_get_value(greeting , env))
            {
                return axiom_text_get_value(greeting, env);
            }
        }
    }
    return NULL;
}

```

Dieser Client wird dann wie folgt kompiliert.

```

gcc -o hello -I$AXIS2C_HOME/include/axis2-1.2/ \
-L$AXIS2C_HOME/lib -laxutil -laxis2_axiom -laxis2_parser \
-laxis2_engine -lpthread -laxis2_http_sender \
-laxis2_http_receiver hello.c -ldl -Wl,--rpath \
-Wl,$AXIS2C_HOME/lib

```

Ein Aufruf des Client sieht dann wie folgt aus.

```

./hello http://localhost/axis2/services/hello
Using endpoint : http://localhost/axis2/services/hello

Received greeting: "Hello Client!" from service

```

4.2.3 Aufgetretene Probleme

Im Folgenden sind die wesentlichen Nachteile von Axis2/C aufgeführt.

- WSDL2C unterstützt kein `encoded`-Stil, daher werden keine Perl-Web Services unterstützt (siehe auch Abschnitt 3.5).
- Ausgangspunkt für die Web-Service-Entwicklung ist eine manuell erstellte WSDL-Datei. Etwas Analoges zu `Java2WSDL` auf Basis einer Java-Schnittstelle, z.B. `C2WSDL`, gibt es nicht.
- Es werden keine WSDL-Dateien publiziert. Beim Aufruf der URL `http://localhost/axis2/services/echo?wsdl` kommt folgende Fehlermeldung.

```
XML Parsing Error: syntax error
Location: http://localhost/axis2/services/echo?wsdl
Line Number 1, Column 1:Unable to retrieve wsdl for this service
^
```

- Es wird kein Hot-Deployment unterstützt, daher ist bei Änderungen an bestehenden bzw. neuen Web Services der Neustart des Apache-Web Servers notwendig.

4.3 Apache Axis2/Java 1.4

Die Open-Source-Software Apache Axis2 (<http://ws.apache.org/axis2/>) beinhaltet Werkzeuge für die automatische Generierung von Client-Stubs aus WSDL. Einschränkungen von Apache Axis2 liegen in den unterstützten WSDL-Styles, da Axis2 keine "encoded Styles" unterstützt, sondern nur RPC/Literal und Document/Literal. Weitere Infos zu den WSDL-Styles sind u. a. zu finden unter <http://www.ibm.com/developerworks/webservices/library/ws-whichwsdl/>.

Die Generierung von Java-Client-Stubs mit Hilfe von Axis2 und Eclipse gestaltet sich aufgrund der verfügbaren Axis2-Plugins für Eclipse als komfortabel. Eine einfache Anleitung dazu ist z.B. zu finden unter <http://wso2.org/library/1719> und unter http://ws.apache.org/axis2/tools/1_3/eclipse/wsd2java-plugin.html. Die von Axis2 gelieferten Fehlermeldungen bei Problemen während der Generierung von Client-Stubs auf Basis von WSDL sind jedoch nicht immer aussagekräftig. Das Kommandozeilentool `WSDL2Java` (siehe http://ws.apache.org/axis2/1_3/reference.html) gibt im Falle von Fehlern die geworfenen Java-Exceptions auf der Kommandozeile aus. Diese müssen entsprechend interpretiert werden, was mit hohem Aufwand verbunden ist. Eine vernünftige Dokumentation der Funktionsweise des Werkzeugs und der möglichen auftretenden Fehler ist nicht vorhanden. Stattdessen muss in Foren oder im Quelltext nach Lösungswegen gesucht werden.

Ein einfacher Client, der eine über Axis2 generierte Stub-Klasse für den `WebService`-Zugriff verwendet, sieht beispielsweise wie folgt aus:

```
import de.meteocontrol.ws.service.kd5x6.SampleServiceStub;

import java.rmi.RemoteException;

import org.apache.axis2.AxisFault;
import org.apache.axis2.transport.http.HTTPConstants;
```

```
import
org.apache.axis2.transport.http.HttpTransportProperties.Authenticator;
import org.apache.log4j.BasicConfigurator;

/**
 * Web service test client
 */
public class TestClient
{
    public static void main( String[] args )
    {
        // Set up a simple configuration for log4j that logs on the console.
        BasicConfigurator.configure();

        try
        {
            SampleServiceStub stub = new SampleServiceStub(
                "https://ws.abc.de/index.php?service=xyz" );

            Authenticator authenticator = new Authenticator();
            authenticator.setUsername( "xxx" );
            authenticator.setPassword( "yyy" );
            stub._getServiceClient().getOptions().setProperty(
                HTTPConstants.AUTHENTICATE, authenticator );

            SampleServiceStub.HelloWorldResponse response = stub.helloWorld(
                new SampleServiceStub.HelloWorld() );
            System.out.println( "Result (helloWorld): " +
response.get_return() );

            SampleServiceStub.Hello hello = new SampleServiceStub.Hello();
            hello.setSName( "Dies ist ein Text!" );
            SampleServiceStub.HelloResponse response2 = stub.hello( hello );
            System.out.println( "Result (hello): " + response2.get_return() );

            SampleServiceStub.GetFaultResponse response3 = stub.getFault(
                new SampleServiceStub.GetFault() );
            System.out.println( "Result (getFault): " + response3.get_return()
);
        }
        catch( AxisFault e )
        {
            e.printStackTrace();
        }
        catch( RemoteException e )
        {
            e.printStackTrace();
        }
    }
}
```

Der Zugriff auf von meteocontrol bereitgestellte Web-Services verlief problemlos über eine verschlüsselte HTTPS-Verbindung mit Benutzernamen/Passwort-Authentifizierung. Allerdings musste die vorliegende WSDL-Datei leicht angepasst werden, um den Java-

Client-Stub generieren zu können, da die WSDL-Datei „literal|encoded“ als Style verwendet. Dies führte bei der Generierung zu Problemen, die in dem verwendeten Beispiel mittels einer Umbenennung zu „literal“ gelöst werden konnten, so dass ein Client-Stub generiert werden konnte. Allerdings besteht weiterhin das Problem für komplexere WSDL-Dateien, das diese in Verbindung mit Axis2/Java nicht im „encoded style“ vorliegen dürfen, da eine Generierung für diese Encodierung nicht unterstützt wird.

4.4 Apache CXF 2.1

Das Open-Source-Framework Apache CXF (<http://cxf.apache.org/>) beinhaltet wie Apache Axis2 Werkzeuge zur Generierung von Client-Stubs auf Basis von WSDL. Im Gegensatz zu Axis2 werden jedoch Fehlermeldungen vom Werkzeug WSDL2Java geliefert, die besser nachvollziehbar sind. Eine Beschreibung der Nutzung von Apache CXF in Verbindung mit Eclipse ist verfügbar unter <http://www-128.ibm.com/developerworks/edu/os-dw-os-eclipse-soatp.html>. Apache CXF unterstützt wie Apache Axis2 keine "encoded Styles".

Apache CXF bietet nicht nur die Generierung von Java-Client-Stubs auf Basis von WSDL, sondern bietet die folgenden weiteren Tools zur Generierung von Stubs in unterschiedlichen Programmiersprachen:

- wsdl2corba (zur Generierung der Interface Description Language (IDL))
- wdl2java (Java)
- wsdl2js (JavaScript)
- wsdl2service erstellt eine neue WSDL-Datei, die eine HTTP oder JMS Service-Definition enthält.
- wsdl2soap erstellt eine neue WSDL-Datei mit einem SOAP-Binding.
- wsdl2xml erstellt eine neue WSDL-Datei mit einem XML-Binding.

Zusätzlich bietet Apache CXF einen WSDL-Validierer (wsdlvalidator) zur Überprüfung von WSDL-Dateien.

4.5 Apache Axis (Java) 1.4

Im Gegensatz zu Apache Axis2 unterstützt Apache Axis (<http://ws.apache.org/axis/>) auch "encoded Styles". Seit April 2006 gab es jedoch keine neuen Versionen zu Axis. Es ist davon auszugehen, dass sich die zukünftigen Entwicklungsarbeiten auf Apache Axis2 konzentrieren werden. Für eine WSDL-Datei im encoded Style konnte mit Hilfe von Apache Axis ein Client-Stub in der Programmiersprache Java generiert werden. Der Zugriff auf den von meteocontrol angebotenen Testservice über einen auf dem Client-Stub basierenden Testclient war jedoch nicht erfolgreich. Da Apache Axis voraussichtlich nicht mehr weiterentwickelt wird, wurde auf weitergehende Tests verzichtet.

4.6 Perl

4.6.1 Installation

Damit Web-Services in Perl implementiert werden können, müssen zunächst folgende Perl-Pakete auf dem System installiert sein:

- XML-Parser (<http://search.cpan.org/~msergeant/XML-Parser-2.36/>)
- XML-Writer (<http://search.cpan.org/~josephw/XML-Writer-0.603/>)

- XML-XPath (<http://search.cpan.org/~msergeant/XML-XPath-1.13/>)
- SOAP-Lite (<http://search.cpan.org/~byrne/SOAP-Lite-0.60a/>)
- POD-WSDL (<http://search.cpan.org/~tareka/Pod-WSDL-0.05/>)
- CGI-Lite (<http://search.cpan.org/~smylers/CGI-Lite-2.02/>)

Eine Möglichkeit, diese Pakete zu installieren, ist, diese herunterzuladen, zu entpacken und folgende Befehle in dem entpackten Verzeichnis auszuführen.

```
perl Makefile.PL
make
make install
```

4.6.2 Beispiel-Service

Es wird davon ausgegangen, dass ein Web-Server wie Apache mit einem CGI-Verzeichnis, z.B. `/usr/lib/cgi`, vorhanden ist. Darin das Verzeichnis `echoService` anlegen und folgende Dateien für einen einfachen Echo-Web Service erstellen.

- `MyService.pm`: Dies ist die Implementierung des Web Services. Es gibt die Methode `echo`, die einfach den übergebenen String zurückgibt.

```
package MyService;

use strict;
use warnings;

sub echo
{
    shift;
    my $parameter=shift;
    return $parameter;
}

1;
```

- `proxy.pl`: Dies ist ein SOAP-Proxy, der Client-Anfragen an die Web Service-Implementierung weiterleitet. In dem Beispiel wird dazu eine `dispatch_to`-Methode verwendet, die nur den Aufruf der `echo`-Methode zulässt, auch wenn `MyService.pm` mehrere Methoden hätte.

```
#!/usr/bin/perl

use strict;
use warnings;
use MyService;

use SOAP::Transport::HTTP;

SOAP::Transport::HTTP::CGI
-> dispatch_to(
    'MyService::echo'
)
```

```
-> handle;  
1;
```

- `client.pl`: Dies ist der Client, der die `echo`-Methode des Web Service über den SOAP-Proxy aufruft. Der Aufruf `perl client.pl` sollte den String `Hello World!` ausgeben.

```
#!/usr/bin/perl  
  
use strict;  
use warnings;  
  
use SOAP::Lite;  
#use SOAP::Lite +trace => 'debug';  
  
my $result = SOAP::Lite  
-> uri('http://127.0.0.1/MyService')  
-> proxy('http://127.0.0.1/cgi-bin/echoService/proxy.pl')  
-> echo('Hello World!')  
-> result;  
  
print $result."\n"
```

Wird die Zeile „`use SOAP::Lite +trace => 'debug';`“ auskommentiert, so werden die übermittelten SOAP-Nachrichten mit ausgegeben.

Als nächstes wird eine WSDL-Datei erzeugt, die zur Generierung von Client-Stubs in anderen Programmiersprachen verwendet werden kann. Die Web Service-Implementierung `MyService.pm` wird dazu um so genannte POD-Annotationen (POD = plain old documentation) erweitert. In diesem Beispiel wird die Methode `echo` entsprechend annotiert.

```
package MyService;  
  
use strict;  
use warnings;  
  
=begin WSDL  
    _IN in @string  
    _RETURN @string  
=cut  
sub echo  
{  
    shift;  
    my $parameter=shift;  
    return $parameter;  
}  
  
1;
```

Die Annotationen sind eigentlich selbsterklärend, weitere Informationen dazu gibt es unter <http://search.cpan.org/~tareka/Pod-WSDL-0.05/lib/Pod/WSDL.pm>. Eine WSDL-Datei kann jetzt über ein weiteres Perl-Skript, hier `genWSDL.pl` erzeugt werden.

```
use Pod::WSDL;

my $pod = new Pod::WSDL(source => 'MyService',
    location => 'http://127.0.0.1/cgi-bin/echoService/proxy.pl',
    pretty => 1,
    withDocumentation => 1);

print $pod->WSDL
```

Der Parameter `source` gibt das einzulesende Perl-Modul an, `location` die URL des SOAP-Proxies, `pretty` auf 1 gesetzt bewirkt eine schön formatierte XML-Ausgabe (sonst alles in einer Zeile) und `withDocumentation` auf 1 gesetzt erzeugt auch Kommentare in der WSDL-Datei. Der Aufruf `perl genWSDL.pl` erzeugt jetzt eine entsprechende WSDL-Datei und gibt sie auf `STDOUT` aus.

Üblicherweise wird über die URL eines Web Service die WSDL-Datei ausgegeben, wenn man den Zusatz „`?wsdl`“ anhängt. Damit das funktioniert, wird der SOAP-Proxy `proxy.pm` wie folgt geändert.

```
#!/usr/bin/perl

use strict;
#use warnings # parse_form_data wirft sonst Warnungen beim Parsen von
SOAP-Nachrichten;

use CGI::Lite;

my $cgi = new CGI::Lite;
my %form = $cgi->parse_form_data;

my $wsdl="false";
while ((my $key, my $value) = each(%form)){
    if ($key eq "wsdl") {
        $wsdl="true";
        last;
    }
}

if ($wsdl eq "true") {
    print "Content-type: text/plain", "\n\n";
    use Pod::WSDL;

    my $pod = new Pod::WSDL(source => 'MyService',
        location => 'http://127.0.0.1/cgi-bin/echoService/proxy.pl',
        pretty => 1,
        withDocumentation => 1);

    print $pod->WSDL;
```

```

} else {
    use MyService;
    use SOAP::Transport::HTTP;
    SOAP::Transport::HTTP::CGI
    -> dispatch_to(
        'MyService::echo'
    )
    -> handle;
}

1;

```

Das Skript macht eigentlich nichts weiter, als zu überprüfen, ob in der URL der Parameter `wSDL` gesetzt wurde. Wenn ja, dann verhält sich der SOAP-Proxy als normales CGI-Skript und erzeugt die WSDL-Datei analog zu `genWSDL.pl`. Ansonsten wird ein SOAP-Aufruf erwartet und entsprechend weitergeleitet. Mit einem Browser sollte die URL `http://127.0.0.1/cgi-bin/echoService/proxy.pl?wSDL` aufrufbar sein und WSDL-Code angezeigt werden. Diese WSDL-Beschreibung kann zur Generierung von Client-Stubs genutzt werden.

4.6.3 Aufgetretene Probleme

Im Folgenden sind die wesentlichen Nachteile von Web Services mit Perl aufgeführt.

- Es gibt im Prinzip kein WSDL (außer POD-WSDL) bei Perl-Web Services, daher fehlen auch die entsprechenden Werkzeuge wie `Java2WSDL` und `WSDL2Java`, um Client-Stubs bzw. Server-Skeletons zu erzeugen.
- Mit dem Paket POD-WSDL werden derzeit nur WSDL-Dokumente mit dem Stil "RPC/encoded" erzeugt, insbesondere weil SOAP-Lite nur den Stil "encoded" unterstützt (weitere Informationen zu Stilen siehe <http://www.ibm.com/developerworks/webservices/library/ws-whichwSDL/>). Leider wird dieser Stil nicht von Axis2/Java und Axis2/C unterstützt, so dass damit keine Clients generiert werden können. Mit dem älteren Axis (quasi Axis1/Java) sowie AxisCPP (nicht überprüft aber theoretisch möglich) ist eine Client-Generierung mit diesem Stil möglich.

4.7 HTTPS-Authentifizierung mit Grid-Zertifikaten

Um die Nutzung von Web Services abzusichern, gibt es verschiedene Möglichkeiten. Beispielsweise könnte die Angabe von einem Benutzernamen und Passwort beim Aufruf der Funktion zwecks Authentifizierung verlangt werden. Des Weiteren könnte die Kommunikation über einen verschlüsselten HTTPS-Kanal abgesichert werden, wobei die dabei eingesetzten X.509-Zertifikate auch zur Authentifizierung herangezogen werden können. Im (D-)Grid-Kontext bzw. bei den eingesetzten Grid-Middleware `gLite`, `Globus Toolkit 4` und `UNICORE` ist es üblich, dass eine wechselseitige Authentifizierung über so genannte Grid-Zertifikate (X.509-Zertifikate) durchgeführt wird, d.h. sowohl Nutzer als auch der Server besitzen ein Zertifikat, welches von einer CA signiert wurde, der beide Parteien vertrauen. `Globus Toolkit 4` und `UNICORE 6` bauen dabei HTTPS-Verbindungen auf. Da es sich bei Grid-Zertifikaten um Standard-X.509-Zertifikate handelt, können diese auch für eine HTTPS-Authentifizierung mit einem Web Server verwendet werden. Im Folgenden wird beschrieben, wie der Web Server `Apache2` konfiguriert werden muss, um eine Authentifizierung mit Grid-Zertifikaten zu unterstützen und wie ein mit `Axis2c`

generierter Web Service-Client ebenfalls ein Grid-Zertifikat verwendet, um sich damit gegenüber dem mit einer HTTPS-Authentifizierung gesicherten Web Service zu authentisieren.

4.7.1 Konfiguration des Apache2 Web Servers

Im Folgenden wird beschrieben, wie der HTTPS/SSL-Web-Server von Apache2 so eingerichtet werden kann, dass eine wechselseitige Authentifizierung auf Basis von (D-)Grid-Zertifikaten möglich ist. Da es sich bei Grid-Zertifikaten auch um X.509-Zertifikate handelt, kann diese Anleitung auch auf Anwendungen außerhalb des Grid-Kontextes übertragen werden.

Es wird davon ausgegangen, dass der Apache2 Web-Server bereits installiert und lauffähig ist, sowie sich dieser auf einem System mit einem gültigen Server-Grid-Zertifikat befindet bzw. das Verzeichnis `/etc/grid-security/` existiert. Die Konfigurationsdateien von Apache2 befinden sich in `/etc/apache2`. Um einen SSL-Server einzurichten, sofern noch nicht vorhanden, wird in `/etc/apache2/vhosts.d/` eine entsprechende Konfigurationsdatei angelegt, z.B. `vhost-ssl`. Der Inhalt dieser Datei kann beispielsweise wie folgt aussehen.

```
<IfDefine SSL>
<IfDefine !NOSSL>

<VirtualHost srvgrid01.offis.uni-oldenburg.de:443>

    DocumentRoot "/var/www/"
    ServerName srvgrid01.offis.uni-oldenburg.de
    ServerAdmin dgrid-admin@offis.de
    ErrorLog /var/log/apache2/ssl_error.log
    TransferLog /var/log/apache2/ssl_access.log

    SSLEngine on
    SSLCipherSuite
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
    SSLCertificateFile /etc/grid-security/hostcert.pem
    SSLCertificateKeyFile /etc/grid-security/hostkey.pem
    SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
downgrade-1.0 force-response-1.0

</VirtualHost>

</IfDefine>
</IfDefine>
```

Die meisten Einstellungen enthalten dabei Standardwerte, lediglich `SSLCertificateFile` und `SSLCertificateKeyFile` sind an dieser Stelle interessant, weil dort ein so genanntes (Grid-)Server-Zertifikat und der dazugehörige private Schlüssel verwendet wurden. Mit dieser Konfiguration kann ein Browser eine HTTPS-Verbindung zu `https://srvgrid01.offis.uni-oldenburg.de` aufbauen. Da per Default die Grid-CA des DFN, die dieses Server-Zertifikat signiert hat, nicht als vertrauenswürdige CA in jedem Browser eingetragen ist, wird es einen entsprechenden Warnhinweis geben. In der jetzigen Konfiguration überprüft nur der Browser die Identität

des Web-Servers bzw. ob er der CA vertraut, die dessen Zertifikat signiert hat, aber nicht umgekehrt.

Im nächsten Schritt geht es darum, den Zugriff auf den HTTPS-Server einzuschränken, so dass nicht jeder Benutzer darauf zugreifen kann. Es wird davon ausgegangen, dass ein Browser vorhanden ist, in dem ein so genanntes (Grid-)User-Zertifikat importiert wurde. In dem Beispiel soll nicht der gesamte Server, sondern lediglich das `cgi-bin`-Verzeichnis (in dem sich ein Perl-Web Service befindet) abgesichert werden. Folgendes muss dazu zu der obigen Konfiguration hinzugefügt werden.

```
<IfDefine SSL>
<IfDefine !NOSSL>

<VirtualHost srvgrid01.offis.uni-oldenburg.de:443>

...

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
<Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
    SSLRequireSSL
    SSLVerifyClient require
    SSLVerifyDepth 10
    SSLCACertificatePath /etc/grid-security/certificates/
    SSLOptions +StdEnvVars
</Directory>

</VirtualHost>

</IfDefine>
</IfDefine>
```

Auch in diesem Fall sind fast nur Standardeinstellungen übernommen worden. Mit `SSLCACertificatePath` wird ein Verzeichnis angegeben, in dem sich Zertifikate von vertrauenswürdigen CAs befinden. In dem Beispiel wurde `/etc/grid-security/certificates/` verwendet, in dem sich die Zertifikate aller vertrauenswürdigen Grid-CAs. z.B. die des DFN, befinden. Die Einstellung `SSLOptions +StdEnvVars` bewirkt, dass vor der Ausführung eines CGI-Skriptes etliche SSL-Umgebungsvariablen hinzugefügt werden. In `SSL_CLIENT_S_DN` beispielsweise steht der DN (*distinguished name*) des Benutzers, was im CGI-Skript zur weiteren Einschränkung von Benutzerrechten verwendet werden kann.

Neben der Einschränkung auf vertrauenswürdige CAs kann bereits in der Apache2-Konfiguration der Zugriff auch auf Benutzerebene anhand des DN des Benutzers weiter eingeschränkt werden. Die obige Konfiguration wird dann wie folgt erweitert.

```
...

ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
```

```

<Directory "/usr/lib/cgi-bin">
  ...
  SSLOptions +StdEnvVars +FakeBasicAuth
  AuthName "Secured Area"
  AuthType Basic
  AuthUserFile /usr/lib/cgi-bin/.passwd
  Require valid-user
</Directory>
...

```

Die zusätzliche SSL-Option `FakeBasicAuth` bewirkt, dass die Authentifizierungsmöglichkeit 'Basic' (normalerweise ein Popup-Fenster mit Feldern für Benutzername und Passwort) im SSL-Kontext verwendet werden kann. Die Datei `/usr/lib/cgi-bin/.passwd` in `AuthUserFile` enthält alle zugelassenen Benutzer (inklusive Passwort) und kann mit dem Programm `htpasswd` erstellt werden. Dort werden als Benutzernamen die DNs der zugelassenen Benutzer und als Passwort immer 'password' (ist eine Konvention) eingetragen.

```

htpasswd -c .passwd "/C=DE/O=GridGermany/OU=OFFIS e.V./OU=Grid-
RA/CN=Guido Scherp"
New password: password
Re-type new password: password
Adding password for user /C=DE/O=GridGermany/OU=OFFIS e.V./OU=Grid-
RA/CN=Guido Scherp

```

Die Option '-c' bewirkt, dass die Datei neu erzeugt wird und kann beim Hinzufügen weiterer Benutzer weggelassen werden. Bei jedem Zugriff auf einen Pfad unterhalb von `https://srvgrid01.offis.uni-oldenburg.de/cgi-bin/`, muss der Nutzer ein Zertifikat vorweisen, welches von einer CA signiert wurde, der der Apache2-Web Server vertraut, z.B. der vom DFN.

4.7.2 Konfiguration von Axis2C

Um eine HTTPS-Authentifizierung auf Basis von (Grid-)Zertifikaten zu ermöglichen, muss die zentrale Konfigurationsdatei `$AXIS2C_HOME/axis2.xml` angepasst werden (siehe auch in der offiziellen Dokumentation unter http://ws.apache.org/axis2/c/docs/axis2c_manual.html#ssl_client). Jeder Web Service-Client auf Basis von Axis2C nutzt (per Default) diese Konfigurationsdatei.

```

<transportSender name="https" class="axis2_http_sender">
  <parameter name="PROTOCOL" locked="false">
    HTTP/1.1
  </parameter>
</transportSender>

<parameter name="SERVER_CERT">
  /etc/grid-security/certificates/c25b265c.0
</parameter>

<parameter name="KEY_FILE">

```

```
$USER_HOME/userkeyandcert.pem  
</parameter>  
  
<parameter name="SSL_PASSPHRASE">geheim</parameter>
```

Der Parameter `SERVER_CERT` ist das SSL-Zertifikat, welches der Zielrechner verwendet, in diesem Fall das Grid-Server-Zertifikat vom Apache2-Web Server. Es kann keine CA oder ein CA-Verzeichnis angegeben werden, sondern lediglich direkt das Zertifikat, dem vertraut wird. Hinter dem Parameter `KEY_FILE` verbirgt sich eine Datei, in der sich sowohl das Grid-Benutzer-Zertifikat befinden, als auch der dazugehörige private Schlüssel (einfach zu erzeugen durch `cat userkey.pem usercert.pem > userkeyandcert.pem`). Dies wird vom Web Service-Client zur Authentisierung gegenüber dem HTTPS-Web Server Apache2 verwendet. Der Parameter `SSL_PASSPHRASE` beinhaltet das Passwort, mit dem der private Schlüssel in `KEY_FILE` verschlüsselt wurde.



Anbindung von meteocontrol-Applikationen an einen Web-Service

Arbeitspaket 3.3

Jochen Rehwinkel
meteocontrol GmbH
Spicherer Straße 48
86157 Augsburg

Internet: <http://www.meteocontrol.de>

1 Einführung

In Arbeitspaket 3.3 wird eine Erweiterung ("Wrapping") der vorhandenen Systeme zur Vereinheitlichung des Zugriffs auf die angebotenen Dienste vorgenommen. Für meteocontrol geht es hierbei hauptsächlich um die Erstellung von gridfähigen Schnittstellen und Web-Services für die Solaranlagen simulation und andere schon bestehende Portaldienste. Es wurden drei Kernbereiche festgelegt, bei denen ein standardisierter Datenaustausch und eine standardisierte Datenspeicherung bei meteocontrol notwendig ist:

- Bereich Ertragssimulationen für Solaranlagen.
- Bereich Safer'Sun: Konfiguration und Messdaten der Safer'Sun Datenbank für Solaranlagenüberwachung
- Bereich Wetterdaten: Wetterdatenlieferung aller Modelle (z.B. ECMWF, MOS) für alle Parameter historisch gemessen und prognostiziert

Die technische Umsetzung der Anforderungen erfolgt mit Web-Services (SOAP) und der Programmiersprache PHP

2 Die Ertragssimulation

2.1 Projektplan zur Simulationsprogrammierung

Die Haupttätigkeit von meteocontrol im AP 5.3. ist die Erstellung und Parallelisierung einer eigenen Ertragssimulation von Solaranlagen. Die Anbindung der Simulationen an einen Web-Service ist Bestandteil von AP 3.3 Die Entwicklung wird in drei Versionen vollzogen:

- Version 1: Erstellung einer Simulation die den jährlichen erwarteten Ertrag einer noch nicht gebauten Solaranlage errechnet. Es wird ein Programm entwickelt das aufgrund einer vorgegebenen Solaranlagenkonfiguration (Wechselrichter, Module, Verschaltung, etc.) und den meteocontrol Wetterdaten (Einstrahlung, Temperatur) des geplanten Standorts die erwartete erzeugte Energie simuliert. Version 1 wird vor allem von Kunden von meteocontrol benötigt, es ist also notwendig eine Live-Schnittstelle zur Simulation zu bieten. Dies geschieht in Form eines Web-Service Servers. Die Programmierarbeiten für Version eins sind nicht genau auf AP 3.3 und AP 5.3 zu trennen. Aus diesem Grund wird in diesem Dokument ein Überblick über die gesamte Version 1 gegeben.
- Version 2: Die Simulation in Version 2 greift direkt auf Solaranlagenkonfigurationen im System von meteocontrol zu. Es handelt sich hierbei um die ca. 5000 Solaranlagen die momentan überwacht werden. Ziel ist hier eine tägliche Simulation des erwarteten Ertrags des Vortages jeder Anlage im System. Im meteocontrol Internetportal Safer'Sun kann dann die Simulation mit den tatsächlich gemessenen Erträgen verglichen werden. Weicht die Simulation stark vom tatsächlichen Ertrag ab, so ist ein Defekt bei der Solaranlage zu vermuten und es wird ein Alarm ausgelöst. Die Tätigkeiten innerhalb Version 1 und 2 überschneiden sich, da teilweise der gleiche Programmcode benutzt werden kann. Die Programmierung der Version 2 findet in AP 5.3 statt.
- Version 3: Die Programmierung von Version 3 zielt auf die Parallelität des Simulationsvorgangs ab und soll auf dem Rechencluster des WISENT-Partners OFFIS realisiert werden. Version 2 wird durch Erweiterung zu Version 3. Es ergeben sich aber auch Synergieeffekt mit Version 1, da auch hier entwickelter

Programmcode weiterverwendet werden kann. Die Tätigkeiten für Version 3 zählen auch zu AP 5.3

3 Der Solar-Web Service Version 1

3.1 Die Funktionen des Web-Service

Bei der Programmierung des Web-Service Server wurde die PHP-Bibliothek NUSOAP verwendet. NUSOAP bietet alle technischen Möglichkeiten, die eine standardisierte Web-Service Kommunikation erfordert. Der Web-Service Server bietet folgende Funktionen, die dann vom Web-Service Client aufgerufen werden:

- function doLogin
 - Übergabeparameter: Benutzerkennung, Passwort
 - Rückgabewert: Session-ID
 - Es wird die Benutzerkennung und das Passwort des Kunden übergeben. Username und Passwort sind in der generellen Web-Service Konfigurationsdatenbank abgespeichert. Es wird eine Session-ID zurückgegeben. Bei jedem Aufruf einer anderen Web-Service Funktion wird die Session-ID mit übergeben. So kann jeder Aufruf einer Funktion dem Kunden zugeordnet werden und dementsprechend verrechnet werden.
- function getAllModulTyp
 - Übergabeparameter: Session-ID
 - Rückgabewert: Alle für die Simulation verfügbaren Solarmodule und deren ID's
 - Der Web-Service Client kann hier alle dem Kunden verfügbaren Solarmodule abfragen. Es wird der Hersteller und der Modelltyp übermittelt. Diese Information kann beim Client z.B. dafür benutzt werden im Programm zur Angebotserstellung eine Auswahlliste von allen konfigurierbaren Solarmodultypen bereits zustellen.
- function getAllWrTyp
 - Übergabeparameter: Session-ID
 - Rückgabewert: Alle für die Simulation verfügbaren Wechselrichtermodelle und deren ID's
 - Der Web-Service Client kann hier alle dem Kunden verfügbaren Wechselrichtermodelle abfragen. Es wird der Hersteller und der Modelltyp übermittelt. Diese Information kann beim Client z.B. dafür benutzt werden im Programm zur Angebotserstellung eine Auswahlliste von den konfigurierbaren Wechselrichtern bereitzustellen.
- function getSimErtrag
 - Übergabeparameter :Session-ID, Postleitzahl, Neigung, Ausrichtung, Wechselrichter-ID, Wechselrichteranzahl, Modul-ID, Modulanzahl
 - Rückgabewert: Erwarteter Jahresertrag in kWh
 - Der Simulation werden alle benötigten Eingangsparameter übergeben. Die Postleitzahl dient zur Ermittlung des Standorts mit Längen- und Breitengrad. Mit Neigung ist die Neigung der Solarmodule in Grad gemeint. Die Ausrichtung bezieht sich auf die Himmelsrichtung in die die Module zeigen, sie wird ebenfalls in Grad angegeben. Die Wechselrichter-ID sowie die Modul-ID dienen zur Identifikation der verwendeten Hardware der Solaranlage. Die ID's wurden in der Regel vorher vom Client über die Funktionen getAllWrTyp und getAllModulTyp beschafft. Zusätzlich wird von Client noch übergeben wie viele Module und Wechselrichter verwendet

werden. Der Rückgabewert ist das eigentliche Resultat der Simulation. Es kann folgende Aussage getroffen werden: „Für die Solaranlage mit der übermittelten Konfiguration wurde ein erwarteter Jahresertrag von XXX,XX kWh simuliert.“

3.2 Die Konfigurationsdatenbank und das Logging

3.3 Web-Service Container Version 1

Die Entwicklung des Web-Service Containers bei meteocontrol erfolgte in zwei Schritten. Version 1 ist eine rudimentäre Web-Service Steuerung, die aber voll funktionsfähig ist. Im Laufe des WISENT-Projekts wurde noch eine zweite weiter entwickelte Version erstellt siehe nächstes Kapitel. Im folgenden wird der Aufbau der Datenbank der Web-Service Containers beschrieben

- T_KUNDE: Hier wird generelle Information zum Aufrufer des Web-Service gespeichert, sowie Passwort und Benutzerkennung.
- T_WEBSERVICE: Hier werden die Web-Service-Server sich eingetragen. Die gesamte Solarsimulation Version 1 bedeutet einen Eintrag in dieser Tabelle.
- T_FUNKTION: Hier wird jede Funktion eines Web-Services hinterlegt
- T_KOSTEN: Hier wird jede Funktion eines Web-Service für jeden Kunden mit Kosten belegt. Es ist also möglich eine genaue Preisdifferenzierung pro Kunde vorzunehmen.
- T_REI_KUNDEN_MODUL: Hier wird hinterlegt welcher Kunde auf welche Solarmodule zugreifen darf. Die Funktion getAllModulTyp liefert alle Einträge eines Kunden.
- T_REL_KUNDEN_WR: Hier wird hinterlegt welcher Kunde auf welche Wechselrichtertypen zugreifen darf. Die Funktion getAllWRTyp liefert alle Einträge eines Kunden.
- T_LOGGING: Hier wird hinterlegt welcher Kunde zu welchem Zeitpunkt eine bestimmte Funktion eines Web-Service aufgerufen hat. Zusätzlich werden die dadurch entstandenen Kosten mitprotokolliert (siehe Abbildung 3).

←T→	ID_LOGGING	TIMESTAMP	ID_KUNDE	ID_FUNKTION	SESSIONID	KOSTEN
<input type="checkbox"/>  	5629	2007-07-05 16:44:06	1	1	1183646646_1test	0
<input type="checkbox"/>  	5630	2007-07-05 16:48:04	1	1	1183646884_1test	0
<input type="checkbox"/>  	5631	2007-07-05 16:48:07	1	8	1183646884_1test	0
<input type="checkbox"/>  	5632	2007-07-05 16:54:10	1	1	1183647250_1test	0
<input type="checkbox"/>  	5633	2007-07-05 16:54:13	1	8	1183647250_1test	0
<input type="checkbox"/>  	5634	2007-07-06 10:15:50	1	1	1183709750_1test	0
<input type="checkbox"/>  	5635	2007-07-06 10:15:57	1	8	1183709750_1test	0
<input type="checkbox"/>  	5636	2007-07-06 10:25:40	1	1	1183710340_1test	0
<input type="checkbox"/>  	5637	2007-07-06 10:25:42	1	8	1183710340_1test	0
<input type="checkbox"/>  	5638	2007-07-06 10:33:27	1	1	1183710806_1test	0
<input type="checkbox"/>  	5639	2007-07-06 10:33:29	1	8	1183710806_1test	0
<input type="checkbox"/>  	5640	2007-07-06 10:59:51	1	1	1183712391_1test	0

Abbildung 3: Auszug aus der Web-Service Tabelle T_LOGGING

```

//-----
$server->register('getAllModulTyp',           // method name
    array('sSessionId' => 'xsd:string'),    // input parameters
    array('sModulTyp' => 'xsd:string'),     // output parameters
    'urn:anlagenertrag',                  // namespace
    'urn:anlagenertrag#getModulTyp',      // soapaction
    'rpc',                                 // style
    'encoded',                             // use
    'Returns available Modules'           // documentation
);

//-----
$server->register('getAllWrTyp',             // method name
    array('sSessionId' => 'xsd:string'),    // input parameters
    array('sWrTyp' => 'xsd:string'),       // output parameters
    'urn:anlagenertrag',                  // namespace
    'urn:anlagenertrag#getWrTyp',        // soapaction
    'rpc',                                 // style
    'encoded',                             // use
    'Returns available Converters'        // documentation
);

//-----
$server->register('getSimulation',           // method name
    array('sSessionId' => 'xsd:string',
        'iPlz' => 'xsd:int',
        'iNeigung' => 'xsd:int',
        'iAusrichtung' => 'xsd:int',
        'iIdModul' => 'xsd:int',
        'iIdWr' => 'xsd:int',
        'iModulAnzahl' => 'xsd:int',
        'iWRAnzahl' => 'xsd:int'),         // input parameters
    array('iSimErtrag' => 'xsd:double'),   // output parameters
    'urn:anlagenertrag',                  // namespace
    'urn:anlagenertrag#getWrTyp',        // soapaction
    'rpc',                                 // style
    'encoded',                             // use
    'Returns simulated energy yield'      // documentation
);

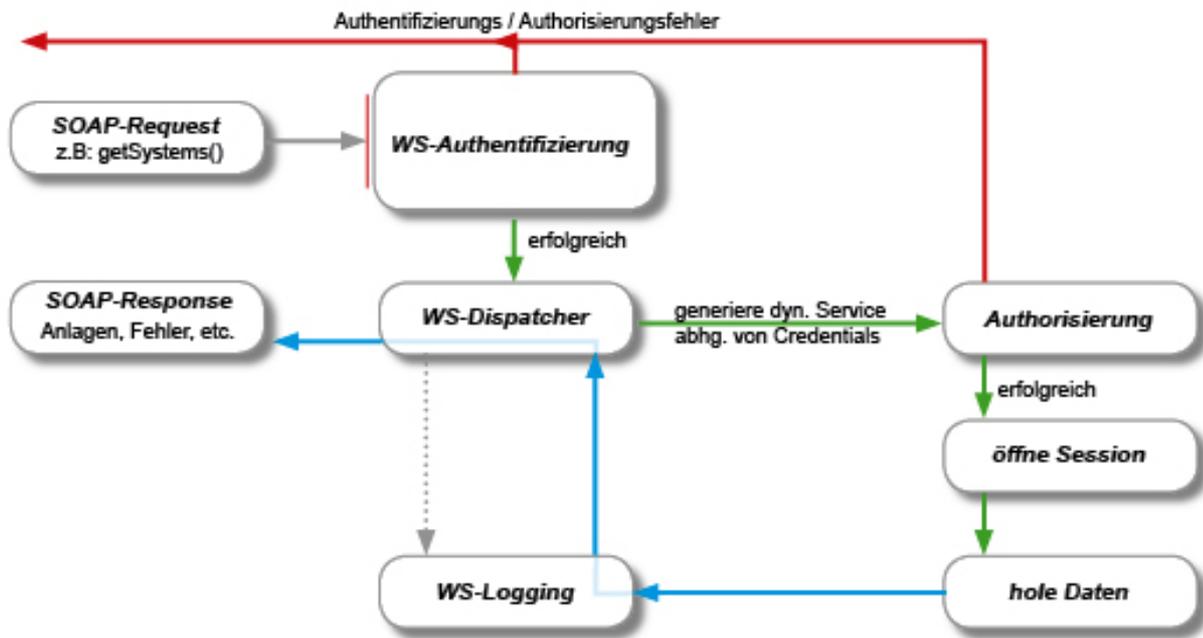
```

Abbildung 4: Programmcode (Auszug) Web-Service Server für Version 1

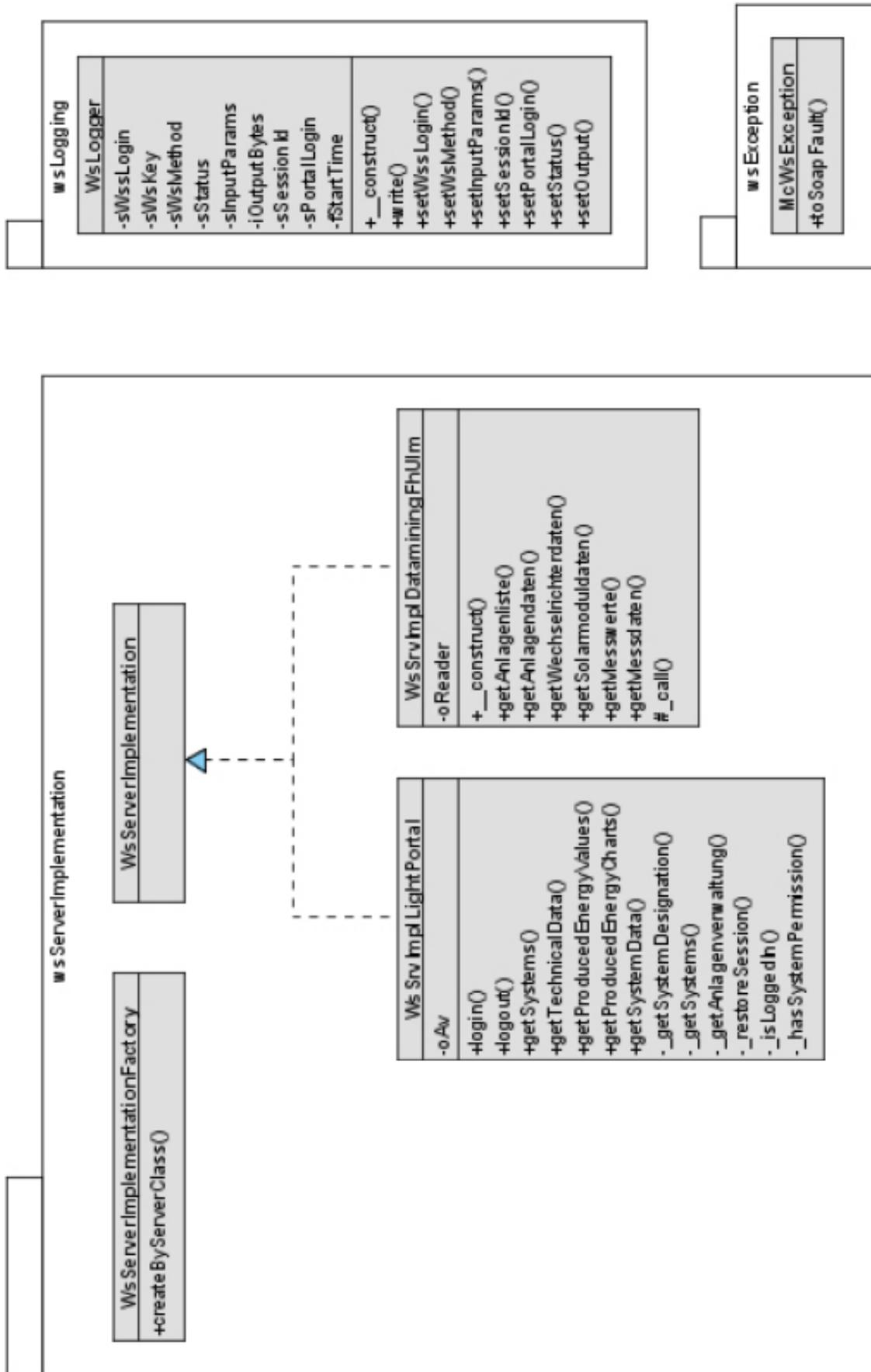
4 Web-Service Container Version 2

Aufgrund der Entwicklungen bei meteocontrol und einer immer stärkeren Nachfrage von Kunden nach Web-Services war es notwendig eine weitere generischere Version des Web-Service Containers zu implementieren. Version 2 zeichnet sich durch einen strukturierten Programmcode aus. Die einzelnen Web-Services lassen sich einfacher auf den einzelnen Kunden anpassen. So ist es nun durch eine Konfiguration in der Datenbank sehr einfach möglich Web-Services für Kunden freizuschalten. Einzelne Web-Service-Funktionen können zugeschaltet werden, je nach Wunsch des Kunden. Version 2 löste Version 1 komplett ab. Alle Web-Services wurde auf Version 2 umgestellt.

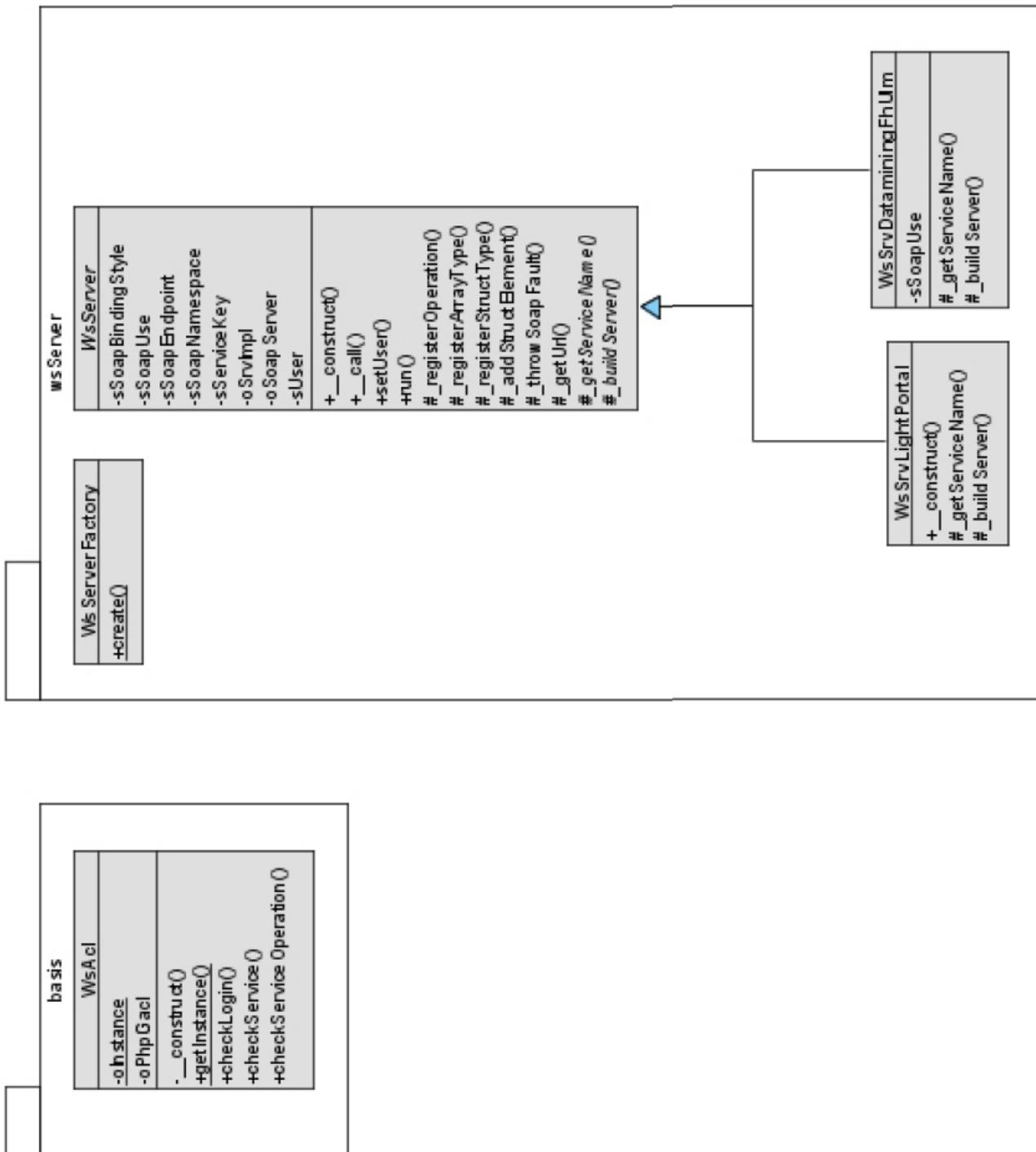
Anbindung von meteocontrol-Applikationen an einen Web-Service (AP 3.3)



Aufbau und Datenfluss Web-Service-Container Version 2



Klassendiagramm Web-Service-Container 2 und Web-Services, Teil 1



Klassendiagramm Web-Service-Container 2 und Web-Services, Teil 1

5 Dataming Web-Service für die FH-Ulm

Zu den Tätigkeiten im A.P 3.3 gehörte die Anbindung der FH-Ulm an die Safer'Sun Datenbank. Die FH-Ulm speist die Konfigurationsdaten und Messdaten aller von meteocontrol überwachten Solaranlagen in eine Data-Warehousing System ein. Ziel ist es Auswertungen über alle Daten zu machen um eine bessere Fehlerleranalyse zu erreichen. Zusätzlich sollen Aussagen getroffen werden über optimale Anlagenkonfigurationen und die Leistungsfähigkeiten einzelner Wechselrichter- und Modulfabrikate. Es wurde ein Web-

Service in PHP realisiert der die wichtigen Konfigurationsparameter und Messwerte aller Solaranlagen je in ein PHP-Objekt serialisiert. Jede Web-Service-Server Funktion liefert ein eigenes serialisiertes Objekt zurück. Auf der Seite der FH-Ulm wird das Objekt wieder in ein PHP-Objekt deserialisiert. Diese Vorgehensweise wurde verwendet um dem großen Datenvolumen einer XML-SOAP Anfrage zu entgehen. Bei 7000 Anlagen und vielen unterschiedlichen Messwerttypen (z.B. Energie, Temperatur, etc.) und einem Messintervall von in der Regel 15 Minuten kommen viele zu übertragende Messpunkte zusammen. Die FH-Ulm kann täglich über den Web-Service Aufruf die aktuellen Werte des Vortages abrufen.

Das obige Klassendiagramm zeigt unter anderem die Web-Service-Server Klasse `WsSrvImplDataminingFHUlm`. Diese Klasse ist in Version 2 des Web-Service-Containers integriert. `WsSrvImplDataminingFHUlm` stellt den Web-Service-Server für die FH-Ulm dar, sie beinhaltet folgende Funktionen, die vom Web-Service-Client aufgerufen werden können:

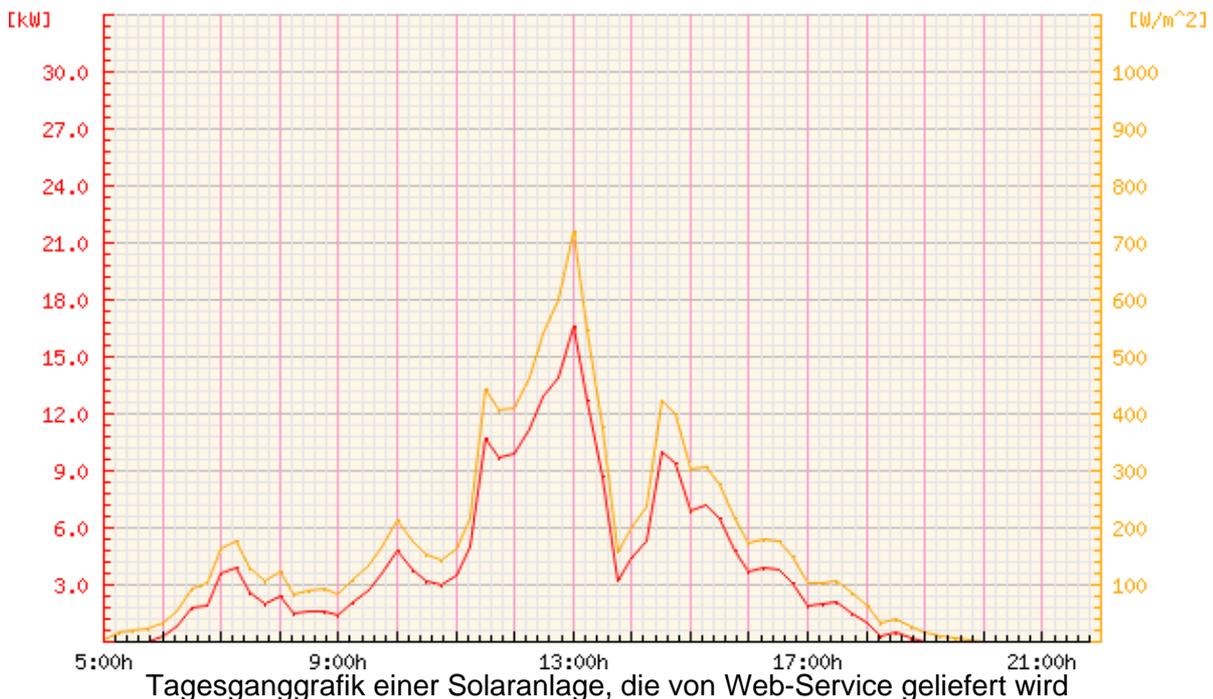
- `getAnlagenliste`: Es werden alle ID's von Anlagen zurückgeliefert die im System von meteocontrol vorhanden sind und einen Status besitzen, der anzeigt das gültige Daten vorliegen.
- `GetAnlagendaten`: Es wird die Anlagenkonfiguration (z.B. Längengrad, Breitengrad, Verschaltung, Neigung, Ausrichtung, etc.) übertragen
- `GetWechselrichterdaten`: Es werden die Konfigurationen und Spezifikationen der Wechselrichter einer Anlage übertragen.
- `GetSolarmoduldaten`: Es werden die Konfigurationen und Spezifikationen der Solarmodule einer Anlage übertragen.
- `GetMesswerte`: Es werden die Messwerttypen, also die Art der Messwerte die eine Anlage liefern kann übertragen.
- `GetMessdaten`: Die einzelnen gemessenen Werte werden übertragen

Der Datamining-Web-Service für die FH-Ulm ist im täglichen Einsatz. Die FH-Ulm arbeitet noch an der Auswertung der Daten. Meteocontrol erhofft sich aus den Ergebnissen Aussagen über die Qualität der gemessenen Daten, Aussagen über die Qualität von bestimmten Produkten und eine weitere Möglichkeit der Fehleranalyse für die Safer'Sun Solaranlagen.

6 Web-Service für Light Portal

Als Light Portal wird im Folgenden ein Internetportal zur Visualisierung von Solaranlagen bezeichnet, das nicht von meteocontrol programmiert wird. Das Light Portal ist die Möglichkeit für Kunden von meteocontrol über Web-Service Anlagendaten von Safer'Sun abzurufen und selbst zu visualisieren. Der Kunde ruft täglich die Messdaten seiner von meteocontrol überwachten Anlagen ab und speist sie in sein eigenes System ein. Mit den Daten kann ein eigenes Produkt erzeugt werden, das wiederum an Endkunden vertrieben werden kann. Für meteocontrol ist es nicht möglich auf jeden einzelnen Wunsch von Kunden für das Internetportal Safer'Sun einzugehen, aus diesem Grund wurde beschlossen den Kunden die Möglichkeit zu geben über Web-Service auf die Daten der eigenen Solaranlagen zuzugreifen. Es werden aber nicht nur Messwerte übertragen, sondern auch komplette schon im Serverpark von meteocontrol erzeugte Grafiken. Da der Web-Service für Kunden der meteocontrol implementiert wurde werden nicht wahllos alle Messwerte übertragen es gibt eine feinere Aufteilung als für den Web-Service der FH-Ulm. Folgende Funktionen wurden implementiert:

- Login: Funktion zur Authentifizierung des Kunden, er kann nur Daten abrufen die seinem Login zugeordnet sind.
- Logout: Beenden einer Web-Service Session.
- GetSystems: Liefert die Anlagenkeys aller Anlagen, die dem Kunden zur Verfügung stehen
- GetTechnicalData: Technische Konfiguration der Solaranlagen, vereinfacht gegenüber dem FH-Ulm-Service
- GetProducedEnergyValues: Erzeugte Energie der Anlage
- GetProducedEnergyCharts: Grafiken der erzeugten Energie
- GetSystemData: Liefert weitere nichttechnische Konfigurationsdaten der Anlage (z.B. Standort, Bezeichnung)



Der Web-Service für das Light Portal wird schon von Kunden von meteocontrol verwendet. U.a. wurde ein Projekt zur Visualisierung von Solaranlagen auf Grundschulen in England realisiert. Es kommen immer weitere Anfragen zum Thema Web-Service. Die Kundenwünsche sind hier sehr unterschiedlich. Es werde in Zukunft noch weitere Web-Services entstehen die Safer'Sun Daten zu Verfügung stellen und auf Version 2 des Web-Service Containers aufbauen. Der Web-Service ist ein fester Bestandteil der Produktpalette von meteocontrol geworden und hätte ohne WISENT nicht in dieser Form realisiert werden können.

Die technische Umsetzung der Anforderungen erfolgt mit Web-Services (SOAP) und der Programmiersprache PHP.

7 Web-Service Wetterdaten

Für die Bereitstellung der Wetterdaten der einzelnen Wetterstationen über Web-Services wurde eine komplett neue Schnittstelle programmiert. Das daraus hervorgehende Produkt hat Marktreife erreicht und wird bereits von Kunden eingesetzt. Es wird dem Client die Möglichkeit gegeben die in der Datenbank von meteocontrol gespeicherten Mess- und Prognosedaten in vor Ort beim Kunden bestehende Informationssysteme einzubinden. Ein

mögliches Anwendungsgebiet ist die Einbindung der aktuellen Wetterdaten, bzw. der Wetterprognosedaten in Gebäudeleitsysteme. Es ist jederzeit Live-Zugriff auf die aktuellsten Werte möglich. Da die Antwortzeit des Web-Service bei der bestehenden Datenbankstruktur nicht tolerierbar war, wurden Veränderungen im Datenbankdesign, im Wetterdatenimport und in den Datenbankzugriffsklassen vorgenommen. Es kann auf ca. 10000 Wetterstationen weltweit zugegriffen werden. Eine einzelne Wetterstation liefert bis zu 20 verschiedene Messparameter (z.B. Lufttemperatur, Windgeschwindigkeit). Folgende Parameter können über den Web-Service gesetzt bzw. Abgerufen werden:

- Zeitraum
- Wetterstation (weltweit)
- Datentyp (historisch, Prognose)
- Messwerttyp (z.B. Lufttemperatur)

```
$oSoapClient = new soapclient($sUrlClient, true);
$oSoapProxy = $oSoapClient->getProxy();

$aAusgabe = $oSoapProxy->getWdat(10852, "2007-08-06 00:00:00", "2007-08-10 23:00:00", "MOS", "SAD");
echo "Ausgabe Mos (SAD): \n";
print_r($aAusgabe);
echo "\n";

exit;

$aAusgabe = $oSoapProxy->getWdat(10852, "2007-07-30 00:00:00", "2007-07-31 23:00:00", "MOS", "FFG");
echo "Ausgabe Mos (FFG): \n";
print_r($aAusgabe);
echo "\n";

$aAusgabe = $oSoapProxy->getWdat(10852, "2007-08-07 00:00:00", "2007-08-12 23:00:00", "MOS", "DRR1");
echo "Ausgabe Mos (DRR1): \n";
print_r($aAusgabe);
echo "\n";

$aAusgabe = $oSoapProxy->getWdat(10852, "2007-07-15 00:00:00", "2007-07-15 23:00:00", "SYNOP", "TT");
echo "Ausgabe Synop (TT): \n";
print_r($aAusgabe);
echo "\n";

$aAusgabe = $oSoapProxy->getWdat(10852, "2007-07-30 00:00:00", "2007-07-31 23:00:00", "MOS", "TT");
echo "Ausgabe Mos (TT): \n";
print_r($aAusgabe);
echo "\n";
```

PHP Programmcode Web-Service-Client für Wetterdaten (Auszug)

Der Wetterdaten Web-Service ist ein neues Produkt von meteocontrol das auch schon von Kunden genutzt wird. Die Zukunft des Web-Service liegt vor allem in der Bereitstellung von Wetterdaten und deren Einbettung in andere Systeme (z.B. Gebäudeleitsysteme).



**Deutsches Zentrum
für Luft- und Raumfahrt e.V.**
in der Helmholtz-Gemeinschaft



Effiziente Interaktion mit großen Datenobjekten mit Hilfe von Quicklooks

Arbeitspaket 4

Gerhard Gesell, Marion Schroedter-Homscheidt
Deutsches Zentrum für Luft- und Raumfahrt DLR, Deutsches
Fernerkundungsdatenzentrum DFD
Münchner Straße 20
82234 Weßling

Internet: <http://www.dlr.de/caf/>

Sönke Brummerloh
OFFIS e.V.
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de>

1 Einführung

1.1 Die Rolle von Quicklooks

In der Fernerkundung im Allgemeinen und beim DFD im Besonderen ist es angesichts großer Datenmengen üblich, so genannte Quicklooks (QL) bereitzustellen, die einen raschen Überblick über vorhandene Daten erlauben. Beispiele dafür sind in der Auflösung reduzierte Bilder, anhand derer sich die Wolkenbedeckung in einem Datensatz schnell erkennen lässt, oder Indexwerte wie die Anzahl verfügbarer Messwerte pro Zeitraum. Der Nutzer kann nach der Untersuchung von Quicklooks eine Entscheidung über die für ihn brauchbaren Daten treffen und sie explizit bestellen.

1.2 Neue Anforderungen und bisherige Lösungsansätze

Bis vor kurzem hatten diese Quicklooks einen geringen Datenumfang und waren somit auch über Internet mit zufrieden stellenden Antwortzeiten erhältlich und auf dem Client-Rechner des Nutzers analysierbar. Durch die spezielle raum-zeitliche Kubus-Datenstruktur bei Meteosat Second Generation (MSG) Satelliten ergeben sich jedoch neue Anforderungen.

Es wurde versucht, statische Quicklooks durch multimediale dynamische Ansichten zu ersetzen (z. B. Filme im MPEG-Format). Diese erreichen aber auch bei geringer Auflösung schnell erhebliche Datenmengen mit bis zu 500 MB. Im Hause DLR-DFD ist dies noch akzeptabel, da diese Informationen mit Gigabit-LAN-Netzwerken übertragen werden. Via Internet (z. B. im Rahmen von vLEM) jedoch verlieren die Filme ihre Funktion eines Quicklooks, da sie nicht mehr schnell und einfach verfügbar sind. Es muss also eine neue Darstellungs- oder Lieferform gefunden werden. Diese Situation wird noch verschärft durch die erhebliche Mehrdimensionalität der Qualitäts- und Metainformation (z. B. Verfügbarkeit der Messung, Angabe verschiedener Fehlerursachen, Wolkenbedeckungsgrad, Verfügbarkeit verschiedener zur Berechnung der Einstrahlung benötigten Zusatzdaten, Genauigkeitsmaß für die Messung und die Zusatzdaten) und den Bedarf sowohl an Einzelinformationen für jede Messung als auch an gemittelten Informationen.

2 Fragestellungen der Nutzer

In diesem Kapitel werden die Anforderungen bzw. Fragestellungen der Nutzer bezüglich Quicklooks, sortiert nach möglichen Produkten, aufgelistet.

2.1 Allgemeine Gedanken zu allen Produkt-QLs

- Sowohl die Standard-Produkte mit z. B. einem Monat Kubus-Länge als auch speziell auf den Nutzer mit Hilfe des Tailor-Postprozessors zugeschnittene Produkte sollen per QL analysiert werden können. Insbesondere beim Tailor erlaubt dies, den Datentransfer nur zu starten, wenn der QL zufriedenstellend ausfällt.
- Alle jemals erstellten QLs sollten geordnet und suchbar archiviert werden – sowohl Standard-QLs als auch on-request erzeugte QLs.
- Ein Tailor-Postprocessing sollte erst angestoßen werden, wenn Nutzer anhand der bereits vorhandenen Standard-QLs oder bereits vorhandener on-request erzeugter QLs festgestellt hat, dass der Datensatz für ihn geeignet ist.
- Bei einer on-request Erzeugung von QLs müssen Schwellen für z. B. Mindestanzahl verfügbarer Tage vom Nutzer setzbar sein.

- Für on-request Prozessierung muss ein Template vom Nutzer ausgefüllt werden. Dieses Template definiert, welche Eigenschaften bei der Erzeugung der QLs wiedergegeben werden sollen. Dieses Template sollte automatisch im Rahmen eines Postprozessors ausgewertet werden können.
- Bei on-request müsste ein Archiv schon bereits erzeugter QLs angelegt und suchbar sein zur Vermeidung von erneuter Prozessierung.
- Es ist ein Abgleichmechanismus zwischen lokalen, beim Nutzer existierenden QL-Archiven und den beim Datenprovider neu erzeugten QLs vorzusehen. Dabei sollte sichergestellt werden, dass nur geänderte QLs zum Nutzer übertragen werden (z. B. bei überlappenden Zeitserien verschiedener Bestellungen).
- Eine Möglichkeit wären user-spezifische QL-Archive im WDC. Ein solches Archiv kann alle QL-Anforderungen eines Nutzers enthalten und ihm permanent zur Verfügung stellen.
- Bei Farbbalken sind z. B. Rot-Grün-Blindheit zu berücksichtigen und Möglichkeit gesättigte Farben zu wählen.
- Informationen zur empirischen und analytischen Genauigkeit sind nur QL-gerecht, wenn diese auf Pixelbasis vorliegen. Im Allgemeinen sind dies aber mittlere Werte, die nicht für jedes einzelne Pixel vorliegen.

2.2 Produkt Wolken

2.2.1 Überblicksinformation

- Überblick über die meteorologische Situation als Film, möglichst ‚echt‘ wirkende Farbdarstellung der Satellitenmessung
- Kurze Filme mit hoher zeitlicher Auflösung, lange Filme mit reduzierter zeitlicher Auflösung
- Eventuell verschiedene regionale Ausschnitte mit hoher raum-zeitlicher Auflösung
- Filme verschiedener geophysikalischer Parameter wie z. B. Wolkenbedeckung, Wolkenstockwerk, optische Dicke, Wolkenklassifikation
- Information, dass Wolken nicht klassifizierbar sind oder in einem bestimmten Stockwerk liegen
- Information, wo wolkenfreie Regionen vorliegen
- Möglichkeit, auch Bilder in voller Auflösung zu betrachten

2.2.2 Plausibilität

- Plausibilität der abgeleiteten Werte (z. B. Zenitwinkel der Sonne, Nachbarschaftsbeziehungen, hohe wolkenoptische Dicke und dünne Wolke passt nicht zusammen, ...). Sobald ein Kriterium nicht erfüllt ist, wird ein Pixel als nicht mehr plausibel definiert. Solch ein QL ermöglicht es zu entscheiden, wie viele plausible Pixel in einer Zeitserie an einem bestimmten Ort vorliegen und ob für den Nutzer die Zeitserie überhaupt interessant ist.

2.2.3 Vollständigkeit

- Wie viele fehlende Werte liegen in einem Produkt an einem Ort vor?
- Wie viele sinnvoll auswertbare Werte liegen in einem Produkt an einem Ort vor? Dabei

sollte die Plausibilitätsschwelle im on-request-Fall auch durch den Nutzer wählbar sein.

- Wie viele wolkenfreie Werte kommen in einem Monat an einem Ort vor? Verteilen sich die wolkenfreien Tage gleichmäßig in der Zeitserie oder gibt es Häufungen?

2.2.4 Genauigkeitsinformationen

- Kann eine pixelweise Genauigkeit angegeben werden?
- Wie viele und wie genaue Pixel liegen in einer Zeitserie an einem bestimmten Ort vor?

2.3 Produkt Schnee

- Wie sieht die Schneeverteilung eines Tages aus?
- Wie aktuell ist die Information im kumulierten Schneeprodukt?
- Wie sieht die Messung des einzelnen Tages aus? Welche Information wurde in das kumulierte Schneeprodukt übernommen?
- Verlauf über die Wintersaison

2.4 Produkt Wasserdampf

- Wie sieht die Wasserdampfverteilung eines Tages aus?
- Verlauf innerhalb eines Monats

2.5 Produkt Lufttemperatur

- Die 2m-Lufttemperatur liegt - analog wie das Wolkenprodukt - vor, sobald Tageslicht verfügbar ist.
- Da nur wolkenfreie Pixel eine Messung zulassen, ist die Verfügbarkeit von Messungen jeweils darzustellen.
- Eine tägliche Verfügbarkeitsinformation ist sinnvoll, monatliche Verfügbarkeitswerte sind dagegen wenig aussagekräftig.
- Überblick als Film, kurze Filme mit hoher zeitlicher Auflösung, lange Filme mit reduzierter zeitlicher Auflösung
- Evt. verschiedene regionale Ausschnitte mit hoher raum-zeitlicher Abdeckung
- Möglichkeit, Bilder auch in voller Auflösung zu betrachten

3 Technische Anforderungen an QLs

Neben den Anforderungen und Fragestellungen der Nutzer gibt es eine Reihe technischer Anforderungen, die berücksichtigt werden müssen.

3.1 Bedingungen des technischen Umfelds

- Alle Eingangsinformation für die QL-Erstellung sollten in der bisher abgespeicherten PRIMAER- oder QL-Komponente enthalten sein, da kein „major reprocessing“ im Rahmen von WISENT möglich ist.
- Desweiteren können auch alternative Darstellungsformen definiert werden, die tieferliegende Eingangsinformationen benötigen. Diese könnten aber erst im nächsten „major reprocessing“ realisiert werden.

- Farbkomposite sollten als JPEG, andere QLs möglichst als GIF (d.h. weniger als 256 Zahlenwerte, d.h. verlustfreie Darstellung anstreben) erzeugt werden. Filme daraus sollten entsprechend als MPG-Video oder als animiertes GIF-Bild erzeugt werden.
- GIFs mit verschiedenen Farbtabelle erlauben mehrere QLs im 8-Bit-Zahlenraum abzulegen und je nach geladener Farbtabelle verschieden darzustellen. Zu klären ist, welche Software-Werkzeuge dieses erlauben.
- Elementar ist die Batchfähigkeit aller verwendeten Softwarewerkzeuge, da alle Prozessschritte aus Skript aufrufbar sein müssen.
- Alle Softwarewerkzeuge sollten unter SUN SOLARIS laufen. SUSE10 als Betriebssystem wäre zur Not auch möglich, wäre aber nur schlecht integrierbar in Prozessketten und würde diese komplizierter und unzuverlässiger machen.
- Zu prüfen ist die Verwendung von de-speckle, dither, oder Tiefpassfiltern für alle Einzelbilder zur Datenreduktion vor der Filmerstellung.
- Gewünscht sind stand-alone Skripte, die später in das Processing integriert werden können.
- In Farbbalken müssen verschiedene Objektklassen separat identifizierbar bleiben.
- Welches Kompressionsverfahren bietet das Optimum zwischen Filmgröße und Datenverlust?
- Frei verfügbare Software-Tools wären von Vorteil, sind aber keine Anforderung.
- Das Schneeprodukt liegt ‚nur‘ in täglicher Auflösung vor. Daher ist die Datenmenge der QLs generell kleiner (Faktor 84). Auf Tages- oder Monatsaggregate kann daher verzichtet werden. Stattdessen wird ein saisonaler Film vorgeschlagen.
- Das Wasserdampfprodukt liegt ‚nur‘ in täglicher Auflösung vor. Daher ist die Datenmenge der QLs generell kleiner (Faktor 84). Auf Tages- oder Monatsaggregate kann daher verzichtet werden. Stattdessen wird ein saisonaler Film vorgeschlagen.

4 Implementierungsvorschläge

In diesem Kapitel werden zunächst Standard-Quicklooks vorgestellt bevor auf einzelne Implementierungsmöglichkeiten eingegangen wird.

4.1 Standard-Quicklooks

Es wird eine Gruppe von Standard-QLs vorgeschlagen.

4.1.1 Produkt Wolken

- Einzelbilder in $\frac{1}{4}$ reduzierter räumlicher Auflösung, alle 15 Minuten, abspeichern in der DISK-QL-H-RES-Komponente
- Film von Farbkompositen, Länge 1 Monat, reduzierte zeitliche Auflösung von 1-2 Stunden, reduzierte räumliche Auflösung, abspeichern in der DISK-QL-L-RES-Komponente
- Film von Farbkompositen, Länge 1 Tag, volle zeitliche Auflösung von 15 Minuten, reduzierte räumliche Auflösung, abspeichern in der DISK-QL-L-RES-Komponente
- Farbkomposite in voller räumlicher Pixelauflösung, nur Einzelbilder um 1200 UTC, abspeichern in der DISK-QL L-RES-Komponente

- Film der Wolkenbedeckung, Länge 1 Monat, Küstenlinien überlagern zur Orientierung, abspeichern in der DISK-QL-L-RES-Komponente
- Film des Wolkenstockwerks, Länge 1 Monat, Küstenlinien überlagern zur Orientierung, abspeichern in der DISK-QL-L-RES-Komponente
- Film der optischen Dicke, Länge 1 Monat, Küstenlinien überlagern zur Orientierung, abspeichern in der DISK-QL-L-RES-Komponente
- Film der Wolkenklassifikation incl. nicht klassifizierbarer Pixel, Länge 1 Monat, Küstenlinien überlagern zur Orientierung, abspeichern in der DISK-QL-L-RES-Komponente
- Farbkomposite mit überlagerter Maske z. B. zu einem Wolkenstockwerk, zu nicht klassifizierbaren Wolken, Einzelbilder, keine Filme, 1200 UTC ausreichend, abspeichern in der DISK-QL-L-RES-Komponente
- Farbkomposite mit wolkenfreien Regionen bzw. Ausmaskierung aller Wolken, Einzelbilder, keine Filme, 1200 UTC ausreichend, abspeichern in der DISK-QL-L-RES-Komponente
- Plausibilitäts-QL: Maskendarstellung erst mal für Einzelbilder, daraus dann einen Film oder ein Aggregat für 1 Monat Daten mit der Anzahl der plausiblen/nicht plausiblen Pixelwerte an jedem Pixel, Küstenlinien überlagern zur Orientierung, abspeichern in der DISK-QA-Komponente
- Darstellung der Anzahl vorhandener Daten bzw. Anzahl der Fehlwerte bzw. %-Angabe der Fehlwerte in Relation zur Zeitserienlänge in einem Monat an einem Pixel, Einzelbild für einen Kubus. Farbbalken sollte unterscheiden, ob 100% vollständig sind, 1 Bild fehlt, ≤ 10 Bilder fehlen, ≤ 50 (d.h. unter 1 Tag Lücke) Bilder fehlen (genaue Abstufung tbd, evt. %-Angaben?), Küstenlinien überlagern zur Orientierung, abspeichern in der DISK-QA-Komponente
- Darstellung der Anzahl wolkenfreier Daten in einem Monat an einem Pixel, Einzelbild für einen Kubus, Küstenlinien überlagern zur Orientierung, abspeichern in der DISK-QL-L-RES-Komponente
- Darstellung der Geschlossenheit der Verteilung wolkenfreier Datenpunkte in einem Monat zur Beschreibung, ob konstante Bewölkung oder ein Wolkenpeak mit einigen Tagen Dauer im Kubus vorliegt, Küstenlinien überlagern zur Orientierung, abspeichern in der DISK-QL-L-RES-Komponente
- Ampeldarstellung verschiedener Klassen (high relative accuracy, medium, low, not available als grün, gelb, rot, grau) erst mal für Einzelbilder, daraus Film oder Aggregat für 1 Monat Daten Anzahl der grünen/gelben/roten/grauen Pixelwerte, Küstenlinien überlagern zur Orientierung, abspeichern in der DISK-QA-Komponente

4.1.2 Produkt Schnee

- Einzelne QLs für jeden Tag der kumulativen Schneebedeckung.
- Einzelne QLs für jeden Tag der jeweils vom Satelliten aktuell gemessenen Schneeverteilung (incl. Lücken durch Wolken), Küstenlinien überlagern zur Orientierung
- Einzelne QLs für jeden Tag, die die Zeit seit dem letzten erfolgreichen Aktualisieren des kumulativen Schneeprodukts an einem Ort beschreibt, Küstenlinien überlagern zur Orientierung

- Filme des Verlaufs des kumulativen Produkts über eine Wintersaison z. B. von Oktober bis April.

4.1.3 Produkt Wasserdampf

- Einzelne QLs der Wasserdampfverteilung für einen Tag, Küstenlinien überlagern zur Orientierung
- Film der Wasserdampfkonzentration über einen Monat, Küstenlinien überlagern zur Orientierung:
 - Diese QLs sollen für den Standardzeitraum der in DIMS abgelegten Produkte jeweils beim Erzeugen des Produkts erzeugt und gespeichert werden.
 - Je nach Größe der QL-Produkte werden diese im DIMS und/oder auf den Webseiten des WDC-RSAT angeboten.

4.2 Quicklooks als Videos

Wenn Quicklooks in Form von Bilderserien erstellt werden, ist die zugrunde liegende technische Frage: Wie können diese Bilder möglichst effizient in ihrem Speicherbedarf reduziert werden, ohne die Darstellungsqualität deutlich zu verschlechtern? Hier liegt im Prinzip das gleiche Problem wie beim Komprimieren von Filmen bzw. Video-Dateien vor, denn auch Videos bestehen aus vielen einzelnen Bildern.

Der Vorgang, Daten (z. B. Bilder, Videos und Audiosequenzen) von einem Format in ein anderes zu überführen (konvertieren), wird auch als Codieren bezeichnet. Der Algorithmus bzw. die Bibliothek um eine Codierung und Decodierung durchzuführen, wird neuerdings häufig auch als Codec bezeichnet. Bei dem Wort Codec handelt es sich um ein Kunstwort aus den englischen Wörtern **c**oder und **d**ecoder¹. Zum Codieren von Videos steht eine Vielzahl leistungsfähiger Codecs zur Verfügung. Daher ist die Idee entstanden, diese zum Erstellen von Quicklooks in Form von Videos zu verwenden. Bei der Auswahl der Codecs wurde darauf geachtet, dass diese auf allen verbreiteten Betriebssystemen wie Microsoft Windows, Linux und MacOS X verfügbar sind. Proprietäre Codecs (z. B. *Windows Media Video* und *Quick Time*), die nur für einige Betriebssysteme verfügbar sind, wurden daher nicht getestet.

Für die Evaluation wurden vier Codecs gewählt. Zum einen die drei MPEG-4 kompatiblen Codecs FFmpeg MPEG-4, Xvid² und x264³. Zum anderen wurde der Sorenson H.263 kompatible Codec der FFmpeg-Bibliothek⁴ genutzt, um Flash-Videos zu erzeugen. Die Wahl fiel auf diese Codecs, da die mit ihnen erzeugten Videos auf den meisten modernen Computern ohne die Installation eines neuen Codecs abgespielt werden können. Open-Source-Codecs wie Dirac⁵ oder Theora⁶ wurden nicht getestet, weil zum Abspielen zugehöriger Videos die Codecs in der Regel installiert werden müssten.

Der MPEG-4 Standard (ISO/IEC 14496) gibt eine Spezifikation vor, die von kompatiblen Codecs für Audio- und Video-Daten implementiert wird. Der MPEG-4 Standard gliedert sich in eine Reihe von Unterpaketen (engl. Parts). FFmpeg MPEG-4 und Xvid entsprechen dem älteren MPEG-4 Part 2 Standard, während x264 dem neueren MPEG-4

¹ c't 6/2005, S. 248

² <http://www.xvid.org>

³ <http://www.videolan.org/developers/x264.html>

⁴ <http://ffmpeg.mplayerhq.hu/>

⁵ <http://dirac.sourceforge.net>

⁶ <http://www.theora.org>

Part 10 Standard entspricht. Der Codec für die Flash-Videos basiert auf dem H.263 Standard und ist damit der einzige getestete Codec, der nicht einen MPEG-4 Standard umsetzt. Da der Flash-Codec auf einem älteren Standard als die anderen Codecs basiert, ist zu erwarten, dass er schlechtere Ergebnisse liefern wird. Er wurde dennoch getestet, weil jeder aktuelle Internet Browser über ein Flash-Plugin verfügt. Das Plugin ermöglicht es, Flash-Videos direkt aus dem Internet zu laden und im Internet Browser abzuspielen⁷. Dadurch ist gewährleistet, dass ein Nutzer keinen neuen Codec installieren muss. Die Möglichkeit, Videos über das Flash-Plugin abzuspielen, nutzen unter anderem Web-basierte Video-Portale wie z. B. YouTube⁸.

Eine Video-Sequenz allein kann nicht von einer Software zum Abspielen von Videos angezeigt werden. Damit die Video-Sequenzen abgespielt werden können, werden sie zusammen mit zugehörigen Audio-Sequenzen und Metainformationen in sogenannten *Containerformaten* gekapselt. Das offizielle Containerformat von MPEG-4 ist das MP4-Format. Es werden zusammen mit MPEG-4 auch andere Containerformate genutzt. Weitverbreitet bei MPEG-4-Videos ist das von Microsoft eingeführte *Audio Video Interleave-Format* (AVI). Auf Grund seiner weiten Verbreitung wurde es in den Tests dem neueren MP4 vorgezogen. Bei Flash-Videos wird das von Adobe Systems entwickelte Containerformat *Flash Video* (FLV) verwendet.

Um zu überprüfen, wie gut die vier Codecs Bilder codieren können, wurde eine beim DLR-DFD typische Bilderserie mit allen vier Codecs mit unterschiedlichen Parametern codiert. Von besonderem Interesse waren die Codierungsdauer sowie die Bildqualität im Verhältnis zur Dateigröße. Die Codierungen wurden auf Grund ihres hohen Zeit- und Rechenaufwands auf dem WISENT-Cluster im OFFIS durchgeführt. Als Codierungswerkzeug wurde das Kommandozeilen-basierte Werkzeug *mencoder* eingesetzt, das Teil der MPlayer⁹-Installation ist. Die MPlayer-Software steht unter der GNU General Public License¹⁰ (GPL) und ist auf vielen Betriebssystemen lauffähig. Auf Grund seiner vielen Konfigurationsmöglichkeiten und der relativ einfachen Handhabung wurde *mencoder* den beiden anderen unter Linux verbreiteten Codierungswerkzeugen FFmpeg¹¹ und Transcode¹² vorgezogen.

4.2.1 Technischer Hintergrund

Die einzelnen Bilder eines Videos werden als Frames bezeichnet. Moderne Codecs speichern allerdings nicht jedes einzelne Bild vollständig ab. Sie versuchen vielmehr, nur die Bildpunkte zu speichern, die sich zwischen zwei Frames geändert haben. Dabei werden in gewissem Maße auch Verschiebungen von Bildelementen erkannt und diese als Verschiebungsvektor gespeichert. Auf diese Weise kann der Speicherbedarf deutlich reduziert werden.

Die Größe einer Video-Datei resultiert aus der Bitrate des Videos. Die Bitrate wird in Bit pro Sekunde (Bit/s) gemessen und gibt an, wie viele Bit pro Sekunde gespeichert werden, um das Video darstellen zu können. Je größer die Bitrate, desto größer ist auch die Video-Datei. Ziel der Codecs ist es, durch geschickte Algorithmen die Bitrate möglichst optimal auszunutzen.

Bei Video-Codierungen kann zwischen einer konstanten und einer variablen Bitrate gewählt werden. Bei einer konstanten Bitrate wird jedes Frame mit der gleichen Bitanzahl

⁷ c't 21/2006, S. 216ff

⁸ <http://www.youtube.com>

⁹ <http://www.mplayerhq.hu>

¹⁰ <http://www.gnu.org/licenses/gpl.html>

¹¹ <http://ffmpeg.mplayerhq.hu>

¹² <http://www.transcoding.org/cgi-bin/transcode>

codiert. Das hat den Vorteil, dass die maximale Größe der Video-Datei sicher vorhersagt werden kann:

$$\text{Bitrate} \times \text{Spieldauer in Sekunden} = \text{Größe der Videodatei}$$

Eine konstante Bitrate hat den Nachteil, dass für sich stark ändernde Frame-Folgen nicht genug Bit zur Verfügung stehen. Hierbei kommt es zu Qualitätsverlusten. Dieses Problem kann durch eine variable Bitrate vermindert werden, ohne die Videodateigröße zu erhöhen. Bei einer Codierung mit variabler Bitrate versucht der Codec die durchschnittliche Bitrate auf einem angegebenen Wert zu halten. Sich stark ändernde Frame-Folgen werden dazu mit einer höheren Bitrate codiert als solche mit geringen Änderungen. Variable Bitraten führen daher normalerweise zu einer durchgängig besseren Bildqualität. Allerdings ist zu Beginn der Codierung noch nicht bekannt, welche Frames eine höhere Bitrate benötigen und bei welchen Frames eine eher geringere Bitrate ausreicht. Daher kann es passieren, dass die Bitrate nicht optimal angepasst ist und die Video-Datei deutlich mehr oder weniger Speicherplatz benötigt, als zunächst angenommen. Aus diesem Grund wird für Codierungen mit variabler Bitrate gewöhnlich das sogenannte „twopass-Verfahren“ eingesetzt. Beim twopass-Verfahren verarbeitet der Codec das zu codierende Material zweimal. Dabei wird im ersten Durchgang das Material zunächst analysiert und die Ergebnisse pro Frame in einer Datei gespeichert. Beim zweiten Durchlauf werden die Analyseergebnisse dann für eine bzgl. der Bildveränderungen zwischen den Frames optimierte Bitrate genutzt. Auf diese Weise wird eine gleichbleibende Video-Qualität möglich und die endgültige Größe des Videos kann sicher vorhergesagt werden. Allerdings braucht die twopass-Codierung in der Regel fast doppelt so lange wie eine Codierung, die das zu codierende Material nur einmal durchläuft.

Die Framerate gibt an, wie viele Frames pro Sekunde (fps) beim Abspielen eines Videos angezeigt werden. Jedem Frame stehen somit eine vorgegebene Menge Speicherbit zur Verfügung:

$$\text{Speicherbit pro Frame} = \frac{\text{Bitrate}}{\text{Framerate}}$$

Wenn lediglich einzelne Bilder in ein Video umgewandelt werden und jedes Bild beispielsweise für eine Sekunde gezeigt werden soll, ist es ineffizient, ein Video mit den in Europa üblichen 25 fps zu erstellen. Jedes Bild würde dann in 25 gleiche Frames umgewandelt. Dies würde dazu führen, dass die Bitrate auf 25 Frames aufgeteilt wird und die effektive Bitrate pro Sekunde niedriger wäre, als wenn pro Sekunde nur ein Frame vorliegt. Um eine möglichst direkte Übersetzung von Bitrate in Bildqualität zu gewährleisten, wurde daher bei den durchgeführten Tests statt 25 fps eine Framerate von 1 fps gewählt.

Die „Peak Signal to Noise Ratio“ (PSNR) ist ein Maß, das angibt, wie stark ein codiertes Frame vom Originalframe bzw. Bild abweicht. Diese mathematische Abweichung wird während der Codierung ermittelt und als globale PSNR am Ende ausgegeben. Die PSNR wird in Dezibel (dB) gemessen und bietet einen guten Anhaltspunkt, wie gut die Qualität des neuen Videos ist. Es gilt hier: Je höher die PSNR ist, desto geringer fällt die mathematische Abweichung zwischen Original und neuem Video aus. Ein Mensch kann die Qualität allerdings anders beurteilen, als die PSNR wiedergibt. So fließen z. B. Blockartefakte sehr negativ in die PSNR ein, obwohl sie für einen Menschen nicht in jedem Fall als störend empfunden werden.¹³

¹³ c't 10/2003, S. 146ff

4.2.2 Testaufbau

Für die Tests wurde die dynamische Programmiersprache Groovy¹⁴ genutzt. Sie wurde auf Grund ihrer Prägnanz und ihrer guten Integration mit der Programmiersprache Java gewählt. Ein Groovy-Skript diente dazu, die Job-Skripte in der Skriptsprache Bash für das am OFFIS betriebene Cluster zu erzeugen. Die Job-Skripte bestanden nur aus Anweisungen für den Scheduler und dem Aufruf des eigentlichen Video-Tests. Das Abschicken der Jobs an das Cluster, die Video-Tests selber und die grafische Aufbereitung der Messwerte erfolgte über Groovy-Skripte.

Zum Erstellen der Videos wurde eine von einem Satelliten aufgenommene Bilderserie im JPEG-Format verwendet (Abb. 1). Von dieser Bilderserie wurden in jedem Codierungsdurchlauf 2173 Bilder mit einem Gesamtspeicherbedarf von 300 MB verarbeitet. Jedes dieser Bilder hatte eine Auflösung von 928 x 928 Pixel.

Der Zeitbedarf einer Codierung kann selbst bei der Verwendung von gleichen Parametern leicht variieren. Dies ist im Wesentlichen zurückzuführen auf unterschiedliche Beanspruchung von Netzwerk, Festplatte und CPU z. B. durch zur gleichen Zeit aktiver Software bzw. Prozessen. Um Schwankungen in der Ausführungszeit möglichst gering zu halten, wurden die Tests diesbezüglich optimiert:

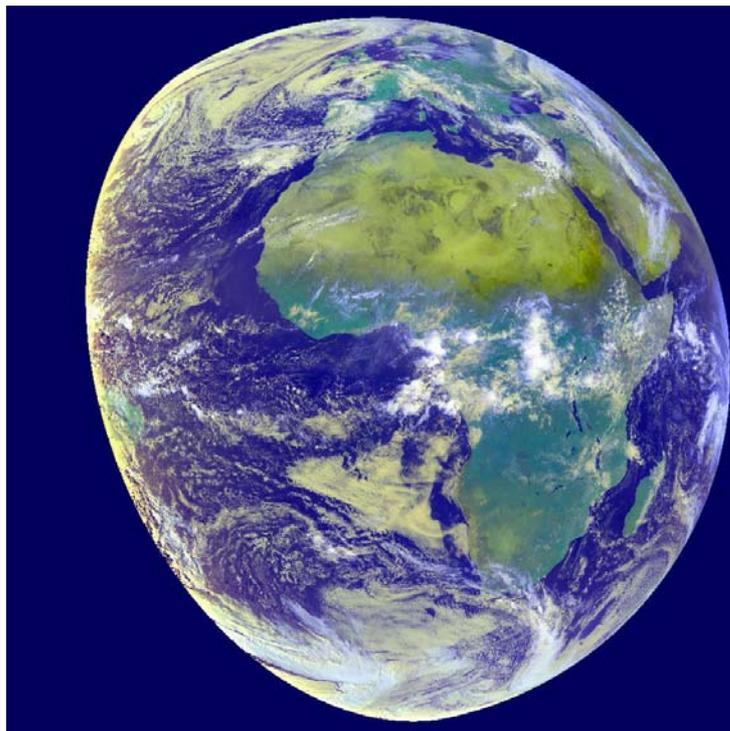


Abbildung 1: Ein Bild der codierten Bilderserie

Zunächst wurden die JPEG-Bilder über ein zentrales über NFS eingebundenes Netzlaufwerk bereitgestellt und von dort während der parallel durchgeführten Codierungen geladen. Dieser gemeinsame Zugriff führte zu deutlichen Abweichungen in der Laufzeit. Aus diesem Grund wurden zu Beginn jedes Jobs, vor den eigentlichen Codierungen, die Bilddateien in ein lokales Verzeichnis auf einem Rechenknoten kopiert. Jede Codierungskonfiguration wurde 50-mal ausgeführt, um einen verlässlichen durchschnittlichen Mittelwert trotz kleinerer Schwankungen bilden zu können.

¹⁴ <http://groovy.codehaus.org>

Nachdem eine Video-Datei erstellt war, wurde das erste Frame zum manuellen Überprüfen der Video-Qualität als unkomprimierte JPEG-Datei gespeichert. Danach wurde die Video-Datei gelöscht, denn ein Archivieren aller Video-Dateien hätte über 700 GB an Speicherplatz benötigt. Das Ergebnis jedes Jobs sind eine Bilddatei und Logging-Informationen. Die Logging-Informationen bestehen aus dem Testnamen, der Codierungsdauer in Millisekunden, der tatsächlichen Bitrate, der Größe der erzeugten Video-Datei in Byte, der PSNR dieses Videos und den verwendeten Optionen.

Beispiel eines Log-Eintrags:

```
<Benchmark>
  <testName>xvid_CBR_200</testName>
  <duration>189695</duration>
  <bitrate>206.729</bitrate>
  <filesize>56152889</filesize>
  <psnr>29.91</psnr>
  <options />
  <codecOptions>-xvidencopts bitrate=200:psnr</codecOptions>
</Benchmark>
```

Jeder Video-Codec wurde mit neun verschiedenen Konfigurationsklassen getestet, die jeweils bzgl. der Bitrate bzw. der gewünschten Qualität variiert wurden. Insgesamt ergaben sich über 7000 unterschiedliche Konfigurationen, die jeweils als Job auf dem Cluster ausgeführt wurden. Die Konfigurationsklassen waren im Einzelnen:

CBR, scaled 50:	Eine Codierung mit konstanter Bitrate ohne optimierende Einstellungen. Die Auflösung wurde auf 50% der Originalgröße (von 928 x 928 Pixel auf 464 x 464 Pixel) reduziert.
CBR, scaled 75:	Eine Codierung mit konstanter Bitrate ohne optimierende Einstellungen. Die Auflösung wurde auf 75% der Originalgröße (von 928 x 928 Pixel auf 696 x 696 Pixel) reduziert.
CBR:	Eine Codierung mit konstanter Bitrate ohne optimierende Einstellungen.
CBR, hq:	Eine Codierung mit konstanter Bitrate und leicht optimierenden Einstellungen bzgl. der Bildqualität.
CBR, vhq:	Eine Codierung mit konstanter Bitrate und sehr stark optimierenden Einstellungen bzgl. der Bildqualität.
Fixed Quality:	Das Video wird auf Basis eines Werts, der die gewünschte Qualität der Videos beschreibt, codiert. Bei x264 sind Werte zwischen 1 (beste Qualität) und 50 (schlechteste Qualität) möglich. Bei den anderen Codecs sind Werte zwischen 1 (beste Qualität) und 30 (schlechteste Qualität) möglich.
Twopass:	Die Bilder werden für die Codierung zweimal durchlaufen, um eine möglichst gute Annäherung an die gewünschte variable Bitrate zu erreichen.
Twopass, hq:	Die Bilder werden für die Codierung zweimal durchlaufen, um eine möglichst gute Annäherung an die gewünschte variable Bitrate zu erreichen. Dabei werden leicht optimierende Einstellungen bzgl. der Bildqualität genutzt.
Twopass, vhq:	Die Bilder werden für die Codierung zweimal durchlaufen, um eine möglichst gute Annäherung an die gewünschte variable Bitrate zu erreichen. Dabei werden sehr stark optimierende Einstellungen bzgl.

der Bildqualität genutzt.

Für die Bitraten-basierten Codierungen wurde die Bitrate zwischen 10 Bit/s und 2500 Bit/s variiert. Bei den Qualitäts-basierten Codierungen (Fixed Quality) wurden alle Qualitätswerte verwendet.

Die Parameter für die hq und vhq Konfigurationsklassen stammen aus den Empfehlungen des MPlayer-Handbuchs¹⁵.

4.2.3 Auswertung

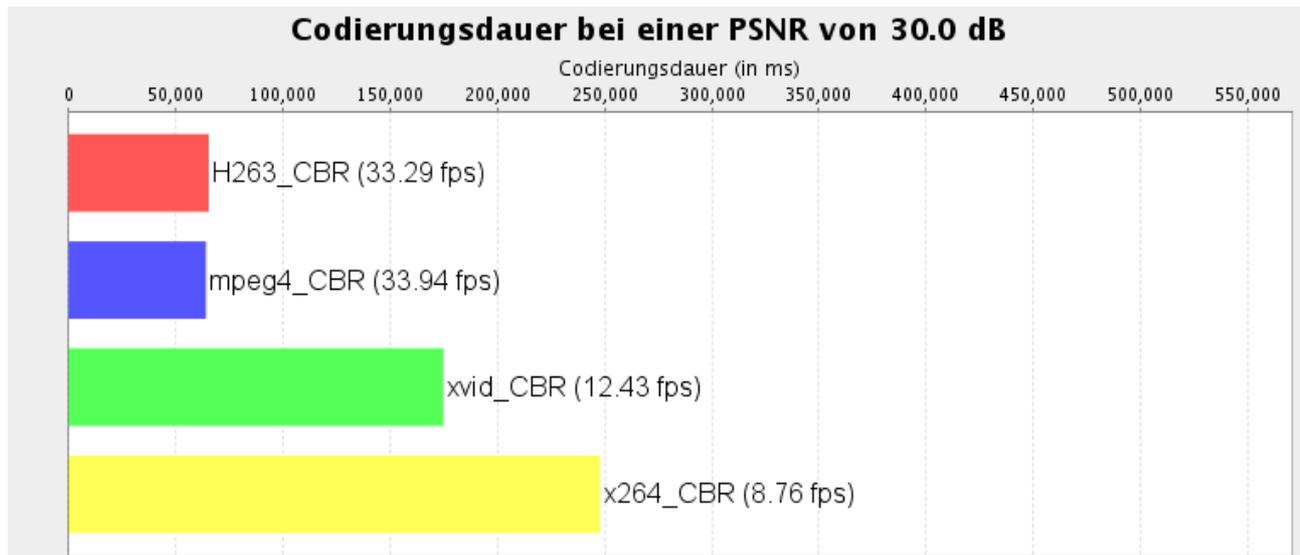


Abbildung 2: Codierungsdauer bei einer PSNR von 30 dB bei der CBR-Codierung ohne optimierende Einstellungen. Die Werte in Klammern geben an, wie viele Bilder pro Sekunde durchschnittlich verarbeitet wurden.

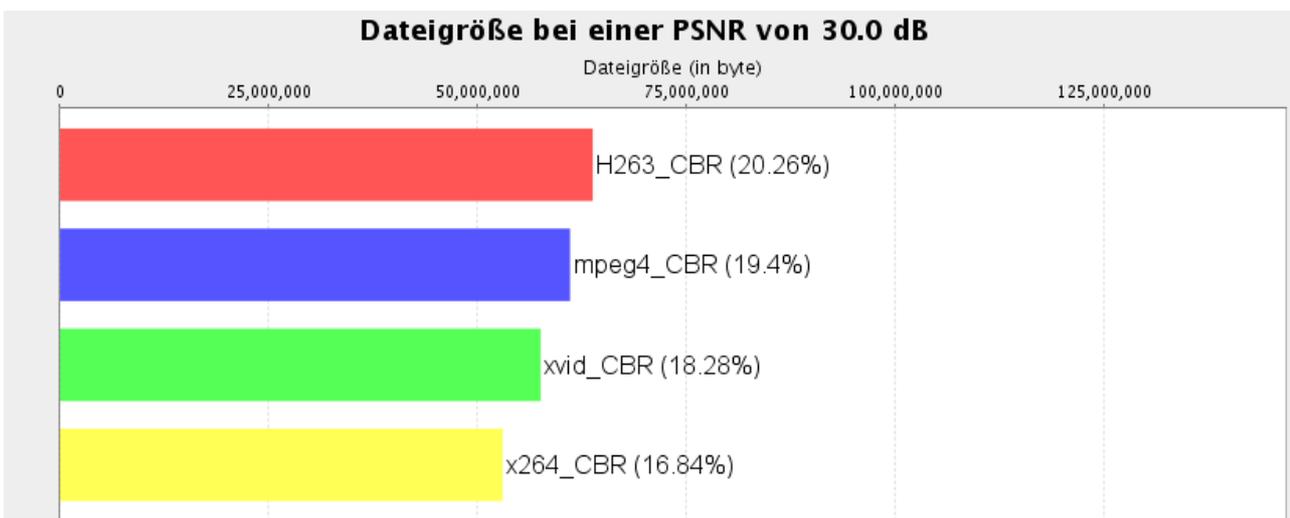


Abbildung 3: Dateigröße bei einer PSNR von 30 dB bei der CBR-Codierung ohne optimierende Einstellungen. Die Werte in den Klammern geben an, wie viel Prozent das Video von dem Speicherbedarf der Originalbilderserie einnahm.

¹⁵ <http://www.mplayerhq.hu/DOCS/HTML/en/encoding-guide.html>

Die Abbildungen 2 und 3 stellen exemplarisch das Verhältnis der Codecs bei einer PSNR von 30 dB bzgl. Dateigröße und Codierungszeit dar. Die Grafiken bei anderen PSNR-Werten zeigen ein ähnliches Verhältnis. Die Abb. 2 verdeutlicht, dass der H.263-Codec und der FFmpeg MPEG-4-Codec am performantesten sind. Beide benötigen nur etwa ein Viertel der Laufzeit, die der x264-Codec benötigt. Xvid-Codec benötigt etwa 70% der Laufzeit vom x264-Codec. Abb. 3 zeigt, dass sich der Platzbedarf bei einer PSNR von 30 dB umgekehrt verhält: Der x264-Codec erzeugt die kleinsten Video-Dateien. Gegenüber dem x264-Codec erzeugt Xvid um ca. 10%, FFmpeg MPEG-4 um ca. 15% und der H.263-Codec um ca. 20% größere Video-Dateien.

Bezüglich anderer PSNR-Werte schwanken die Unterschiede, die Reihenfolge bleibt aber bestehen. Ab einer PSNR von ca. 34 dB liefert Xvid kleinere Dateien als x264.

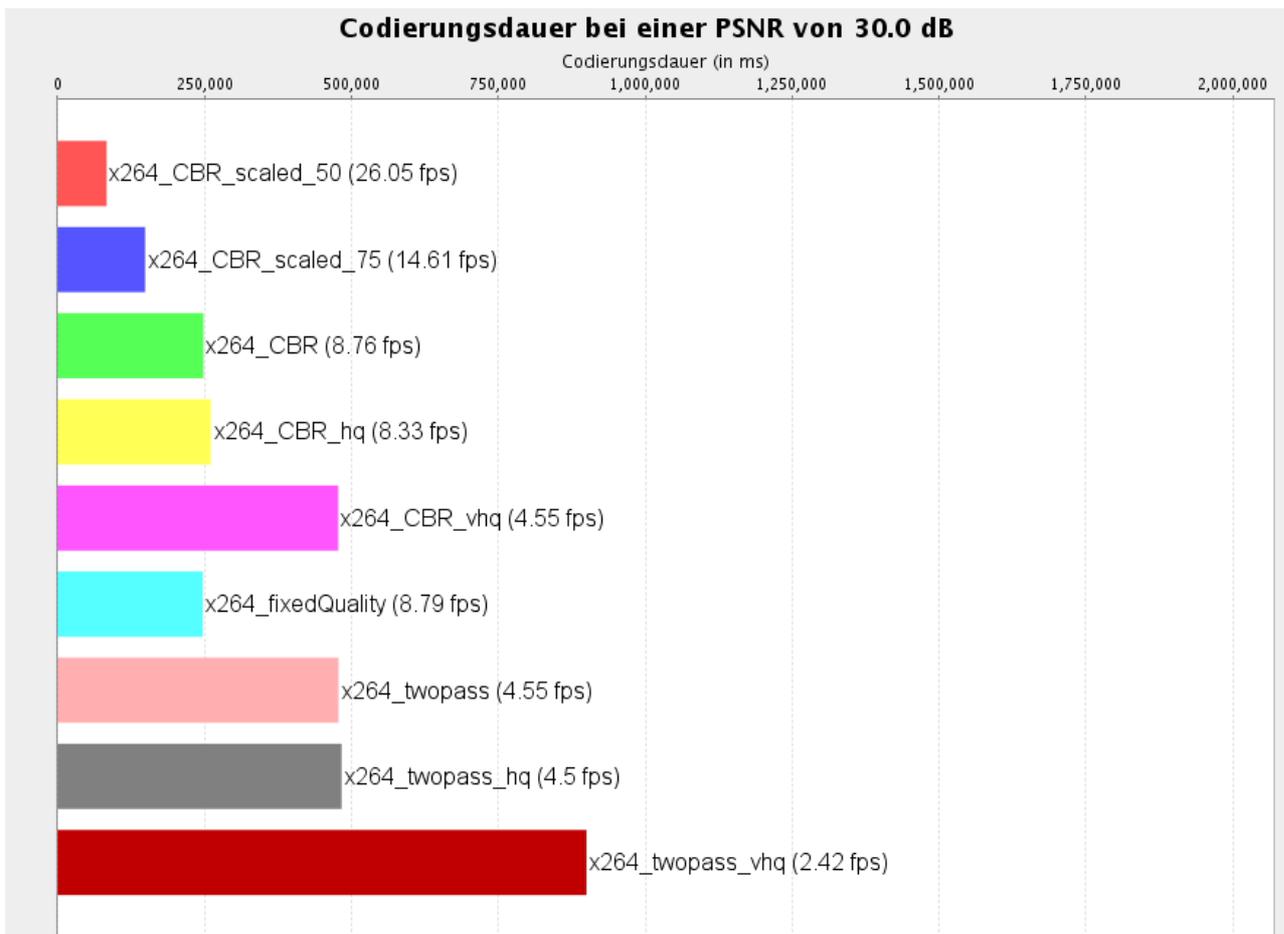


Abbildung 4: Codierungsdauer bei einer PSNR von 30 dB bei der CBR-Codierung ohne optimierende Einstellungen beim x264 Codec. Die Werte in den Klammern geben an, wie viele Bilder pro Sekunde (auf die ganze Laufzeit umgerechnet) verarbeitet wurden.

Durch die gewählten Optionen bei einer Codierung kann es beim gleichen Codec zu großen Unterschieden bzgl. der Laufzeit und der Dateigröße bei gleicher PSNR kommen. In Abb. 4 und 5 werden diese Unterschiede beim x264-Codec bei einer PSNR von 30 dB visualisiert. Bei den anderen drei Codecs ergeben sich beim Verhältnis der verschiedenen Optionen zueinander ähnliche Darstellungen. Für die Darstellung wurde exemplarisch der x264-Codec gewählt, da er die gleichmäßigste Entwicklung über die Messreihen aufwies.

In Abb. 4 ist zu sehen, dass die Grundeinstellung ohne zusätzliche Parameter eine sehr gute Laufzeit hat, jedoch (wie in Abb. 5 dargestellt) auch die größte Video-Datei generiert. Die Größe der Video-Datei kann zum Erreichen einer gleichen PSNR zum einen durch

optimierende Parameter reduziert werden und zum anderen durch das Verwenden des twopass-Modus.

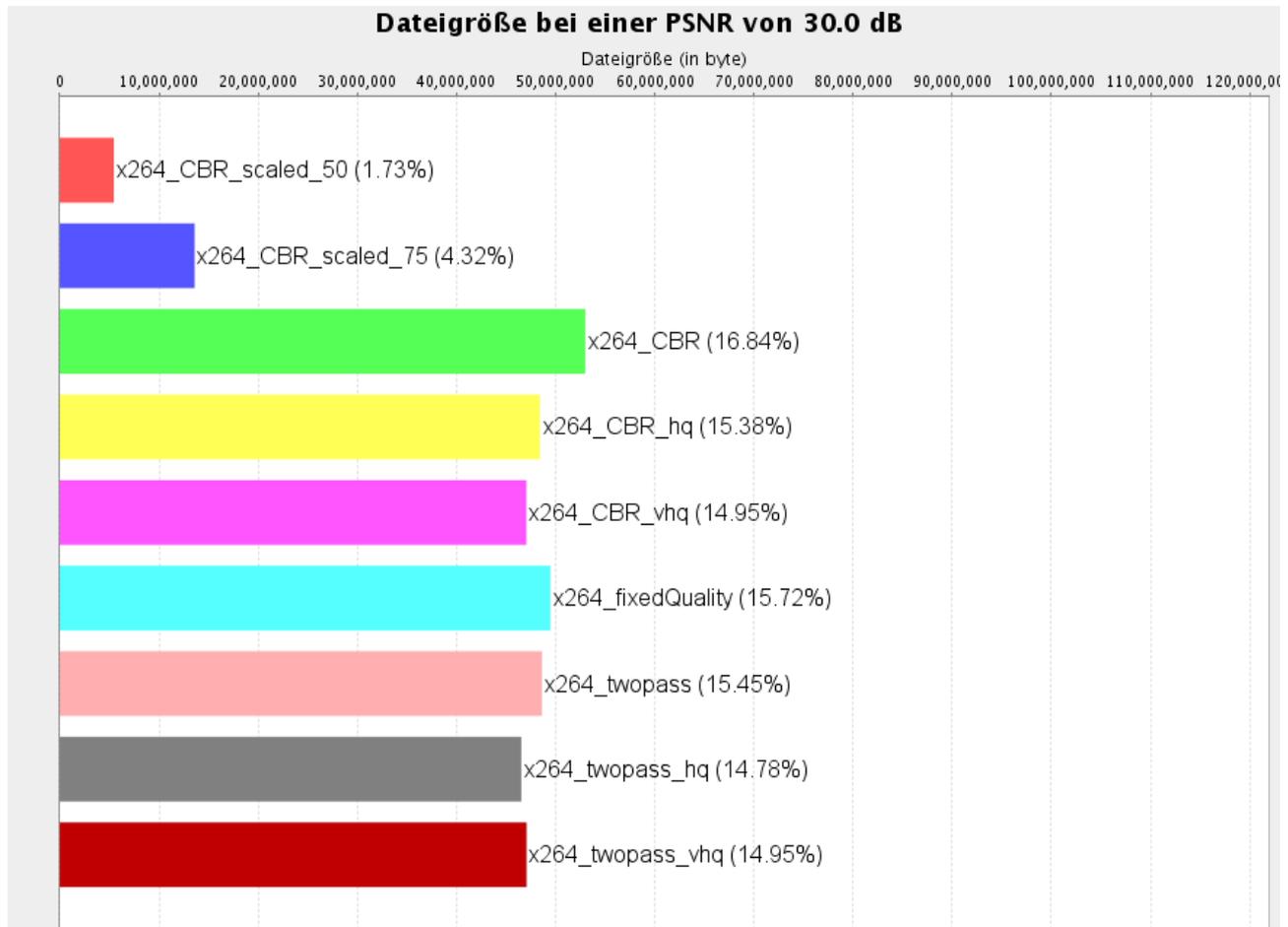


Abbildung 5: Dateigröße bei einer PSNR von 30 dB bei der CBR-Codierung ohne optimierende Einstellungen beim x264 Codec. Die Werte in den Klammern geben an, wie viel Prozent das Video von dem Speicherbedarf der Originalbilderserie einnahm.

Wenn keine exakte Dateigröße benötigt wird, sind leicht optimierende Einstellungen zu empfehlen. Mit geeigneten Parametern ist es möglich, die Dateigröße bei gleicher Qualität zu reduzieren, ohne die Laufzeit deutlich zu erhöhen. Starke Optimierungen sind nur zu empfehlen, wenn die Laufzeit keine oder nur eine geringe Relevanz hat, da sich die Laufzeit durch einige Parameter deutlich erhöht. Eine Abschätzung der Auswirkungen von einigen Parametern auf die PSNR ist im MPlayer-Handbuch¹⁶ zu finden. Falls eine exakte Dateigröße benötigt wird, kann die twopass-Codierung genutzt werden. Dadurch wird die Laufzeit allerdings in etwa verdoppelt. Stark optimierende Einstellungen bewirken beim twopass-Modus gegenüber leicht optimierenden Einstellungen kaum eine Verbesserung. Teilweise kam es sogar, wie in Abb. 5 zu sehen, zum umgekehrten Effekt. Die qualitätsbasierte Codierung bringt bzgl. Speicherplatz und Laufzeit ähnliche Ergebnisse wie die Codierung mit konstanter Bitrate ohne optimierende Einstellungen und ist daher nicht zu empfehlen.

Wenn die Video-Dateien möglichst wenig Speicherplatz verbrauchen sollen, hat sich eine Reduktion der Auflösung als brauchbare Lösung herausgestellt: Die Laufzeit sinkt dadurch erheblich und die erzeugten Video-Dateien benötigen deutlich weniger Speicherplatz. Der

¹⁶ <http://www.mplayerhq.hu/DOCS/HTML/en/encoding-guide.html>

Nachteil ist, dass feine Details durch die geringere Auflösung wegfallen. Wenn dies akzeptabel ist, stellt eine Verringerung der Auflösung eine optimale Möglichkeit dar, um die Laufzeit zu verringern und gleichzeitig die Video-Dateigröße zu reduzieren.

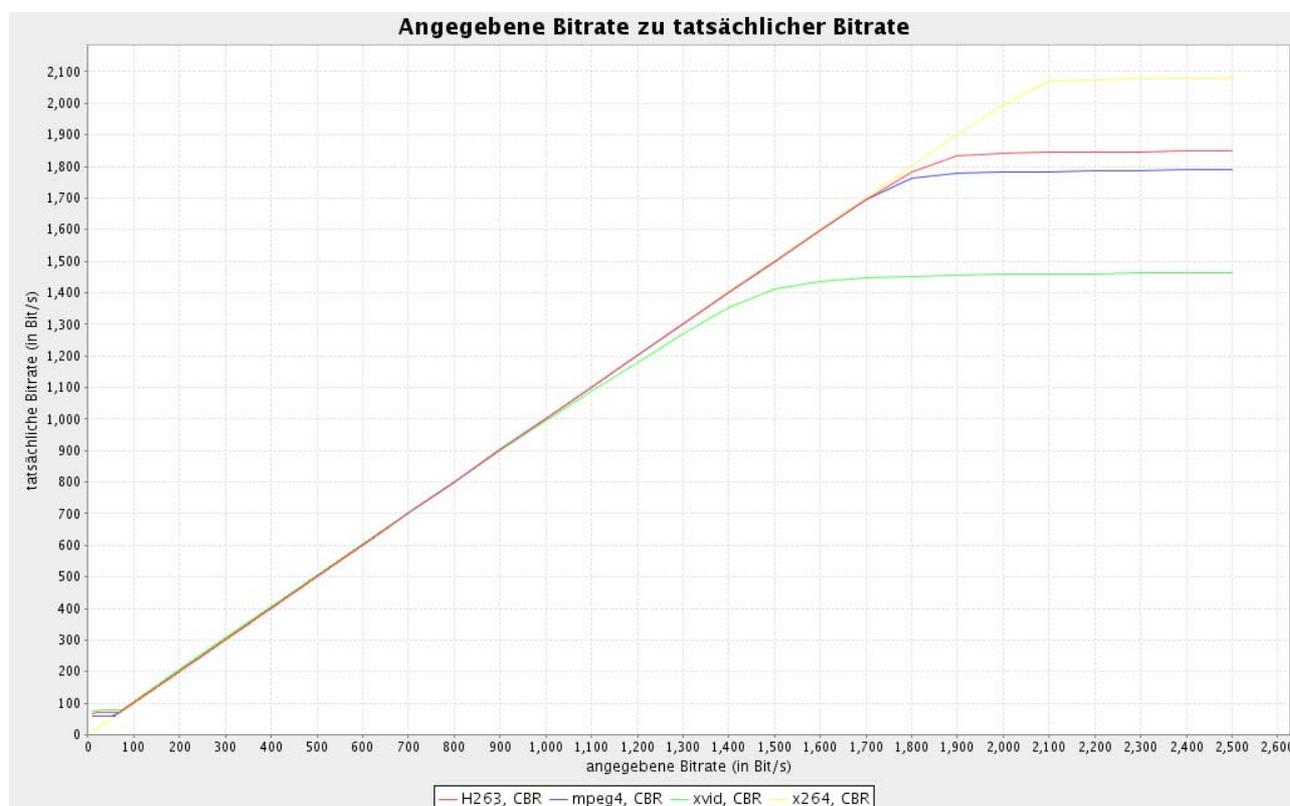


Abbildung 6: Verhältnis der angegebenen Bitrate zu der tatsächlich verwendeten Bitrate

Abb. 6 zeigt, dass der x264-Codec die geforderte Bitrate auch bei niedrigen Werten korrekt umsetzt. Bei den anderen drei Codecs ist dies nicht möglich, da sie anscheinend an eine interne Qualitätsgrenze gebunden sind, die sie nicht unterschreiten.

Ab einer PSNR von ca. 30 dB war die Bildqualität annehmbar. Ab einem PSNR von ca. 32 dB war das generierte Video kaum von den Ausgangsbildern zu unterscheiden. Trotzdem war eine Abweichung zum Originalbild feststellbar. Es handelte sich dabei um ein leichtes Weichzeichnen von feinen Strukturen und eine Farbverfälschung. Um die Ursache dieser Verfälschungen festzustellen, wurden die Bilder testweise mit dem FFmpeg-Werkzeug unter Linux und mit VirtualDub unter Microsoft Windows in Videos umgewandelt. Es stellte sich heraus, dass es bei der automatischen Umwandlung der JPEG-Bilder in ein Motion-JPEG-Video durch mencoder und FFmpeg (dessen Bibliothek von mencoder genutzt wird) zu einer fehlerhaften Farbkonvertierung kommt. Wie in der Mailing-Liste von mencoder festgestellt werden kann, scheint mencoder tatsächlich nicht behobene Fehler bei der Farbkonvertierung zu enthalten.

Nach einer Begutachtung der Videoqualität durch menschliche Betrachter stellte sich heraus, dass nur x264 ab einer gewissen Bitrate feine Details und Farbschattierungen nicht mehr weichzeichnet. Bei den anderen Codecs trat ein Weichzeichnen selbst bei sehr hohen Bitraten auf. Das Weichzeichnen kann durch die Wahl geeigneter Quantizer reduziert werden. Als eine geeignete Einstellung wurde der MPEG-Quantizer identifiziert. Bei den Codecs FFmpeg-MPEG4 und Xvid kann dieser von der Standardeinstellung „H.263“ auf „MPEG“ geändert werden, um ein besseres Ergebnis zu erhalten. Die Laufzeit ändert sich durch diese Einstellung kaum. Der x264-Codec arbeitet bereits mit

entsprechend guten Quantizer-Einstellungen, während beim H.263-Codec kein MPEG-Quantizer gewählt werden kann. Zur feineren Abstimmung können manuell eine Reihe weiterer Quantizer-Parameter spezifiziert werden.

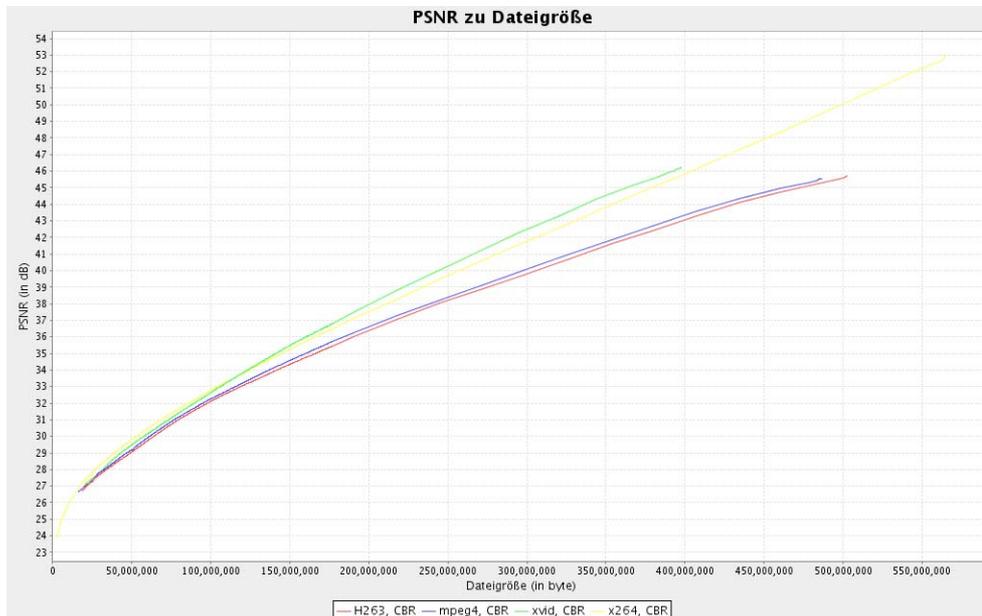


Abbildung 7: Verhältnis von PSNR zu Dateigröße

In Abb. 7 ist zu sehen, dass der x264-Codec bei geringen Bitraten die höchste Qualität bei kleinster Speicherplatzausnutzung gegenüber den anderen drei Codecs hat. Ab etwa einer PSNR von 34 dB erweist sich der Xvid-Codec als besser.

4.2.4 Codec-Empfehlungen

Videos als Quicklooks sind dann eine gute Alternative zu Einzelbildern, wenn kleinere Fehler durch Farbverfälschungen und Weichzeichnen kein Problem darstellen. Andernfalls ist von Videos als Darstellungsform abzuraten.

Grundsätzlich müssen bei der Auswahl eines geeigneten Codecs mehrere Kriterien berücksichtigt werden. Es hat sich herausgestellt, dass die Codecs bei gleicher Video-Qualität eine unterschiedliche Laufzeit haben und unterschiedlich große Video-Dateien generieren. Außerdem muss zum Abspielen bei manchen Formaten mit größerer Wahrscheinlichkeit ein Codec nachinstalliert werden als bei anderen Formaten. Auf Grund der weiten Verbreitung von Flash-Plugins kann davon ausgegangen werden, dass Flash-Videos am ehesten abgespielt werden können.

Video-Dateien, die mit einem zum MPEG-4 Part 2 Standard kompatiblen Codec codiert sind, sollten ebenfalls auf einer Vielzahl von Systemen abspielbar sein. Der in den AVI-Dateien angegebene Codierungscodec sollte bei der Verwendung der MPEG-4 Part 2 kompatiblen Codecs allerdings auf „DX50“ gesetzt werden. Zum Abspielen wird dann der kompatible weitverbreitete Codec DivX 5 genutzt. Beim relativ neuen x264-Codec kann momentan davon ausgegangen werden, dass er auf vielen Systemen noch installiert werden muss.

Als der leistungsfähigste Codec hat sich x264 herausgestellt. Er produziert bei niedrigen Bitraten die besten Ergebnisse. Bei höheren Bitraten produzierte bei den verwendeten Ausgangsbildern allerdings Xvid bei vergleichbarer Qualität die kleineren Video-Dateien.

Bei dem verwendeten Bildmaterial waren solche hohen Bitraten nicht notwendig, um von der Bildqualität her sehr gute Ergebnisse zu erhalten.

FFmpeg MPEG-4 und H.263 erzeugen bei vergleichbarer Qualität durchweg größere Dateien als x264 und Xvid. Dafür benötigen FFmpeg MPEG-4 und H.263 bei Codierungen ohne optimierende Einstellungen nur ca. 25% der x264-Codec-Laufzeit. Je nachdem, ob Speicherplatz oder Zeit der begrenzende Faktor ist, sollte daher zwischen x264 und Xvid oder FFmpeg MPEG-4 und H.263 gewählt werden.

Falls viel gleichartiges Bildmaterial in Videos umgewandelt werden soll, ist es empfehlenswert Videos testweise zu erstellen, um zu ermitteln wie gut die Codecs das Material codieren können und mit welchen Einstellungen die PSNR bzw. die Bildqualität ausreichend ist.

Eine alternative Möglichkeit Bilderserien zu bündeln sind animierte GIF-Dateien. Diese haben sich gegenüber modernen Videoformaten bei den durchgeführten Test nur in Ausnahmefällen als praktikabel erwiesen. Dies liegt daran, dass das Erzeugen von GIF-Dateien zu langen Bilderserien mit z. B. ImageMagick¹⁷ sehr viel Arbeitsspeicher benötigt. Zum Anderen sind GIF-Bilder in ihrer Darstellung auf 256 Farben beschränkt. Viele Bilddateien verwenden mehr als 256 Farben. Durch eine Reduktion der Farbpalette würde es daher zu einer schlechteren Bildqualität kommen. Außerdem führt die Farbreduktion zu Bildern, die sich schlecht komprimieren lassen.

Falls die Ausgangsbilder allerdings mit 256 Farben auskommen, werden GIF-Bilder eine attraktive Alternative, die gegenüber z. B. JPEG-Bildern nur 15 bis 20% des Speicherplatzes benötigt. Da die GIF-Bilder in einem solchen Fall gegenüber komprimierten Videos in Originalqualität vorliegen, sind sie hier ausnahmsweise die bessere Wahl. Ein Gewinn an Speicherplatz durch ein animiertes GIF-Bild gegenüber einzelnen GIF-Bildern ist vorhanden aber gering.

4.2.5 Erweiterungen

Für weitere Tests könnten die Codecs, soweit sie es erlauben, auf mehrere Threads aufgeteilt und so das Erstellen eines Videos parallelisiert werden. Auf einem Computer mit mehreren CPU-Kernen wäre ein Geschwindigkeitsgewinn zu erwarten. Um auf mehrere Threads aufgeteilt werden zu können, müssen die Codecs und mencoder allerdings speziell dafür kompiliert werden.

Generell könnten die optimierenden Parameter besser an das Bildmaterial angepasst werden. Mit einer manuellen Feinabstimmung sind mit Sicherheit noch Optimierungen bzgl. Qualität, Codierungsgeschwindigkeit und Speicherbedarf der Videos möglich. Speziell beim twopass-Verfahren könnte beim ersten Durchlauf durch Weglassen einiger Parameter, die sich negativ auf die Laufzeit auswirken, die Codierungsgeschwindigkeit erhöht werden.

Zur besseren Qualitätsabschätzung könnten statt die PSNR neuere Verfahren wie z. B. der Structural Similarity Index¹⁸ (SSIM) verwendet werden. Dieses neuere Verfahren ist allerdings nicht in die Codecs integriert. Es müsste das fertige Video nach der Codierung mit den Originalbildern verglichen werden, was zu einer deutlich längeren Testdauer führen würde.

Falls es den Nutzern zumutbar ist, speziell Codecs zum Anschauen der Videos zu installieren, könnten Open-Source-Codecs wie z. B. Theora oder Dirac getestet werden.

¹⁷ <http://www.imagemagick.org>

¹⁸ <http://www.ece.uwaterloo.ca/~z70wang/research/ssim>

Da die Codecs weiterentwickelt und optimiert werden, wäre es denkbar, die getesteten Codecs in regelmäßigen Abständen erneut Tests zu unterziehen. Wie Tests des Computermagazins c't gezeigt haben (c't 10/2003, S. 146ff und c't 10/2005, S. 146ff) können mit neueren bzw. optimierten Codecs bessere Ergebnisse erwartet werden.

In den durchgeführten Tests wurde mencoder verwendet. Falls die Codierung unter Windows durchgeführt werden soll, könnte eine andere Software von der eine geringere Farbverfälschung als von mencoder zu erwarten ist (z. B. AviSynth¹⁹) verwendet werden.

4.2.6 Technische Anleitungen

Im Folgenden sind für jeden Codec beispielhaft für jeweils alle Testklassen die mencoder Aufrufe dargestellt.

FFmpeg MPEG4 Codec

CBR, scaled 50:	mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -vf scale=464:464 -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:psnr -ffourcc DX50 -nosound -ofps 1 -of avi -o output/mpeg4_CBR_scaled_50_1000.avi
CBR, scaled 75:	mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -vf scale=696:696 -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:psnr -ffourcc DX50 -nosound -ofps 1 -of avi -o output/mpeg4_CBR_scaled_75_1000.avi
CBR:	mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:psnr -ffourcc DX50 -nosound -ofps 1 -of avi -o output/mpeg4_CBR_1000.avi
CBR, hq:	mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:mpeg_quant:mbd=2:trell:v4mv:last_pred=2 :dia=1:vmax_b_frames=2:vb_strategy=1:cmp=3:subcmp=3:precmp=0:vq comp=0.6:turbo:psnr -ffourcc DX50 -nosound -ofps 1 -of avi -o output/mpeg4_CBR_hq_1000.avi
CBR, vhq:	mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:mpeg_quant:mbd=2:mv0:trell:v4mv:cbp:las t_pred=3:predia=2:dia=2:vmax_b_frames=2:vb_strategy=1:precmp=2:cm p=2:subcmp=2:preme=2:qns=2:psnr -ffourcc DX50 -nosound -ofps 1 -of avi -o output/mpeg4_CBR_vhq_1000.avi
Fixed Quality:	mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vqscale=1:psnr -ffourcc DX50 -nosound -ofps 1 -of avi -o output/mpeg4_fixedQuality_1.avi
Twopass:	mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:psnr:vpass=1 -ffourcc DX50 -nosound - ofps 1 -of avi -o output/mpeg4_twopass_1000.avi mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:psnr:vpass=2 -ffourcc DX50 -nosound - ofps 1 -of avi -o output/mpeg4_twopass_1000.avi
Twopass, hq:	mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:mpeg_quant:mbd=2:trell:v4mv:last_pred=2

¹⁹ <http://avisynth.org>

	<pre>:dia=- 1:vmax_b_frames=2:vb_strategy=1:cmp=3:subcmp=3:precmp=0:vqcomp =0.6:turbo:psnr:vpas=1 -ffourcc DX50 -nosound -ofps 1 -of avi -o output/mpeg4_twopass_hq_1000.avi mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:mpeg_quant:mbd=2:trell:v4mv:last_pred=2 :dia=- 1:vmax_b_frames=2:vb_strategy=1:cmp=3:subcmp=3:precmp=0:vqcomp =0.6:turbo:psnr:vpas=2 -ffourcc DX50 -nosound -ofps 1 -of avi -o output/mpeg4_twopass_hq_1000.avi</pre>
Twopass, vhq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:mpeg_quant:mbd=2:mv0:trell:v4mv:cbp:las t_pred=3:predia=2:dia=2:vmax_b_frames=2:vb_strategy=1:precmp=2:cm p=2:subcmp=2:preme=2:qns=2:psnr:vpas=1 -ffourcc DX50 -nosound - ofps 1 -of avi -o output/mpeg4_twopass_vhq_1000.avi mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1000:mpeg_quant:mbd=2:mv0:trell:v4mv:cbp:las t_pred=3:predia=2:dia=2:vmax_b_frames=2:vb_strategy=1:precmp=2:cm p=2:subcmp=2:preme=2:qns=2:psnr:vpas=2 -ffourcc DX50 -nosound - ofps 1 -of avi -o output/mpeg4_twopass_vhq_1000.avi</pre>

H.263 Codec

CBR, scaled 50:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -vf scale=464:464 -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:psnr -nosound -ofps 1 -of lavf -o output/H263_CBR_scaled_50_1000.flv</pre>
CBR, scaled 75:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -vf scale=696:696 -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:psnr -nosound -ofps 1 -of lavf -o output/H263_CBR_scaled_75_1000.flv</pre>
CBR:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:psnr -nosound -ofps 1 -of lavf -o output/H263_CBR_1000.flv</pre>
CBR, hq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:mbd=2:trell:v4mv:last_pred=2:dia=- 1:cmp=3:subcmp=3:precmp=0:vqcomp=0.6:turbo:psnr -nosound -ofps 1 - of lavf -o output/H263_CBR_hq_1000.flv</pre>
CBR, vhq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:mbd=2:mv0:trell:v4mv:cbp:last_pred=3:predia= 2:dia=2:precmp=2:cmp=2:subcmp=2:preme=2:qns=2:psnr -nosound - ofps 1 -of lavf -o output/H263_CBR_vhq_1000.flv</pre>
Fixed Quality:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vqscale=1:psnr -nosound -ofps 1 -of lavf -o output/H263_fixedQuality_1.flv</pre>
Twopass:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:psnr:vpas=1 -nosound -ofps 1 -of lavf -o</pre>

Effiziente Interaktion mit großen Datenobjekten mit Hilfe von Quicklooks (AP 4)

	<pre>output/H263_twopass_1000.flv mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:psnr:vpas=2 -nosound -ofps 1 -of lavf -o output/H263_twopass_1000.flv</pre>
Twopass, hq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:mbd=2:trell:v4mv:last_pred=2:dia=- 1:cmp=3:subcmp=3:precmp=0:vqcomp=0.6:turbo:psnr:vpas=1 -nosound -ofps 1 -of lavf -o output/H263_twopass_hq_1000.flv mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:mbd=2:trell:v4mv:last_pred=2:dia=- 1:cmp=3:subcmp=3:precmp=0:vqcomp=0.6:turbo:psnr:vpas=2 -nosound -ofps 1 -of lavf -o output/H263_twopass_hq_1000.flv</pre>
Twopass, vhq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:mbd=2:mv0:trell:v4mv:cbp:last_pred=3:predia= 2:dia=2:precmp=2:cmp=2:subcmp=2:preme=2:qns=2:psnr:vpas=1 - nosound -ofps 1 -of lavf -o output/H263_twopass_vhq_1000.flv mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc lavc -lavcopts vcodec=flv:vbitrate=1000:mbd=2:mv0:trell:v4mv:cbp:last_pred=3:predia= 2:dia=2:precmp=2:cmp=2:subcmp=2:preme=2:qns=2:psnr:vpas=2 - nosound -ofps 1 -of lavf -o output/H263_twopass_vhq_1000.flv</pre>

x264 Codec

CBR, scaled 50:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -vf scale=464:464 -ovc x264 -x264encopts bitrate=1000:psnr -nosound -ofps 1 -of avi -o output/x264_CBR_scaled_50_1000.avi</pre>
CBR, scaled 75:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -vf scale=696:696 -ovc x264 -x264encopts bitrate=1000:psnr -nosound -ofps 1 -of avi -o output/x264_CBR_scaled_75_1000.avi</pre>
CBR:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc x264 -x264encopts bitrate=1000:psnr -nosound -ofps 1 -of avi -o output/x264_CBR_1000.avi</pre>
CBR, hq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc x264 -x264encopts bitrate=1000:subq=5:8x8dct:frameref=2:bframes=3:b_pyramid:weight_b:p snr -nosound -ofps 1 -of avi -o output/x264_CBR_hq_1000.avi</pre>
CBR, vhq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc x264 -x264encopts bitrate=1000:subq=6:partitions=all:8x8dct:me=umh:frameref=5:bframes= 3:b_pyramid:weight_b:psnr -nosound -ofps 1 -of avi -o output/x264_CBR_vhq_1000.avi</pre>
Fixed Quality:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc x264 -x264encopts crf=1:psnr -nosound -ofps 1 -of avi -o output/x264_fixedQuality_1.avi</pre>
Twopass:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile x264_twopass_1000.log -ovc x264 -x264encopts bitrate=1000:psnr:pass=1 -nosound -ofps 1 -of avi -o</pre>

Effiziente Interaktion mit großen Datenobjekten mit Hilfe von Quicklooks (AP 4)

	<pre>output/x264_twopass_1000.avi mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile x264_twopass_1000.log -ovc x264 -x264encopts bitrate=1000:psnr:pass=2 -nosound -ofps 1 -of avi -o output/x264_twopass_1000.avi</pre>
Twopass, hq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile x264_twopass_hq_1000.log -ovc x264 -x264encopts bitrate=1000:subq=5:8x8dct:frameref=2:bframes=3:b_pyramid:weight_b:p snr:pass=1 -nosound -ofps 1 -of avi -o output/x264_twopass_hq_1000.avi mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile x264_twopass_hq_1000.log -ovc x264 -x264encopts bitrate=1000:subq=5:8x8dct:frameref=2:bframes=3:b_pyramid:weight_b:p snr:pass=2 -nosound -ofps 1 -of avi -o output/x264_twopass_hq_1000.avi</pre>
Twopass, vhq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile x264_twopass_vhq_1000.log -ovc x264 -x264encopts bitrate=1000:subq=6:partitions=all:8x8dct:me=umh:frameref=5:bframes= 3:b_pyramid:weight_b:psnr:pass=1 -nosound -ofps 1 -of avi -o output/x264_twopass_vhq_1000.avi mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile x264_twopass_vhq_1000.log -ovc x264 -x264encopts bitrate=1000:subq=6:partitions=all:8x8dct:me=umh:frameref=5:bframes= 3:b_pyramid:weight_b:psnr:pass=2 -nosound -ofps 1 -of avi -o output/x264_twopass_vhq_1000.avi</pre>

Xvid Codec

CBR, scaled 50:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -vf scale=464:464 -ovc xvid -xvidencopts bitrate=1000:psnr -nosound -ofps 1 -of avi -o output/xvid_CBR_scaled_50_1000.avi</pre>
CBR, scaled 75:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -vf scale=696:696 -ovc xvid -xvidencopts bitrate=1000:psnr -nosound -ofps 1 -of avi -o output/xvid_CBR_scaled_75_1000.avi</pre>
CBR:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc xvid -xvidencopts bitrate=1000:psnr -nosound -ofps 1 -of avi -o output/xvid_CBR_1000.avi</pre>
CBR, hq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc xvid -xvidencopts bitrate=1000:vhq=2:bvhq=1:chroma_opt:quant_type=mpeg:psnr - nosound -ofps 1 -of avi -o output/xvid_CBR_hq_1000.avi</pre>
CBR, vhq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc xvid -xvidencopts bitrate=1000:chroma_opt:vhq=4:bvhq=1:quant_type=mpeg:psnr - nosound -ofps 1 -of avi -o output/xvid_CBR_vhq_1000.avi</pre>
Fixed Quality:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -ovc xvid -xvidencopts fixed_quant=1:psnr -nosound -ofps 1 -of avi -o output/xvid_fixedQuality_1.avi</pre>
Twopass:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile</pre>

	<pre>xvid_twopass_1000.log -ovc xvid -xvidencopts bitrate=1000:psnr:pass=1 -nosound -ofps 1 -of avi -o output/xvid_twopass_1000.avi mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile xvid_twopass_1000.log -ovc xvid -xvidencopts bitrate=1000:psnr:pass=2 -nosound -ofps 1 -of avi -o output/xvid_twopass_1000.avi</pre>
Twopass, hq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile xvid_twopass_hq_1000.log -ovc xvid -xvidencopts bitrate=1000:vhq=2:bvhq=1:chroma_opt:quant_type=mpeg:psnr:pass=1 - nosound -ofps 1 -of avi -o output/xvid_twopass_hq_1000.avi mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile xvid_twopass_hq_1000.log -ovc xvid -xvidencopts bitrate=1000:vhq=2:bvhq=1:chroma_opt:quant_type=mpeg:psnr:pass=2 - nosound -ofps 1 -of avi -o output/xvid_twopass_hq_1000.avi</pre>
Twopass, vhq:	<pre>mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile xvid_twopass_vhq_1000.log -ovc xvid -xvidencopts bitrate=1000:chroma_opt:vhq=4:bvhq=1:quant_type=mpeg:psnr:pass=1 - nosound -ofps 1 -of avi -o output/xvid_twopass_vhq_1000.avi mencoder mf://input/jpg/*.jpg -mf fps=1:type=jpg -passlogfile xvid_twopass_vhq_1000.log -ovc xvid -xvidencopts bitrate=1000:chroma_opt:vhq=4:bvhq=1:quant_type=mpeg:psnr:pass=2 - nosound -ofps 1 -of avi -o output/xvid_twopass_vhq_1000.avi</pre>

4.3 Super-Overlays in Google Earth

Als eine mögliche neue Darstellungsform von Quicklooks werden im Folgenden Super-Overlays in Google Earth vorgestellt. Google Earth ist eine von Google für Linux, MacOS X und Microsoft Windows frei verfügbare Anwendung zum Betrachten der Erdoberfläche (Abb. 8). Zunächst wird die Erde in einer Weltraumansicht dargestellt. In dieser Ansicht lässt sich der Globus frei drehen und zoomen. Dabei ist es möglich, nahezu stufenlos bis auf eine Auflösung von wenigen Metern zu zoomen.

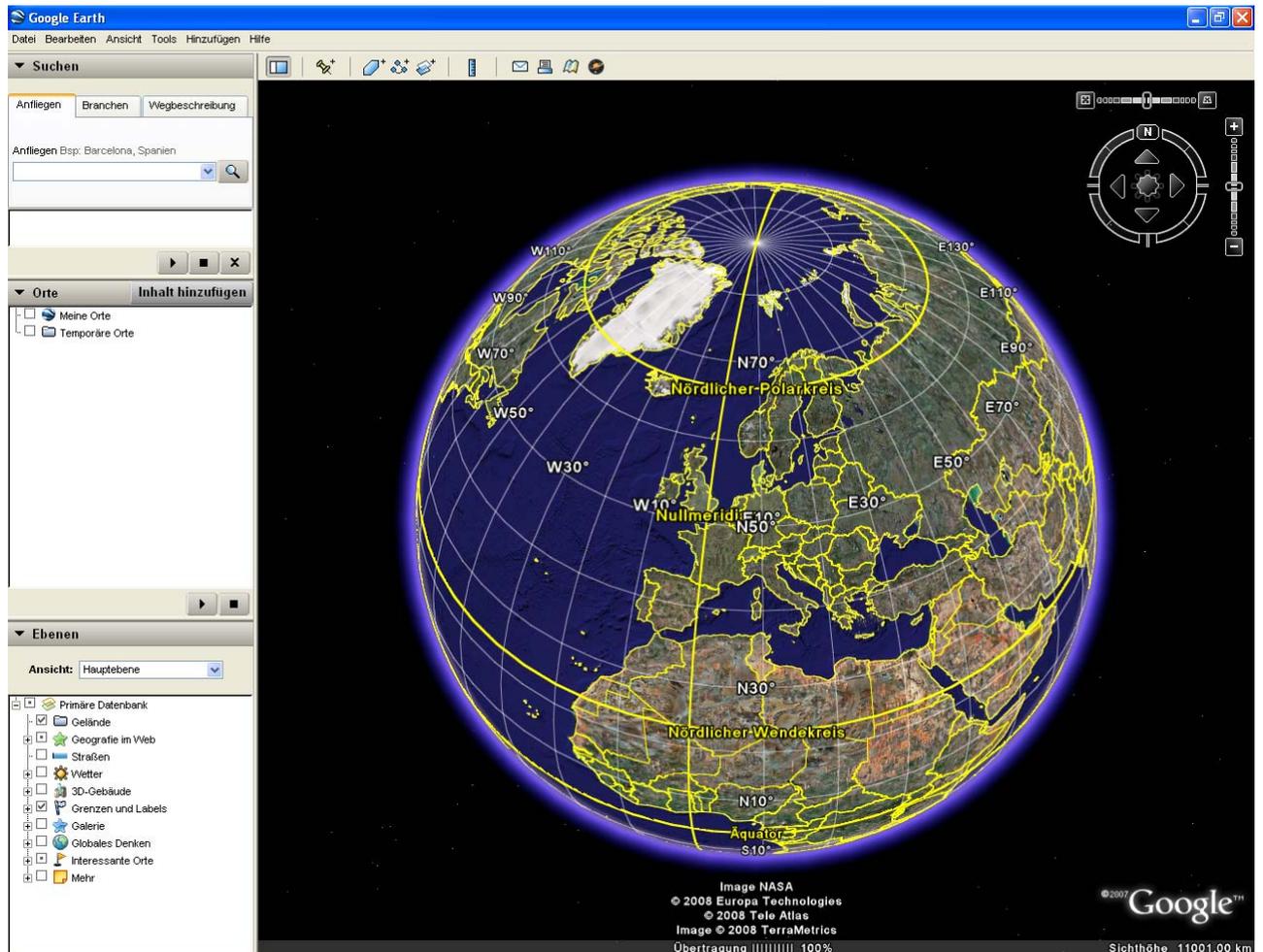


Abbildung 8: Die Startansicht von Google Earth

Um dies effizient zu bewerkstelligen, lädt Google Earth nur soviel Details, wie für die momentane Ansicht notwendig sind, und verwendet geschickte Caching-Strategien. Sobald ein Anwender die Ansicht ändert, werden die notwendigen Details nachgeladen²⁰.

Die Karte von Google Earth kann um weitere Objekte (z. B. 3D-Modelle von Gebäuden oder Grafiken, die über die Erdoberfläche gelegt werden) ergänzt werden. Dazu werden XML-basierte KML-Dateien²¹ in Google Earth geladen. Zusätzliche Grafiken, die über die Erdoberfläche gelegt werden, werden als *Ground Overlays* bzw. allgemein als *Overlays* bezeichnet. In einer KML-Datei können mehrere *Region*-Elemente definiert werden. Ein *Region*-Element umfasst eine Fläche auf der Google Earth Karte. Für jede *Region* berechnet Google Earth die Fläche, die diese in der aktuellen Ansicht einnehmen würde. Dabei kann für jede *Region* festgelegt werden, ab wie vielen minimal geladenen Bildpunkten sie aktiviert bzw. dargestellt wird und ab wie viel maximal geladenen Bildpunkten die *Region* wieder deaktiviert bzw. ausgeblendet wird. So ist es möglich, bestimmte Grafiken erst ab einer bestimmten Zoom-Stufe anzuzeigen und sie ab einer bestimmten Zoom-Stufe auch wieder auszublenden. Auf diese Weise kann in einer *Region* auf weitere KML-Dateien verwiesen werden, mit dem Ergebnis, dass diese erst beim Aktivieren der *Region* nachgeladen werden. So ist es z. B. möglich bei geringer Zoom-Stufe eine grob aufgelöste Grafik darzustellen. Sobald der Anwender weiter hineinzoomt, wird eine weitere KML-Datei mit einer detaillierteren Grafik geladen²². So können

²⁰ <http://www.cs.virginia.edu/~gfx/Courses/2002/BigData/papers/Texturing/Clipmap.pdf>

²¹ <http://code.google.com/apis/kml/documentation/>

²² http://code.google.com/apis/kml/documentation/kml_21tutorial.html#workingregions

Hierarchien von Region-Elementen angelegt werden, um schrittweise immer detailliertere Ausschnitte einer großen Grafik zu laden. Eine solche Hierarchie wird als *Super-Overlay* bezeichnet.

Der Vorteil solcher Super-Overlays ist, dass nur geladen wird, was der Nutzer sich genauer anschauen möchte. Dadurch sind die Ladezeiten für den Nutzer und das generelle Datenaufkommen geringer.

Über Super-Overlays können auch hochaufgelöste Satellitenbilder bzw. Quicklooks von Satellitenbildern angezeigt werden. Der Nutzer kann in Google Earth die Lage der Grafik feststellen und dann nur in den Teil der Grafik hineinzoomen, der für ihn relevant ist.

4.3.1 Anlegen eines Super-Overlays

Zum Anlegen eines Overlays wird als Ausgangsbasis eine hoch aufgelöste Grafik, sowie ihre genaue Lage benötigt. Als Beispielgrafik wurde eine Luftaufnahme von Sri Lanka²³ verwendet (Abb. 9). Sri Lanka wurde gewählt, da es sich leicht auf der Erdoberfläche identifizieren lässt. Dies war hilfreich, da die Grafik zunächst in Google Earth geladen werden musste, um manuell die genaue Lage auf dem Globus zu ermitteln. Bei Kartenausschnitten ohne markante Punkte oder dichter Wolkenbedeckung kann die genaue Lage eventuell nicht eindeutig in Google Earth bestimmt werden. Dies wird unter Umständen erschwert, wenn die Grafik noch gedreht werden muss. Bei Sri Lanka kann die Lage auf Grund der Inselumrisse leicht festgestellt werden. Für alle im operationellen Einsatz genutzten Satellitenbilder muss daher eine vollständige Geolokation vorhanden sein – idealerweise sollten sie in der Google Earth intern genutzten WGS84-Projektion des World Geodetic System vorliegen.

23

http://www.ssec.wisc.edu/~gumley/modis_gallery/images/SriLanka.2004361.0515.143.250m.jpg, (Liam Gumley, Space Science and Engineering Center, University of Wisconsin-Madison)

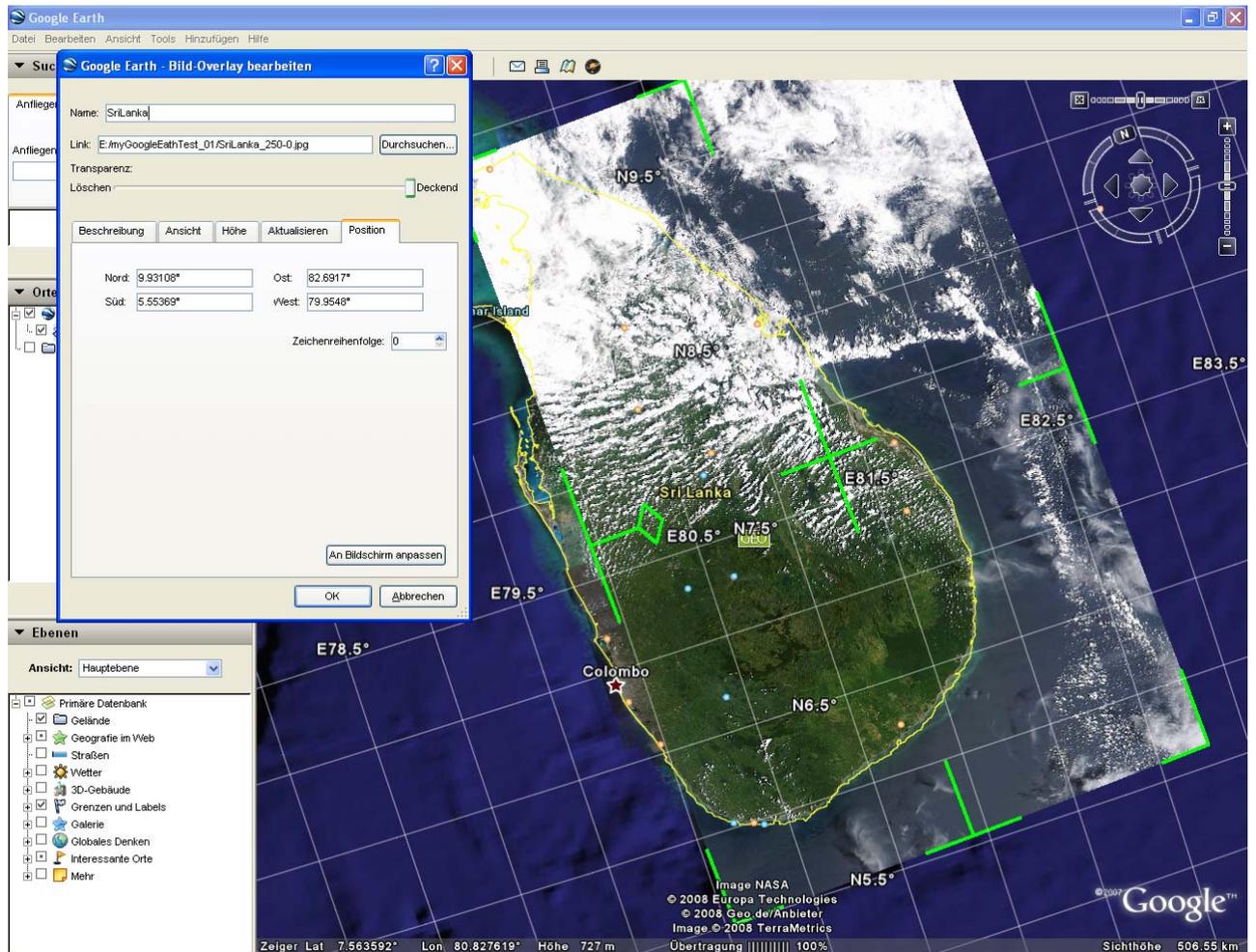


Abbildung 9: Die Lage der Overlay-Grafik von Sri Lanka wurde bestimmt

Nachdem die korrekte Lage der Grafik bestimmt wurde, kann mit den Nord-, Ost-, Süd- und West-Grenzen weitergearbeitet werden. Für ein unkompliziertes Erstellen eines Super-Overlays kann die kommerzielle Anwendung Superoverlay²⁴ eingesetzt werden (Abb. 10), von der allerdings nur eine Version für das Betriebssystem Microsoft Windows existiert. Dazu werden die Koordinaten der Grafik von Google Earth manuell nach Superoverlay übertragen. Dabei ist zu beachten, dass eventuell die Formatierung der Zahlen geändert werden muss, da Superoverlay nur in der Lage ist, Zahlen korrekt zu verarbeiten, die den lokalen Einstellungen entsprechen. Für die deutsche Dezimalschreibweise müssen die Zahlen beispielsweise ein Dezimalkomma statt einem Punkt aufweisen. Ein Dezimalpunkt führte zu fehlerhaften KML-Dateien. Die KML-Datei kann durch einen Klick erzeugt werden. Superoverlay sorgt dann für das korrekte Anlegen der Overlay-Grafiken in verschiedenen Auflösungen.

Die Super-Overlay-Struktur lässt sich auch manuell erzeugen. In diesem Fall muss die KML-Datei bzw. die KML-Dateien mit einem Texteditor angelegt werden. Die Grafiken können z. B. über die Kommandozeile mit ImageMagick²⁵ erzeugt werden. Dies führt allerdings zu einer großen Anzahl an Grafik-Dateien, die manuell verwaltet werden müssen.

²⁴ <http://xbbster.googlepages.com/superoverlay>

²⁵ <http://www.imagemagick.org>

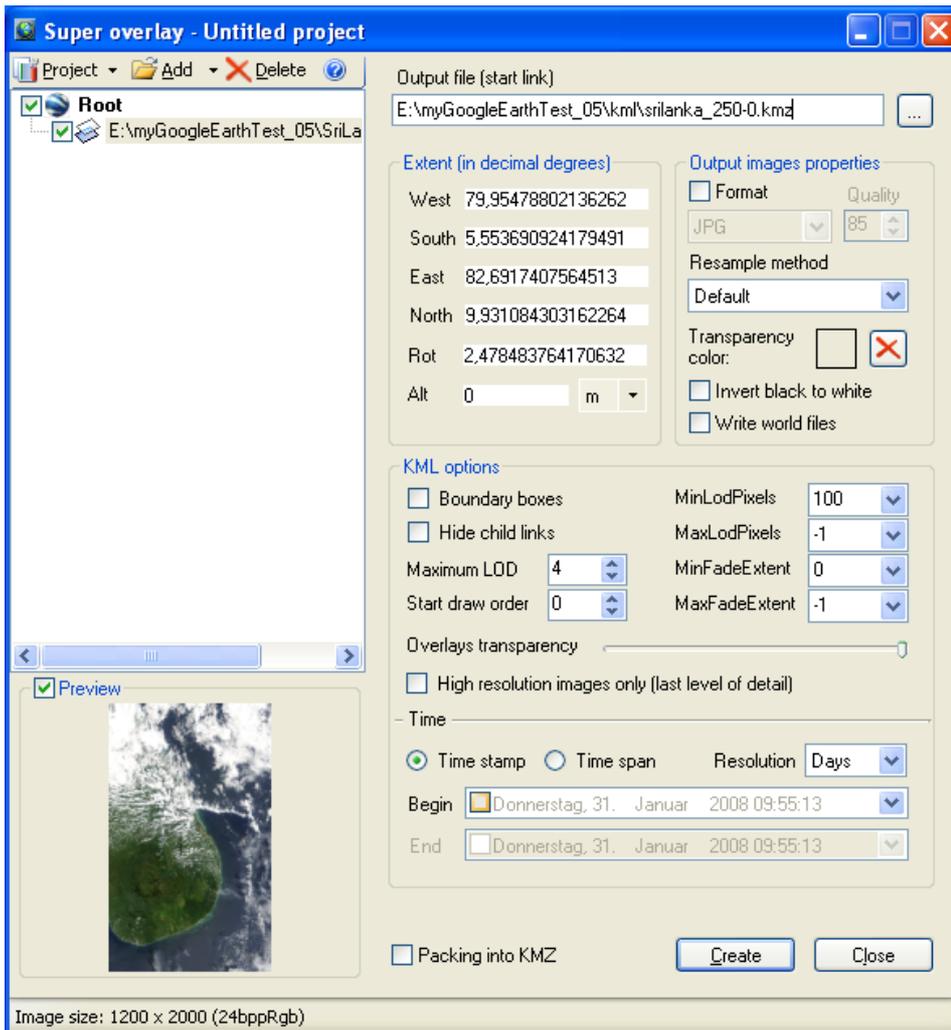


Abbildung 10: Super-Overlay Erstellung in Superoverlay

Das Sri Lanka-Bild hat eine Auflösung von 1200 x 2000 Pixel und kann in mehrere Teilbilder mit einheitlicher Größe umgewandelt werden. Die erste Ebene aus einem einzigen Bild könnte z. B. die Auflösung 300 x 500 Pixel haben. Dann sollten alle folgenden Teilbilder zur einfacheren Handhabung auch die Auflösung 300 x 500 Pixel haben. Das Gesamtbild in der zweiten Ebene hätte dann eine Auflösung von 600 x 1000 Pixel. In der dritten und detailliertesten Ebene würde das Gesamtbild aus 16 Teilbildern bestehen und hätte die volle Auflösung von 1200 x 2000 Pixel (Abb. 11 und 12).

Für ein performanteres Laden in Google Earth empfiehlt es sich allerdings, eine kleinere Auflösung der Teilbilder von z. B. 150 x 250 Pixel zu wählen. Dann würde es vier Ebenen geben, wobei die detaillierteste Ebene aus 64 Teilbildern bestehen würde. Das heißt, es gäbe 85 einzelne Bilder. Für jedes dieser Bilder müsste eine entsprechende Angabe im KML-Format geschrieben sowie die korrekte Lage berechnet und angegeben werden. Diese Arbeit von Hand durchzuführen ist sehr zeitaufwändig und fehleranfällig.

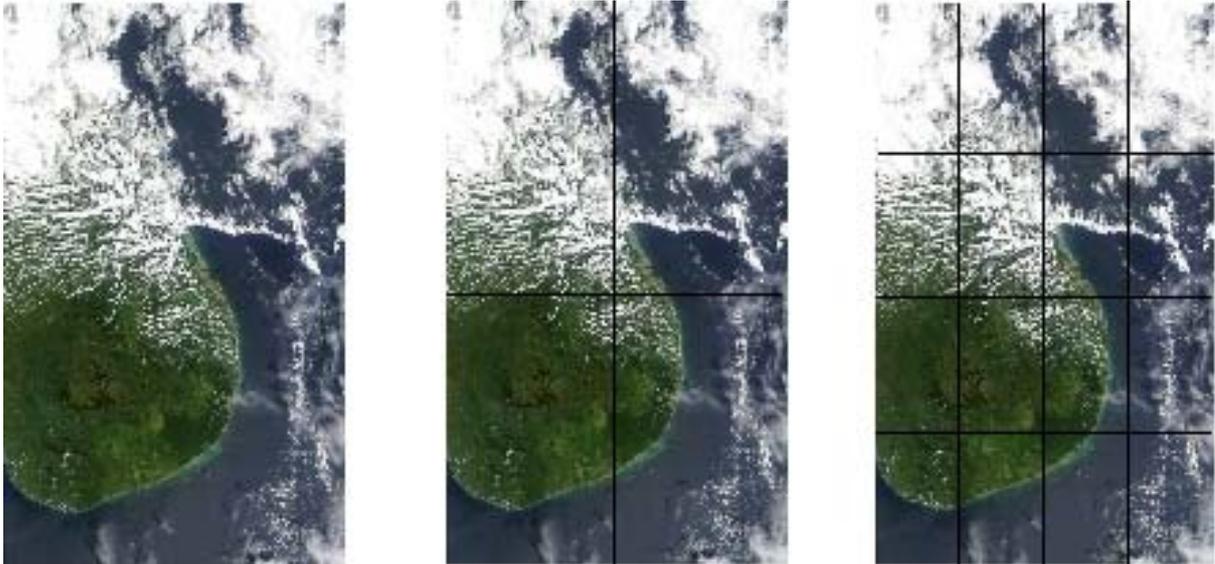


Abbildung 11: links: 300x500 Pixel, Mitte: 600x1000 Pixel, rechts: 1200x2000 Pixel

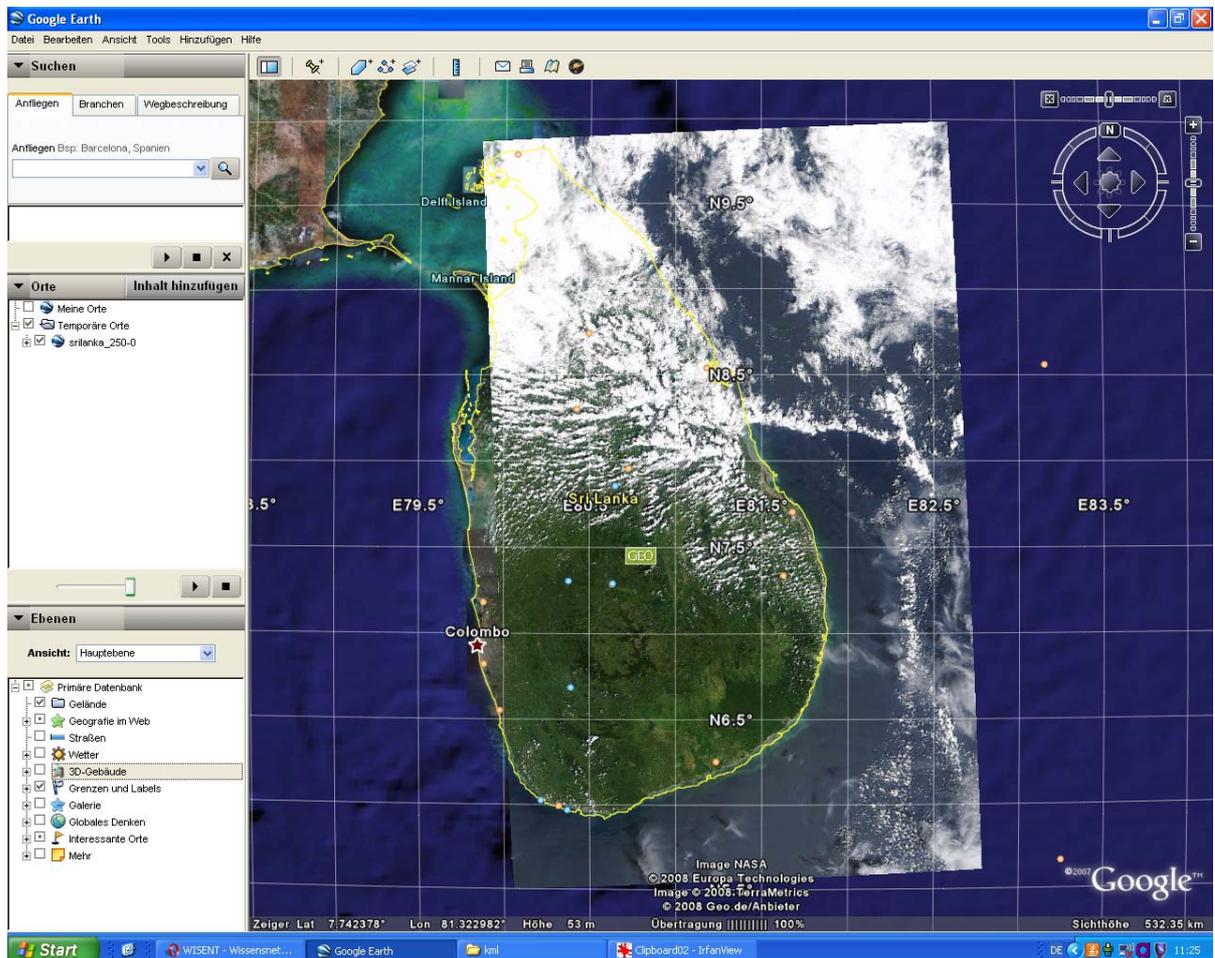


Abbildung 12: Die Overlay-Grafik in der Level 3 Ansicht

4.3.2 Super-Overlay-Generator

Um Super-Overlays skriptbasiert bzw. von der Kommandozeile aus erstellen zu können, wurde eine Java-Anwendung geschrieben. Diese wurde in der Programmiersprache Groovy implementiert und ist dadurch unter anderem unter Linux/Unix, MacOS X und

Microsoft Windows lauffähig. Der Super-Overlay-Generator wurde als Open-Source-Projekt unter SourceForge veröffentlicht: <http://sourceforge.net/projects/superoverlaygen/>

Über den Super-Overlay-Generator ist es möglich, sowohl zu einer einzelnen Grafik, als auch zu einer Serie von Grafiken ein Super-Overlay zu erstellen. Bei einer Bilderserie können die einzelnen Dateien mit Zeitbezügen versehen werden. Dadurch ist es möglich, ganze Bilderserien in Google Earth darzustellen und als Animation ablaufen zu lassen. Google Earth bietet dazu automatisch eine entsprechende Scrollbar sowie Funktionalität zum Starten und Stoppen.

Insbesondere bei rotierten Grafiken sind bei manchen Grafiken Übergänge zwischen den einzelnen Grafiken beim Hereinzoomen erkennbar. Hier könnte die Anwendung noch optimiert werden.

4.3.3 Möglichkeiten und Probleme

Ein Problem, das mit allen Bildern besteht, die in Super-Overlays umgewandelt werden sollen, ist, dass die genaue Lage bekannt sein muss und unter Umständen zunächst manuell in Google Earth ermittelt werden muss. Satellitenaufnahmen haben darüber hinaus das Problem, dass sie in der Regel in einer Projektion vorliegen, die nicht der in Google Earth verwendeten WGS84 entspricht. Satellitenaufnahmen müssen daher zunächst in eine verwendbare Projektion konvertiert werden. Dies hat gegebenenfalls manuell zu geschehen.

Durch den hohen manuellen Aufwand ist der Einsatz von Super-Overlays nur praktikabel, wenn es sich um regelmäßig erstellte Bilder handelt, deren Lage und Projektion nur einmal ermittelt werden muss und die sich über die Zeit nicht ändert. Nachdem die notwendige Projektion ermittelt wurde, kann die Konvertierung der Projektion und das Anlegen der Super-Overlays in Zukunft automatisch erfolgen. Zur Automatisierung der Super-Overlay-Erstellung wurde eine eigene Anwendung implementiert, die über die Kommandozeile oder über Skripte angesteuert werden kann.

4.3.4 Bewertung

- Der visuelle Eindruck der Darstellung in Google Earth ist sehr gut.
- Die zu ladende Datenmenge lässt sich über Super Overlays gut einschränken, da nur geladen wird was sich der Nutzer genauer anschaut.
- Die Erzeugung von Super Overlays kann allerdings recht arbeitsintensiv sein, wenn mangels geeigneter Metadaten für jedes Super Overlay die exakten Koordinaten der Grafik in Google Earth interaktiv ermittelt werden müssen.
- Alle Satellitenbilddaten müssen vor der Darstellung in Google Earth in die Standardprojektion von Google Earth (WGS84) umgerechnet werden.

Das Werkzeug eignet sich gut zur Visualisierung von einzelnen Beispielen und löst das gestellte Problem in hervorragender Weise. Mit dem Super-Overlay-Generator ist das Erstellen von Super-Overlays jetzt auch batch-fähig.

4.4 Nutzerspezifische Farbtabelle im GIF-Format

Auf tieferer Datenebene, d.h. unmittelbar nach der Datenprodukt-Erzeugung bzw. auch schon während der Prozessierung lassen sich QL-Informationen aus einer Kombination von QS-Metadaten (AP3.1) und aktuellen Produkt-Primärdaten ableiten. Dabei gewinnt der Begriff Quicklook eine erweiterte Bedeutung insofern, als viele Einzelinformationen nicht mehr sinnvoll einzeln dargestellt werden können, sondern durch besondere Aufbereitung komprimiert und in einer jeweils sinnvollen Zusammenschau eine Auswahl an Einzelinformationen als QL angesehen werden soll.

Für die am DFD erzeugten Produkte des HELIOSAT-Wolkenprozessors werden QL-relevante Informationen zu GIF-Dateien zusammengefasst. Das GIF-Format bietet für hinreichend flächenhafte QL-Informationen eine gute Kompression und ist sowohl als Einzelbild als auch als Film leicht darstellbar. GIF-Dateien sind stets 8-Bit tief, daher lassen sich in einer solchen Datei maximal 8 unabhängige Informationsebenen unterbringen. Diese Eigenschaften werden am DFD nun genutzt, um Informationen möglichst dicht gepackt so unterzubringen, dass sie auf Nutzerseite leicht rekombiniert und in möglichst vielen Kombinationen und Darstellungsformen angesehen werden können. Im Folgenden werden zur Veranschaulichung dieser speziellen, erweiterten QL-Sichtweise einige QL-Informationen zur Wolkenanalyse einer MSG-SEVIRI-Szene erläutert:

Für die SEVIRI-Szene vom 23. Juli 2004, 12:00 UT, werden die folgenden, die Wolkenanalyse betreffenden Qualitätsinformationen zu einem 8-Bit GIF-Quicklook als Beispiel zusammengestellt. Jede der folgenden Qualitätseigenschaften kann als binäre Information im Sinne einer Ja/Nein-Antwort auf eine der acht Bit-Ebenen gelegt werden. Dabei entspricht ein „Ja“ einem gesetzten, ein „Nein“ einem ungesetzten Bit auf der jeweiligen Ebene:

- GV Good-Value: Pixel ist nutzbar (ja oder nein)
- WS Wolkenstockwerk wurde erkannt
- WOT Wolkenoberflächentemperatur wurde berechnet
- WOD Wolkenoptische Dicke wurde berechnet
- GSS Größte Schneesicherheit, Pixel-Schnee-Eigenschaften maximal
- GWS Größte Wolkensicherheit, Pixel-Wolken-Eigenschaften maximal
- WT Wolkentyp wurde ermittelt
- SR Sonnenreflex möglich (aktuelles Beispiel enthält allerdings keinen SR)

In einer 8-Bit tiefen (GIF-)Datei sind theoretisch 255 von Null verschiedene Pixelwerte möglich. Bei der Zusammenstellung von acht unabhängigen Informationsebenen treten aber in der Regel nur relativ wenige der kombinatorisch möglichen Pixelwerte tatsächlich auf. Das liegt zum Teil daran, dass etliche Informationskombinationen keinen physikalischen Sinn ergeben und daher auch nicht auftreten dürfen. Je nach Ausschnitt, der gewählt wurde, treten aber auch nicht unbedingt immer alle physikalisch sinnvollen Kombinationen auf. Tab. 1 gibt für dieses SEVIRI-Wolkenanalysebeispiel eine Übersicht über die im GIF-Quicklook auftretenden Pixelwerte und deren Zusammensetzung aus den Einzelinformationen. In Klammern neben den Informationsklassen sind die Bit-Ebenen n in Form ihres numerischen Beitrages zum Pixelwert 2^{n-1} angegeben.

PW	GV(1)	WS(2)	WOT(4)	WOD(8)	GSS(16)	GWS(32)	WT(64)	SR(128)
0								
1	*							
3	*	*						
15	*	*	*	*				
17	*				*			
19	*	*			*			
35	*	*				*		
39	*	*	*			*		
47	*	*	*	*		*		
75	*	*		*			*	
79	*	*	*	*			*	
99	*	*				*	*	
107	*	*		*		*	*	
111	*	*	*	*		*	*	

Tabelle 1: Beispiel für acht Wolkenanalyse-Qualitätsklassen

Das komplette GIF-Quicklook (Abb.14) ist einem Farbkomposit der SEVIRI-Szene (Abb. 13) gegenübergestellt. Das GIF-QL ist schwarz-weiß dargestellt, nur der Pixelwert 0 (Good-value GV = „nein“) erscheint rot.



Abbildung 13: Ausschnitt SEVIRI-Komposit 23. Juli 2004, 12:00 UT

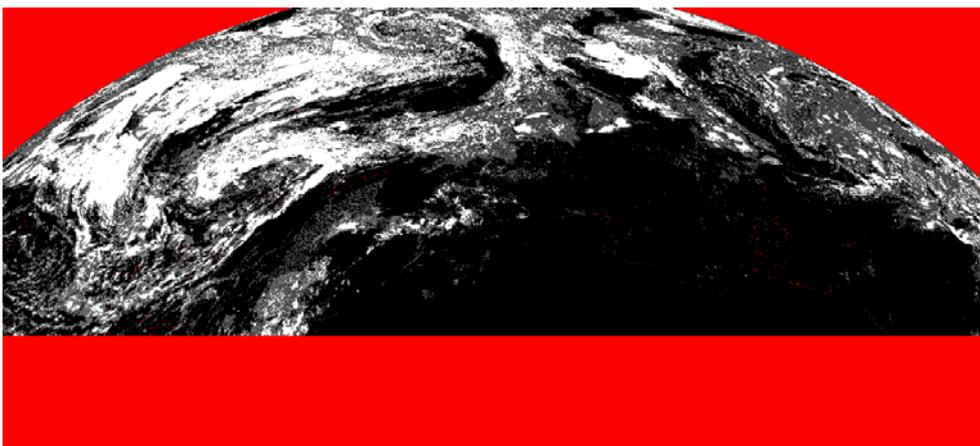


Abbildung 14: 8-Bit GIF-Quicklook mit Informationen laut Tab. 1

Da die Ansicht des kompletten, also alle acht Informationsebenen umfassenden Quicklooks nicht besonders anschaulich ist, können bestimmte Teilkombinationen leicht extrahiert und als so genannte Rekombinations-Quicklooks dargestellt werden. Die Abbildungen 15 und 16 zeigen jeweils drei der enthaltenen Informationen als Farbbilder, deren Rot-, Grün- und Blauanteil jeweils eine Information steuert. Tab. 2 zeigt hierzu, welche Information in welchem Bild welchem Farbkanal zugeordnet ist. Der Nutzer kann also aus dem kompakten 8-Bit tiefen GIF-Quicklook die für ihn interessanten Informationen zu individuellen drei-kanaligen Farbquicklooks extrahieren, indem zum Beispiel lediglich verschiedene Farbtabelle zu den GIF-QLs zugeladen werden. Dieses Verfahren ist auch für animierte GIF-Dateien anwendbar, so dass auch filmartige Darstellungen leicht erzeugt werden können. Dabei hält sich die QL-Datenmenge, die zum Nutzer übertragen werden muss, im Vergleich zu herkömmlichen QLs in Grenzen.

	GV(1)	WS(2)	WOT(4)	WOD(8)	GSS(16)	GWS(32)	WT(64)	SR(128)
Abb. 4.x3		■				■	■	
Abb. 4.x4		■		■		■		

Tabelle 2: Farbzuordnung (RGB) der Information in Abb. 15 und 16

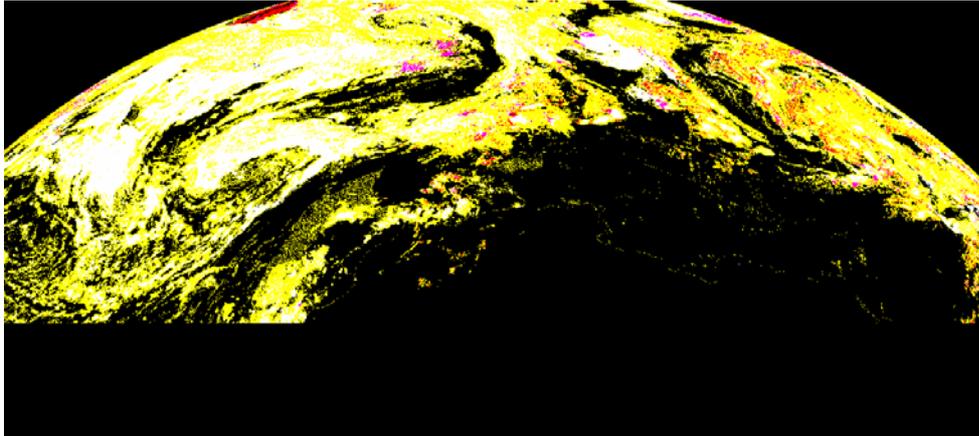


Abbildung 15: QL-Farbkomposit mit 3 Informationsebenen laut Tab. 2

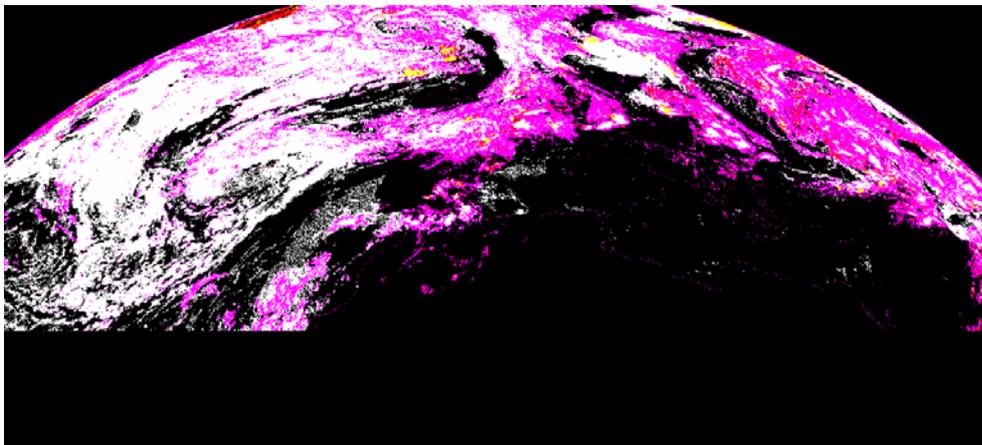


Abbildung 16: QL-Farbkomposit mit 3 Informationsebenen laut Tab. 2

Die Farbkanäle Rot, Grün und Blau werden additiv gemischt, d.h. Rot und Grün ergibt Gelb, Rot und Blau ergibt Magenta, Grün und Blau ergibt Cyan. Die Farben in den Abbildungen 15 und 16 bedeuten somit folgendes:

Abbildung 15:

Weiß bedeutet, dass das Wolkenstockwerk und der Wolkentyp erkannt wurden, bei gleichzeitiger größtmöglicher Wolkensicherheit. Gelb bedeutet, dass zwar das Wolkenstockwerk bei größter Wolkensicherheit erkannt wurde, jedoch kein Wolkentyp ermittelt werden konnte. Bei den rot eingefärbten Pixeln wird zwar ein Wolkenstockwerk angegeben, es ist jedoch nicht sehr sicher, ob es sich überhaupt um eine Wolke handelt. Für die magenta eingefärbten Pixel wird sowohl Wolkenstockwerk als auch Wolkentyp angegeben, jedoch ist die Klassifikation als Wolke unsicher.

Abbildung 16:

Weiß bedeutet, es liegen Wolkenstockwerk und optische Dicke vor bei größter Wolkensicherheit. Demgegenüber bieten die gelben Pixel zwar das Stockwerk und die optische Dicke, aber keine besondere Wolken-Klassifikationssicherheit. Die roten Pixel sind dieselben wie in Abb. 15. Die magentafarbenen Pixel bieten das Wolkenstockwerk bei größtmöglicher Wolkensicherheit, jedoch keine optische Dicke.

Dass keine cyanfarbenen Pixel in beiden Abbildungen auftreten, bedeutet, dass die jeweils anderen beiden Parameter nur berechnet werden, wenn ein Wolkenstockwerk angegeben wird.

Mit dieser Methode der kompakten Speicherung von Informationen als GIF-Datei werden regelmäßig zwei Qualitätskategorien bildlich dargestellt:

Die Kategorie „Verfügbarkeit“ wird durch die in Tab. 1 beschriebenen Informationseinheiten in Form eines Quicklooks angeboten. Ein zweites QL-GIF enthält folgende Informationen der Qualitätskategorie „Plausibilität“:

SZ	Sonnenzenitwinkel liegt im nutzbaren Bereich (Default $\mu_0 \geq 0.3$)
WTT	Wolkentyp und –stockwerk temperaturplausibel
WTS	Wolkentyp und –stockwerk strahlungsplausibel
ODS	Wolkenoptische Dicke spektral kontextkompatibel
ODK	Wolkenoptische Dicke klassifikationskompatibel
SRS	Schnee ist saisonal und regional plausibel
ZAP	Zusammenfassung anderer zweifelhafter Werte und Klassifikationen
SRA	Sonnenreflex Alternativpixel (z. B. evtl. doch Wolke)

Um die spezifischen Informationsinteressen der Nutzer noch gezielter beantworten zu können, stehen auf Anfrage zwei Mechanismen zur Informationsextraktion und weiteren Informationsgewinnung zur Verfügung:

- Aus den beiden QL-GIFs können bestimmte Informationskombinationen in Form vorgefertigter Tabellen (auch GIF-Farbtabelle) und Überlagerungen extrahiert werden.
- Wenn der Nutzer Informationen benötigt, die nicht im Standardumfang bereits angeboten werden, können diese durch Nachprozessierung bereits archivierter Szenen nachträglich verfügbar gemacht werden, sofern sich die Informationen aus den archivierten Primärdaten ableiten lassen.

Das Informationsangebot lässt sich also einerseits durch Nachprozessieren, d.h. durch Konfiguration eines Postprozessors für bestimmte Szenen erweitern. Andererseits können darüber hinaus erweiterte Informationsinteressen der Nutzer in den Prozessierungsablauf eingebunden werden und beim nächsten „major reprocessing“ in den Standard aufgenommen werden.

5 Weitere Realisierungsbeispiele

5.1 Standard-Quicklooks

5.1.1 Wolken

Die in Abschnitt 4.4 beschriebenen Informationen in Form kompakter GIF-Dateien lassen sich in beliebiger möglicher Kombination auf Farbkomposite überlagern, um die Wolkensituation schnell und übersichtlich darzustellen. Standardmäßig wird folgendes angeboten:

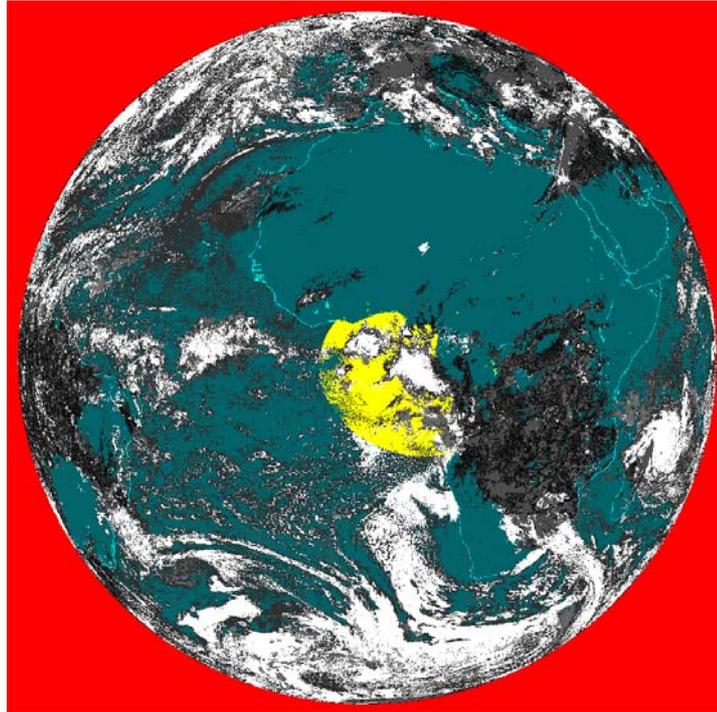


Abbildung 17: Standard-GIF-QL „Vollständigkeit“

Das Standard-GIF-QL „Vollständigkeit“ (Abb. 17) wird so dargestellt, dass Bad-Value-Pixel (Pixelwert 0) rot, nicht weiter klassifizierte Pixel (Pixelwert 1) dunkelgrün und der Pixelwert 111 (siehe Tab. 1) weiß dargestellt werden. Der Rest der Pixel wird ihrem Pixelwert entsprechend in mehr oder weniger hellem Grau dargestellt. Dieser Darstellung werden zur besseren Orientierung Küstenumrisse überlagert.

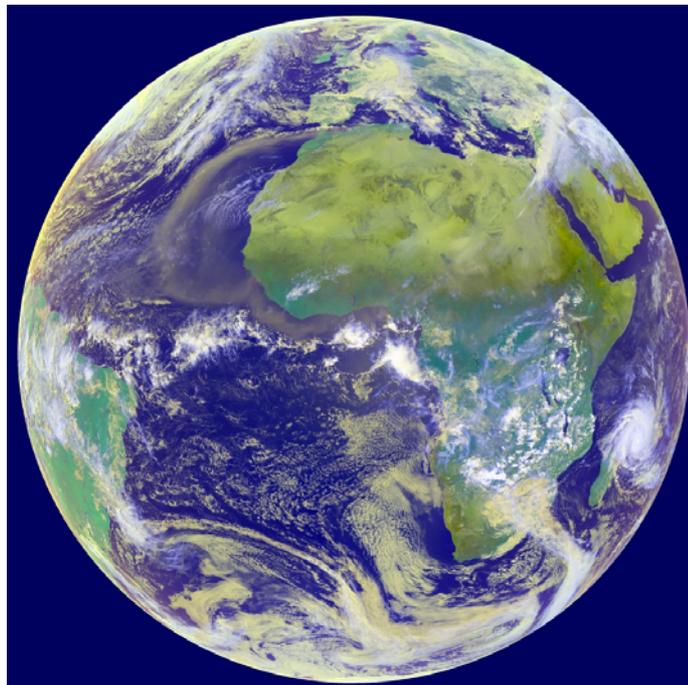


Abbildung 18: Farbkomposit zum GIF-QL Abb. 17 ohne Überlagerung

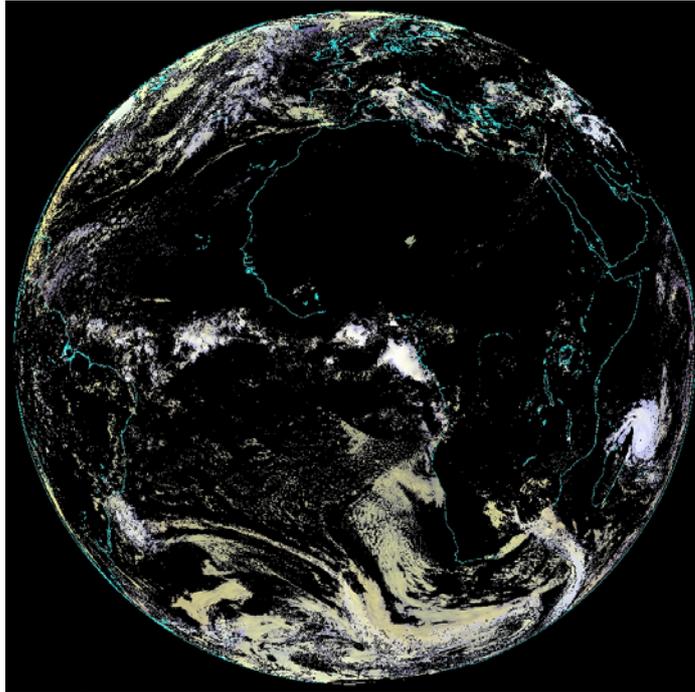


Abbildung 19: Überlagerung von Abb. 17 auf Abb. 18 durch Ausblenden

Abb. 19 ist das Farbkomposit aus Abb. 17 überlagert mit dem Pixelwert 111 aus dem Standard-GIF-QL „Vollständigkeit“. Die Überlagerung erfolgt durch Ausblenden, d.h. auf Null (Schwarz) Setzen aller Pixel, die nicht den Pixelwert 111 im GIF-QL haben. Wiederum sind Küstenumrisse überlagert. Auf diese Weise lassen sich diejenigen Wolken schnell identifizieren, die die größte Klassifikationssicherheit bieten und für die die Oberflächentemperatur, optische Dicke und Typklassifizierung vorliegen.

5.1.2 Schnee

- Tagesweise QLs der kumulativen oder gefilterten Schneebedeckung werden mit Hilfe der Generic Mapping Tools erzeugt. Ein solches GIF-Bild ist 480 kB groß (Abb. 20).

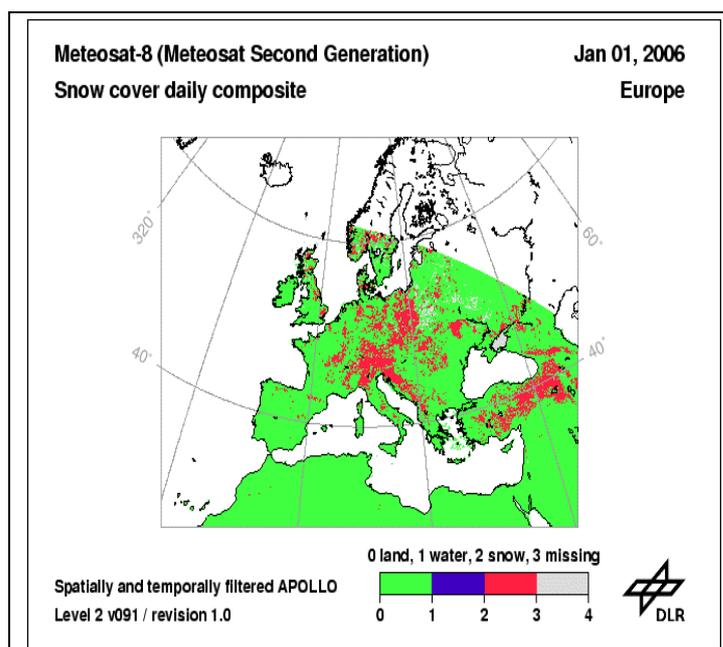


Abbildung 20: Beispiel für ein 1-Tages-Schnee-QL

- Saisonale Filme der Schneebedeckung können als animiertes GIF in ausreichender Auflösung aus den Einzelbildern erstellt werden. Dabei entsteht z. B. aus 120 Einzelbildern mit jeweils 480 kB eine Gesamtdatei von nur 2.4 MB bei kaum merkbarem Verlust an Informationsgehalt.

Verwendet wird die Konvertiersoftware gifsicle (<http://www.lcdf.org/gifsicle/>), frei verfügbar, unter Unix und Linux lauffähig und voll batchfähig.

Ein solches Beispiel befindet sich auf der WISENT-Webseite.

5.1.3 Wasserdampf

- Quicklooks der täglichen Wasserdampfverteilung werden ebenfalls mit Hilfe der Generic Mapping Tools erstellt.

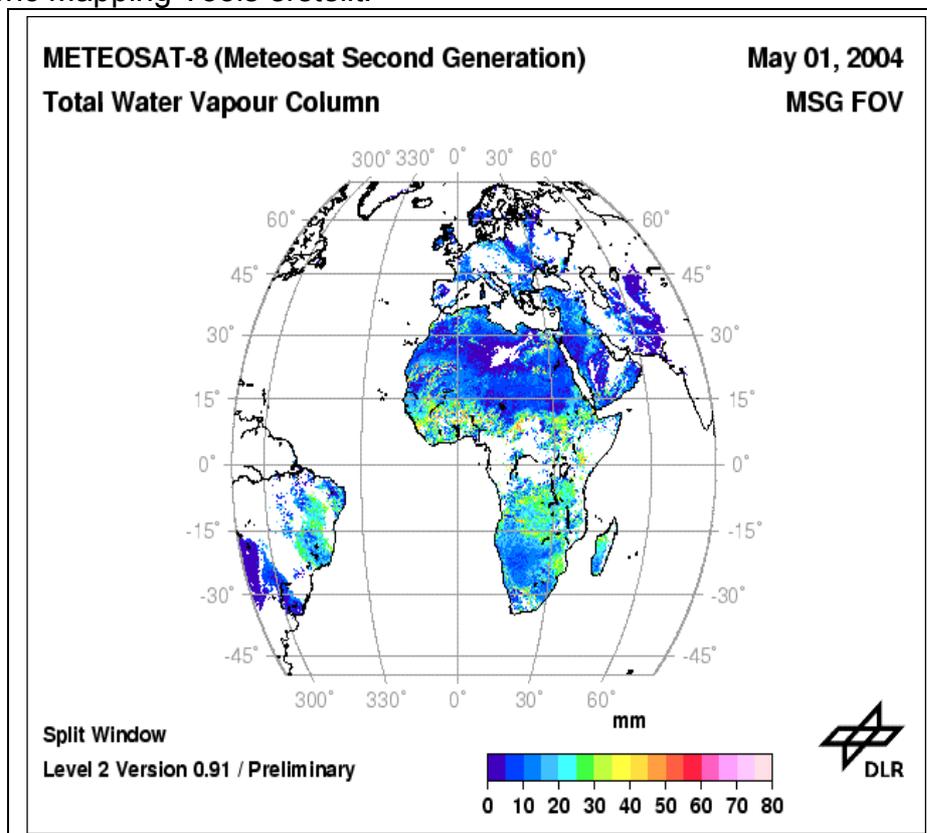


Abbildung 21: Beispiel für ein 1-Tages-Wasserdampf-QL

- Da sich die Art der Darstellung mit den Schnee-QL vergleichen lässt, wurde auch hier für den monatsweisen Film der Wasserdampfkonzentration die Erstellung eines animierten GIFs mit dem Tool gifsicle ausgewählt. Es entstehen Dateien mit einer Größe von 790 kB.

Ein solches Beispiel befindet sich auf der WISENT-Webseite.

6 Abkürzungen

CBR	Constant Bit Rate
DFD	Deutsches Fernerkundungsdatenzentrum (German Remote Sensing Data Centre)

DIMS	Data Information and Management System
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DMS	Data Management System
EOWEB	Earth Observation on the WEB (Web Gateway of ISIS and MUIS)
FTP	File Transfer Protocol
HDF	Hierarchical Data Format
IIF	Item Information File
JPEG	Joint Photographic Experts Group Datenformat
KML	Keyhole Markup Language
LAN	Local Area Network
MSG	Meteosat Second Generation Satellit
NFS	Network File System
PcR	Processing Request
PdR	Production Request
PL	Product Library
PS	Processing System
PSM	Processing System Management
PSNR	Peak Signal to Noise Ratio
QL	Quicklook
SEVIRI	Spinning Enhanced Visible and Infrared Imager
SSIM	Structural Similarity Index
TAR	Tape Archive
TIFF	Tagged Image File Format
viEM	Virtuelles Institut für Energiemeteorologie
WDC	Weltdatenzentrum für Fernerkundung der Atmosphäre
WGS	World Geographical System
WWW	World Wide Web
XML	Extended Markup Language

Erfahrungen mit visuellen Datenformaten

Arbeitspaket 4

Sönke Brummerloh
OFFIS – Institut für Informatik
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de>

1 Einleitung

In Arbeitspaket 4 (Effiziente Interaktion mit großen Datenobjekten mit Hilfe von Quicklooks) wurden animierte GIF-Bilder, Videos und Super Overlays eingesetzt, um das Transfervolumen von speicherintensiven Grafiken zu reduzieren. Die Ausgangsgrafiken lagen dabei in allen Fällen als JPEG-Dateien vor. Die gesammelten Erfahrungen werden im Folgenden zusammengefasst.

2 Videos

Das Ziel von Video-Codecs ist es, die Einzelbilder eines Videos in einer möglichst guten Bildqualität zu speichern und dabei den Speicherbedarf des fertigen Videos möglichst gering zu halten. Das macht moderne Video-Codecs zu einem attraktiven Werkzeug, um Bilderserien effizient zu komprimieren.

Zur Abschätzung der Leistungsfähigkeit wurden vier moderne weitverbreitete Codecs evaluiert. Dies waren zum einen die drei MPEG-4 kompatiblen Codecs FFmpeg MPEG-4, Xvid¹ und x264². Zum anderen wurde zum Erzeugen von Flash-Videos der Sorenson H.263 kompatible Codec der FFmpeg-Bibliothek³ genutzt. Die Wahl fiel auf diese Codecs, da die mit ihnen erzeugten Videos auf den meisten modernen Computern ohne die Installation eines neuen Codecs abgespielt werden können. Open-Source-Codecs wie Dirac⁴ oder Theora⁵ wurden nicht getestet, weil zum Abspielen zugehöriger Videos die Codecs in der Regel installiert werden müssten.

Um zu überprüfen, wie gut die vier Codecs Bilderserien codieren können, wurde eine beim DLR-DFD typische Bilderserie mit allen vier Codecs mit unterschiedlichen Parametern codiert. Von besonderem Interesse waren die Codierungsdauer sowie die Bildqualität im Verhältnis zur Dateigröße. Die Bildqualität wurde als „Peak Signal to Noise Ratio“ (PSNR) gemessen.

Als Codierungswerkzeug wurde das Kommandozeilen-basierte Werkzeug *mencoder* eingesetzt, das Teil der MPlayer⁶-Installation ist. Die MPlayer-Software steht unter der GNU General Public License⁷ (GPL) und ist auf vielen Betriebssystemen lauffähig. Auf Grund seiner vielen Konfigurationsmöglichkeiten und der relativ einfachen Handhabung wurde *mencoder* den beiden anderen unter Linux verbreiteten Codierungswerkzeugen FFmpeg⁸ und Transcode⁹ vorgezogen.

Die folgenden Ergebnisse gelten streng genommen nur für das verwendete Bildmaterial und die getesteten Software Versionen. Eine generelle Tendenz für die Codecs bzgl. der untersuchten Merkmale sollte sich aber verallgemeinern lassen.

2.1 Ergebnisse

Bei den Tests hat sich ergeben, dass der H.263-Codec und der FFmpeg MPEG-4-Codec am Performantesten sind. Beide benötigen nur etwa ein Viertel der Laufzeit, die der x264-Codec benötigt. Der Xvid-Codec benötigt etwa 70% der Laufzeit vom x264-Codec. Der

¹ <http://www.xvid.org>

² <http://www.videolan.org/developers/x264.html>

³ <http://ffmpeg.mplayerhq.hu/>

⁴ <http://dirac.sourceforge.net>

⁵ <http://www.theora.org>

⁶ <http://www.mplayerhq.hu>

⁷ <http://www.gnu.org/licenses/gpl.html>

⁸ <http://ffmpeg.mplayerhq.hu>

⁹ <http://www.transcoding.org/cgi-bin/transcode>

Platzbedarf bei einem PSNR von 30 dB verhält sich dagegen umgekehrt: Der x264-Codec erzeugt die kleinsten Video-Dateien. Gegenüber dem x264-Codec erzeugt Xvid um ca. 10%, FFmpeg MPEG-4 um ca. 15% und der H.263-Codec um ca. 20% größere Video-Dateien.

Als der leistungsfähigste Codec hat sich x264 herausgestellt. Er produziert bei niedrigen Bitraten die besten Ergebnisse. Bei höheren Bitraten produzierte bei den verwendeten Ausgangsbildern allerdings Xvid bei vergleichbarer Qualität die kleineren Video-Dateien. Bei dem verwendeten Bildmaterial waren solche hohen Bitraten nicht notwendig, um von der Bildqualität her sehr gute Ergebnisse zu erhalten.

Mit der Grundeinstellung ohne zusätzliche Parameter hat ein Codec im Vergleich zu Einstellungen mit weiteren Parametern eine sehr gute Laufzeit, erzeugt jedoch auch die größte Video-Datei. Die Größe der Video-Datei kann beim gleichen PSNR zum einen durch optimierende Parameter reduziert werden und zum anderen durch das Verwenden des twopass-Modus, bei dem das Ausgangsmaterial erst vollständig analysiert wird, bevor das Video generiert wird.

Wenn keine exakte Dateigröße benötigt wird, sind leicht optimierende Einstellungen zu empfehlen. Mit geeigneten Parametern ist es möglich, die Dateigröße bei gleicher Qualität zu reduzieren, ohne die Laufzeit deutlich zu erhöhen. Starke Optimierungen sind nur zu empfehlen, wenn die Laufzeit keine oder nur eine geringe Relevanz hat, da sich die Laufzeit durch einige Parameter deutlich erhöht. Eine Abschätzung der Auswirkungen von einigen Parametern auf die PSNR ist im MPlayer-Handbuch¹⁰ zu finden. Falls eine exakte Dateigröße benötigt wird, kann die twopass-Codierung genutzt werden. Dadurch wird die Laufzeit allerdings in etwa verdoppelt. Stark optimierende Einstellungen bewirken beim twopass-Modus gegenüber leicht optimierenden Einstellungen kaum eine Verbesserung. Eine qualitätsbasierte Codierung bringt bzgl. Speicherplatz und Laufzeit ähnliche Ergebnisse wie die Codierung mit konstanter Bitrate ohne optimierende Einstellungen und ist daher nicht zu empfehlen.

Wenn die Video-Dateien möglichst wenig Speicherplatz verbrauchen sollen, hat sich eine Reduktion der Auflösung als brauchbare Lösung herausgestellt: Die Laufzeit der Codierung sinkt dadurch erheblich und die erzeugten Video-Dateien benötigen deutlich weniger Speicherplatz. Der Nachteil ist, dass feine Details durch die geringere Auflösung wegfallen. Wenn dies akzeptabel ist, stellt eine Verringerung der Auflösung eine optimale Möglichkeit dar, um die Laufzeit zu verringern und gleichzeitig die Video-Dateigröße zu reduzieren.

Bei der eingesetzten Version (Mplayer-1.0rc2) von mencoder kam es zu einer leichten Farbverfälschung. Diese resultiert aus der automatischen Umwandlung der JPEG-Bilder in ein Motion-JPEG-Zwischenformat durch mencoder bzw. FFmpeg (dessen Bibliothek von mencoder genutzt wird).

Nach einer Begutachtung der Videoqualität durch menschliche Betrachter stellte sich heraus, dass nur x264 ab einer gewissen Bitrate feine Details und Farbschattierungen nicht mehr weichzeichnet. Bei den anderen Codecs trat ein Weichzeichnen selbst bei sehr hohen Bitraten auf. Das Weichzeichnen kann durch die Wahl geeigneter Quantizer reduziert werden. Als eine geeignete Einstellung wurde der MPEG-Quantizer identifiziert. Bei den Codecs FFmpeg-MPEG4 und Xvid kann dieser von der Standardeinstellung „H.263“ auf „MPEG“ geändert werden, um ein besseres Ergebnis zu erhalten. Die Laufzeit ändert sich durch diese Einstellung kaum. Der x264-Codec arbeitet bereits mit entsprechend guten Quantizer-Einstellungen, während beim H.263-Codec kein MPEG-

¹⁰ <http://www.mplayerhq.hu/DOCS/HTML/en/encoding-guide.html>

Quantizer gewählt werden kann. Zur feineren Abstimmung können manuell eine Reihe weiterer Quantizer-Parameter spezifiziert werden.

2.2 Fazit

Videos als Quicklooks sind dann eine gute Alternative zu Einzelbildern, wenn kleinere Fehler durch Farbverfälschungen und Weichzeichnen kein Problem darstellen. Andernfalls ist von Videos als Darstellungsform abzuraten.

Grundsätzlich müssen bei der Auswahl eines geeigneten Codecs mehrere Kriterien berücksichtigt werden. Es hat sich herausgestellt, dass die Codecs bei gleicher Video-Qualität eine unterschiedliche Laufzeit haben und unterschiedlich große Video-Dateien generieren. Außerdem muss zum Abspielen bei manchen Formaten mit größerer Wahrscheinlichkeit ein Codec nachinstalliert werden als bei anderen Formaten.

Auf Grund der weiten Verbreitung von Flash-Plugins kann davon ausgegangen werden, dass Flash-Videos am ehesten abgespielt werden können. Video-Dateien, die mit einem zum MPEG-4 Part 2 Standard kompatiblen Codec codiert sind, sollten ebenfalls auf einer Vielzahl von Systemen abspielbar sein. Das in den AVI-Dateien angegebene Videoformat sollte bei der Verwendung der MPEG-4 Part 2 kompatiblen Codecs allerdings auf „DX50“ gesetzt werden. Zum Abspielen wird dann der kompatible weitverbreitete Codec DivX 5 genutzt. Beim relativ neuen x264-Codec kann momentan davon ausgegangen werden, dass er auf vielen Systemen noch installiert werden muss.

FFmpeg MPEG-4 und H.263 erzeugen bei vergleichbarer Qualität durchweg größere Dateien als x264 und Xvid. Dafür benötigen FFmpeg MPEG-4 und H.263 bei Codierungen ohne optimierende Einstellungen nur ca. 25% der x264-Codec-Laufzeit. Je nachdem, ob Speicherplatz oder Codierungszeit der begrenzende Faktor ist, sollte daher zwischen x264 und Xvid oder FFmpeg MPEG-4 und H.263 gewählt werden.

Falls viel gleichartiges Bildmaterial in Videos umgewandelt werden soll, ist es empfehlenswert Videos testweise zu erstellen, um zu ermitteln wie gut die Codecs das Ausgangsmaterial codieren können und mit welchen Einstellungen der PSNR bzw. die Bildqualität ausreichend ist.

Da die Codecs weiterentwickelt und optimiert werden, ist es sinnvoll die getesteten Codecs in regelmäßigen Abständen erneut Tests zu unterziehen. Wie Tests des Computermagazins c't gezeigt haben (c't 10/2003, S. 146ff und c't 10/2005, S. 146ff) können mit neueren bzw. optimierten Codecs bessere Ergebnisse erwartet werden.

3 Animierte GIF-Bilder

Eine alternative Möglichkeit Bilderserien zu bündeln sind animierte GIF-Dateien. Diese haben sich gegenüber den getesteten Videoformaten nur in Ausnahmefällen als praktikabel erwiesen. Dies liegt zum einen daran, dass das Erzeugen von GIF-Dateien zu langen Bilderserien mit z. B. ImageMagick¹¹ sehr viel Arbeitsspeicher benötigt. Zum anderen sind GIF-Bilder in ihrer Darstellung auf 256 Farben beschränkt. Viele Bilddateien verwenden mehr als 256 Farben. Durch eine Reduktion der Farbpalette würde es daher zu einer schlechteren Bildqualität kommen. Außerdem führt die Farbreduktion zu Bildern, die sich schlecht komprimieren lassen. Das führt dazu, dass die animierten GIF-Bilder größer werden als Videos in besserer Qualität. Es kann sogar dazu kommen, dass die GIF-Bilder mehr Speicherplatz benötigen als die ursprünglichen JPEG-Bilder mit mehr Farben.

Falls die Ausgangsbilder allerdings mit 256 Farben auskommen, werden GIF-Bilder eine attraktive Alternative, die gegenüber z. B. JPEG-Bildern nur 15 bis 20% des

¹¹ <http://www.imagemagick.org>

Speicherplatzes benötigt. Da die GIF-Bilder in einem solchen Fall gegenüber komprimierten Videos in Originalqualität vorliegen, sind sie hier ausnahmsweise die bessere Wahl. Ein Gewinn an Speicherplatz durch ein animiertes GIF-Bild gegenüber einzelnen GIF-Bildern ist vorhanden aber gering.

4 Super Overlays

Als eine mögliche neue Darstellungsform von Quicklooks werden im Folgenden Super-Overlays in Google Earth vorgestellt. Google Earth ist eine von Google für Linux, MacOS X und Microsoft Windows frei verfügbare Anwendung zum Betrachten der Erdoberfläche. In Google Earth wird zunächst die Erde in einer Weltraumansicht dargestellt. In dieser Ansicht lässt sich der Globus frei drehen und zoomen. Dabei ist es möglich, nahezu stufenlos bis auf eine Auflösung von wenigen Metern zu zoomen.

Die Karte von Google Earth kann mit Grafiken überlagert werden. Dazu werden XML-basierte KML-Dateien¹² in Google Earth geladen. Zusätzliche Grafiken, die über die Erdoberfläche gelegt werden, werden als *Ground Overlays* bzw. allgemein als *Overlays* bezeichnet. In einer KML-Datei können mehrere *Region*-Elemente mit *Overlays* definiert werden. Ein *Region*-Element umfasst eine Fläche auf der Google Earth Karte. Für jede *Region* berechnet Google Earth die Fläche, die diese in der aktuellen Ansicht einnehmen würde. Dabei kann für jede *Region* festgelegt werden, ab wie vielen minimal geladenen Bildpunkten sie aktiviert bzw. dargestellt wird und ab wie viel maximal geladenen Bildpunkten die *Region* wieder deaktiviert bzw. ausgeblendet wird. So ist es möglich, bestimmte Grafiken erst ab einer bestimmten Zoom-Stufe anzuzeigen und sie ab einer bestimmten Zoom-Stufe auch wieder auszublenden. Auf diese Weise kann in einer *Region* auf weitere KML-Dateien verwiesen werden, mit dem Ergebnis, dass diese erst beim Aktivieren der *Region* nachgeladen werden. So ist es z. B. möglich bei geringer Zoom-Stufe eine grob aufgelöste Grafik darzustellen. Sobald der Anwender weiter hineinzoomt, werden weitere KML-Dateien mit detaillierteren Grafiken geladen¹³. Dadurch können Hierarchien von *Region*-Elementen angelegt werden, um schrittweise immer detailliertere Ausschnitte einer großen Grafik zu laden. Eine solche Hierarchie wird als *Super-Overlay* bezeichnet.

Der Vorteil solcher *Super-Overlays* ist, dass nur geladen wird, was der Nutzer sich genauer anschauen möchte. Dadurch sind die Ladezeiten für den Nutzer und das generelle Datenaufkommen geringer. Da eine Grafik effektiv in mehreren Auflösungen auf dem Server hinterlegt ist, wird auf diesem mehr Speicherplatz benötigt.

Für alle Bilder, die in *Super-Overlays* umgewandelt werden sollen, muss deren genaue Lage auf der Erde bekannt sein. Falls die Lage nicht bekannt ist, muss sie zunächst manuell in Google Earth ermittelt werden. Satellitenaufnahmen haben darüber hinaus das Problem, dass sie in der Regel in einer Projektion vorliegen, die nicht der in Google Earth verwendeten WGS84 entspricht. Satellitenaufnahmen müssen daher zunächst in eine verwendbare Projektion konvertiert werden. Dies hat gegebenenfalls manuell zu geschehen.

Durch diesen hohen manuellen Aufwand ist der Einsatz von *Super-Overlays* nur praktikabel, wenn es sich um regelmäßig erstellte Bilder handelt, deren Lage und Projektion nur einmal ermittelt werden muss und die sich über die Zeit nicht ändert. Nachdem die notwendige Projektion ermittelt wurde, kann die Konvertierung der Projektion und das Anlegen der *Super-Overlays* in Zukunft automatisch erfolgen. Zur Automatisierung der *Super-Overlay*-Erstellung wurde eine eigene Open-Source

¹² <http://code.google.com/apis/kml/documentation/>

¹³ http://code.google.com/apis/kml/documentation/kml_21tutorial.html#workingregions

Anwendung¹⁴ implementiert, die über die Kommandozeile oder Skripte angesteuert werden kann. Diese ist auch in der Lage Bilderserien mit Zeitbezügen in ein Super-Overlay umzuwandeln.

Zusammenfassung:

- Der visuelle Eindruck der Darstellung in Google Earth ist sehr gut.
- Die zu ladende Datenmenge lässt sich über Super Overlays gut einschränken, da nur geladen wird, was sich der Nutzer genauer anschaut.
- Die Erzeugung von Super Overlays kann allerdings recht arbeitsintensiv sein, wenn mangels geeigneter Metadaten für jedes Super Overlay die exakten Koordinaten der Grafik in Google Earth interaktiv ermittelt werden müssen.
- Alle Satellitenbilddaten müssen vor der Darstellung in Google Earth in die Standardprojektion von Google Earth (WGS84) umgerechnet werden.

5 Fazit

Alle Komprimierungsansätze haben ihre Berechtigung: Bilderserien mit über 256 Farben lassen sich gut über Videos komprimieren. Für Bilderserien mit bis zu 256 Farben können auch gut animierte GIF-Dateien verwendet werden. Wenn Grafiken mit hoher Ausdehnung visualisiert werden sollen, deren genaue Position bekannt ist und diese in einer für Google Earth geeigneten Projektion vorliegen, können sie gut als Super Overlays dargestellt werden.

Zu Beachten ist, dass jeder dieser Ansätze rechenintensiv ist. Besonders wenn viele Bilderserien prozessiert werden sollen, sollte ausreichend Rechenkapazität und Zeit zur Verfügung stehen.

¹⁴ <http://sourceforge.net/projects/superoverlaygen/>



Strahlungstransportrechnung – Erfahrungsbericht Condor

Arbeitspaket 5.1

Sina Lohmann, Luca Bugliaro, Tobias Zinner, Winfried Beer, Robert Buras
DLR Oberpfaffenhofen
Institut für Physik der Atmosphäre
82234 Weßling

Internet: <http://www.dlr.de/pa/>

Jan Ploski
OFFIS – Institut für Informatik
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de/>

1 Motivation

1.1 Strahlungstransportrechnungen bei DLR-IPA

In der Abteilung für Fernerkundung der Atmosphäre bei DLR-PA werden unterschiedliche Aspekte der Wolken-Strahlung-Wechselwirkungen untersucht. Einer davon, der eine wichtige Rolle in vIEM spielt, ist die Entwicklung, Bewertung und Validierung von Verfahren zur Bestimmung von Solarstrahlung am Erdboden aus Satellitenbeobachtungen. Wolken weisen nämlich die höchste zeitliche und räumliche Variabilität auf und haben den größten Einfluss auf die Solarstrahlung am Boden. Dazu werden detaillierte Strahlungstransfersimulationen mithilfe des libRadtran (library for radiative transfer [Mayer und Kylling 2005]) Softwarepakets durchgeführt. Wolken- und Atmosphärendaten des LM (Lokalmodell) Vorhersagemodells des Deutschen Wetterdienstes (DWD) und Zusatzdaten zur Beschreibung des Bodenreflexionsvermögens werden als Input dafür benutzt, um Satellitenmessungen von MSG/SEVIRI und einfallende Solarstrahlung am Boden zu simulieren. Nach Anwendung der Solarstrahlungsalgorithmen auf diese berechneten Satellitenmessungen können die Resultate mit den ebenfalls berechneten Solarstrahlungsdaten verglichen werden, um die Genauigkeit der Retrievalverfahren zu bestimmen. Außerdem hat man dann die Möglichkeit, die Fälle im Detail zu untersuchen, welche erhebliche Abweichungen zeigen, weil alle Inputfaktoren, d.h. alle atmosphärischen Parameter, bekannt sind. Damit können diese Algorithmen verbessert werden.

Zur Berechnung der Satellitenbeobachtungen sowie der einfallenden Solarstrahlung am Boden für eine gewisse Region gibt es grundsätzlich zwei Möglichkeiten. Üblicherweise wird die Gesamtsimulation in einzelne Atmosphärensäulen zerlegt und angenommen, dass diese horizontal homogen und unabhängig voneinander sind. Unter diesen Annahmen können sogenannte eindimensionale (1D) Strahlungstransferrechnungen durchgeführt werden. Auf libRadtran aufbauend ist zu diesem Zweck bei DLR-PA das SIMSAT (Satellite Simulator) Modul geschrieben worden. Wenn hingegen keine Vereinfachungen gemacht werden und Photonen auf ihrem Weg durch die Atmosphäre genau verfolgt werden, sind diese sogenannten dreidimensionalen (3D) Strahlungstransferrechnungen aufwändiger und werden mit dem MYSTIC (Monte Carlo code for the physically correct tracing of photons in cloudy atmospheres [Mayer 2000]) Modul, auch auf libRadtran basierend, durchgeführt.

Wenn Satellitenbeobachtungen für ausgedehnte Gebiete simuliert werden, sind schon 1D Rechnungen sehr rechenzeitintensiv. Bisher wurden solche Arbeiten entweder auf einzelnen Standard-Arbeitsplatzrechnern unter Linux/UNIX oder auf einem 32 x 2 GHz PC-Cluster durchgeführt. Für einzelne Anwendungen wurden auch die gerade außerhalb der täglichen Kernarbeitszeiten wenig genutzten Arbeitsplatzrechner für verteiltes Rechnen genutzt. Allerdings wurde dies bisher durch personalintensives einzelnes Ansprechen der Rechner vorgenommen, eine automatische technische Lösung bestand nicht. Deswegen wurde im Rahmen von WISENT die Vernetzung der einzelnen Knoten des Linux-Clusters sowie die Vernetzung von 30 Arbeitsplatz-PCs mit Hilfe der Condor-Software vorgenommen.

1.2 Strahlungstransportrechnungen bei DLR-TT

Einer der Schwerpunkte bei der Verarbeitung energiemeteorologischer Daten bei DLR-TT ist die Prozessierung von Fernerkundungsdaten, um Karten der solaren Einstrahlung zu erstellen. Die Rohdaten werden im DIMS des DFD geholt und bei DLR-TT verarbeitet. Dabei geht es oft auch um große Regionen, wie z.B. den Mittelmeerraum. Auf einem einzelnen Rechner würde die Verarbeitung der Daten mehrere Jahre dauern. Ein Vorteil der genutzten Daten ist, dass sie sich gut in Teilrechnungen aufteilen lassen. Eine

Gesamtrechnung lässt sich in unabhängige Regionen oder Zeitabschnitte (z.B. eine 10-Jahresrechnung in die einzelnen Jahre oder Monate) aufteilen. Eine manuelle Parallelsierung in mehrere Läufe von Teilbildern oder Teilzeiträumen ist personell aufwändig und fehleranfällig.

Die oben genannten Anforderungen lassen sich durch die Durchführung von vielen unabhängigen, nebenläufigen Simulationen erfüllen, die mit verschiedenen Randbedingungen und Parametervariationen der Ausgangskonfiguration angestoßen werden. Exemplarisch wurde der Strahlungsprozessor als zu parallelisierende Komponente ausgewählt. Die entwickelte Lösung lässt sich aber auch leicht auf andere rechenintensive Komponenten übertragen.

2 Condor Installation

2.1 Condor Installation und MYSTIC

Die Installation des Condor-Paketes mit der vorbereiteten Anleitung von OFFIS bereitete keine Probleme. Testläufe mit einfachen kleinen Beispielprogrammen, die im Paket enthalten sind, verliefen ebenfalls ohne Störungen.

Zum Testen der Berechnung von rechenzeitintensiveren Anwendungen wurden 25 Jobs mit dreidimensionalen Strahlungstransferrechnungen vorbereitet, wobei diese sich in einigen atmosphärischen Parametern unterschieden, um die Berechnung eines realistischen Wolkenfeldes zu simulieren. Alle Jobs wurden auf einer Maschine mit 3 GHz CPU bereits im Vorfeld berechnet. Die Dauer der Berechnungen variierte zwischen 4 Stunden und 85 Stunden.

Die Jobs wurden zunächst unter Verwendung des vanilla Universums an das Condor-Netzwerk übergeben. Die Verteilung der Jobs auf die vorhandenen Knoten lief problemlos. Folgende Beobachtungen konnten während der Tests gemacht werden:

- Bei Simulation von mehreren Knoten unter Verwendung von mehreren Rechenkernen auf Servern wird der Gesamtspeicher auf die entstehenden virtual machines aufgeteilt. Jede virtual machine erscheint als unabhängiger Knoten mit einer begrenzten Kapazität an Speicher im Condor-Netzwerk. Dies kann dazu führen, dass bei Angabe von einem Mindestspeicher in den Condor-requirements der betreffende Rechner nicht mehr verwendet wird, obwohl er eigentlich verfügbar wäre und auch entsprechende Voraussetzungen bzgl. CPU und Speicher hätte. Dieses Problem wurde gelöst, indem nur eine begrenzte Anzahl an virtual machines pro Rechner in das Condor-Netzwerk eingetragen wurden.
- Die Definition von „idle“ innerhalb Condors erwies sich während der Testläufe als zu weit gefasst, da Mausbewegungen beispielsweise beim Öffnen eines Browsers die Condor-Jobs nicht stoppen. Erst durch Tastatureingaben oder ssh-Zugriff werden Jobs angehalten. Dies konnte dazu führen, dass ein Rechner auf die Eingaben des Users sehr schwerfällig reagierte.
- Startet ein User sehr viele Jobs, ist das gesamte Netzwerk solange für andere User blockiert, bis diese Jobs abgearbeitet sind. Es können zwar über user priorities einzelne User vorrangig behandelt werden, eigentlich sollte dies aber nicht der Fall sein. Des Weiteren können über die Angabe von Rank-Einstellungen vorgenommen werden, wie genau Jobs zu verteilen sind. Nachteil ist aber, dass dies vom User im

Command-file getan werden muss. Der User muss also einerseits wissen, wie er die Einstellungen am besten konfiguriert und andererseits auch bereit sein, seinen eigenen Job nicht mit maximaler Ausdehnung im Netz laufen zu lassen. Wünschenswert wären Einstellungen, die vom Administrator bei der Einrichtung des Netzwerkes vorgenommen werden können, z.B. maximale Anzahl von Jobs pro User.

- Jobs auf Rechnern, die als Arbeitsplatzrechner dienen, wurden oft in kurzen Abständen suspended und wieder unsuspending. Zum Teil wurden die Jobs schließlich nach mehreren Stunden abgebrochen und neu verteilt, wobei sie jedoch häufig wieder auf den Rechner geschoben wurden, auf dem sie gerade abgebrochen worden waren. Die Berechnung dieser Jobs dauerte dementsprechend sehr lange. Um die Verteilung der Jobs zu verbessern und auch dem suspend/unsuspend vorzubeugen, wurde die ContinueldleTime auf 5 Minuten hochgesetzt und ein rank=kflops in die command-Datei eingefügt, um bei der Neuverteilung schnelle Rechner zu bevorzugen.

Um bei Benutzung des DLR-PA-Netzwerkes zu gewährleisten, dass nach stundenlanger Bearbeitung eines Jobs nicht alle Ergebnisse verloren gehen, wenn ein User an seinen Arbeitsplatzrechner zurückkehrt, wurde die Verwendung des standard Universums und damit die Möglichkeit des Checkpointing in Betracht gezogen. Kleinere C-Programme konnten unter Verwendung von condor_compile kompiliert und verlinkt werden. Für das Gesamtpaket von libRadtran erwies sich dies jedoch als äußerst schwierig. Um überhaupt einen fehlerfreien Kompilierungsvorgang mit condor_compile durchführen zu können, mussten folgende Anpassungen in der libRadtran-library vorgenommen werden:

- Nach Ausführen des configure-Skriptes mussten in Makeconf von libRadtran folgende Einträge geändert bzw. hinzugefügt werden:
 - FC = condor_compile /usr/bin/g77
 - LDF77 = condor_compile /usr/bin/g77
 - CC = condor_compile /usr/bin/gcc
 - CC_OHNE_CONDOR = /usr/bin/gccDies wird benötigt, um dependencies zu erstellen.
- In der libRadtran library im Verzeichnis tools und src_c mussten in den Makefiles in den entsprechenden Zeilen zur Erstellung der dependencies die Angabe CC durch CC_OHNE_CONDOR ersetzt werden.
- LibRadtran enthält eine Funktion Namens set_seed. Leider kommt eine Funktion mit demselben Namen offenbar auch in Condor vor, was wiederum zu Fehlermeldungen bei der Kompilation und Verlinkung führte. In drei Teilprogrammen (src_c/randlcg.c, src_c/randlcg.h, src_c/r250.c) von libRadtran musste daher die Funktion set_seed umbenannt werden.

Auch nach Durchführung der genannten Veränderungen war es nicht problemlos möglich, die libRadtran library durch den Befehl make zu kompilieren. Als Hürde erwies sich die Einbeziehung der netcdf library. Der Pfad für diese kann zwar bei Ausführen des configure-Skriptes explizit angegeben werden, so dass sich libRadtran unter normalen Umständen problemlos kompilieren lässt. Unter Einbeziehung von Condor führte dies jedoch nicht zum Erfolg. Eine vollständige Kompilation konnte nur erreicht werden, wenn die netcdf-library im System statisch verlinkt war, aber auch dann nicht auf jedem System.

So enthalten SuSE-Versionen ab 10.x zwar netcdf, führen aber trotzdem zu Fehlermeldungen und Abbruch der Kompilation. Funktionierte die Ausführung des make-Befehls, war die Dauer des Kompilationsvorganges sehr stark erhöht. Gegenüber Kompilation ohne Einbeziehung von Condor verlängerte sich diese mindestens um einen Faktor 5.

Noch ein Faktor musste berücksichtigt werden: wenn libRadtran mit Condor kompiliert wird und im standard universe arbeitet, hat MYSTIC keinen Zugriff auf den Rechner, wo der Job tatsächlich läuft, weil die Umgebung des Rechners simuliert wird, wo der Job eingereicht wurde. Das macht die Abfrage von host id und ähnlichem unmöglich und hat aber einen Einfluss auf die Erzeugung der Zufallszahl, die zum Initialisieren des Monte Carlo Prozesses gebraucht wird. Fangen zwei MYSTIC-Rechnungen mit derselben Zufallszahl an, sind die Resultate identisch. Das Problem kann man umgehen, indem man diese Zufallszahl MYSTIC explizit übergibt.

Nach obigen Veränderungen konnte bei DLR-PA die libRadtran library zwar auf einigen Rechnern mit condor_compile kompiliert werden, die Bearbeitung von Jobs im standard Universum schlug jedoch noch immer fehl. Bei Beauftragung des Condor-Netzwerkes wurden die Jobs zwar an die Knoten verteilt, jedoch unmittelbar darauf sofort abgebrochen und an einen anderen Rechner vergeben, auf diesem wiederum sofort abgebrochen etc. Dies gilt auch für Rechner mit derselben Konfiguration, unter der die Kompilation erfolgreich war.

Auf dem DLR-PA-Cluster war die Kompilation von libRadtran nach den oben genannten Veränderungen problemlos möglich. Auch die Bearbeitung von Jobs unter Verwendung des standard Universums funktionierte auf diesem System. Im Abstand von 3 Stunden wurden Checkpoints angelegt und bei Bedarf der Job auf einen anderen Knoten verlagert.

Aufgrund der Schwierigkeiten bei der Kompilierung steht zu vermuten, dass die Probleme, das standard Universum bei DLR-PA zu verwenden, mit den unterschiedlichen Versionen des Betriebssystems und den damit verbundenen Differenzen in der library-Struktur der einzelnen Knoten zusammenhängen. Des Weiteren wird bei DLR-PA nicht die aktuellste Version der Condor-Software verwendet, da diese sich nicht auf allen Knoten installieren lässt. Es steht zu vermuten, dass die Probleme mit dem standard Universum durch ein update von Condor zumindest eingegrenzt werden können. Dies ist jedoch derzeit bei DLR-PA aus den genannten Gründen nicht möglich.

Nachtrag: Mittlerweile sind die Probleme bezüglich der Verwendung des standard universe größtenteils behoben, sowohl auf dem DLR-PA-Cluster wie auch auf dem DLR-PA-Netzwerk. Erstens wurde ein Update der Condor-libraries durchgeführt. Die condor-kompilierte Version von libRadtran kann jetzt mit Hilfe von CCC zum Linken (ausschließlich) durchgeführt werden, während alle anderen Aufrufe des C-Compilers mit dem üblichen CC erfolgen. Dabei steht CCC für die Condor-kompilierte Version von gcc und CC für gcc:

- CCC = condor_compile /usr/bin/gcc
- CC = /usr/bin/gcc

Dieser Vorgang ist automatisiert worden, so dass bei der Installation des libRadtran Strahlungspaketes lediglich "configure" mit der Option "--with-condor" vor "make"

ausgeführt werden muss, wenn eine Condor-compilierte Version von libRadtran erwünscht ist. Schließlich bildet das Initialisieren der Zufallszahlen für MYSTIC keine Probleme mehr, da der Initialisierungswert (seed) als die Summe von process id und Zeit in Sekunden gesetzt wird, und ersteres im Falle eines Jobs im standard Universum mit der Prozessnummer gleichgesetzt wird. Somit werden zwei Läufe nur im (unwahrscheinlichen) Fall gleiche Ergebnisse liefern, wenn z.B. Prozess 9 eine Sekunde vor Prozess 8 den Initialisierungswert festlegt. Um aber auch dieses Problem zu beseitigen, kann die Festlegung der seeds vom Nutzer selbst vorgenommen werden (s.u.).

2.2 Condor Installation bei DLR-TT

Die Installation einer aktuellen Condor-Version verlief auf den von DLR-TT verwendeten Rechnern problemlos, obwohl auch hier verschiedene Linuxversionen zum Einsatz kommen. Erste Tests auf einer Maschine mit dem in AP 5.4 erstellten Jobgeneratoren waren direkt erfolgreich. Probleme traten nur beim start der Jobs auf der zweiten Maschine auf, so dass hier die ausführenden Programme nicht im NFS zu finden waren. Eine Analyse der Logdateien ergab, dass die Jobs auf der zweiten Maschine als user „nobody“ gestartet worden, der natürlich nicht in den „/home/“ Verzeichnissen lesen und ausführen durfte. Es war hier eine Anpassung der Parameter „UID_DOMAIN“ und „FILESYSTEM_DOMAIN“. Die richtige Konfiguration dieser Parameter war aus der Condordokumentation für den hiesigen Administrator nicht klar genug ersichtlich. Klarere Beispiele für die Setzung der Parameter hätten wahrscheinlich geholfen. Nach einigem Versuchen konnte die richtige Setzung gefunden werden und seit dem läuft Condor problemlos und wird routinemäßig bei DLR-TT für die Berechnung von Strahlungskarten und dem Wolkeinindex eingesetzt.

3 COCOS

Um die Vorbereitung der parallelen Verarbeitung von MYSTIC-Jobs zu erleichtern, wurde das COCOS-Script erstellt. Das COndor-uvspec-COmbination-Script ist ein shell script, das dazu dient, große Aufgaben in kleinere Teilaufgaben geeignet zu zerteilen, automatisch auf mehrere Rechner zu verteilen und abschließend die Ergebnisse korrekt zusammenzufassen. Dabei werden automatisch große Wolkenzenen in Ausschnitte unterteilt und berechnet, große Photonenanzahlen in kleinere Pakete aufgeteilt oder Verzeichnisse von Eingabedateien abgearbeitet. Der Benutzer erstellt dazu eine Eingabedatei für MYSTIC, an die lediglich einige Anweisungen für Condor angefügt werden. Die Zerteilung und Verteilung auf mehrere Rechner und gegebenenfalls das Zusammensetzen der Ergebnisse erfolgt automatisch. Natürlich werden auch alle Inputdateien (z.B. Albedodaten) entsprechend behandelt. Dabei werden tmp-Verzeichnisse angelegt und am Ende wieder gelöscht. Besonders bei der Zerteilung von Wolken ist aber immer noch vom Nutzer darauf zu achten, dass genug Speicherplatz vorhanden ist, um auch die tmp-Dateien und die Ergebnisse zu lagern.

COCOS erhält als Argument die Eingabedatei, die alle MYSTIC- und Condor-Anweisungen enthält. Neben den gewohnten Spezifikationen für das Strahlungstransportmodell werden zusätzlich die entsprechenden Einstellungen spezifiziert, nach denen mit Hilfe von Condor verteilt gerechnet werden soll. Alle diese Anweisungen beginnen mit „#condor_“, so dass bei herkömmlicher unverteilter Rechnungen (ohne COCOS) diese Zeilen als Kommentar gelesen werden. Insbesondere sind Angaben zur Größe der Condor-Jobs möglich (Größe des Ausschnitts, Überlapp, Anzahl Photonen, usw.), mit welchen Parametern Condor arbeiten soll (d.h. requirements, rank, universe usw.) und in welchem Verzeichnis COCOS die temporären Dateien für

Wolken, Albedo, libRadtran- Inputfiles usw. angelegt werden sollen. COCOS übernimmt auch die Aufgabe, eindeutige Initialisierungswerte (seeds) des Zufallsgenerators für das Monte Carlo- Modell MYSTIC zu übergeben, um zu verhindern, dass zwei Teil-Jobs redundante Resultate liefern (z.B. bei der Berechnung von Photonenpaketen für sonst gleichen Input).

4 SIMSAT

Als zusätzliche Anwendung wurde das SIMSAT Paket zur Simulation von Satellitenstrahldichten mit eindimensionalen Strahlungstransferlösern mit den Condor-Funktionalitäten erweitert. Der technische Hauptunterschied zu den oben erwähnten dreidimensionalen Rechnungen besteht darin, dass SIMSAT kein C-Programm ist, sondern ein C-Shell-Skript, das unter Linux läuft und unterschiedliche C-Programme und weitere Shell-Skripte aufruft. Unter den C-Programmen befindet sich natürlich auch libRadtran zur Lösung der Strahlungstransportgleichung. Besondere Aufmerksamkeit musste hier den temporären Dateien gewidmet werden, die von SIMSAT während eines Laufes erzeugt werden und Gefahr laufen, von konkurrierenden SIMSAT-Läufen überschrieben zu werden. Mit einer sauberen Namensgebung und den mit MYSTIC gesammelten Erfahrungen konnte SIMSAT mit Condor im vanilla Universum auf den DLR-PA-Arbeitsplatzrechnern und auf dem DLR-PA-Cluster benutzt werden. Unterschiedliche Tests wurden dabei erfolgreich durchgeführt.

Um das verteilte Rechnen mit SIMSAT komfortabel zu gestalten wurde es hier aus dem COCOS-Prinzip ausgegangen, wonach Condor-spezifische Anweisungen mit #condor_ anfangen und in einem normalen SIMSAT-Inputfile eingebettet werden sollen. Die Zerteilung der Inputdateien erfolgt dann ähnlich wie in COCOS automatisch, wie auch das Verteilen der einzelnen Jobs auf die jeweiligen Rechner und das Zusammenstellen der Ergebnisse. Allerdings tauchen die drei Hauptschritte (Input erzeugen, rechnen, Output sammeln) als einzelne Condor-Jobs innerhalb eines Condor DAG (Directed Acyclic Graph) und werden von DAGMan verwaltet.

Schließlich wurde die Nutzung des von OFFIS betreuten Linux-Clusters mit Condor-GT4 angestrebt und realisiert. Nach erfolgreicher Beantragung eines Grid-Zertifikats erwies sich die DLR-Firewall als das Hauptproblem. Mit Hilfe von OFFIS konnten diese Schwierigkeiten beseitigt werden, indem

- ein Rechner außerhalb der eigentlich Firewall in einer so genannten demilitarisierten Zone (DMZ) als „submission host“ eingesetzt wurde.
- GridFTP die erlaubten Ports für den benötigten Datentransfer explizit mitgeteilt wurden.
- die genaue Identifikation des DMZ-Rechners seitens des OFFIS-Clusters sichergestellt wurde.
- die Hostnamen für die Dämonen des Globus Toolkit näher spezifiziert und Dummy-Netzwerkinterfaces eingerichtet wurden, um Probleme mit der IP-Address-Substitution durch die externe Firewall zu beheben.

Die für das SIMSAT-Paket notwendigen Veränderungen lassen sich wie folgt zusammenfassen:

- Der „submission host“ und die CPUs, die die Strahlungstransfer-rechnungen durchführen, verfügen nicht über einen gemeinsamen Speicher, auf den sie zugreifen könnten. Deswegen musste auf der einen Seite gewährleistet werden, dass alle nötigen Files dem OFFIS-Cluster zur Verfügung stehen; und auf der anderen Seite, dass die unterschiedlichen Verzeichnisstrukturen auf dem DMZ- und den OFFIS-Rechnern berücksichtigt wurden.
- SIMSAT wurde so angepasst, dass Grid-Ressourcen wie das oben erwähnte GridFTP-Client zum Einsatz kommen, wenn Input- und Outputdaten von und zu dem einreichenden DMZ-Rechner transferiert werden.
- Wegen der kurzen Rechenzeit eines einzelnen Jobs und der gleichzeitig hohen Anzahl solcher Jobs, erwies sich als sehr nützlich, das von OFFIS entwickelte Perl-Modul Multijob zu benutzen, das mehrere Jobs bündelt und sie auf eine beschränkte Anzahl Prozessoren verteilt. Damit konnte erreicht werden, dass die NFS-Last auf dem OFFIS-Cluster begrenzt blieb.

Das so modifizierte SIMSAT-Paket wurde zusammen mit libRadtran letztendlich „per Hand“ auf dem OFFIS-Cluster installiert und dort mittels Condor-GT4 vom „submission host“ angesprochen.

5 Resultate und Nachhaltigkeit

Insgesamt erwies sich das Condor-Paket bei DLR-PA zwar als lauffähig, hat aber gerade bei der Verwendung auf den sehr heterogenen Arbeitsplatzrechnern noch deutliche Schwächen. Auf dem DLR-PA-Cluster erwies sich die Condor-Software als sehr geeignet zur Bearbeitung von Jobs. Dort hat sie daher die bisher verwendete Software abgelöst.

Aufgrund der erfolgreichen Erfahrungen mit Condor auf dem DLR-PA-Linux-Cluster besteht seitens DLR-PA großes Interesse, Strahlungstransportrechnungen auch auf D-GRID-Rechenressourcen durchführen zu können. Obwohl sich dies bedingt durch die DLR-Firewall als aufwendig erwies, erlaubt es für viele Jobs die Rechenzeit erheblich zu verkürzen und stellt momentan die einzige Möglichkeit dar, aufwendige dreidimensionale Strahlungstransportrechnungen für ausgedehnte Gebiete und in hoher räumlicher Auflösung durchzuführen. Für die Zukunft ist DLR-PA daher sehr interessiert, weiterhin Zugriff auf D-GRID-Rechenressourcen zu haben. Damit wäre ein nachhaltiger Einsatz der während WISENT durchgeführten Entwicklungen bei DLR-PA garantiert.

Referenzen

[Mayer 2000] Mayer, B.: I3RC phase 1/2 results from the MYSTIC Monte Carlo model. Abstract for the I3RC (Intercomparison of 3D radiation codes) workshops, Tucson, Arizona, November 17-19, 1999 and November 15-17, 2000. University of Arizona Press, ISBN 0-9709609-0-5.

[Mayer und Kylling 2005] Mayer, B. und Kylling, A.: Technical note: The libRadtran software package for radiative transfer calculations - description and examples of use, Atmos. Chem. Phys. 5, 2005, S. 1855–1877.



Mesoskalige Modellierung

Arbeitspaket 5.2

Thomas I. Petroligis
University of Oldenburg
Carl-von-Ossietzky-Str 9-11
26111 Oldenburg

Internet: <http://www.uni-oldenburg.de>

Jan Ploski
OFFIS – Institut für Informatik
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de>

1 Einleitung

Energiemeteorologie ist ein interdisziplinäres Forschungsfeld, das physikalische und meteorologische Methoden mit Expertenwissen in Bereichen Wind- und Solarenergieerzeugung, Energieversorgung, Informatik und Umweltwissenschaften verbindet. Energiemeteorologen befassen sich mit Fragestellungen wie z. B. Wind- und Solarenergieerzeugungsvorhersagen. Ihre Arbeit führt zu daten- und rechenintensiven anwendungsspezifischen Modellen, die globale Vorhersagen einbinden und herunterskalieren, um damit Nutzer wie Energieunternehmen, Windparkbetreiber, Energiemarkthändler und Infrastrukturinvestoren mit notwendigen Informationen zu versorgen.

Über die letzten Jahre hat sich Energiemeteorologie von einem neuen wissenschaftlichen Forschungsfeld zu einer etablierten, marktrelevanten Disziplin entwickelt, mit kommerziellen Unternehmen, die Datenprodukte sowie einschlägige Beratungsdienstleistungen für Entscheidungsträger anbieten. Eins der wichtigsten Verfahren, das Wissen in Energiemeteorologie darzustellen, ist dabei die Installation, Implementierung und Optimierung von Wettervorhersagemodellen.

Globale numerische Wettervorhersagemodelle werden durch die internationale Forschungsgemeinschaft in langfristigen Projekten entwickelt. Daten aus solchen etablierten Modellen werden weltweit in standardisierten Formaten verbreitet. Die Outputs dieser Modelle können jedoch in energiemeteorologischen Anwendungen leider nicht unmittelbar eingesetzt werden. Die Gründe dafür sind die grobe Raumauflösung (wegen Mängel an Rechenressourcen) und das Fehlen von detaillierten Parametrisierungen, die bekannte regionale Phänomene widerspiegeln würden. Dementsprechend werden von Meteorologen ebenfalls mesoskalige Modelle gepflegt, die an individuelle Anforderungen einer Anwendung (wie z. B. kurzfristige Windgeschwindigkeitsvorhersage) angepasst werden können und dabei Informationsquellen verwenden, die für die globalen Modelle nicht zur Verfügung stehen. Nachdem die mesoskaligen Modelle konfiguriert und gründlich validiert worden sind, können sie in Prozessierungsketten integriert werden, an deren Ende sich kommerzielle Anwender wie Windparkbetreiber befinden. Alternativ können ihre Outputs als spezialisierte Datenprodukte am Markt angeboten werden, die dann durch neuartige Anwendungen aufgegriffen werden können.

Die großskalige Realisierung der oben skizzierten Vision stellt besondere Anforderungen an die Datenverarbeitungsinfrastruktur einschließlich Rechenressourcen, Speicherressourcen und Spezialsoftware, sowie an das meteorologische Expertenwissen zum Finden von passenden Modellkonfigurationen, die genaue und aussagefähige Vorhersagen liefern. Um diese Anforderungen zu erfüllen, wurden im Arbeitspaket 5.2 von WISENT verschiedene Methoden entwickelt und umgesetzt. Dazu gehören die Implementierung und Validierung von Weather Research & Forecasting Model (WRF), die Anwendung der Lagged Average Forecasting-Technik (LAF), die optimale Kombination von mesoskaligen WRF-Vorhersagen mit globalen NCEP-Vorhersagen unter Anwendung von LAF, sowie die Entwicklung zahlreicher Softwarewerkzeuge und des WISENT-eigenen Wettervorhersageportals (WFP).

Im Laufe des Arbeitspakets fand eine intensive Zusammenarbeit zwischen Informatikern aus OFFIS und Meteorologen aus dem Forschungszentrum ForWind der Universität Oldenburg statt. Der Rest des Dokuments (in englischer Sprache) beschreibt daher nach

einer Einführung in Merkmale des WRF-Modells (Abschnitt 3) die erfolgten Projektaktivitäten aus Sicht der Energiemetereologieforschung (Abschnitt 4) und der Softwaretechnik (Abschnitt 5).

Anmerkung: Um Verwirrung zu vermeiden, wird im Rest des Dokuments konsequent die Schreibkonvention durchgesetzt, dass das Wort "Grid" (mit Großbuchstaben G) ein Netzwerk von Rechen- und Speicherressourcen bezeichnet, während das Wort "grid" eine mehrdimensionale Datenstruktur bezeichnet, die von WRF-Programmen verarbeitet wird.

2 Introduction

Energy meteorology is an interdisciplinary field of research utilizing physical and meteorological methods combined with expertise in the field of wind and solar energy conversion, energy supply structures, computational techniques, and environmental science. Researchers work on problems such as improvement of wind and solar energy output forecasts. This work is brought to fruition in form of customized data- and compute-intensive models, which incorporate global forecasts and downscale them to address issues important for electrical utilities, wind farm operators, energy market traders, and infrastructure investors.

Over the past several years, energy meteorology has advanced from a new scientific field to an established market-relevant discipline, with commercial companies offering data products and relevant consulting services to support decision makers. One of the primary means of representing knowledge related to meteorology in general and energy meteorology in particular is the installation, implementation, and optimization of weather prediction models.

Global numerical weather prediction models are developed by the international research community and are the fruit of many man-years of collaborative scientific and engineering effort. The data obtained from such established models is disseminated world-wide in standardized formats. Unfortunately, the outputs of global models cannot be directly employed in energy meteorological applications because of their coarse spatial resolution (due to limited computing resources) and the lack of detailed parametrizations that would reflect the empirical knowledge of important regional phenomena. To address this issue, the meteorological research community also maintains mesoscale models, which can be executed according to the individual demands of an application – such as the regional short-term prediction of wind speed – and which can be fine-tuned using information sources not available to their global counterparts. Once configured and thoroughly validated, mesoscale models can be incorporated into processing chains oriented toward commercial end-users, such as wind farm operators interested in energy output predictions, or their outputs can be packaged into specialized data products for the general market, to provide a basis for new applications.

Realizing the vision outlined above places substantial demands on the processing infrastructure – including the computational resources, storage, and custom software – and on the meteorological know-how required to find viable model configurations that yield accurate and informative predictions. To address these demands, in WP 5.2 of WISENT various methods have been developed and implemented, such as the implementation and validation of the Weather Research & Forecasting Model (WRF), the application of Lagged Average Forecasting (LAF) technique, the optimal combination of mesoscale WRF and global NCEP LAF forecasts, and the development of the Weather Forecasting Portal (WFP).

During the entire work package the computer scientists from OFFIS collaborated intensively with the meteorologists from the research center ForWind at the University of Oldenburg. Accordingly, after introducing the WRF model's features in Section 3, the remaining document explains our work activities from the energy meteorology (Section 4) and software engineering perspective (Section 5).

Note: To avoid confusion, the following convention is kept in the rest of the document: the capitalized word “Grid” is used to refer to a network of computing and storage resources, while the lowercase word “grid” is used to refer to the multi-dimensional data structure processed by WRF programs.

3 Overview of the Weather Research & Forecasting (WRF) Model

WRF is a mesoscale NWP (Numerical Weather Prediction) model, suitable for research and operations, capable of resolving and simulating atmospheric phenomena with scales ranging from cloud to fully developed weather systems.

The following subsections are organized as follows: In Section 3.1, the reasons behind the selection of WRF as WISENT's mesoscale platform are explained, while in Section 3.2 the main characteristics of WRF Version 2.2 are presented.

3.1 Selection of WRF as the mesoscale platform in WISENT

The Weather Research & Forecasting (WRF) model is state-of-the-art, transportable, and efficient in a parallel computing environment. It is designed to be modular, and a single source code is maintained that can be configured for both research and operations. That is why the WRF model is well suited for a wide range of applications, from idealized research simulations to operational forecasting. It offers numerous physics options. Advanced data assimilation systems are being developed and tested in tandem with the model. The model is designed to improve forecast accuracy across scales ranging from cloud to synoptic.

Its horizontal grid resolutions of 1-10 kilometers makes WRF also well suited for newly emerging NWP applications in the non-hydrostatic regime, so it does not come as a surprise why WRF has been considered as an excellent model candidate for wind flow simulations (wind power prediction). For this reason and for other research & operational modeling aspects explained below, WRF has become the mesoscale platform of WISENT. WRF's main components, i.e. preprocessing, solver(s), physics & postprocessing modules are shown in Figure 3.1.1.

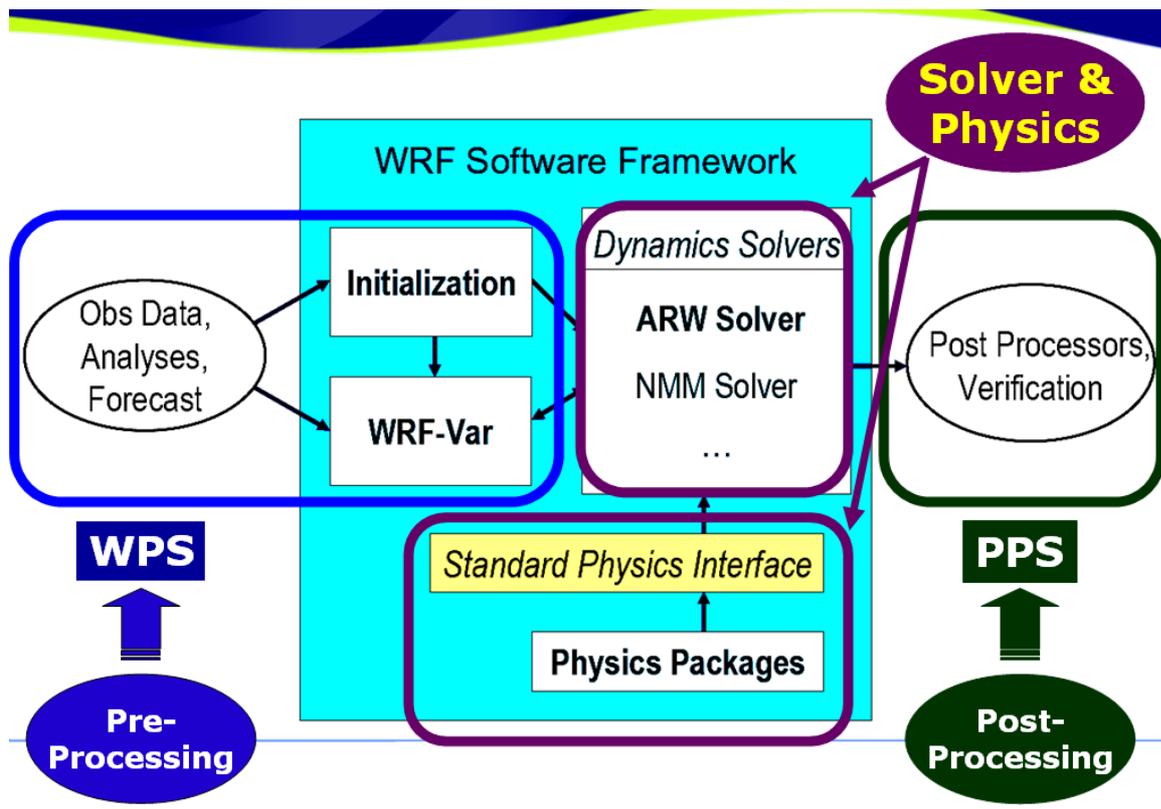


Figure 3.1.1: Weather Research & Forecasting Model's main components

WRF is maintained and supported as a community mesoscale model to facilitate wide use in research, particularly in the university community. Advances achieved in the wider research community will have a direct path to operations. The WRF software infrastructure supports two dynamical cores, the Advanced Research WRF (ARW), whose development has been led by the National Center for Atmospheric Research (NCAR), and the Non-hydrostatic Mesoscale Model (NMM) core developed by the National Centers for Environmental Prediction (NCEP).

The ARW dynamical core has been chosen as the WRF mesoscale platform in WISENT for a lot of reasons, but most importantly because WRF is considered as the next generation mesoscale community model. It has replaced ETA at NCEP since 2006, and RUC (Rapid Update Cycle) model in 2007-2008. It is considered as the successor to MM5 (Fifth-generation Mesoscale Model / joint effort of NCAR & Pennsylvania State University), which is no longer being developed. The basic differences between MM5 and WRF can be seen in Figure 2.1.2.

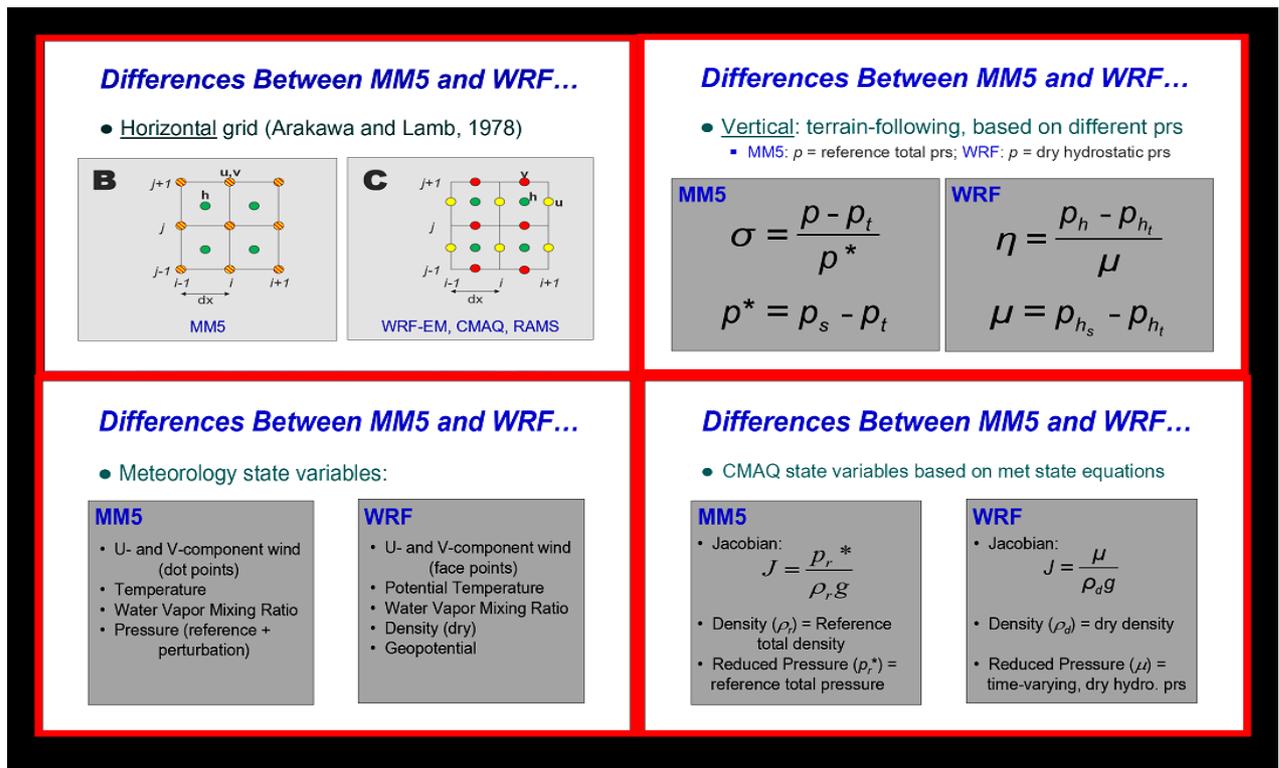


Figure 3.1.2: Basic differences between MM5 and WRF

WRF uses the fully-compressible, non-hydrostatic equations, and is conservative for scalar variables. The horizontal coordinate is the Arakawa staggered C-grid, while in the vertical a terrain-following mass coordinate is used. WRF employs advanced numerics. The time integration is 3rd-order Runge-Kutta in time with split acoustic/gravity mode handler, while 2nd- to 6th-order integration is used in the horizontal and vertical. Sub-grid-scale turbulence, divergence damping, and external mode filtering are available. Both 1-way and 2-way nesting is supported, with an option for moving nests. The grid ratio can be any integer. A number of microphysical schemes are available, including Thompson (six classes plus ice number concentration). Available cumulus parameterizations include Kain-Fritsch (Eta version enhanced for shallow convection), Betts-Miller-Janic, and Grell-Devenyi ensemble scheme. Surface layer schemes based on similarity theory have been adapted from Eta and MM5 models. Multi-layer Land-surface models include Noah and RUC scheme. Planetary Boundary Layer (PBL) schemes include MRF, Yonsei University (YSU), and Mellor-Yamada-Janic (MYJ). WRF provides efficient execution on a range of computing platforms (distributed and shared memory, vector and scalar computing). The scientific code is separated from the parallelization and other architecture-specific codes. WRF is highly modular, single-source Fortran-90 code for maintainability. WRF's model coupling API (Application Programming Interface) enables it to be coupled with other models such as ocean, and land models.

Since the first release of an early version of the ARW in December 2000, over 4,200 users have registered to download the model code as of September 2006. Over half of these users are distributed across some 82 foreign countries. Community researchers have contributed strongly to evaluating the capabilities and limitations of the WRF system through real-time forecast experiments, model inter-comparison studies, and case-study

analyses. The WRF model is also transitioning into use in a number of operational forecast centers NCEP is currently running versions of WRF in their High Resolution Window Domains, as members of their Short-Range Ensemble Forecasts, and in their North American Meso (NAM) Model, and is adapting WRF-based implementations for the Rapid Refresh Model (RRM) and Hurricane Forecast Model (HFM) over the next years. The Air Force Weather Agency (AFWA) began operational use of WRF for forecasting in their worldwide theaters in July 2006. In addition to applications in the U.S., operational centers in South Korea, India, Israel, China, Taiwan, and Greece are implementing new forecast systems based on WRF.

3.2 Main characteristics of WRF Version 2.2

The Weather Research and Forecasting (WRF) Version 2.0 was released in May 2004, followed by Version 2.1 in August 2005, and Version 2.2 in December 2006. Along with Version 2.2 of the model, a new preprocessing system (WPS) was released at the same time. Several new features became available in Version 2.2. Four-dimensional data assimilation by grid-nudging was added by Penn State. This provides a capability for keeping the model domain constrained by analyses over a period. Observational nudging was added by NCAR/RAL, providing a capability of ingesting individual observations as the model runs. An urban canopy model was added as an option in the Noah LSM scheme. The urban canopy model, designed for use with high-resolution grids, has a sophisticated treatment of sub-grid roads, walls, and roofs in the surface radiation and fluxes. The CAM3 radiation package from the NCAR CCSM climate model has been added. This includes accurate, and sophisticated shortwave and longwave schemes that also can handle a variety of aerosols for atmospheric chemistry applications, although the aerosol link has not been completed yet, and it just uses specified constant aerosol amounts. A fully revised and rewritten microphysics package was introduced in V2.2. Among these changes, the Betts-Miller-Janjic cumulus scheme was made more active again, more similar to the 2.0 version. The Ferrier microphysics was adapted to work better with radiation for ARW, which previously did not distinguish the cloud and precipitation species from the Ferrier total condensate. This would mostly reduce the shortwave blocking effect of clouds with that scheme. The Dudhia shortwave scheme has a new parameter to allow tuning of the clear-sky scattering effect that can account for aerosols to some extent. A sixth-order filter was added as an option to selectively remove grid-scale noise. This is a numerical filter and can be used in conjunction with the physical diffusion options already available. Positive-definite advection was added as an option for all WRF scalars. This scheme prevents the generation of negative values by the advection process, and is therefore beneficial for conservation because it means that the non-conservative option is no longer required. The upper absorbing layer was modified so that it could be used with real-data cases.

The WRF Preprocessing System (WPS) was also released with Version 2.2. This was designed to take over the functionality of the Standard Initialization (which is also still supported), and has been developed with generality in mind, also building on the idea of modularity in the preprocessing stages. The WPS components and functions (shown in Figure 3.2.1) are:

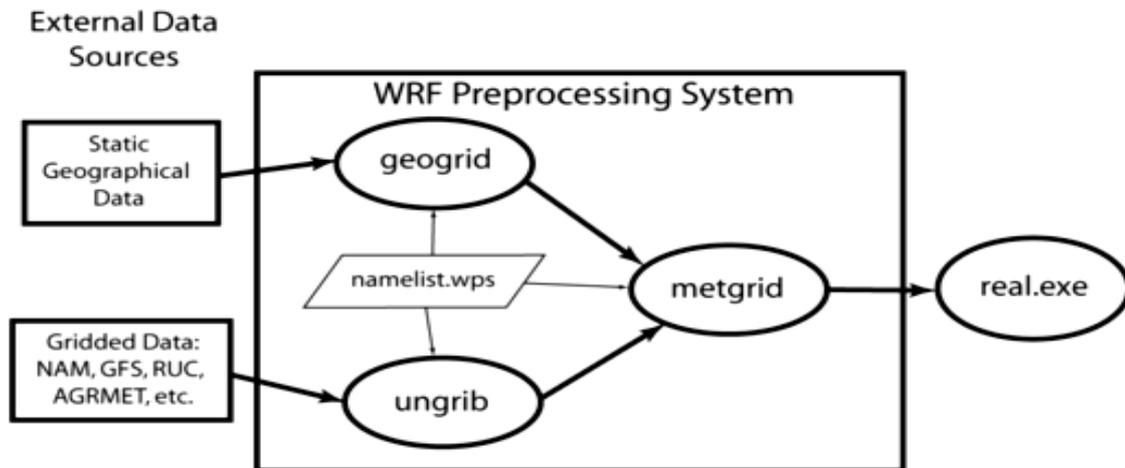


Figure 3.2.1: WRF Preprocessing System's main components

- *GEOGRID*: Sets up domain location information and interpolates geographic (static) data onto the model grid.
- *UNGRIB*: Converts Grib data to a simple “intermediate” format.
- *METGRID*: Interpolates intermediate data to the model grid, and combines it with geogrid output.

The new version of the `real.exe` program now does the vertical interpolation step when WPS (`metgrid`) data is input, but can also handle SI (Standard Initialization) data that is already interpolated. Since Version 2.2, WPS has been extended to work with the NMM dynamical core, and this is already available in a prerelease version available from the DTC (Developmental Testbed Center at NCAR). For graphics capabilities, ARWpost was released with Version 2.2, replacing `wrf2grads` and `wrf2vis5d`. This converts from WRF and WPS output files (NetCDF) to GrADS and Vis5D formats. RIP routines were also adapted to work with WPS outputs, and with idealized WRF output.

In brief, the Weather Research and Forecasting model on which we focused our efforts in Grid context, is a mesoscale NWP model, suitable for research and operations, capable of running on a variety of platforms, either serially or in parallel, with or without multithreading.

WRF (V2.2) is fully compressible; Euler non-hydrostatic with a run-time hydrostatic option available and it is conservative for scalar variables. Its prognostic variables are: the velocity components u and v in Cartesian coordinates, the vertical velocity w , the perturbation potential temperature, the perturbation geopotential, and the perturbation surface pressure of dry air. Optionally, the turbulent kinetic energy and any number of scalars such as water vapor mixing ratio, rain/snow mixing ratio, and cloud water/ice mixing ratio can be also predicted.

The model's vertical coordinate is the terrain-following hydrostatic pressure, with vertical grid stretching permitted. The top of the model is a constant pressure surface. Its horizontal grid is the Arakawa C-grid staggering. Its time integration scheme is a time-split integration using a 3rd order Runge-Kutta scheme with smaller time step for acoustic and gravity-wave modes. Its spatial discretization comprises 2nd to 6th order advection options in horizontal and vertical. WRF provides the user with the capability to use both

turbulent mixing and model filters (WRF tuning parameters). An explicit filter option is also available.

Finally, considering WRF's total characteristics, it is not a surprise why it is considered as a "Swiss knife" model capable to cope with a wide spectrum of time/spatial scales and almost any kind of forecast guidance (Figure 2.2.2).

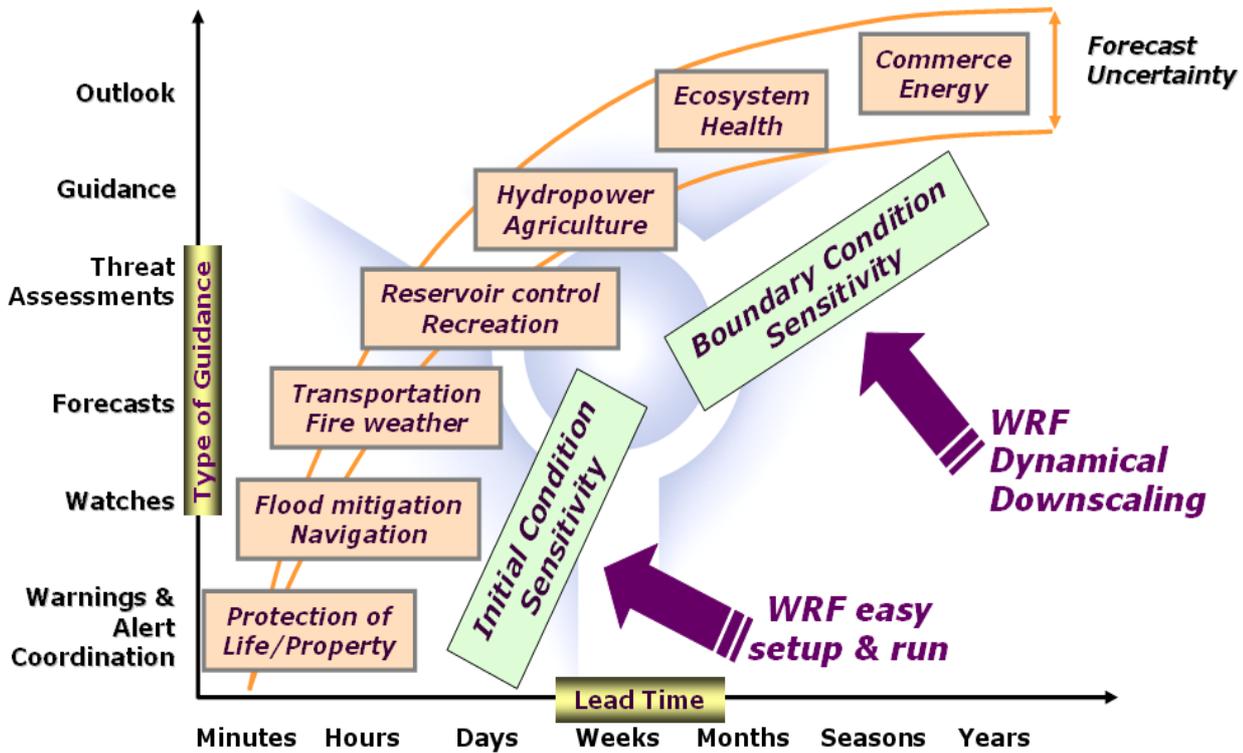


Figure 3.2.2: Graphical representation of WRF's modeling capabilities

4 Implementation and optimization of WRF in WISENT

When implementing a mesoscale model, there exists a long list of modeling aspects that someone has to consider. For instance, what are the important physical processes that should be resolved in the WRF model? Figure 4.1 shows the main physical processes that have to be resolved or parametrized (whatever the model is not able to resolve has to be parametrized by the model) by a mesoscale model such as the WRF.

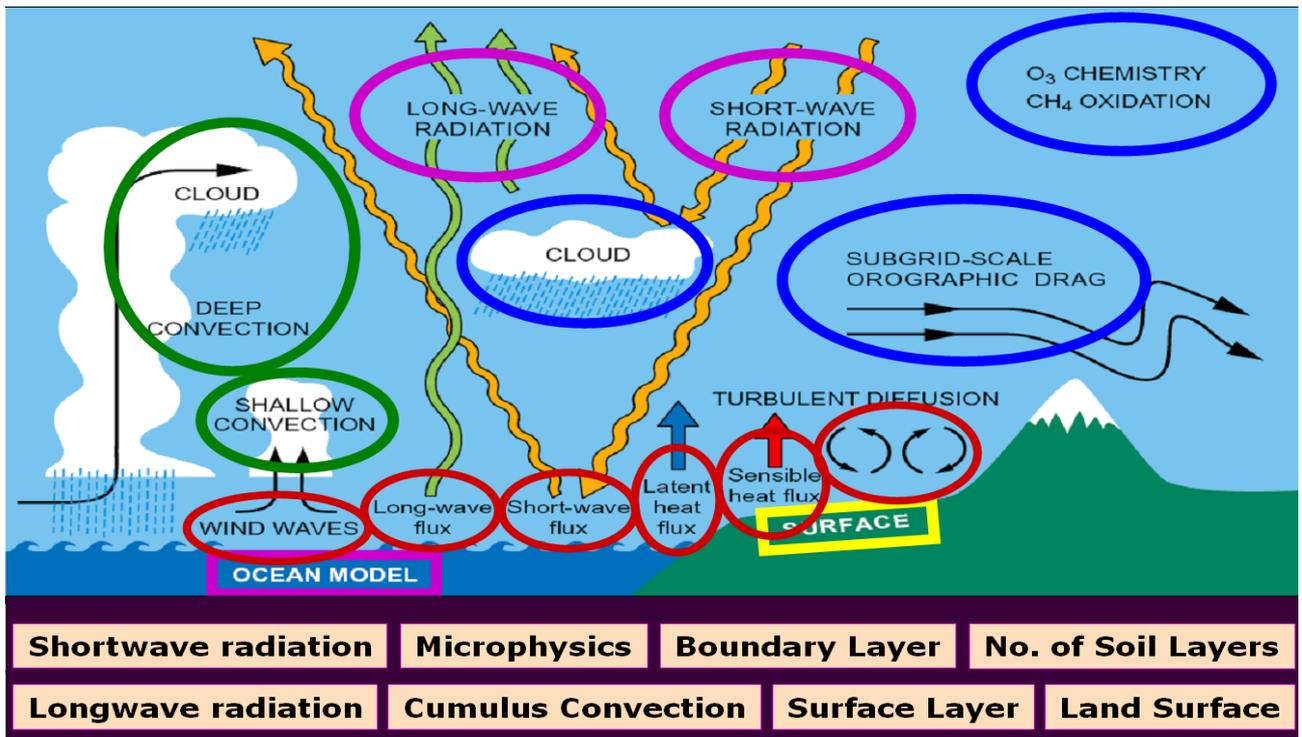


Figure 4.1: Atmospheric processes (physical parametrization schemes)

For instance, the land-surface models (LSMs) use atmospheric information from the surface layer scheme, radiative forcing from the radiation scheme, and precipitation forcing from the microphysics and convective schemes, together with internal information on the land's state variables and land-surface properties, to provide heat and moisture fluxes over land points and sea-ice points. These fluxes provide a lower boundary condition for the vertical transport done in the PBL schemes (or the vertical diffusion scheme in the case where a PBL scheme is not run, such as in large-eddy mode). The radiation schemes provide atmospheric heating due to radiative flux divergence and surface downward longwave and shortwave radiation for the ground heat budget. Longwave radiation includes infrared or thermal radiation absorbed and emitted by gases and surfaces. Upward longwave radiative flux from the ground is determined by the surface emissivity that in turn depends upon land-use type, as well as the ground (skin) temperature. Shortwave radiation includes visible and surrounding wavelengths that make up the solar spectrum. Hence, the only source is the Sun, but processes include absorption, reflection, and scattering in the atmosphere and at surfaces. For shortwave radiation, the upward flux is the reflection due to surface albedo. Within the atmosphere the radiations responds to model-predicted cloud and water vapor distributions, as well as specified carbon dioxide,

ozone, and (optionally) trace gas concentrations. All the radiation schemes in WRF currently are column (one-dimensional) schemes, so each column is treated independently, and the fluxes correspond to those in infinite horizontally uniform planes, which is a good approximation if the vertical thickness of the model layers is much less than the horizontal grid length.

Other important aspects to consider implementing WRF are its capabilities under constraints imposed by an operational or semi-operational environment. WISENT's WRF mesoscale platform should be able to provide high-resolution wind/solar power related NWP fields on a demanding tactical basis. This is not a new problem in (operational) modeling. Operational forecasters in general have typically relied on meteorological models produced by government agencies in the US, Japan, and Europe as part of routine operations. Although these models can provide high quality guidance, relying on multiple external sources presents a set of challenges for software engineers, IT staff, and the operational forecasters. Common sense suggests the use of a single global model, such as the NCEP's Global Forecast System (GFS) provided freely by the US National Weather Service. However, as explained in this document's introduction, the global models do not provide sufficient resolution to resolve important details in areas of highly-variable topography or near coastlines, so the installation and optimization of a mesoscale model such as the WISENT's WRF platform should be considered as a must.

The following subsections are organized as follows: In Section 4.1, the initial implementation and validation of WRF is presented. In Section 4.2, a study case, the storm Kyrill (hitting Germany on 18 January 2007) is described, which also served for WRF validation. High-resolution simulations of the storm Kyrill have helped us draft future guidelines and make valuable decisions for the further optimization of WRF. In Section 4.3, the methodology of constructing WRF Lagged Average Forecasting (LAF) ensemble schemes for Wind Power Prediction is analyzed. In Section 4.4, the techniques of various combinations of LAFs with deterministic model components and the construction of optimal ensemble schemes are presented. Finally in Section 4.5, future research work plans are proposed and some recommendations are given.

4.1 Initial implementation and validation

WISENT's WRF model characteristics are presented in detail in Section 3.2. The WRF Version 2.2 on which we focused our initial testing efforts in Grid context was installed at the OFFIS cluster in Oldenburg. The WRF code has been available to the general community for download at <http://www.wrf-model.org>. WISENT's WRF version makes use of the ARW (Advanced Research WRF) solver, which is composed of several initialization programs for idealized and real-data simulations, and the numerical integration program. Besides of highlighting the advantages and current challenges of large-scale execution of NWP models on the German Grid, the results of a research study performed using our infrastructure, focusing on wind power prediction issues, have served as an initial WRF validation.

Validation of a model refers to the processes and techniques that are used to ensure that the model represents (simulates) the real system (atmosphere) to a sufficient level of accuracy, while verification refers to the processes and techniques used to ensure that the model is correct and matches certain specifications and assumptions. During our initial validation and verification tests different model configurations were examined. Various

nested one- or two-way formulations of WRF with its inner grid to take values as low as 1km were configured and run in a single or multithreading environment.

For our initial verification purposes, daily WRF runs for March, April & May (MAM) 2007 period were performed with a forecast horizon of up to 48 hours, based on the NCEP analysis cycle of 00 UTC (initial & boundary fields). A two-way nesting was employed, utilizing two domains. The outer (coarse) domain was configured with a 45 km grid resolution, while the inner domain (covering Germany) had a resolution of 15 km (ratio parent to child 3:1).

WRF uses an Arakawa-C stagger scheme allowing an equal number of points in the grid to be distributed across the 16 processors used on the computational platform. The Runge-Kutta solver used allowed a long time step of 90 seconds to be implemented despite the 15 km (inner) grid spacing. In the vertical, 35 full levels (34 computational layers for the mass variables) were used. This formulation was chosen to serve as our “basis” version, for future inter-comparisons. The physical parametrization schemes used in our “basis” version are summarized below:

- Land-surface model: thermal diffusion scheme
- Microphysics: WRF Single-Moment (WSM) 3-class simple ice scheme. A simple efficient scheme with ice and snow processes suitable for mesoscale grid sizes
- Cumulus parametrization: Kain-Fritsch (new ETA) scheme
- Planetary Boundary Layer (PBL): Yonsei University Scheme (YSU). Next generation MRF-PBL. Non-local-K scheme with an explicit entrainment layer and parabolic K profile in unstable mixed layer
- Shortwave radiation: Dudhia scheme. A simple downward integration allowing for efficient cloud and clear-sky absorption and scattering
- Longwave radiation: Rapid Radiative Transfer Model (RRTM) scheme. An accurate scheme using look-up tables for efficiency. Accounts for multiple bands, trace gases, and microphysics species

Using objective verification scores such as Bias (Mean Error), MAE (Mean Absolute Error) and RMSE (Root Mean Square Error), WRF 24 and 48 hour forecasts were verified against corresponding NCEP analysis fields. A similar set of scores was used for NCEP 24 and 48 hour forecasts. Figure 4.1.1 contains the mean error (bias) for WRF and NCEP 24 (a) and 48 (b) hour forecasts for wind speed at 10 meters valid for MAM 2007 period.

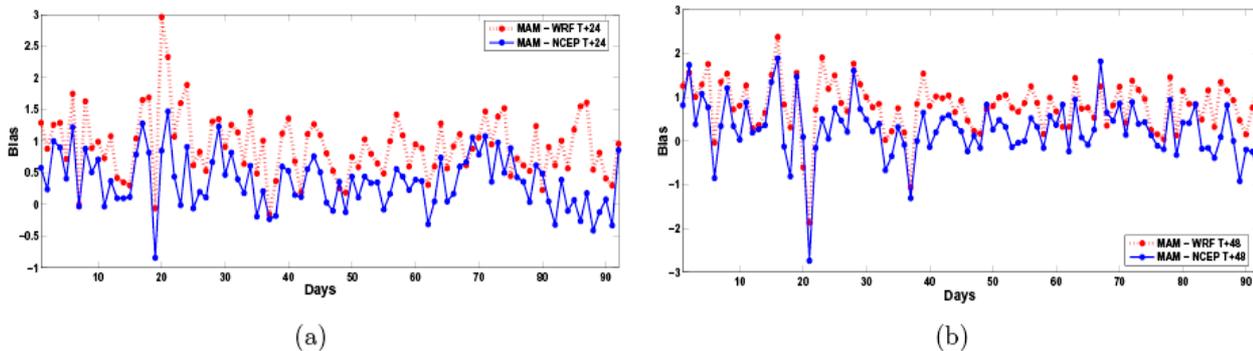


Figure 4.1.1: 10 meter wind speed bias for WRF and NCEP 24 (a) and 48 (b) hour forecasts (MAM 2007)

Although the installed WRF model can be easily tailored to local needs and interests, serving all future potential Grid users, initial results showed that, meteorologically, the WRF forecasts did not perform as well as expected. However, running a mesoscale model capable of simulating correctly local topographic effects takes time and a lot of tuning. WRF wind forecasts exhibit a positive bias at 10 meter height level (WRF flow too strong) in both the 24 and 48 hour forecast horizons.

Nevertheless, it is important to bear in mind that most of the currently operational mesoscale models (as the old NCEP/ETA for example) and their associated postprocessed fields (e.g., 2-meter temperature, 10 meter winds) have undergone extensive tuning since their implementations, most of them for several years, whereas the WRF model is a relatively new model and was used by us in an “off-the-shelf” configuration. So, even in its early stages of development, the performance of the WRF model overall has been very encouraging.

The problem with low level wind forecasts may be linked to the implementation of the PBL, land surface, and radiation schemes and their interactions within the WRF model (cf. Section 4), which poses interesting new challenges for the potential users of WRF. Furthermore in order to account for systematic forecasting errors that may be due to the NWP model or to the modeling approach, there is a need for integration of the so-called Model Output Statistics (MOS) for postprocessing wind/wind power forecasts (as will become evident in the following subsections).

4.2 The Kyrill storm study case

Driven by a very strong jet stream, storm Kyrill hit Europe on 18 and 19 January. It caused considerable damage in England, Germany and the Netherlands and, unusually for a European storm, extended its impact as far east as the Czech Republic and Poland (Figure 4.2.1).

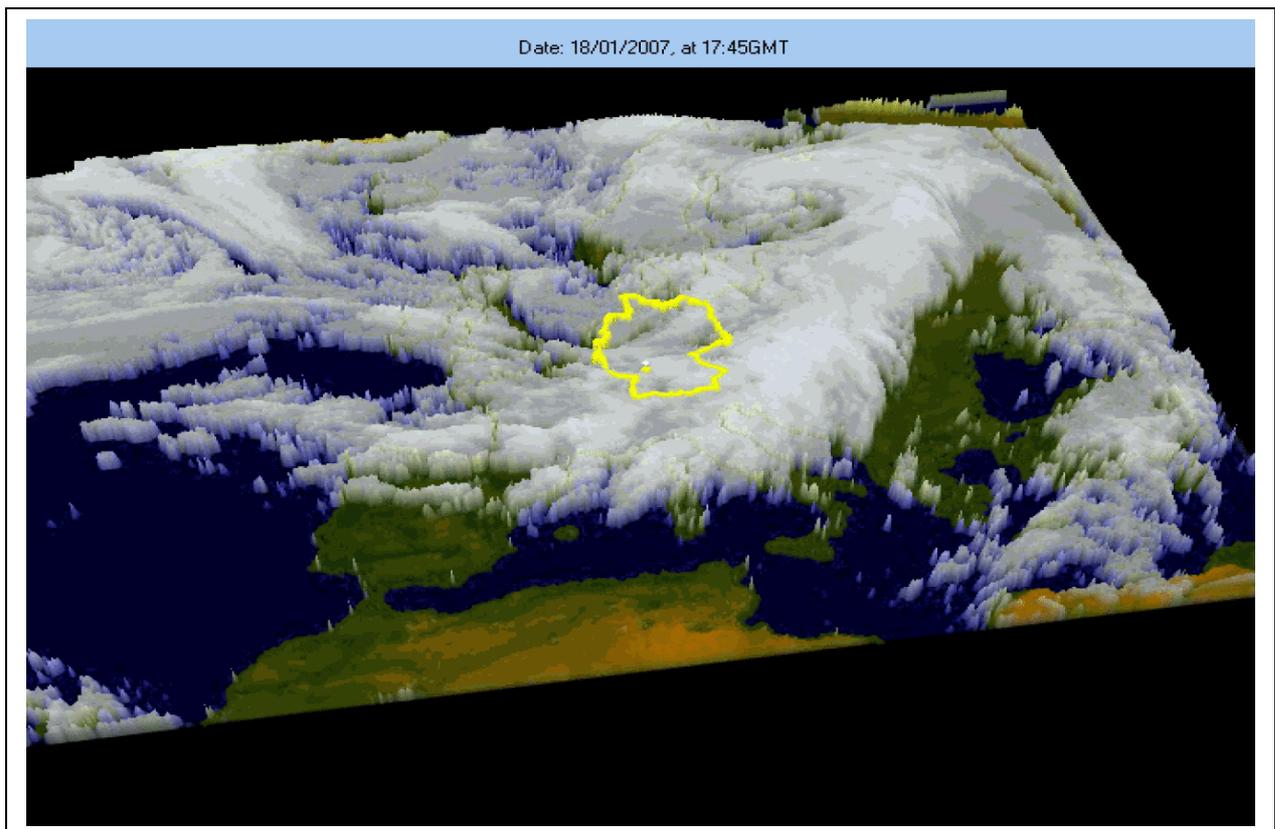


Figure 4.2.1: Satellite image of storm Kyrill hitting Germany (18 January 2007, source: <http://www.schuai.net/VirtualWave3D/Kyrill/>)

Kyrill has been the worst windstorm to hit Europe since the storms Lothar and Martin in 1999. Kyrill's wind speeds reached around 130km/h (80mph) and its gale force winds battered some areas for more than 24 hours. The strongest wind speeds were recorded in northern England. Over half a million homes and businesses were left without electricity at the height of the storm, and 14 people were killed. Kyrill then tracked east/southeast after moving across the United Kingdom, battering Germany, the Netherlands, the Czech Republic and Poland.

Germany suffered the largest percentage of insured losses from Kyrill after the storm intensified and produced wind gusts of up to 195km/h. The powerful winds felled trees, and property damage was reported across the whole of Germany. Thirteen people were killed and over a million households were without power. Heavy rain accompanied the storm, triggering flooding in coastal and low lying areas. All forms of transport were badly disrupted and Berlin's new train station, the biggest in Europe, was evacuated and closed for 14 hours after a two tonne girder fell. The severe and widespread damage triggered significant insurance claims across Europe, and RMS has estimated a total market property loss of 3 billion Euro in all affected countries other than the Czech Republic and Poland. Up to 65% of this loss could come from Germany.

For simulating Kyrill using WISENT's mesoscale platform, various WRF configurations were tested before final setup. It is worth to mention that for a typical two-way WRF two-domain nested formulation, with an inner grid of 5 km and 241x241 grid points covering

the greater area of Germany, and a vertical resolution comprising 35 levels (as shown in Figure 4.2.2), the completion of a 48 hour forecast took 23.1 hours on a single processor.

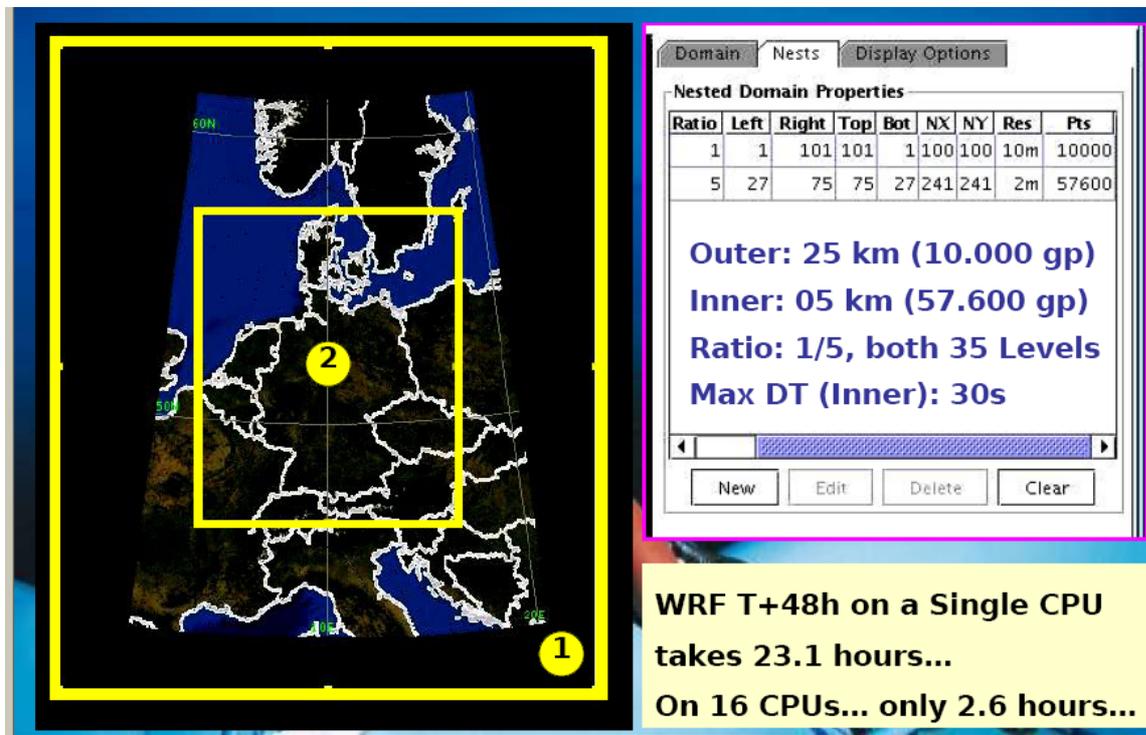


Figure 4.2.2: WISENT's WRF 2-way / 2-domain nested formulation setup for simulation of storm Kyrill

Running WRF (with exactly the same configuration setup) in a parallel (cluster) environment using 16 CPUs with MVAPICH compilation option, the integration time dropped down to 2.6 hours, showing the capabilities of parallelization techniques. The two-way/two-domain nested setup shown in Figure 4.2.2 was used for a detailed simulation of Kyrill. The "basis" set of physical parametrization schemes used in Section 4.1 was utilized. The WRF forecasts were based on NCEP's GFS 18-01-2007 00UTC global forecast fields (initial and lateral conditions). The WRF forecast horizon was set to 24 hours while the maximum time interval for the inner domain was restricted to 30 seconds to ensure computational stability.

Figure 4.2.3 contains WRF's 6-hour (left panel) and 12-hour (right panel) wind speed forecast at the height of 25 meters. The bold dot (shown by an arrow) represents the exact position of the Sylt Island (North Germany). It is evident that from 06UTC (18-01-2007) to 12UTC WRF correctly predicts a slight decrease of the wind speed over Sylt, even if the storm, which is evident (yellow/red color) to the west over North Sea, is reaching values close to 25m/s (90km/h).

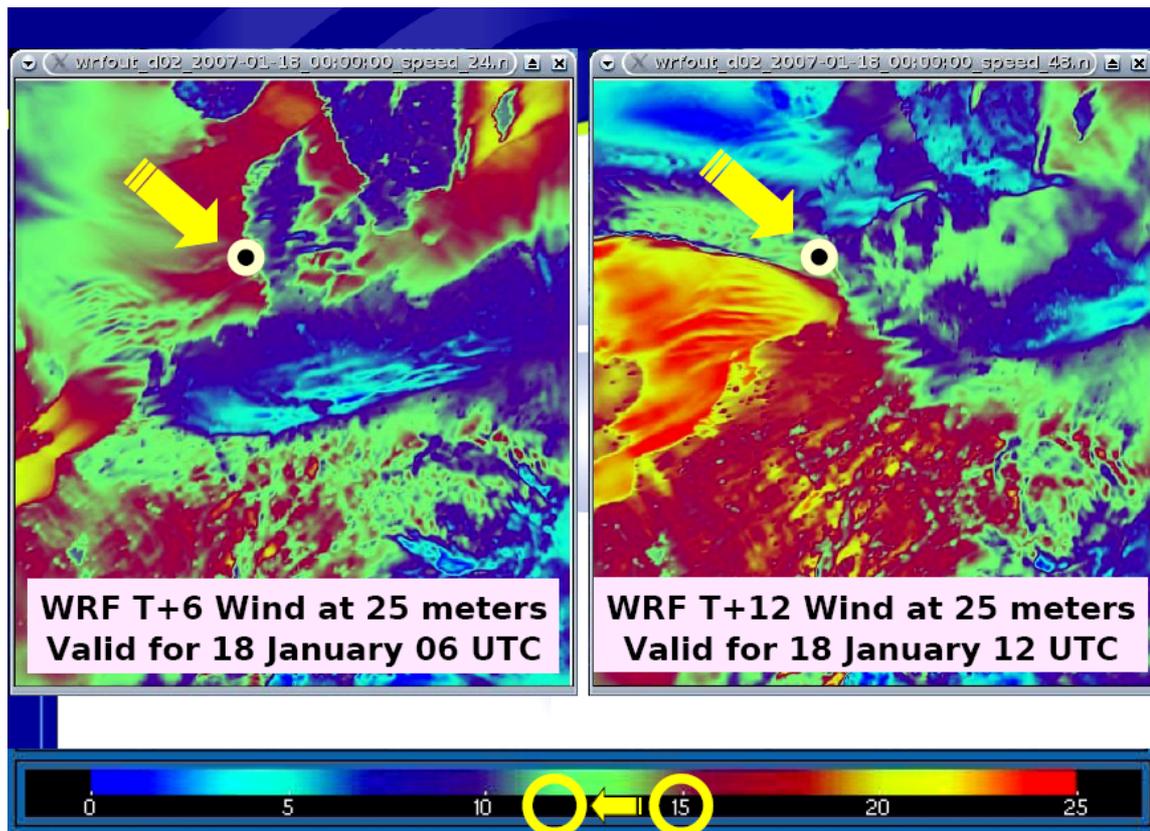


Figure 4.2.3: WRF high-resolution simulation of German storm Kyrill. Left panel contains WRF's 6-hour forecast for wind speed at 25 meter height. Right panel contains WRF's 12-hour forecast. Values are expressed in m/s (i.e. 15m/s corresponds to a 54km/h value)

Figure 4.2.4 contains WRF's 18-hour (left panel) and 24-hour (right panel) wind speed forecast at the height of 25 meters. Following WRF's forecast guidance, it is evident that the storm is reaching Sylt at 18UTC with values close to 80km/h (left panel). Furthermore, after six hours (i.e. at 00UTC of the next day, 19-01-2007) the storm has passed Sylt Island with its maxima now centered far to the South, as the storm is moving east/southeast (shown at the right panel). It is worth to point out that WRF has been capable of simulating the basic characteristics of storm Kyrill with satisfying accuracy.

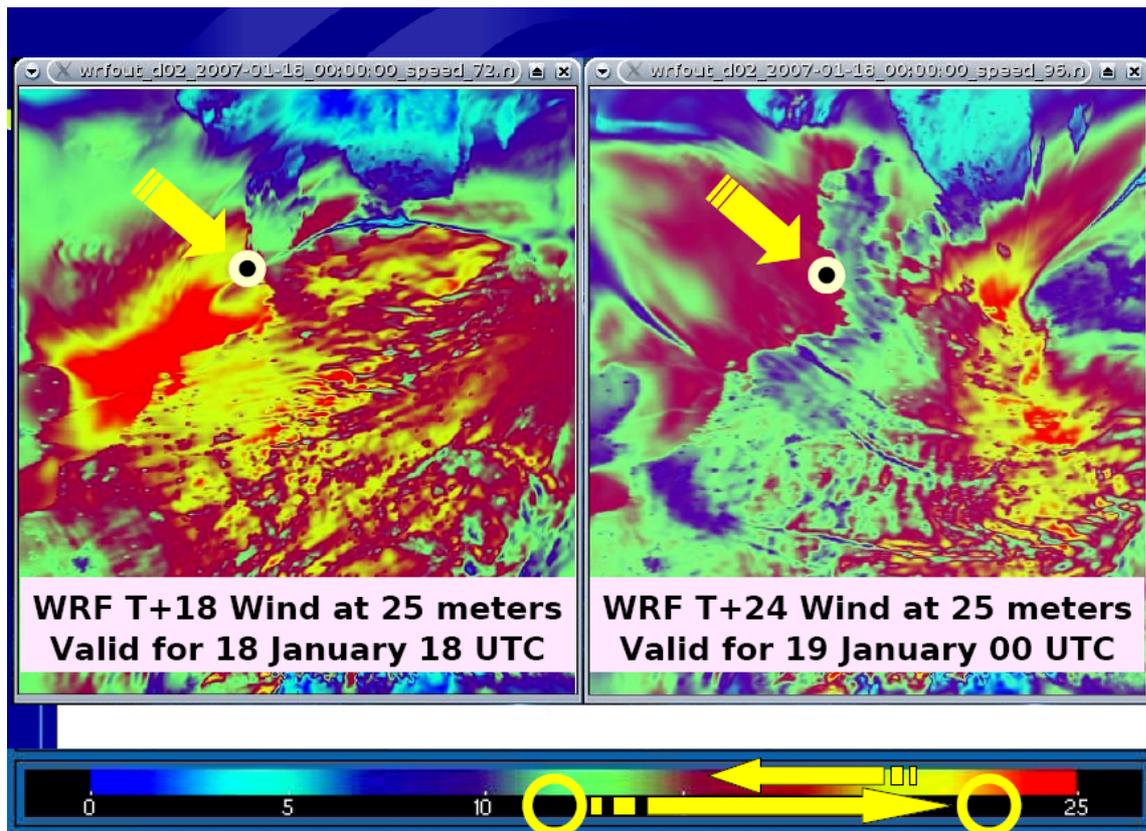


Figure 4.2.4: As in Figure 4.2.3, but the left panel contains WRF's 18-hour forecast, while the right panel contains WRF's 24-hour forecast (wind speed at 25 meters)

Finally, Figure 4.2.5 contains photos of Sylt Island during and after the passage of storm Kyrill together with WRF's verification against NCEP analysis values over Sylt. Wind speeds are referring to the first model level (at 25 meters height) interpolated for the exact position of Sylt. From Figure 4.2.5 it is clear that the model is following the observed decrease of wind speeds for the first 12 hours and it correctly forecasts the very abrupt and significant increase of wind which took place from 12 to 18UTC. It is worth to point out the ability of WRF to provide useful prediction guidance for this critical time interval where the wind speed from a value less than 40km/h (12UTC) has reached an 80km/h value within 6 hours (18UTC). Furthermore, WRF was able to provide useful information for the last interval of simulation as well, since it correctly predicted the decrease of the wind from 80 to 60km/h for the next 6-hour time interval (i.e. from 18 to 24UTC).

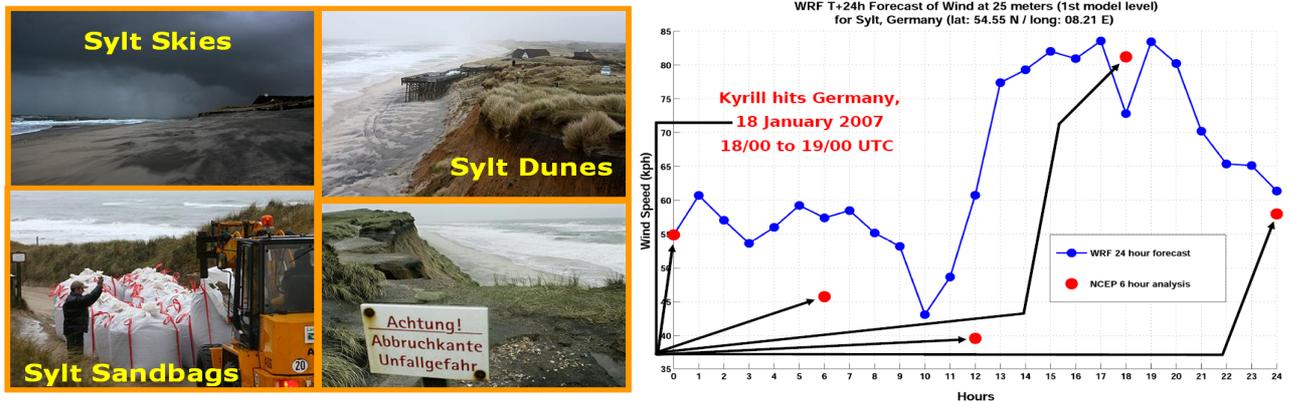


Figure 4.2.5: Photos of Sylt Island during and after Kyrill's passage (left panel) and verification of WRF wind speed forecasts (1st model level) against NCEP analysis values for Sylt Island valid for a period of 24h (2007-01-18)

4.3 Wind power prediction utilizing WRF LAFs

Linking meteorological, physical, and technical topics is a typical attribute of energy meteorology. The retrieval of application specific meteorological information such as WRF's or any other NWP model's wind prediction fields is an important prerequisite. In Figure 4.3.1 the basic characteristics of operational wind power prediction utilizing the HUGIN forecasting tool (platform) are presented. HUGIN was developed at ForWind (University of Oldenburg) and it is regarded as the successor of the well-known and successful Previento model that was also developed at the University Oldenburg and is nowadays commercially used by wind farm operators. HUGIN is capable to provide wind power forecasts at a tactical basis using NCEP's global forecasts.

HUGIN forecasts are available at www.forwind.de. The predictions are updated four times a day. The quality of the forecasts is comparable with other commercial Wind Power Prediction (WPP) models (for day-ahead HUGIN has a normalized RMSE of 5.2% and for 2 day-ahead a RMSE of 6.5% utilizing the NCEP's 00UTC analysis and forecast fields). The same is true when ECMWF forecasts are used.

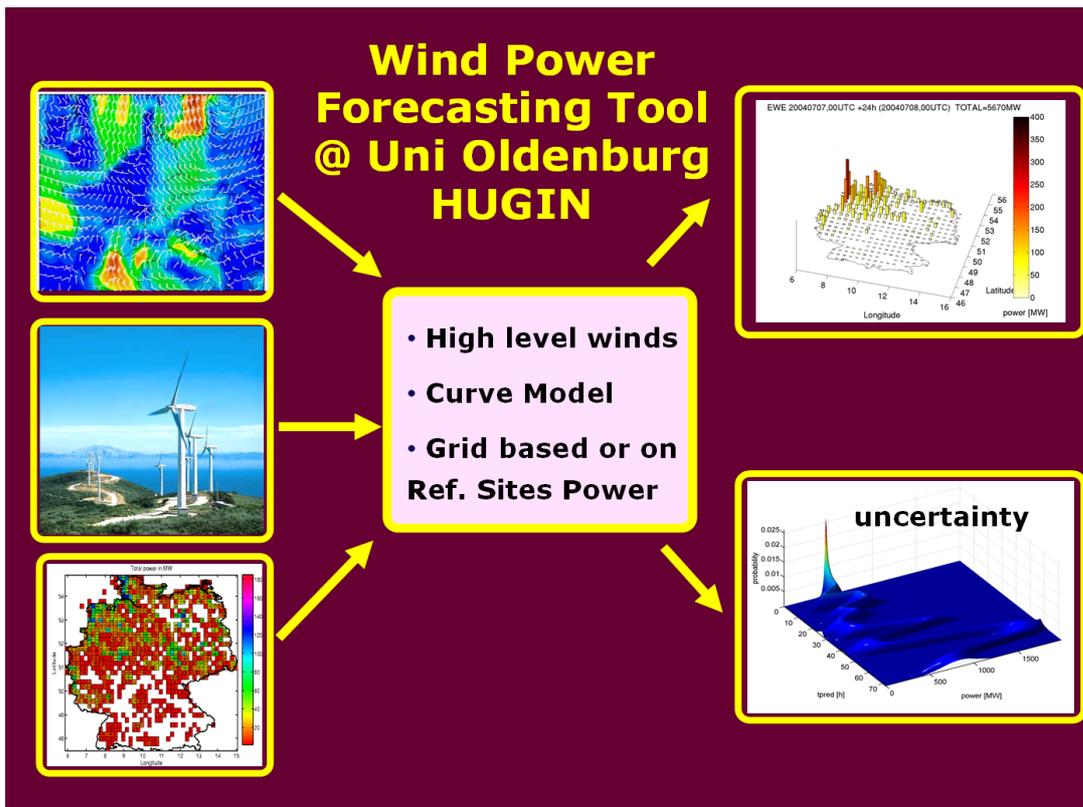


Figure 4.3.1: Typical cycle of operational wind power prediction from HUGIN Wind Power Prediction Platform (University of Oldenburg)

Currently, HUGIN uses mainly the NWP global fields given by the NCEP GFS model as input data (initial and boundary fields). Another option is the use of a “detailed” regional model such as a meso-scale model which can provide HUGIN with high-resolution NWP fields. For investigating the accuracy of wind speed fields produced by WISENT's meso-scale platform (WRF) we have utilized a two-domain, two-way nested set-up (Figure 4.3.2), with a parent (outer) to child (inner) domain ratio 1:3 (30 & 10 km).

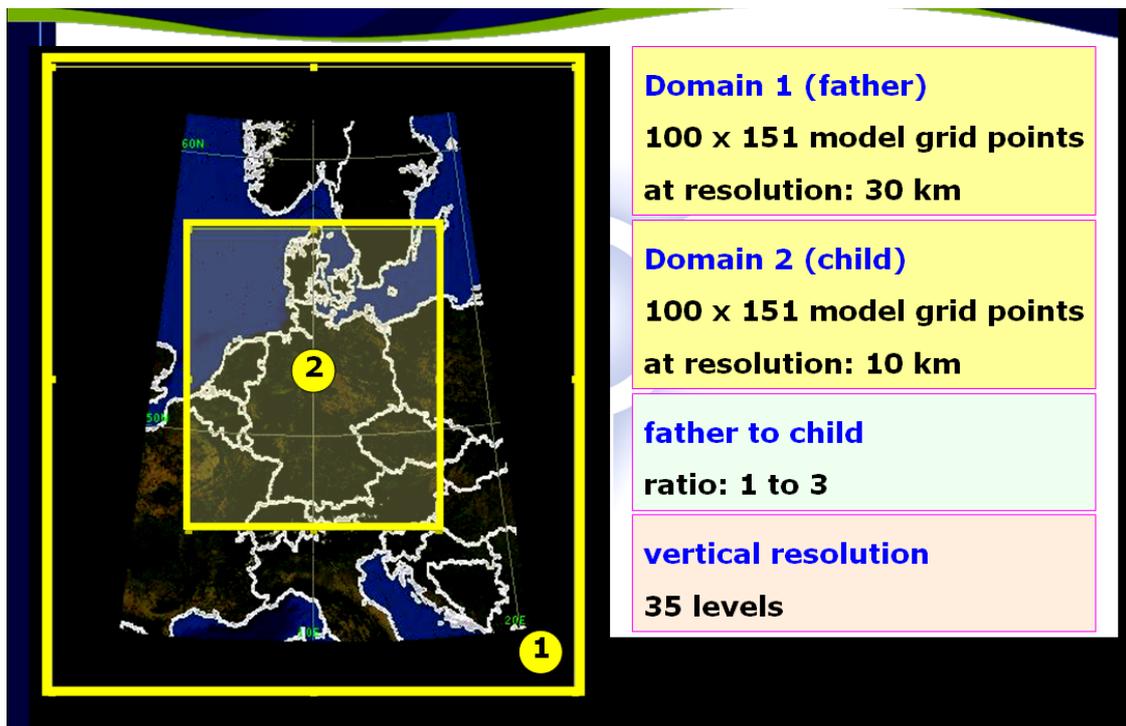


Figure 4.3.2: WISENT's WRF 2-domain, 2-way nested setup

Two physical parametrizations have been employed:

1. the default “off-the-shelf” physics (Figure 4.3.3), i.e. land-surface model: thermal diffusion scheme, microphysics: single-moment WSM 3-class simple ice scheme, surface layer: Monin-Obukhov, cumulus parametrization: Kain-Fritsch new ETA scheme, planetary boundary layer: Yonsei University Scheme, shortwave radiation: Dudhia scheme, longwave radiation: rapid radiative transfer model.
2. another one (marked as FINO setup physics) with a different set of schemes, i.e. microphysics (Ferrier-New ETA scheme), surface layer (Monin-Obukhov & Janjik), planetary boundary layer (Mellor-Yamada-Janjik), land surface (Noah) and number of soil layers (4-Noah).

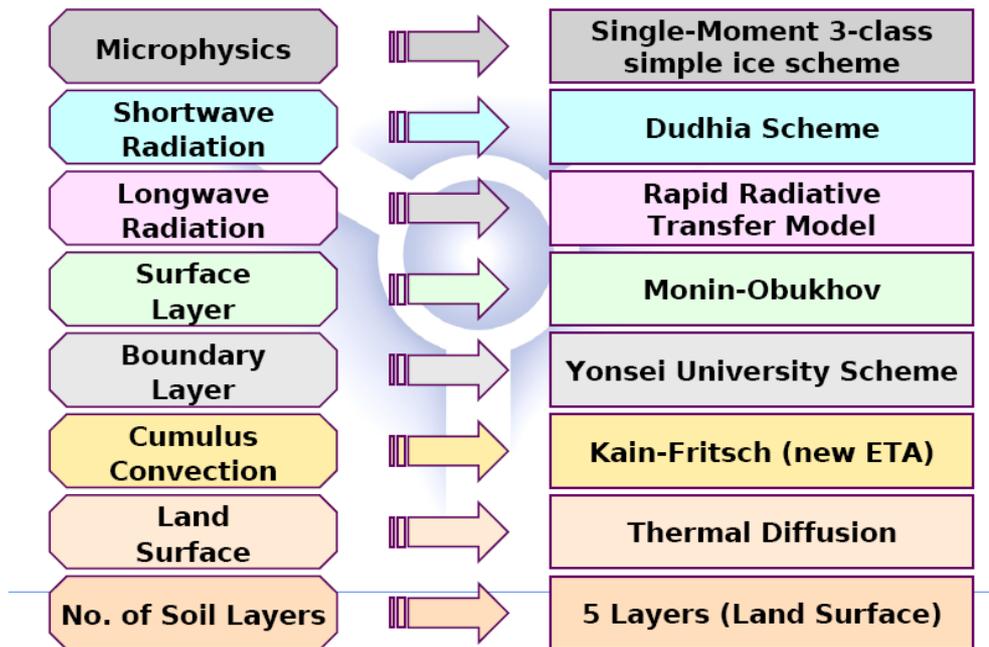


Figure 4.3.3: WISENT's WRF "off-the-shelf" physical parametrization schemes

The chosen postprocessing technique has been the Lagged Average Forecasting (LAF) technique, which relies upon gathering model forecasts that are verified at the same time, but were initiated at sequentially different times as shown schematically in Figure 4.3.4.

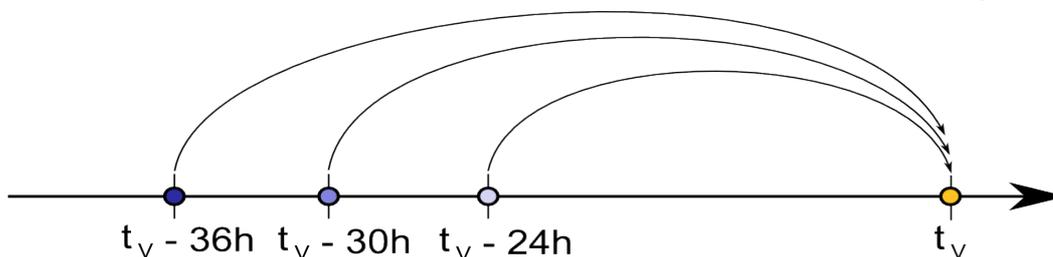


Figure 4.3.4: Schematic formulation of Lagged Average Forecasting components

The LAF technique has been applied in two formulations. The first (basic) one with emphasis on analysis errors comprised 19 LAF members of the same "off-the-shelf" physics scheme (Figure 4.3.5).

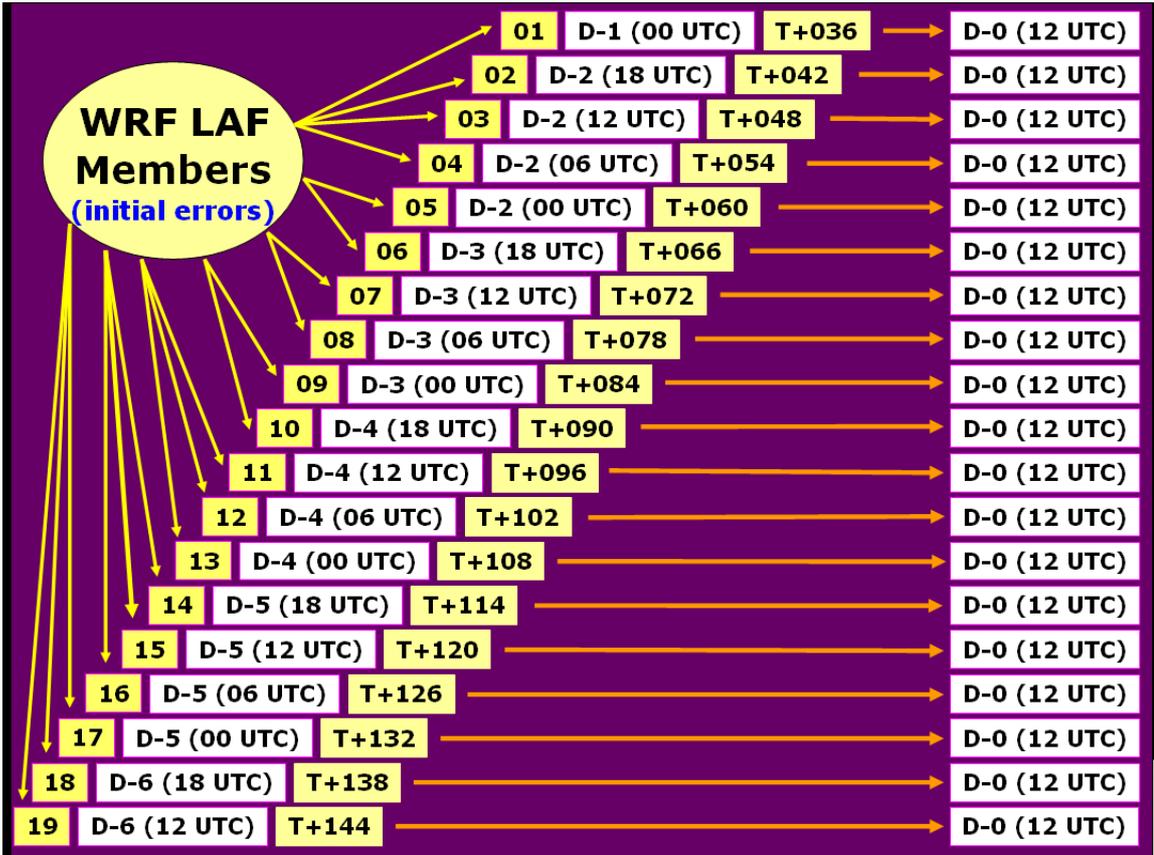


Figure 4.3.5: Schematic formulation of “off-the-shelf” WRF LAFs

The second WRF LAF formulation, allowed us to investigate the impact of utilizing 10 LAF members with different (multi) physical schemes, i.e. half of the members consisted of a different than the basic parametrization (Figure 4.3.6).

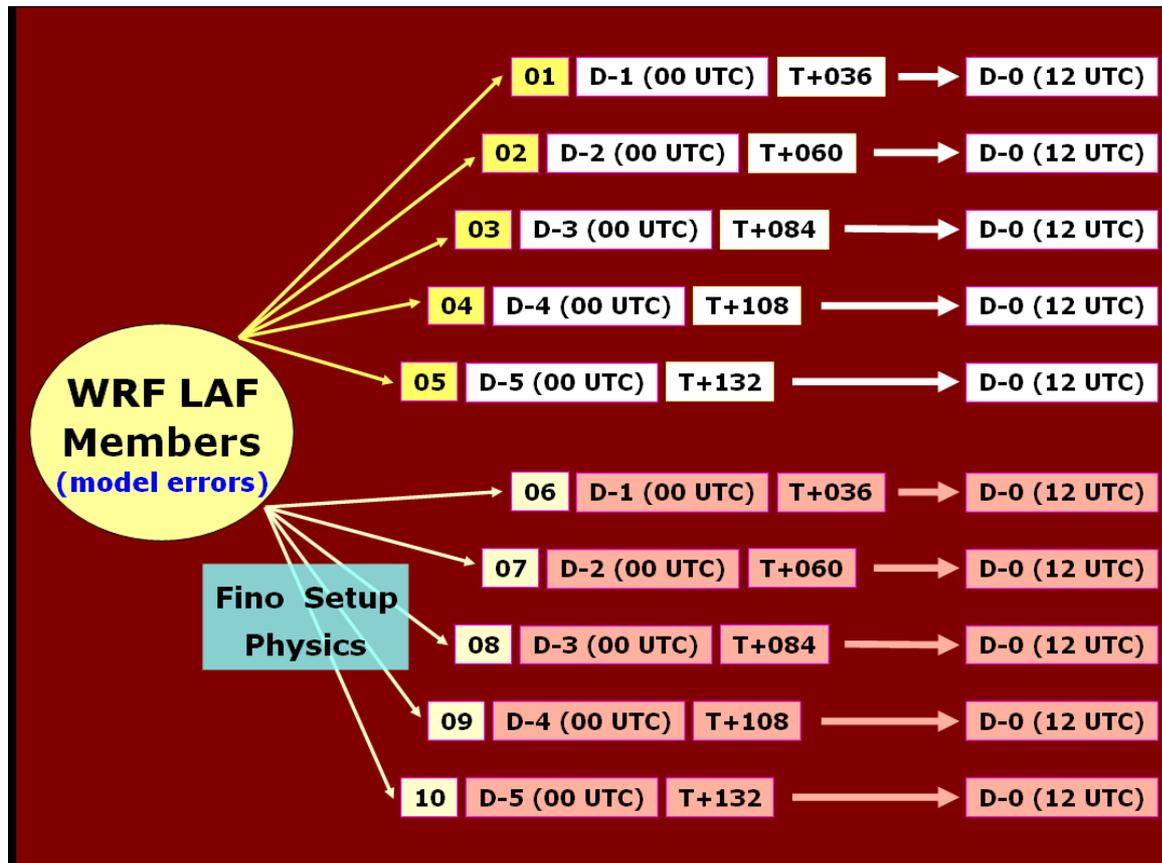


Figure 4.3.6: Schematic formulation of "multi-physics" WRF LAFs

WRF wind speed LAFs at the height of 80 meters (hub height) were constructed for the period of MAM (March, April & May) 2007. The lagging time interval was 6 hours (24 hours, for multi-physics case), while the maximum forecast horizon was set to 144 hours. Both LAF formulations have been verified for the time interval of 36 hours, over 11 wind farms of Germany (using NCEP analysis). EM denotes the Ensemble Mean of LAF members (same physics), while EM-M the EM of LAF members with different (multi-)physics.

Seasonal RMSE scores for each LAF member were compared against their own LAF Ensemble Mean's RMSE (Figure 4.3.7). For both cases EM & EM-M was found to be superior to their own individual members, providing a better forecast guidance for wind power prediction. EM's RMSE (1.77 m/s) resulted in a reduction of 37.9% compared to the mean RMSE (2.85 m/s) of all LAF members (average skill of all ensemble members using equal weights).

Likewise, EM-M resulted in a reduction of 34.6% compared to the Mean RMSE of all LAF-Ms.

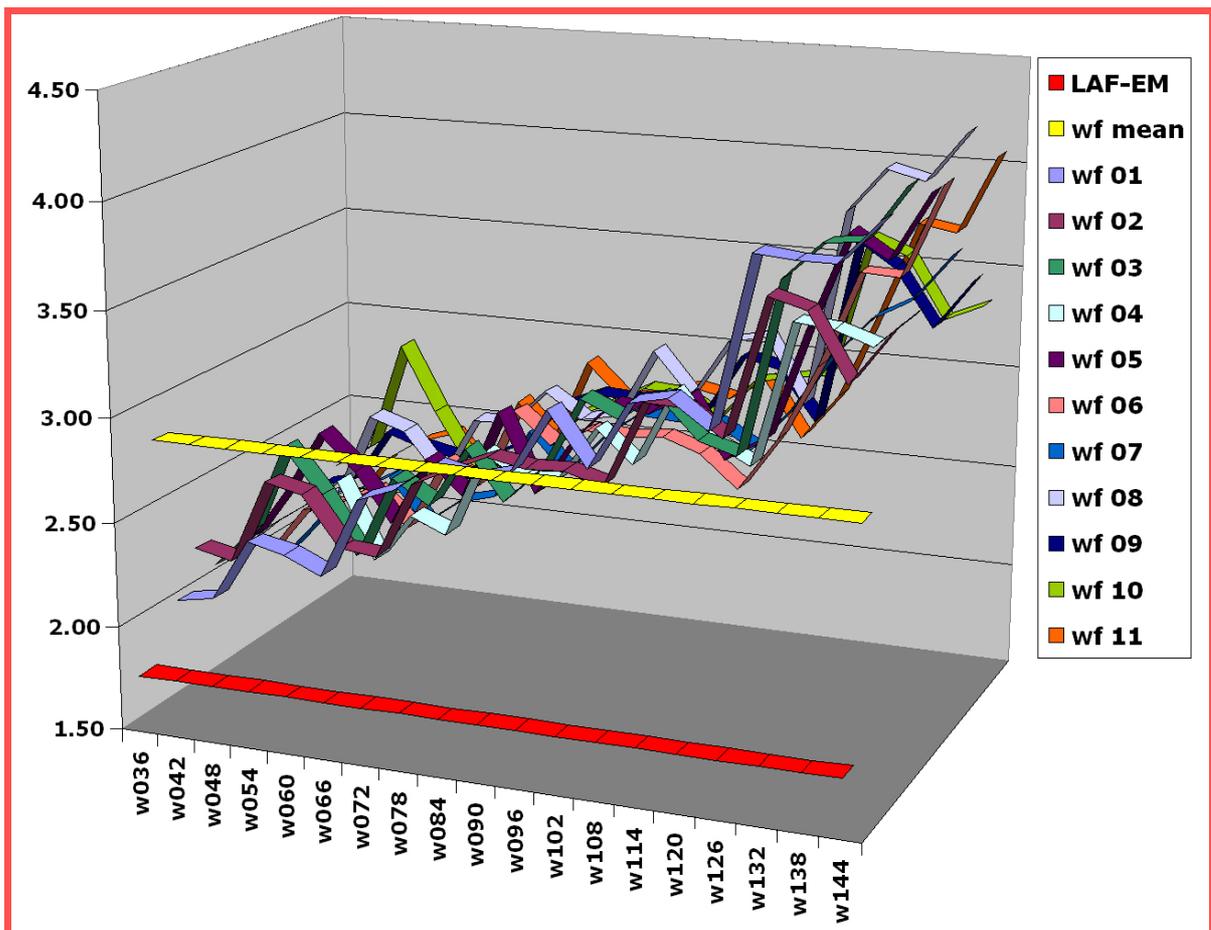


Figure 4.3.7: Seasonal RMSE for all WRF LAFs, Mean LAF & LAF-EM

Due to the limited number of ensemble members in every ensemble prediction system (EPS), the verifying analysis may lay outside the ensemble range. For instance, for in an ideal EPS with 19 members this will happen 2/20, that is ~10% of the time. In reality this number varies (in most of the cases the number is higher). This means that the EPS does not spread out sufficiently. A more detailed way of analyzing the EPS spread is to construct a so called Talagrand diagram (ranked histogram, see Figures 4.3.8 and 4.3.9). It is constructed from the notion that in an ideal EPS system the verifying analysis is equally likely to lie between any two ordered adjacent members, including the cases when the analysis will be outside the ensemble range on either side of the distribution. For every observation point and date all the members are ordered according to their forecast values, from the smallest to the biggest. For an EPS with 19 members, such as our WRF LAF, there are 20 intervals (bins) including the unlimited intervals at the edges of the distribution. The y-axis indicates the number of cases (relative to the total number) in which the analysis is found between two ordered adjacent members on the x-axis. If the diagram has a U-shape, this indicates that the ensemble does not spread out sufficiently. Likewise, the tendency to have a J-shape indicates that the system has a bias. In an ideal EPS system the long-term “Talagrand distribution” should be flat with equally many verifications in each interval. In reality the distribution is often slightly U-shaped due to over-representation of cases when the verification falls outside the ensemble and under-representation when it falls in the ensemble center. For some parameters the U-shape

degenerates into a (reversed) J-shape, which indicates that the system has a bias for this parameter.

For WRF LAF & LAF-M member validation, Talagrand bin diagrams were constructed, comprising 20 & 11 bins respectively (Figures 4.3.8 and 4.3.9). For LAFs, the outer two bins were found to contain 142 values out of 1012, i.e. 14% compared to an ideal of 10% (2/20). Most of the outer values (138 out of 142) were assigned to the outer left bin of the diagram (resulting in a reversed J-shape), reflecting the WRF model's tendency to overestimate wind speeds. This systematic error was removed by subtracting the seasonal (MAM 2007) bias. The seasonal bias was estimated by averaging the differences between forecast and corresponding analysis values for all forecasting intervals (on a seasonal basis). After the seasonal bias correction, the percentage of values falling in both outer bins dropped to 5.6% .

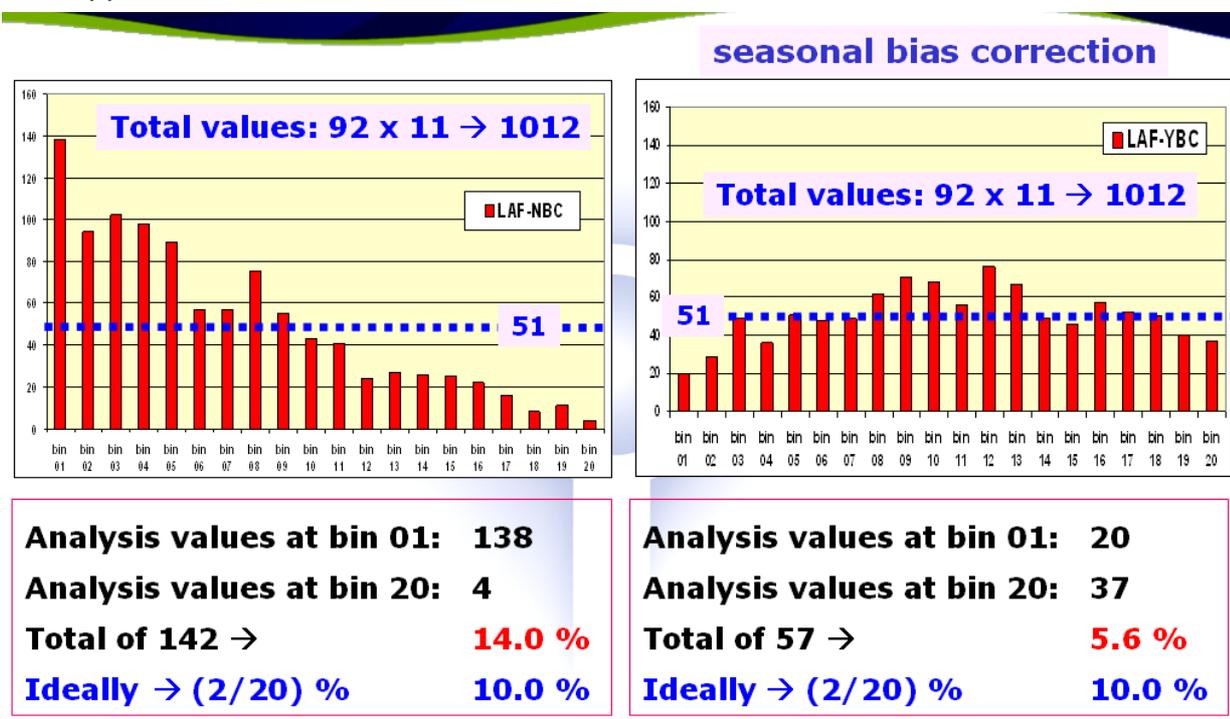


Figure 4.3.8 Talagrand bin diagrams for WRF LAFs (same physics)

For LAF-M forecasts, the outer bins were initially found to contain 277 values out of 1012, i.e. 27.4% (16.6% after bias correction), compared to an ideal of 18.2% (Figure 4.3.9).

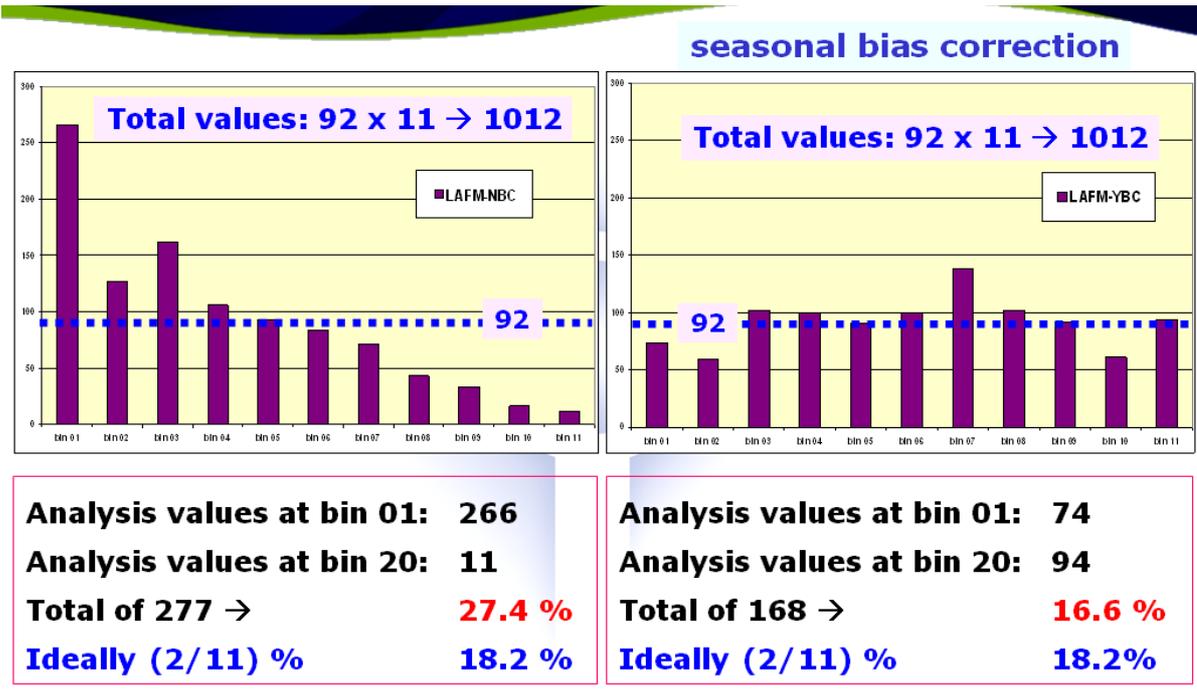


Figure 4.3.9 Talagrand bin diagrams for WRF LAF-Ms (multi-physics)

Another important issue has been the investigation of the improvement of NCEP's operational wind forecasts by combining “deterministic” LAF EMs with NCEP forecasts using equal weights, constructing “poor man/forecaster's” two member EPS & EPS-M schemes.

Although NCEP T+36 hour forecasts were better than LAF forecasts (i.e. better than LAF-EM) with a mean RMSE of 1.62 m/s, compared to 1.77 m/s of LAF's, the EPS final formulation was found to be superior of all, with a mean RMSE of 1.41 m/s leading to a 13% reduction of NCEP's error (Figure 4.3.10).

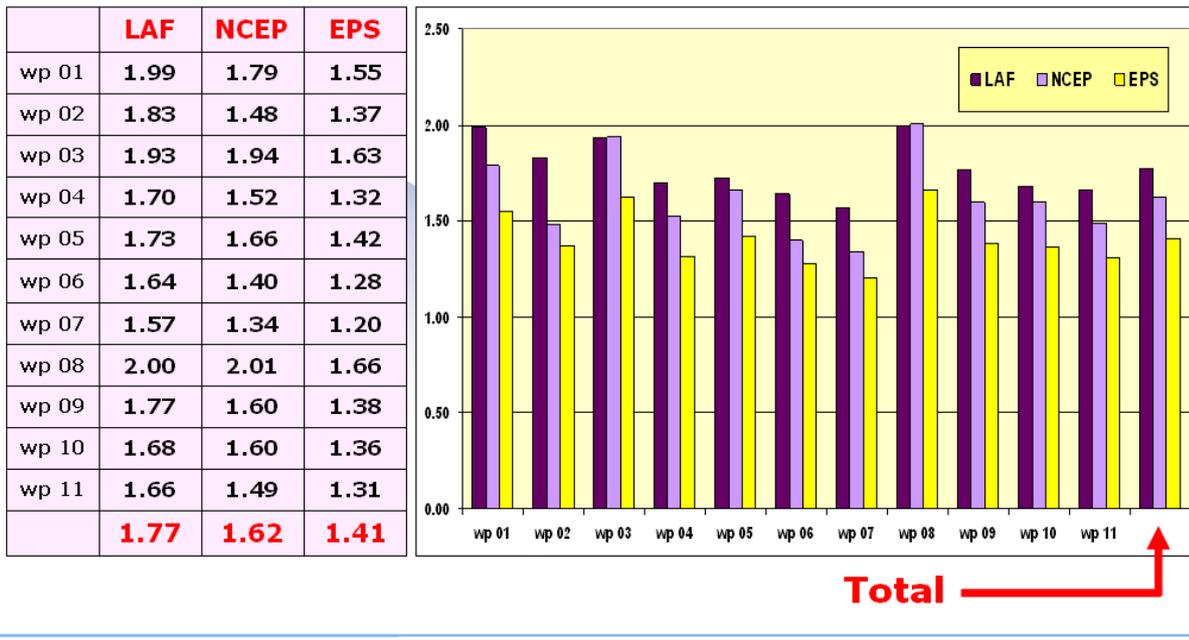


Figure 4.3.10 Seasonal RMSE for WRF LAF-EM, NCEP & “poor man/forecaster’s” EPS

LAF-M Ensemble Mean was found to be slightly worse than the corresponding LAF EM (with an RMSE of 1.80), but even so, the EPS-M formulation led to a 10% reduction of NCEP's original forecast error.

Summarizing, the LAF technique was applied on the German D-Grid WISSENT WRF model in two basic formulations, one with an emphasis on analysis errors (same physics setup) while the other (multi-physics setup) allowed for investigation into the impact of utilizing different physical schemes (note: both sources of error are not disjoint). Results are promising for both formulations.

Both Ensemble Means (EMs) were found to be superior to their members, i.e. both were capable to provide better forecast guidance. In addition, combining any of the “deterministic” EMs with the operational NCEP GFS deterministic forecast, has led to a better than the original GFS global forecast.

4.4 Combination of LAFs – construction of optimal ensemble schemes

Using WISSENT's WFP (Weather Forecasting Portal) a set of different LAF schemes were constructed from NCEP's GFS and WISSENT's WRF deterministic components.

- **LAF06**, the 6-member ensemble scheme constructed using the 6 most recent, lagged NCEP's GFS model forecasts (i.e. the forecasts started at the initial time, and 6, 12, 18, 24 and 30 hours earlier). In such a system, the most recent (GFS) forecast is considered as the “control” forecast (LAF06-Con). LAF06's ensemble-mean (LAF06-EM) is taken by giving the same weight (1/6) to all 6 LAF members.
- **LAF12**, the 12-member ensemble scheme constructed using the 12 most recent, lagged GFS model forecasts (i.e. the forecasts started at the initial time, and 6, 12, 18,

24, 30, 36, 42, 48, 54, 60 & 66 hours earlier). As in LAF06 formulation, the most recent GFS forecast is considered as the “control” forecast (LAF12-Con), while the ensemble-mean (LAF12-EM) is taken by giving the same weight (1/12) to all 12 LAF members.

- **WAF06**, the 6-member ensemble scheme constructed using the 6 most recent, lagged WISENT's WRF model forecasts. As in LAF06, the most recent WRF forecast is considered as the “control” forecast (WAF06-Con), while the ensemble-mean (WAF06-EM) is taken by giving the same weight (1/6) to all 6 WAF members.
- **WAF12**, the 12-member ensemble scheme constructed using the 12 most recent, lagged WISENT's WRF model forecasts. As in LAF12, the most recent WRF forecast is considered as the “control” forecast (WAF12-Con), while the ensemble-mean (WAF12-EM) is taken by giving the same weight (1/12) to all 12 WAF members.
- **CMB12**, the 12-member ensemble constructed by combining LAF06 and WAF06. In this scheme, the “control” forecast (CMB12-Con) is taken by averaging the two most recent GFS and WRF forecasts giving the same weight (1/2) to the members. The ensemble-mean (CMB12-EM) is taken by giving the same weight (1/12) to all 12 members.
- **CMB24**, as in CMB12 but for the 24-member ensemble constructed by combining LAF12 and WAF12. The “control” forecast (CMB24-Con) is calculated as in CMB12, while the ensemble-mean (CMB24-EM) is taken by giving the same weight (1/24) to all 24 members.
- **OLAF06**, as in LAF06 but the ensemble-mean (OLAF06-EM) is taken by utilizing normalized weights based on each member's seasonal (inversed) forecast skill. Once more, the most recent GFS forecast (OLAF06-Con) is considered as the “control” forecast (equal to LAF06-Con).
- **OLAF12**, as in LAF12 but the ensemble-mean (OLAF12-EM) is taken by utilizing normalized weights based on each member's seasonal (inversed) forecast skill. As in OLAF06, the most recent GFS forecast (OLAF12-Con) is considered as the ‘control’ forecast (equal to LAF12-Con).
- **OWAF06**, as in WAF06 but the ensemble-mean (OWAF06-EM) is taken by calculating normalized weights based on each member's seasonal (inversed) forecast skill. The most recent WRF forecast (OWAF06-Con) is considered as the “control” forecast (equal to WAF06-Con).
- **OWAF12**, as in WAF12 but the ensemble-mean (OWAF12-EM) is taken by utilizing normalized weights based on each member's seasonal (inversed) forecast skill. As in OWAF06, the most recent WRF forecast (OWAF12-Con) is considered as the “control” forecast (equal to WAF12-Con).
- **OPT12**, as in CMB12 but combining OLAF06 and OWAF06 (utilizing equal weights). In this scheme, the “control” forecast (OPT12-Con) is taken by combining the two most recent GFS (OLAF06-Con) and WRF (OWAF06-Con) forecasts utilizing normalized weights based on each member's seasonal (inversed) forecast skill. As in CMB12, the ensemble-mean (OPT12-EM) is taken by giving the same weight (1/12) to all 12 members.
- **OPT24**, as in CMB24 but combining OLAF12 and OWAF12 (utilizing equal weights). As in OPT12, the “control” forecast (OPT24-Con) is taken by combining OLAF12-Con

and OWAF12-Con utilizing normalized weights based on each member's seasonal (inversed) forecast skill. As in CMB24, the ensemble-mean (OPT24-EM) is taken by giving the same weight (1/24) to all 24 members.

- **MOPT12**, as in OPT12 but the “control” forecast (MOPT12-Con) and the ensemble mean (MOPT12-EM) are taken by utilizing optimal weights calculated by minimizing the squared residuals of combination.
- **MOPT24**, as in OPT24 but the “control” forecast (MOPT24-Con) and the ensemble mean (MOPT24-EM) are taken by utilizing optimal weights calculated by minimizing the squared residuals of combination.

The setup for all WRF deterministic components is described in the previous subsection, corresponding to the “off-the shelf” physics formulation. A two-domain, two-way nested set-up, with a parent to child ratio 1:3 (30 & 10 km) was utilized centered over Germany. WRF LAFs at the height of 80 meters were constructed for the period of MAM (March, April & May) 2007.

LAF formulations have been verified for all forecast intervals, i.e. from 6 to 120 hours over 11 wind farms of Germany. This can be considered as an extension of the verification which took place according to Section 4.3, where the estimation of the forecast skill of NCEP and WRF LAFs focused only on the 36 hour forecast horizon.

During the investigation, the total forecast time interval of 120 hours (Total-Range Forecast interval, TO-RFO) was conveniently split into three sub-intervals:

- (a) Very Short-Range FOrecast interval (VS-RFO), defined from 6 to 36 hours.
- (b) ShOrt-Range FOrecast interval (SO-RFO), defined from 36 to 72 hours
- (c) Early Medium-Range FOrecast interval (EM-RFO), defined from 72 to 120 hours

By construction,

- OLAF06 is very similar to LAF06 since they share the same “control” forecast and the same members, but they have different ensemble means (i.e LAF06-EM is different from OLAF06-EM).
- OLAF12 is very similar to LAF12 since they share the same “control” forecast and the same members, but they have different ensemble means (i.e LAF12-EM is different from OLAF12-EM).
- OWAF06 is very similar to WAF06 since they share the same “control” forecast and the same members, but they have different ensemble means (i.e WAF06-EM is different from OWAF06-EM).
- OWAF12 is very similar to WAF12 since they share the same “control” forecast and the same members, but they have different ensemble means (i.e WAF12-EM is different from OWAF12-EM).
- OPT12 is similar to CMB12 since they share same members, but they have different ensemble means (i.e CMB12-EM is not equal to OPT12-EM) and different “control” forecasts (i.e CMB12-Con is not equal to OPT12-Con).

- OPT24 is similar to CMB24 since they share same members, but they have different ensemble means (i.e CMB24-EM is not equal to OPT24-EM) and different “control” forecasts (i.e CMB24-Con is not equal to OPT24-Con).
- MOPT12 is similar to OPT12 since they share same members, but they have different ensemble means (i.e OPT12-EM is not equal to MOPT12-EM) and different “control” forecasts (i.e OPT12-Con is not equal to MOPT12-Con).
- MOPT24 is similar to OPT24 since they share same members, but they have different ensemble means (i.e OPT24-EM is not equal to MOPT24-EM) and different “control” forecasts (i.e OPT24-Con is not equal to MOPT24-Con).

The best (optimal) LAF ensemble scheme is determined by inter-comparisons among all possible LAF formulations for the period of MAM (March, April and May) 2007.

All speed values have been transformed to the hub height (80 meters), while the average performance of each individual formulation has been assessed considering the level of ensemble spread and the accuracy of single (control, ensemble-mean and LAF members) and probabilistic forecasts. Talagrand bin diagrams (rank histograms) were also used in order to determine the percentage of the analysis outside the ensemble range on either side of the distribution. Initial results focusing mainly on the “deterministic” side (as the “control” and the ensemble mean forecasts) of all available schemes are summarized in Tables 4.4.1, 4.4.2, 4.4.3, 4.4.4, 4.4.5, 4.4.6 & 4.4.7. The values are the Root Mean Square Error in m/s for the single control forecast and for the different examined ensemble-mean forecasts.

	LAF-Control	LAF06	LAF12
L006	0.82	1.00	1.17
L012	0.94	1.11	1.25
L018	1.12	1.21	1.33
L024	1.27	1.31	1.41
L030	1.37	1.40	1.49
L036	1.52	1.49	1.57
L042	1.71	1.59	1.63
L048	1.78	1.68	1.69
L054	1.88	1.76	1.76
L060	2.08	1.82	1.83
L066	2.08	1.88	1.89
L072	2.28	1.95	1.95
L078	2.24	1.98	2.00
L084	2.35	2.02	2.06
L090	2.48	2.09	2.11
L096	2.50	2.18	2.17
L102	2.60	2.26	2.23
L108	2.58	2.31	2.28
L114	2.62	2.37	2.33
L120	2.84	2.45	2.39
VS-RFO	1.17	1.25	1.37
SO-RFO	1.90	1.74	1.76
EM-RFO	2.50	2.18	2.17
TO-RFO	1.86	1.72	1.77

Table 4.4.1: NCEP GFS LAFs –
seasonal skill over 11 wind farms (North Germany)

	WAF-Control	WAF06	LAF12
W006	1.27	1.38	1.44
W012	1.39	1.51	1.50
W018	1.60	1.63	1.58
W024	1.75	1.71	1.62
W030	1.92	1.75	1.68
W036	2.00	1.77	1.74
W042	2.07	1.80	1.80
W048	2.22	1.86	1.85
W054	2.26	1.91	1.90
W060	2.22	1.95	1.95
W066	2.27	2.03	2.01
W072	2.34	2.11	2.08
W078	2.49	2.17	2.13
W084	2.56	2.22	2.19
W090	2.70	2.28	2.26
W096	2.78	2.36	2.33
W102	2.87	2.41	2.40
W108	2.90	2.47	2.45
W114	2.90	2.51	2.51
W120	3.03	2.61	2.61
VS-RFO	1.65	1.62	1.59
SO-RFO	2.20	1.92	1.90
EM-RFO	2.73	2.35	2.33
TO-RFO	2.19	1.96	1.94

Table 4.4.2: WISENT's WRF LAFs (WAFs) – seasonal skill over 11 wind farms (North Germany)

	CMB-Control	CMB12	CMB24
C006	0.95	1.09	1.20
C012	1.06	1.19	1.28
C018	1.22	1.30	1.36
C024	1.36	1.39	1.42
C030	1.47	1.45	1.50
C036	1.57	1.51	1.57
C042	1.71	1.58	1.64
C048	1.80	1.66	1.70
C054	1.89	1.73	1.76
C060	1.96	1.79	1.83
C066	1.98	1.87	1.89
C072	2.14	1.95	1.96
C078	2.20	2.00	2.02
C084	2.29	2.05	2.08
C090	2.42	2.12	2.13
C096	2.48	2.21	2.19
C102	2.58	2.28	2.25
C108	2.59	2.34	2.29
C114	2.62	2.40	2.34
C120	2.79	2.47	2.41
VS-RFO	1.27	1.32	1.39
SO-RFO	1.87	1.73	1.77
EM-RFO	2.46	2.20	2.19
TO-RFO	1.87	1.75	1.78

Table 4.4.3: Combined LAFs & WAFs (CMBs) –
seasonal skill over 11 wind farms (North Germany)

	OLAF-Control	OLAF06	OLAF12
N006	0.82	0.95	1.05
N012	0.94	1.07	1.15
N018	1.12	1.18	1.25
N024	1.27	1.28	1.34
N030	1.37	1.37	1.43
N036	1.52	1.47	1.52
N042	1.71	1.57	1.60
N048	1.78	1.66	1.66
N054	1.88	1.74	1.73
N060	2.08	1.81	1.80
N066	2.08	1.87	1.86
N072	2.28	1.94	1.92
N078	2.24	1.97	1.97
N084	2.35	2.02	2.03
N090	2.48	2.09	2.09
N096	2.50	2.17	2.15
N102	2.60	2.24	2.21
N108	2.58	2.30	2.26
N114	2.62	2.36	2.33
N120	2.84	2.44	2.42
VS-RFO	1.17	1.22	1.29
SO-RFO	1.90	1.73	1.73
EM-RFO	2.50	2.17	2.16
TO-RFO	1.86	1.71	1.73

Table 4.4.4: LAFs with normalized weights (OLAFs) – seasonal skill over 11 wind farms (North Germany)

	OWAF-Control	OWAF06	OWAF12
R006	1.27	1.33	1.37
R012	1.39	1.47	1.45
R018	1.60	1.61	1.54
R024	1.75	1.70	1.60
R030	1.92	1.74	1.66
R036	2.00	1.76	1.72
R042	2.07	1.79	1.77
R048	2.22	1.85	1.83
R054	2.26	1.90	1.87
R060	2.22	1.94	1.93
R066	2.27	2.02	1.98
R072	2.34	2.10	2.05
R078	2.49	2.16	2.11
R084	2.56	2.20	2.19
R090	2.70	2.27	2.29
R096	2.78	2.35	2.40
R102	2.87	2.40	2.52
R108	2.90	2.46	2.63
R114	2.90	2.50	2.75
R120	3.03	2.63	2.90
VS-RFO	1.65	1.60	1.56
SO-RFO	2.20	1.91	1.88
EM-RFO	2.73	2.34	2.43
TO-RFO	2.19	1.95	1.95

Table 4.4.5: WAFs with normalized weights (OWAFs) – seasonal skill over 11 wind farms (North Germany)

	OPT-Control	OPT12	OPT24
P006	0.91	1.05	1.12
P012	1.02	1.16	1.21
P018	1.18	1.28	1.30
P024	1.32	1.37	1.38
P030	1.43	1.44	1.46
P036	1.54	1.50	1.54
P042	1.69	1.57	1.61
P048	1.78	1.65	1.67
P054	1.87	1.73	1.74
P060	1.96	1.78	1.80
P066	1.98	1.86	1.87
P072	2.14	1.94	1.94
P078	2.19	1.99	1.99
P084	2.29	2.04	2.06
P090	2.42	2.11	2.14
P096	2.48	2.20	2.21
P102	2.57	2.27	2.29
P108	2.58	2.33	2.36
P114	2.61	2.38	2.43
P120	2.79	2.48	2.52
VS-RFO	1.23	1.30	1.33
SO-RFO	1.85	1.72	1.74
EM-RFO	2.45	2.19	2.22
TO-RFO	1.84	1.74	1.76

Table 4.4.6: Combined OLAFs & OWAFs (OPTs) - Seasonal skill over 11 wind farms (North Germany)

	MOPT-Control	MOPT12	MOPT24
M006	0.86	0.96	1.05
M012	0.97	1.07	1.15
M018	1.13	1.18	1.24
M024	1.27	1.28	1.33
M030	1.37	1.36	1.42
M036	1.49	1.45	1.51
M042	1.66	1.55	1.58
M048	1.73	1.64	1.65
M054	1.83	1.71	1.72
M060	1.97	1.78	1.79
M066	1.98	1.85	1.85
M072	2.16	1.93	1.92
M078	2.17	1.96	1.97
M084	2.28	2.01	2.03
M090	2.41	2.08	2.09
M096	2.45	2.17	2.15
M102	2.55	2.24	2.22
M108	2.54	2.29	2.27
M114	2.58	2.36	2.33
M120	2.78	2.44	2.41
VS-RFO	1.18	1.22	1.28
SO-RFO	1.83	1.70	1.72
EM-RFO	2.44	2.16	2.16
TO-RFO	1.82	1.69	1.72

Table 4.4.7: OPTs utilizing optimal weights (MOPTs) - Seasonal skill over 11 wind farms (North Germany)

It should be pointed out that seasonal bias correction has been applied to all schemes, while all negative values (if any after correction) have been set to zero. An example of different stages of forecast error reduction by applying different postprocessing techniques is given in Figure 4.4.1.

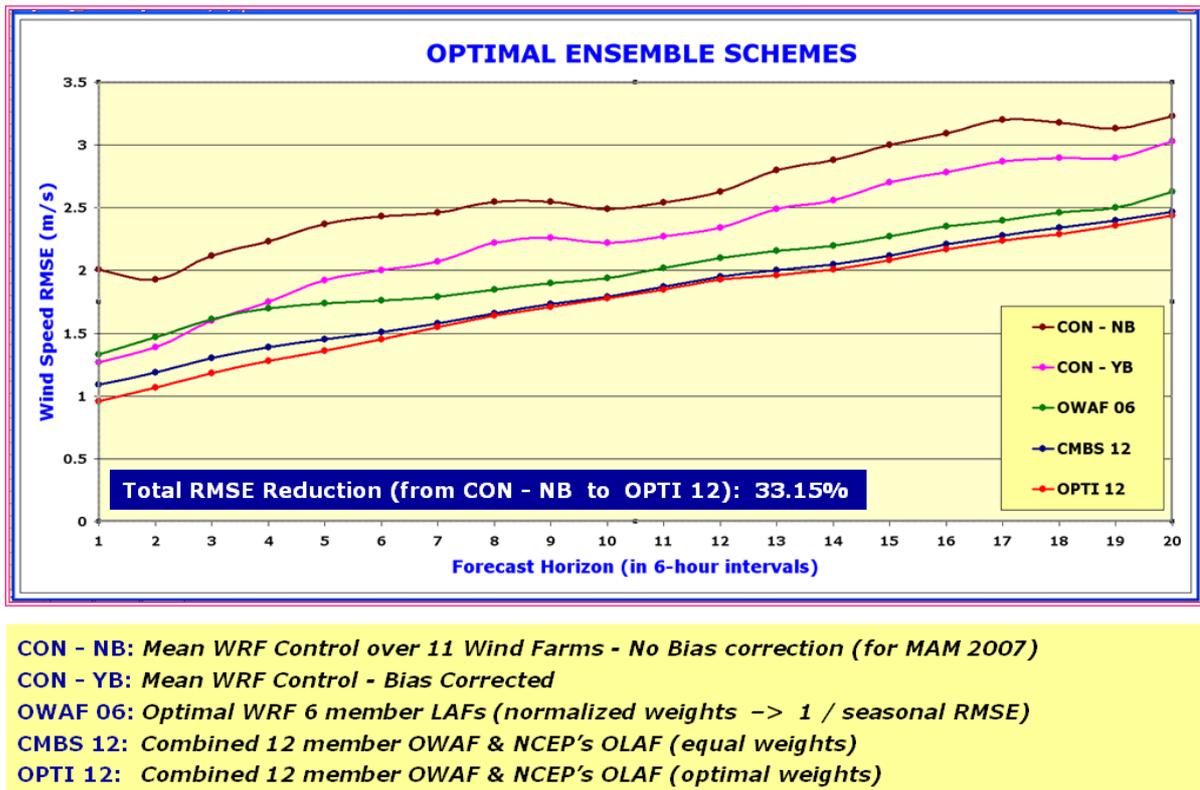


Figure 4.4.1: WISENT WRF WAFs – seasonal skill over 11 wind farms (North Germany)

In Figure 4.4.1, the RMSE of WRF control forecast before bias correction is represented by the brown line (top curve), while the RMSE of bias corrected WRF control is given by the pink line (2nd from the top). The green line (3rd from top) represents OWAF06-EM's RMSE, while the blue (4th from the top) represents the RMSE of CMB12-EM. Finally, the red line (at the bottom) represents OPT12-EM's RMSE. The total reduction of WRF's forecast RMSE from initial stage (no bias correction) to final OPT12-EM (combining OWAF & OLAF and utilizing optimal weights) is equal to 33.15%

Even in the early stages of our investigation, it has become evident that for all weighted ensemble schemes, most of the weight should be given to the most recent forecast ("control"). This was introduced eventually during the formulation of OLAF and OWAF schemes by assigning weights to every ensemble member proportional to the inverse of its seasonal RMSE value.

It should be pointed out that weights have been used only in the definition of the ensemble-mean and not in the definition of the forecast probability distribution functions. Preliminary results indicate that weighted schemes as OLAF06, OLAF12, OWAF06 and OWAF12 have in general superior characteristics compared to the schemes using same "control" and forecast members but different (unweighted) ensemble means (i.e. LAF06, LAF12, WAF12 and WAF24 respectively).

In this framework, OLAF06-EM and OLAF12-EM are superior to LAF06-EM and LAF12-EM respectively for VS-RFO, SO-RFO, and EM-RFO forecast intervals. Likewise OWAF06-EM and OWAF12-EM are superior to WAF06-EM and WAF12-EM respectively for all forecast intervals.

Furthermore, OPT12-EM and OPT24-EM are superior to CMB12-EM and CMB24-EM respectively for all forecast intervals. MOPT12-EM and MOPT24-EM also are superior to OPT12-EM and OPT24-EM respectively for all forecast intervals.

As the quality of the “control” forecast is concerned for the VS-RFO interval, the best forecast guidance is given by LAF06-Con (equal to LAF12-Con, OLAF06-Con and OLAF12-Con). As for the remaining SO-RFO and EM-RFO intervals, the best forecast prediction is provided by MOPT12-Con (equal to MOPT24-Con).

Finally, initial results show that MOPT24 ensemble (optimal combined scheme) has the largest spread than any other scheme, while at the same time it matches more to the error of its corresponding ensemble mean (MOPT24-EM), revealing a more harmonized (tuned) overall performance. It also provided us with the most “reliable” set of probabilistic products.

4.5 Future work

Forecasts of wind conditions and related power generation from a few hours to a few days ahead are paramount for various management tasks related to the integration of wind generation in power systems. A major concern expressed by end-users is related to the accuracy of wind power predictions, whose level is highly variable and too low on average according to their expectations. Especially, the prediction error may reach very high level in case of, for example, fronts or wind speeds near cut-off speed. Depending on the sensitivity of end-users, the quality of the forecasts is not adequate for about 10-20% of times. These are the situations which significantly reduce the value and acceptability of wind power due to their impact on the grid, on penalties when trading, and other consequences.

Although current forecasting technology focuses mainly on deterministic models for the wind power prediction, the concept of complementary tools that can be used jointly to traditional forecasts to assess wind predictability has emerged as an important issue. In situations with low predictability the development of probabilistic wind power forecasts becomes very important and requires probabilistic information for decision making.

Based on results from Section 4.4, the development of a Combined Prediction System (CPS) comprising deterministic and probabilistic meteorological components for skillful, probabilistic wind power predictions is possible based on an optimal combination of different forecast components as:

- NCEP GFS deterministic global forecasts (hor. resolution ~70km)
- NCEP GFS LAFs (6- and 12-member ensemble schemes)
- WRF deterministic high-resolution (mesoscale) forecasts (hor. resolution ~5km)
- WRF LAFs (6- and 12- member ensemble schemes)
- ECMWF IFS deterministic global forecasts (hor. resolution ~25km)
- ECMWF Ensemble Prediction System (EPS), comprising 51 members (res. ~50km)

The optimally combined meteorological forecast system (CPS) linked to wind power forecasting tools and improvements in wind power forecasts should be evaluated on all relevant time scales that are interesting to end-users, such as wind farm operators, transmission system operators and energy traders. The forecast horizon of such a CPS

can be split on different time ranges as defined in sub-section 4.4, i.e. VS-RFO (Very Short-Range FOrecast interval, from 06 to 36 hours), SO-RFO (ShOrt-Range FOrecast interval, from 36 to 72 hours) and EM-RFO (Early Medium-Range FOrecast interval, from 72 to 120 hours) utilizing all different components listed above. Another (extended) time interval can be defined from 120 to 180 hours (M-RFO) utilizing GFS deterministic, GFS probabilistic LAFs, ECMWF IFS deterministic, and ECMWF EPS probabilistic components.

Such a combined system (CPS) can provide probabilistic forecasts which will be superior to the probabilistic forecasts based on any of its components. Special attention should be paid also to forecast extreme events in the early medium-range (EM-RFO time interval) more accurate to facilitate the integration of wind power in the power system in any weather situation. All CPS configurations can be evaluated using newly developed probabilistic skill score measures that are based on observations of different sources. The evaluation should include a range of probabilistic skill scores like Brier Skill Score, ROC Area, Reliability Diagram, and Ignorance Score. Each of the scores will be used to diagnose specific aspects of the quality of the wind power forecast system. Emphasis should be given to the combination of high resolved deterministic forecast products with ensemble predictions by minimizing probabilistic skill scores.

Furthermore, it is recognized that the operation of power systems with a significant share of wind power capacities cannot be optimal if information on uncertainty does not accompany the usually provided point predictions. It would be beneficial to invest research efforts of better estimating and communication the uncertainty of wind power forecasts, in the range of prediction horizons going from few hours to few days ahead. The use of the Ensemble Prediction Systems (EPS) in numerical weather prediction is well established for global models like the ECMWF EPS. It provides useful information on uncertainty of the weather prediction for many different end-users and the public. Most important is the Extreme Forecast Index (EFI) that is established at ECMWF since many years and warns for extreme weather events (mainly storm) in the medium-range (3-5 days).

Probabilistic forecasts of wind generation still are a pretty new form of predictions, and few of the methods described in the state-of-the-art are actually implemented for providing end-users with situation-dependent estimation of forecast uncertainty. The main concern of these forecast users relates to the extremes of probability distributions, i.e. to the ability of probabilistic predictions to give reliable information on potential extremes forecast errors. Focus should be given to methods for probabilistic forecasting of wind generation, with particular emphasis on the tails of predictive distributions.

Finally, wind resource assessment is almost exclusively based on determining long-term wind statistics for a given location or over an area where it is envisaged to install wind capacities. However in a liberalized market environment, the revenue from the selling of wind generation on the market is significantly reduced by the costs for regulation resulting from deviations between contracted and actually delivered volumes of energy. In addition, prediction errors that are strongly correlated over an area result in large imbalances to be faced by the grid operator. Therefore, evaluating the resource in terms of wind power should also include an assessment of the predictability for a site or a region of interest.

5 The WISENT platform for executing WRF models

This section describes the technical details and insights from WRF deployment throughout the WISENT project. This work led to the establishment of a semi-operational Grid-based platform for executing WRF. Our goal has been to make WRF available to the energy meteorology community on a wide scale. Technically, this meant deployment beyond a single high-performance computing cluster or supercomputer, as it typically happens for smaller research groups. Instead, we harnessed the distributed computing and storage resources made available to our project participants through the German Grid (D-Grid) initiative.

The platform developed in WISENT is referred to as “semi-operational” because it supports the execution of customized model runs on demand rather than the regular, fully automated computations typical for an operational platform. This property results mostly from the research-centric character of the work performed in WISENT Work Package 5.2, which produced significant, but sporadic resource demands corresponding to the individual research studies (described in Section 4). Accordingly, the development work was focused on individual, small tools that could be reliably combined to address our users' immediate research needs rather than on a monolithic system with goal of supporting a single application. Importantly, this bottom-up approach offered advantages of risk mitigation in the presence of evolving requirements and helped to keep the development process user-driven, with short delivery times for requested features and minimal communication overheads.

In the later phases of the work package, with future requirements in mind, we have also developed a prototypical framework for daily mesoscale weather forecasting relying on the model configurations and techniques that we examined. However, at the time of writing the operational framework has not been tested extensively yet.

Section 5.1 introduces the high-level WRF workflow which guided our implementation efforts. Section 5.2 describes the performance benchmarking system developed for the purpose of initial testing of distributed WRF model runs and selecting the best configurations and hardware resources. Section 5.3 presents the “back-end” software tools developed to support executing WRF models and postprocessing their output. Section 5.4 briefly describes the prototype of our WISENT Weather Forecasting Portal, which offers an elegant medium for disseminating data products such as application-specific weather forecasts.

5.1 Grid-based WRF workflows

Figure 5.1.1 contains an overview of the WRF semi-operational platform realized throughout the work package. The numbered arrows indicate the data flows that need to occur to execute a typical computational workflow involving WRF. The boxes represent various participating software components and the stick figures represent the different user roles interacting with the system.

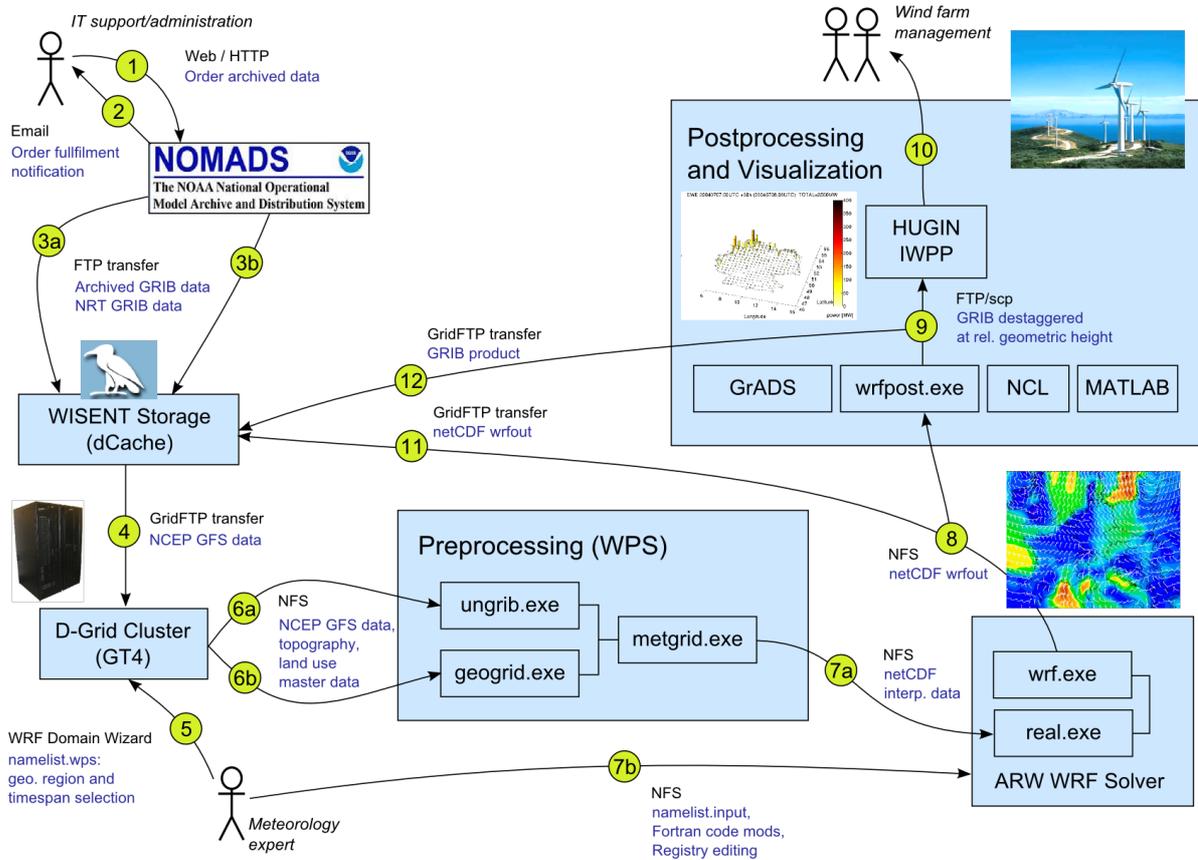


Figure 5.1.1: A typical WRF workflow in WISENT

The workflow begins with a request for the global input data from the NOAA Operational Model Archive Distribution System (NOMADS) server located in the US. Steps 1-2 in the figure exemplarily show a manually performed request for archived data using the NOMADS web interface (<http://nomads.ncdc.noaa.gov/>). Such a request results in transferring the data from offline storage to an FTP server at NCEP followed by an email notification to the requester.

In contrast to archived data, recent data can be procured automatically by a continuously running, fault-tolerant daemon script, which regularly polls the server. In general, fresh GFS data become available four times a day, corresponding to the 4 analysis hours (00 UTC, 06 UTC, 12 UTC, and 18 UTC).

In Step 3, the data (1.5-3 GB per day) is transferred to one of the dCache storage systems configured for use by the WISENT project (located in Jülich and Hannover). This step is intended to improve the input data availability. The central NCEP system, being globally available, is sometimes overloaded or suffers outages. It is therefore more reliable and faster to fetch the data using GridFTP from the dCache sites rather than via HTTP from the original source when they are needed by WISENT jobs (Step 4).

Step 5 corresponds to the manual model configuration required before execution. This step is performed by an expert user familiar with WRF and its configuration. Importantly, it includes choosing the nested geographical domains over which the simulation should be performed.

Step 6, also specific to WRF, provides the input data to the preprocessing stage (cf. Section 3.2), which involves executing three programs to convert the data into a format suitable for the main model run. Horizontal interpolation of the raw data to the user-defined grid is performed during this step. The WPS components are shown together because they share a common characteristic of being I/O intensive with relatively modest computational demands. They are executed in sequence using a single-processor job.

Step 7 delivers the WPS output to the ARW WRF solver (`wrf.exe`), after performing vertical data interpolation (`real.exe`). The numerical integration by the ARW WRF solver is the compute-intensive part of the workflow. The resource demands are highly-dependent on the desired grid size, resolution (distance between grid points), and time step. For cases examined in WISENT, model runs required multi-hour computations utilizing from 8 to 100+ processors.

Step 8 forwards the output data to the postprocessing stage and for archival in the the dCache storage sites (Step 11 and 12). The postprocessing involves computing additional output variables not provided by the native model, such as the wind speed at the wind turbine hub level (80m). Furthermore, the postprocessing may include visualization using tools such as GrADS or NCL or manual data examination using MATLAB.

Step 9 supplies the postprocessed data to the HUGIN system (cf. Section 4.3), capable of converting wind speed predictions into energy output predictions. In Step 10 the resulting data is delivered to the end users. Unlike the preceding steps, for which an actual technical implementation was possible during WISENT, the implementation of Steps 9 and 10 was prevented by organizational and resource constraints.

5.2 WRF performance benchmarks in D-Grid

WRF model code can be compiled in multiple ways, using shared memory (MPI) parallelism, distributed memory parallelism (OpenMP), or a hybrid combination of both. Even though WRF is a widely deployed model, little data exist concerning the recommended software and hardware configurations. Users have to learn by guessing or stay with an initially compiled, suboptimal (but working) configuration. Likewise, because WRF supports both MPI and OpenMP, the IT decision makers responsible for purchasing hardware miss recommendations on whether to invest in a fast network interconnect, prefer more nodes with a slower network, or maybe fewer, more powerful multi-processor nodes. Benchmark results provided by WRF developers concentrate on high-end supercomputing hardware.

After an initial evaluation of model configurations in the local cluster, we took advantage of our access to the heterogeneous hardware resources provided by D-Grid to empirically measure the model performance and scalability using different configurations. The benchmarking system developed in WISENT provides better guidance for users of Linux clusters, similar to those found at many D-Grid sites.

Another motivation for developing a WRF benchmarking system has been its capability of executing typical test cases, thus establishing the appropriateness of the available D-Grid resources for our applications. Benchmark cases that obtained expected results with acceptable performance cleared the way to running more complicated, real-world models in D-Grid.

It is important to note that users cannot be compelled to use the Grid by scalability arguments alone. They expect that Grid jobs execute at least as fast and reliably as local jobs and are reluctant to move their jobs to the Grid if tradeoffs in service quality are involved.

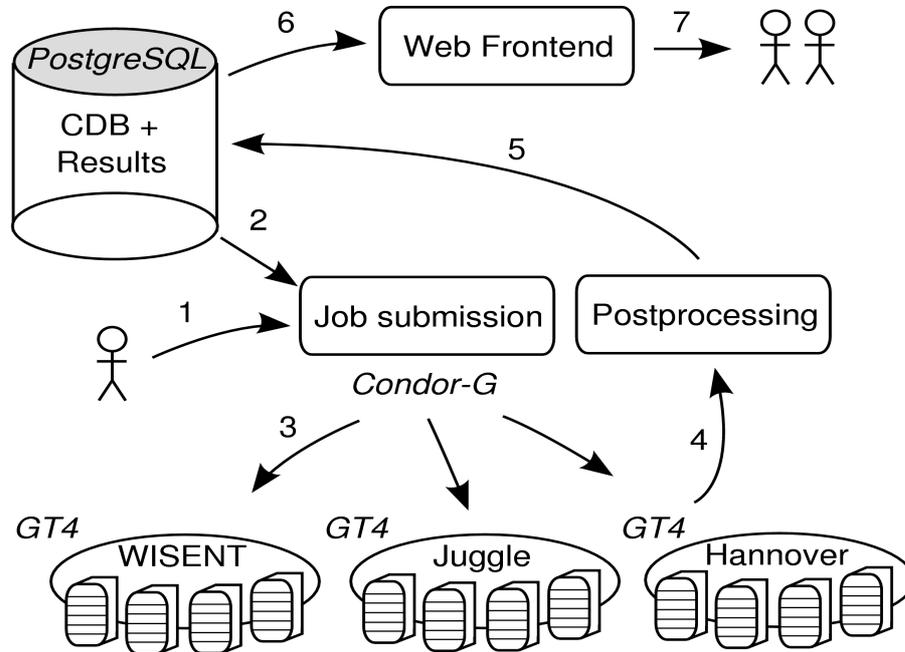


Figure 5.2.1: Main workflow steps of our WRF benchmarking system

Figure 5.2.1 illustrates the overall architecture and the main workflow steps involved in using the WRF benchmarking system. The operator starts a backend script to request the execution of multiple benchmark run jobs (Step 1). Each job is addressed explicitly to a target Grid site, references a desired bench-mark case, the model variant to use (e.g., a version of WRF compiled with MVAPICH), and the number of processors. The information about the availability of model variants and benchmark cases at Grid sites is retrieved from the configuration database (Step 2). The job is submitted to a central Condor-G queue, from which it is forwarded using WS GRAM to the target Globus Toolkit 4 site (Step 3). The job scripts, including the supporting module, are staged in and the benchmark case is run at the target site. After the execution completes, an XML file describing the individual run is staged out (Step 4). A periodic postprocessing job inserts data from the XML file into the results database (Step 5). The data are retrieved upon an external user's request by the web frontend (Step 6) and presented either as a raw CSV file or in form of graphical plots. The user may also view and download the configuration details of the employed WRF variants and benchmarks.

The key characteristics of an individual benchmark case that obviously affect WRF execution times are

- Horizontal grid sizes – derived from the extents of the geographical region over which the simulation is performed and the requested spatial resolution. For each of the grid points, the value of every WRF variable (such as temperature, wind speed) is simulated over time. The bigger the region and resolution, the more points are needed in the geographical grid, and the higher the computational requirements.

- Time step – the interval between the subsequent steps of the simulation; the smaller the time step, the higher the computational requirements.
- Number of (nested) domains – WRF supports nested runs in which a geographical region is subdivided into subregions with a higher spatial resolution and a smaller time step.

What was less evident was the influence of some additional parameters such as the different possible combinations of physics options, the choice of the parallelization mechanism (MPI, OpenMP or hybrid), the particular implementation of that mechanism, and hardware. Our benchmarks collected empirical data to improve the understanding of these relationships.

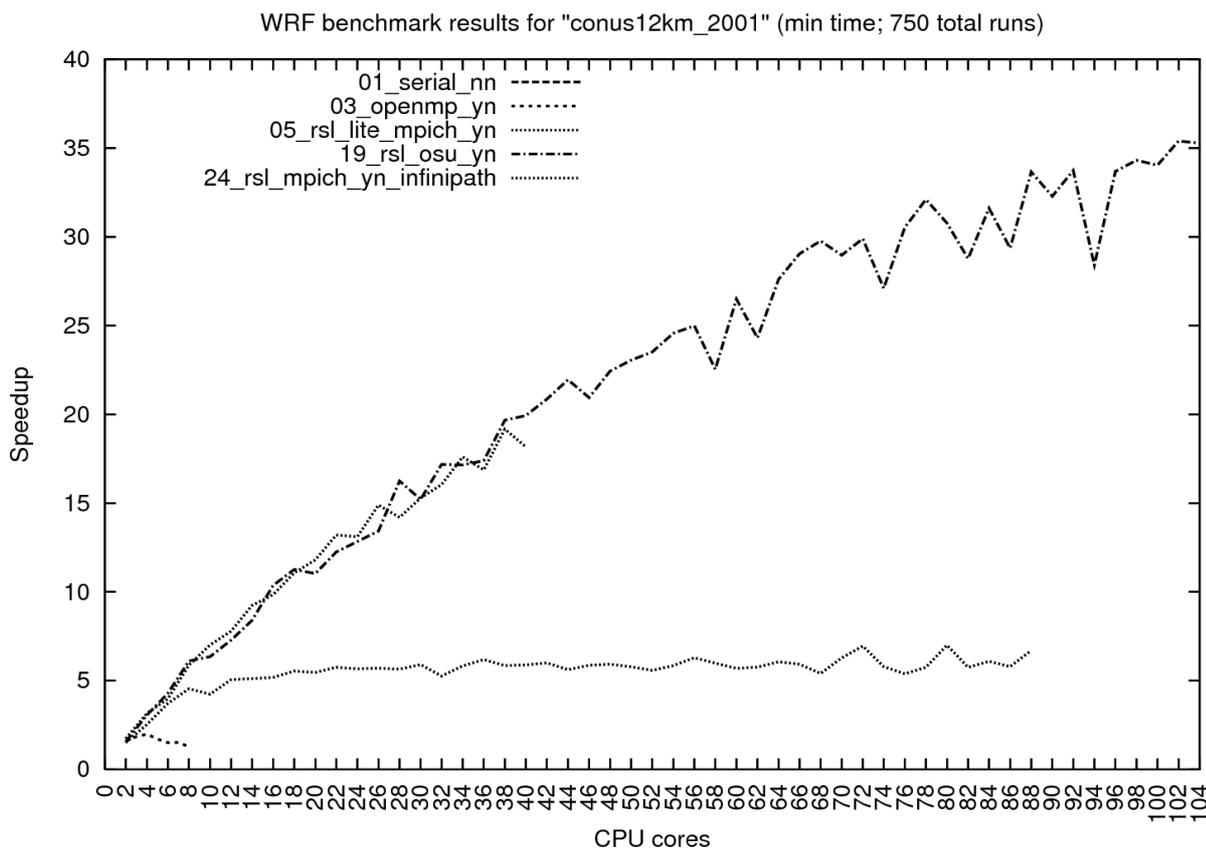


Figure 5.2.2: Benchmark results for conus12km_2001 case

As an example, the results for the conus12km 2001 case are shown in Figure 5.2.2. This benchmark case was originally constructed by the WRF developers at NCAR. However, since the introduction of WRFv2.2 no performance results for this case have been published. This single-domain case covers the continental USA area with 12 km resolution (distance between adjacent grid points), forecasting 3 hours from an original 48 hour model run. The geographical grid size is 425x300x35 points with a corresponding time step of 72s.

The conus12km 2001 case has been executed using Oldenburg and Jülich D-Grid clusters. WRF has been built with the same compiler, but with different parallelization mechanisms. The reported metric is the best speedup observed during a total of 750 runs. This metric, measured in multiple runs with the same number of processors performed at different times of day, provides a reliable impression of WRF's scalability, reducing the

possibility of obtaining results skewed by other temporary activity in the cluster (such as other users' applications executing on the same nodes or accessing the common NFS-mounted file system).

The results for conus12km 2001 reveal the following information, which we used to aid decisions about the number of processors required for more realistic cases in later phases of the work package:

- OpenMP variants of WRF scale very poorly. This confirms anecdotal evidence from the WRF user forum and was later confirmed by discussions through the WRF mailing list. This poor scalability is commonly attributed to limitations in bus latency and bandwidth of multi-core processors.
- WRF's scalability clearly benefits from the reduced latency and increased bandwidth of the InfiniBand interconnect, compared to Gigabit Ethernet, which scales very poorly beyond 8 processors. Therefore, WRF users should seek to deploy their models in InfiniBand equipped clusters. From a technical viewpoint, it places additional requirements on the D-Grid infrastructure, namely the ability to choose between InfiniBand-equipped and Ethernet-equipped worker nodes during job submission. (This requirement was satisfied by clusters configured for running WISENT WRF jobs.)
- There is no noticeable difference between the scalability of the InfiniBand and InfiniPath based solutions. InfiniPath's expected superior performance due to tighter hardware integration did not become apparent in the benchmark.
- The curves contain irregularities which cannot be explained by the change in the number of processors or by environmental factors (e.g., between 94 and 96 processors for model variant 19). Awareness of this characteristic is helpful to avoid a mistake of assuming that adding further processors would not improve performance upon observing a first "performance slip".

In addition to the conus2001_12km case described above, we also ran numerous benchmarks using two other cases, Kyrill and Katrina. The results of these benchmarks were processed automatically and published in an online database available at https://bi.offis.de/wisent/tiki-index.php?page_ref_id=94

In addition to performance benchmarks, as a side-effect of their preparation, technical tutorials were published online, concerning the WRF model and postprocessor compilation using the open-source GNU Fortran compiler (not among officially supported for WRFv2.2), compiling and running hybrid models.

5.3 WISENT WRF software tool-chain

In order to support local and distributed executions of WRF in accordance with the needs of our main application (wind speed prediction), we developed multiple software tools with command-line interfaces. Broadly speaking, our tools can be categorized as follows:

- tools that simplify setting up WRF variants and configurations;
- tools for postprocessing of WRF output;
- tools that help distribute WRF computations to multiple D-Grid sites and collect their results.

Our tools have been developed using a combination of Perl, NCL, and Java technologies, taking into account their individual strengths and weaknesses:

- Perl is universally available in all D-Grid sites (as it is in any UNIX system) and offers an extensive, mature repository of third-party software modules known as CPAN¹. The language is excellent for rapid prototyping and for integrating other software components, especially those accessible through UNIX-style command-line interfaces and producing text-oriented outputs. The main weakness of Perl is, apart from its intricate syntax and lack of static typing, a tendency to favor lightweight, yet unstable software designs over explicitly specified interfaces, which may raise risks of software defects in large, multi-developer projects (the development work in WISENT Work Package 5.2 clearly did not belong to this category).
- NCL² is a domain-specific language designed at National Center for Atmospheric Research (NCAR). It caters to meteorological and other scientific applications by providing built-in support for the ubiquitous NetCDF format used for representing multi-dimensional data sets annotated with metadata. Operations for reading and manipulating such data objects can be expressed very succinctly in NCL and are executed efficiently thanks to the underlying Fortran implementation. NCL also provides specialized functions for the visualization of meteorological data. This specialization turns out to be potentially problematic, as more general-purpose libraries, or even some universal data structures, are unavailable. Furthermore, we noticed that researchers may be more comfortable with imperative Fortran programming and confused by some of the more sophisticated language features of NCL.
- Java is a widely used, object-oriented programming language, which is commonly used in commercial applications and provides a solid base for many Grid and general middleware products. It has excellent development tools, multiple mature implementations, and a large developer community. However, it happens to be little known to scientific users, requires familiarity with object-oriented design principles, does not integrate quite as smoothly as Perl with native UNIX tools and concepts, and its performance (e.g. computation performance, memory footprint) are not optimal for high-end scientific applications.

Besides of the three programming languages, we relied on the following third-party software components:

- TORQUE/Maui as the local job scheduler, endorsed by D-Grid;
- Condor-G as the low-level Grid client and Condor DAGMan as the Grid scheduler;
- Globus Toolkit 4 providing uniform access to multiple D-Grid sites;
- JGlobus library for implementing custom GridFTP client functions;
- MVAPICH (+ OSU mpiexec) and OpenMPI as the MPI implementations;
- WRF Domain Wizard as the graphical user interface for setting up WRF configurations.

1 Comprehensive Perl Archive Network (CPAN), <http://www.cpan.org>

2 NCAR Command Language, <http://www.ncl.ucar.edu>

All of the technologies and software components utilized in the work package are governed by open source or (in case of Java) permissive commercial licenses, which removes concerns about their replicated deployment in D-Grid clusters.

The remaining sections describe our custom software tool-chain and its interplay with the above listed system software components in more detail. This tool-chain, unlike some other software products created in WISENT, was not published as open source projects because of its strong ties to the overall D-Grid system architecture, dependencies on components installed only in the Oldenburg “home” cluster, and relatively narrow thematic scope. Thus, the immediate beneficiaries of the tools are the energy meteorology community and its IT support unit (EnMetSE – Energiemetereologie-Support-Einheit³). Interested parties external to the project have also benefited from our experience through technical notes published on the WISENT web site.

5.3.1 Model run configuration: the `wrf-setup` and `multiday` tools

In a typical small-scale installation of WRF, the model runs might be performed within the directory tree of the WRF model, which is downloaded and compiled directly by the user. This shared-nothing configuration would become difficult to manage and would incur overhead costs when multiple variants of WRF should be accessed collaboratively by multiple users, as is the case when deploying WRF in a Grid environment. The `wrf-setup` tool, developed early in the work package, implements a basic concept of numbered WRF model variants being available in each supported D-Grid cluster. Users can create working directories for their individual model runs and then invoke `wrf-setup` to review a list of variants and to establish the necessary symbolic links for executing the selected variant. Job scripts suitable for local (non-Grid) execution are prepared by `wrf-setup` taking into account configuration details such as the required MPI version or scheduler queue names. These scripts can be customized by the user before submission, for example to specify the desired number of processors and to adjust walltime limits.

The `multiday` tools, which use `wrf-setup` internally, were developed in direct response to the requirements of our research studies performed during the work package. These tools prepare multiple working directories (e.g. 92 directories for the 92 days in the March-April-May period), based on user-provided "namelist" and job templates specifying model run configuration parameters and job scripts. Thus, the burden of replicating a configuration with a varying time parameter is removed from the user. In addition to the job scripts produced by `wrf-setup`, job scripts for the preprocessing (WPS) stage are also generated by the `multiday` tools. As with `wrf-setup`, the user has the flexibility to intervene and change the job scripts or namelists shortly before the model is run. However given the replicated character and consistency constraints on these scripts, it is recommended to perform most modifications on the templates.

5.3.2 Postprocessing for wind speed: the `windvelo` tool

The `windvelo` tool, which has evolved continuously during the work package, performs vertical interpolation from the model height levels output by WRF to the desired geopotential height of 80m relevant for wind energy applications. It also performs horizontal interpolation from the forecast grid to target geographical coordinates (e.g.

³ WISENT-Geschäftsmodell, https://bi.offis.de/wisent/tiki-download_file.php?fileId=818

locations of individual wind farms). The `windvelo` tool uses internally a combination of the WPP postprocessor (version 2 or 3) and NCL scripts. In particular, it converts the GRIB output of the postprocessor back to NetCDF, taking care to preserve or inject metadata attributes required for later data visualization (e.g. using the `flvplot` tool) and verification (e.g. using the MET package).

5.3.3 Verification and skill scores: the `rmse` tool and integration of MET

A WRF output file of a single model run is structured internally into time slots corresponding to snapshots of the simulation taken throughout the run (e. g. each 3 hours). In studies that execute WRF for multiple initial times (e.g. over 92 days), it is thus normal that multiple WRF output files contain the same time slot, forecast from a different initial time. This poses a practical problem for the user who wishes to compute a Root Mean Square Error (RMSE) skill score for the matching time slots from two series of runs. The `rmse` tool supports this task by scanning two directories with collected output files and performing the appropriate comparisons (implemented in NCL).

Additional programs were also implemented in Fortran to perform similar types of verification tasks on a case-by-case basis.

We also evaluated the Model Evaluation Tools (MET)⁴ package developed by Developmental Testbed Center at NCAR based on their proprietary verification framework as a potential replacement for our custom verification scripts. The main advantage of MET is that it is able to compute many skill scores (among them scores suited for meteorological applications beyond our own) and is maintained and tested centrally for the benefit of the whole meteorological modeling community. The disadvantage is its limited flexibility concerning the input data formats (in particular, spurious dependencies on the GRIB format) and the supported use cases. We took care in developing our postprocessing tools to maintain the compatibility with MET for future use, but have not implemented it operationally yet. Furthermore, we communicated our concerns about the data formats to MET developers, who have acknowledged their relevance.

5.3.4 Lag Averaged Forecasting: the `wrf-lag-avg` tool and NCL scripts

The lag averaging technique lies at heart of the research studies performed in the work package. The `wrf-lag-avg` tool, which was superseded by more specialized NCL scripts, serves the purpose of combining together outputs of multiple WRF model runs to compute the desired weighted averages. The technical requirements addressed by these scripts are very similar to those discussed for the `rmse` tool above; they involve access to semantically related data distributed across NetCDF file boundaries.

5.3.5 Model output visualization: the `contour` and `flvplot` tools

The `contour` tool was initially developed to simply create arbitrary contour plots in JPEG format based on NetCDF data (especially WRF outputs). It was later replaced by the `flvplot` tool, capable of producing a series of images (reflecting the time dimension of data) and combining them into a Flash Video (FLV) file, which can be presented online using the Weather Forecasting Portal (see Section 5.4). Development of `flvplot`

4 <http://www.dtcenter.org/met/users/>

benefited from experience with video codecs gathered in Work Package 4 (Efficient Interaction with Large Data Objects).

5.3.6 Grid scheduling with Condor-G: the `update-classads` daemon

The "gridification" of WRF was performed with help of the Condor software, which was first used for pooling desktop resources in the other work packages (notably, Work Package 5.1 at DLR IPA). By "gridification" we mean the ability to execute WRF model runs using multiple D-Grid sites at the granularity level of a single model run. The individual concurrent processes belonging to a single model run must be executed at the same site (e.g. no MPI-over-WAN or similar architectures were evaluated). This level of granularity provided adequate support for the research work performed in WISENT – distributed model runs using varying initialization times and model parametrizations.

The Condor-G extension was examined in detail and put into productive use, which led to several bug reports contributed to the Condor project and consequent fixes, as well as to additional documentation originated from the WISENT project and published online at <https://bi.offis.de/wisent/tiki-index.php?page=Condor-GT4>

The `update-classads` daemon runs continuously in the "home" cluster and maintains a set of advertisements for D-Grid sites that have been configured and tested for running WISENT VO software, in particular the WRF software. Essentially, these are the sites of earlier performance benchmarks, described in Section 5.2.

The advertisements (known as ClassAds in Condor terminology) are published to the Condor scheduler, running on the same machine as the daemon. They contain static and dynamic metadata specified by the WISENT VO, such as the name of the WS GRAM host accepting job submissions for the particular Grid site, the default WRF variant number, and the target queue and MPI parameters to be used at that site. The ClassAds published to the Condor scheduler can be referenced from submitted user jobs directed to Condor's "Grid" universe. The presence of these custom ClassAds enables scheduling of jobs to the Grid using sophisticated matching and ranking mechanisms supported by Condor. They also provide a soft-state view on the D-Grid resources available to WISENT Grid clients such as the `dagrun` tool described in Section 5.3.9.

This straightforward, VO-specific approach to maintaining Grid site metadata represents a viable alternative to centralized metadata management approaches that imply stronger coordination among potentially unrelated VOs. More general metadata, which is published according to standardized schemes by Grid components, can be imported into the local metadata on demand along with additional application-specific, or even user-specific metadata. A potential weakness lies in the fact that the `update-classads` daemon works closely with a single Condor scheduler. In an environment where multiple Condor pools exist, so do multiple schedulers, and the `update-classads` daemon would mostly likely have to be deployed separately in each pool. Even so, it would be difficult to share information across the schedulers to maintain a consistent view on the VO's resources.

5.3.7 Distribution of custom software and configuration data in D-Grid: the `svnsync` daemon

Deploying software to multiple Grid sites with slightly varying configurations (OS kernel version, library versions, installation paths) poses challenges for software configuration

management. Different concepts exist for addressing these challenges, from submitting all required software along with the computational jobs, through full machine virtualization approaches, to keeping configured replicas of VO-specific software at each potential execution site. The WRF software, like any other high-performance scientific application utilizing MPI libraries that depend strongly on the local hardware drivers, requires that the last approach be taken.

D-Grid supports the distributed software installation process at a very basic level by enforcing a convention that the environment variable `$DGRID_VO_DIRECTORY` points to a disk location in each cluster where the authorized VO members may permanently store their software and configuration data. In WISENT we relied on this convention, and extended it by several administrative guidelines regarding the structuring of the VO-specific software directory.

Manual, interactive software compilation and installation is acceptable (and oftentimes a sad necessity) for deploying third-party software packages such as WRF, which do not need to be updated very often after their successful deployment. However, VO-specific custom software such as the scripts described in earlier sections of this document, changes very often and its versions need to be kept synchronized across sites in an automated way.

As with many other system design matters in Work Package 5.2, we decided to follow the simplest approach that would produce the expected results, in order to avoid delays or even failing our project's goals. To this end, we introduced a central VO-specific Subversion repository to which write access is authorized for users authenticated by Grid certificates and read access is authorized for scripts by a password. Immediately after a VO member commits a change to this repository, a post-commit hook is invoked to relay a notification to a continuously running `svnsync` daemon, which in order contacts the Grid sites registered in the Condor pool by the `update-classads` daemon and triggers an update at each site (using the password-based access method). Because no (authorization) mechanism exists in D-Grid for deploying VO-specific services to Globus containers installed at remote Grid sites, the update is requested by logging in to the site using `gsissh` and executing the `svn` program there. In order to use `gsissh`, the `svnsync` daemon must temporarily assume the identity of a VO user by obtaining credentials that had been previously manually delegated to a MyProxy server.

The process described above was quite complicated to set up and remains suboptimal from a security standpoint. The dangers lie in storing a relatively long-lived proxy certificate in the MyProxy server and in the `svnsync` software daemon executing with unrestricted (if only temporary) privileges of the delegating user. On the other hand, it saves significant amounts of developers time which would otherwise be consumed by keeping the contents of different D-Grid sites synchronized. The automated checkouts have proved invaluable for rapid prototyping and testing. Doubtlessly, better alternatives will be proposed and implemented in the future as the D-Grid infrastructure evolves beyond research applications and more fine-grained role-based access control concepts that permit software agents are effectively introduced.

5.3.8 Executing MPI jobs in D-Grid: the `MultiJob` module

WISENT was one of the first projects to use the D-Grid infrastructure for executing MPI applications. In this “pioneer” role, we encountered some technical problems with job scheduling, which we subsequently described together with our solution in a paper published in the FGCS journal⁵, which was accompanied by a feature enhancement request⁶ for the Globus middleware (seconded by our colleagues from the AstroGrid project).

MPI jobs, from a user's point of view, typically consist of an single-processor initialization phase, followed by the actual parallel execution of an MPI program, and ending with another single-processor phase to retrieve the results and clean up the working directory. Unfortunately, Globus Toolkit 4 does not natively support this kind of partitioning and would instead require submitting three distinct jobs, of three different types, to cover the individual phases. This approach is not very efficient and also complicates job handling for users, who can no longer track a single “logical” job in their workflow. As an alternative, we developed a Perl module called `MultiJob` which uses NFS locking mechanisms to perform synchronization between the three stages of an MPI job. The user implements three simple Perl subroutines, called `pre`, `main`, and `post`, which are executed in order at the execution site. `pre` and `post` are executed on a single node only, whereas `main` is executed on every node allocated for the job. These subroutines correspond to the phases of MPI job execution mentioned previously.

The `MultiJob` module was superseded by a more efficient implementation, `UDPMultiJob`, which offers the same user interface, but uses a simple voting protocol utilizing UDP broadcast instead of file locks to achieve the needed synchronization.

These modules turned out to be a practical workaround for Globus Toolkit 4 deficiencies in handling MPI jobs, with one notable technical quirk. The implementation of MPI provided by ParaStation (installed in Jülich) kills the job immediately after the final MPI protocol exchange, making it impossible to execute the `post` subroutine. We contacted the developers, but have not received an agreeable solution, which left us with having to implement application-specific workarounds (run the `post` code from within the MPI program) or to separate the `post` part into another job. It remains to see whether future improvements in Globus Toolkit will make introducing modules such as `MultiJob` unnecessary.

5.3.9 Executing Grid workflows: the `dagrun` tool

While a single WRF model run can be packaged into a `MultiJob` as explained in the previous section, it would be inefficient to rely solely on `MultiJob` module's functionality to attach the preprocessing (WPS) and postprocessing stages (e. g. `windvelo`). The reason is that a `MultiJob` keeps all worker nodes committed during its entire execution, including the single-processor `pre` and `post` stages. If these stages perform time-consuming operations, as may be the case for WPS and certain postprocessing scenarios, the reserved remaining processors would be wasted during `pre` and `post` execution. Furthermore, the mapping between postprocessing operations and model runs is not

5 <http://dx.doi.org/10.1016/j.future.2008.05.003>

6 http://bugzilla.globus.org/globus/show_bug.cgi?id=5698

always one-to-one. For example, in lag averaged forecasting, the outputs from multiple model runs are combined together to form the postprocessed output. These features call for separating the pre- and postprocessing operations into individual jobs, which are nonetheless logically related to other jobs.

A set of such interdependent jobs, which may execute at a single Grid site or be spread across multiple Grid sites, is commonly known as a “workflow”. Many approaches have been devised by various projects for describing and executing Grid workflows. Because of our commitment to Condor as the Grid client, we decided to first examine its built-in solution, known as Condor DAGMan⁷, and found it to be satisfactory for our purposes. DAGMan allows the user to specify dependencies between a set of Condor job using a directed acyclic graph (DAG) structure encoded in a simple text file. The DAG is then submitted as a Condor job to the “scheduler” universe. It starts execution immediately on the submission host and from there it coordinates the ordered submission (and, in case of errors, resubmission) of the contained subjobs. In between of subjob executions, user-defined scripts are invoked on the submission host to evaluate intermediate results and influence the remaining execution of the DAG.

While the interaction between Condor and DAGMan is not especially difficult to understand, using these programs directly requires the user to prepare job command files and the DAG files manually. Given that the job command files for the “grid” universe may contain many technical details quite independent from the user's application (e.g. regarding proxy certificate renewal or retry policy), this process may become tedious and simple mistakes can be made that do not become apparent until runtime, slowing down the development and testing process. To support quicker and safer authoring of DAGs, we developed a tool called `dagrun`, which is similar in spirit to the `MultiJob` module, in that it provides a simple application programming interface (currently implemented in Perl) by which the user can provide application-specific functionality to be executed as a DAG. Essentially, the user should implement:

- a set of subroutines, each of which corresponds to a subjob in the DAG;
- subroutines which contain processing of the intermediate results on the submission host;
- a pair of functions that return metadata about the individual subjobs (e.g. files and directories to be staged from/to submission host) and subjob dependencies (ordering and optional constraints binding a subjob to the same Grid site as its predecessor).

The `dagrun` API permits the user to declare a subjob to be of type “multiple” or “mpi”, to obtain execution semantics supported by the `MultiJob` module. Moreover, it supports parameter sweep jobs by allowing the user to declare that a subjob should be executed n times (in parallel). A subroutine corresponding to such a parameter sweep subjob is supplied an integer representing its ordinal number, from which the variable parameter (e.g. a subset of data to be processed) may be derived. It is noteworthy that the user-provided source code is executed both on the submission host and remote worker node environment, and that any user-defined static metadata are available in both environments. The code may be interpreted as a set of callbacks, which are invoked at different points in the DAG's lifecycle, on different machines.

⁷ http://www.cs.wisc.edu/condor/manual/v7.0/2_10DAGMan_Applications.html

In summary, from an application programmer's standpoint, the `dagrun` tool reduces the task of Grid computing to writing straightforward code in Perl. Most Grid-related technical details are abstracted away by providing sensible defaults reflecting the WISENT Grid environment (established by the `update-classads` daemon and other conventions). On the other hand, all features of a general-purpose programming language, including the powerful standard libraries, remain available. Instead of adjusting the programmer's habits to match the Grid environment, we adjust the Grid environment to match the programmer's habits. An important benefit of this approach is that it allows the implementation of the Grid environment to be replaced without affecting applications, as long as the new implementation can fulfill the API's contract. The apparent dependence on the programming language (Perl) is of less concern, because the same API could be just as well realized in different programming languages.

5.3.10 The `dcache-sync` tool

File staging is a conceptually simple task of transferring the required input files over network from storage to the execution Grid site before a job's execution, and transferring the output files back to storage thereafter. This task becomes less trivial when large amounts of data processed by Grid applications must be taken into account. For example, one of our case studies consumed 300 GB of global input data and produced over 800 GB of output data to be postprocessed using lag averaging. Dealing with such amounts of data naively would lead in the best case to wasted bandwidth, and in the worst case to a failure to meet execution deadlines in operational context.

In practice, we found the file staging support offered by Globus Toolkit and Condor to be insufficient for our application, for the following reasons:

- the WS GRAM protocol of Globus Toolkit places file staging operations in front and after individual jobs; in practice, a file staging operation has to precede and follow a *set* of such jobs, and staged in data must be retained at the execution site across job boundaries;
- the Condor-G component only supports staging from/to the submission host; the Condor Stork component, which supposedly supports third-party file transfers, did not work as expected and appears to be no longer actively maintained;
- the RFT component of Globus Toolkit was not interoperable with the version of dCache installed in D-Grid;
- neither Globus Toolkit nor Condor provide support for managing the lifetime of the staged-in files and disk space management at execution sites; for example, if many jobs required the same large file, it would be transferred repeatedly with each job, wasting bandwidth.

The `dcache-sync` tool, implemented as part of Work Package 3.2 (Standardized Data Storage and Exchange) is our solution which addresses the above shortcomings, discovered in Work Package 5.2. It is functionally similar to the Replica Location Service⁸ and Data Replication Service⁹ of Globus Toolkit 4, in that it supports data provisioning on behalf user jobs from a distributed data archive. Although it is less sophisticated and

8 <http://www.globus.org/toolkit/data/rls/>

9 <http://www.globus.org/toolkit/docs/4.0/techpreview/datarep/>

scalable than these services, it provides a form of WISENT VO-specific disk space management for Grid sites (a least-recently used cache combined with soft disk quotas) and is also compatible with the dCache version deployed in D-Grid (currently, Jülich and Hannover are serving as storage sites for WISENT).

The overall file staging concept is to have the initial subjobs of a DAG invoke `dcache-sync` to download input data from dCache – but only if it is not available locally yet – and to have the final subjobs invoke `dcache-sync` to upload output data to dCache. This serves as an alternative to manual (bulk) file staging using the interactive GridFTP Client. `dcache-sync` refers to data files using a global namespace, which mirrors the directory structure in one or more dCache archives. Each execution site maintains a partial replica of the files from this namespace (at the granularity level of individual files). A locking mechanism supports safe retrieval of remote files by multiple jobs executing concurrently in the same cluster. The full technical specification of the `dcache-sync` tool is available at <https://bi.offis.de/wisent/tiki-index.php?page=dcache-sync>

5.4 WISENT Weather Forecasting Portal (under construction)

The software tool-chain presented in Section 5.3 satisfied the main project goals for Work Package 5.2 by making it possible to execute custom WRF model configurations using the technical infrastructure provided by D-Grid, according to the workflow described in Section 5.1. Following that, in the final phase of the work package, we established an additional, future-oriented goal of developing a weather forecasting web portal that would accelerate the dissemination of data products researched and designed during the work package to a wider potential audience.

The WISENT Weather Forecasting Portal (WFP) will utilize cutting-edge software technology to enhance the delivery of data products and to provide users with new levels of data and model interaction beyond the monolithic portals run by large organizations.

The idea of the portal follows from our observation that the market for energy meteorology applications, despite having grown considerably in the past years, is not yet ripe in terms of product transparency and information dissemination to potential customers:

- (a) potential customers cannot easily understand the different available products – especially their quality – before obtaining commercial assistance from their respective vendors;
- (b) researchers and vendors involved in the development of new innovative data products cannot easily market them to customers by presenting favorable quantitative inter-comparisons using an open, commercial-quality delivery medium; instead proprietary solutions must be “reinvented” by each research group or vendor.

A prototype of WFP has been implemented as a lightweight Rich Internet Application, that is, by emulating a sophisticated desktop application in the runtime environment of a web browser. The front-end uses Google Web Toolkit combined with Adobe Flash technology. The back-end relies on the computing power provided by the German Grid infrastructure for executing the configured WRF models and postprocessing their outputs, and is realized using the WISENT WRF software tool-chain developed within the project.

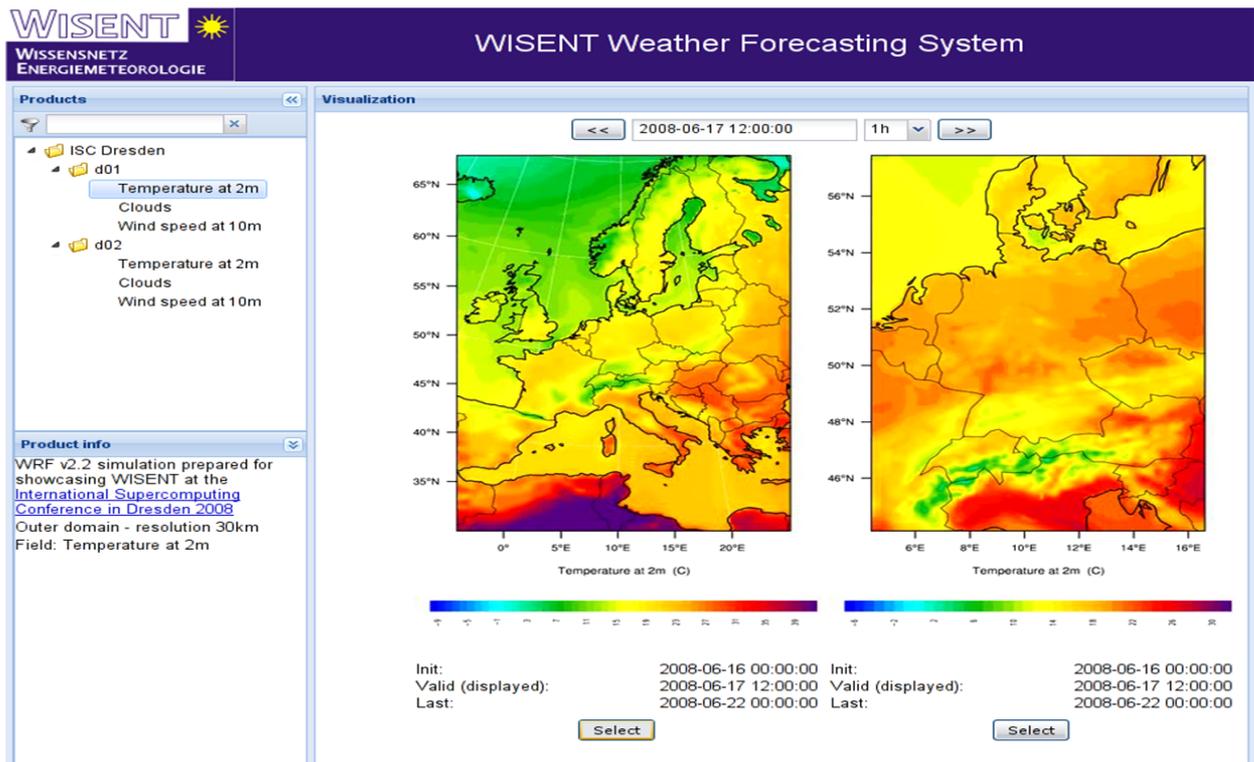


Figure 5.4.1: Screenshot of WISENT's WFP (under construction)

At present, the prototype version of the portal can be used to provide access to either historical or regularly updated time-dependent data products, including preconfigured wind speed forecasts constructed using the Lagged Average Forecasting methodology described in Sections 4.3 and 4.4.

Future work includes extending the portal by incorporating standard verification scores (based on the MET package mentioned in Section 5.3.3). Furthermore, we plan to enable authorized users to perform interactive data queries and to experiment with selected model parameters on a small scale (constrained by D-Grid and WISENT VO Acceptable Use Policies). The portal should provide a starting point where potential users of energy meteorology data may preview the data product portfolio developed by researchers and learn about the product origins and specific characteristics. We envision that from this starting point new collaborations between research and industry may become established for mutual benefits.

6 Summary

The Work Package 5.2 of WISENT focused mainly on the implementation and evaluation of the Weather Research and Forecasting (WRF) Model, Version 2.2 based on the experiences and within the technical infrastructure established in the preceding work packages of the project.

The research goals consisted in improving the wind speed forecasting skill as compared to the previous semi-operational forecasting platform established in ForWind. The technical goals consisted in successfully and efficiently utilizing the heterogeneous computing resources available within D-Grid and in implementing a generally usable, sustainable framework for running the WRF model and postprocessing its raw output for the purposes of energy meteorology community (e.g. vertical interpolation to desired height levels, horizontal interpolation to desired coordinates, lag averaging, computing verification scores and so on). Finally, the organizational goals consisted in establishing a set of standardized procedures, tools, and domain knowledge necessary to perform research and to provide semi-operational wind forecasting services, as well as identifying further opportunities for transferring this knowledge into generalized energy meteorology applications (e.g. solar energy).

The research work resulted in the initial implementation and validation of WRF as the mesoscale platform in WISENT, followed by the examination of the Lagged Average Forecasting methodology for wind speed prediction, which was later refined by combining LAFs into ensemble schemes using optimal weights.

Based on our results, the construction of a Combined Prediction System (CPS) comprising deterministic and ensemble members was possible using an optimal combination of different forecast components such as NCEP GFS deterministic forecasts, NCEP GFS LAFs, WRF deterministic high-resolution (mesoscale) forecasts, and WRF LAFs. This CPS can be extended to include additional components, for example ECMWF IFS deterministic forecasts and ECMWF Ensemble Prediction System (EPS). Such an optimally combined meteorological forecast system (CPS) linked to wind power forecasting tools should be evaluated on all relevant time scales that are interesting to end-users, such as wind farm operators, transmission system operators and energy traders.

The technical work resulted in the "gridification" of WRF – establishing the procedures and tools for maintaining model installations and executing instances of the model at multiple D-Grid sites and for data postprocessing as required by the Lagged Average Forecasting methodology.

7 Papers & Posters

- Thomas I. Petroligis, Jan Ploski, Jens Tambke, Thomas Scheidsteger & Detlev Heinemann, 2009: *Inter-comparison & Optimal Combination of Multi-model Lagged Average Forecasting Ensemble Schemes for Wind Prediction*. European Wind Energy Conference (EWEC), Parc Chanot, 16-19 March 2009, Marseille, France (submitted).
- Jan Ploski, Thomas I. Petroligis, Jens Tambke, Detlev Heinemann & Wilhelm Hasselbring, 2009: *Design & Implementation of the WISENT WFP (Weather Forecasting Portal)*. European Wind Energy Conference (EWEC), Parc Chanot, 16-19 March 2009, Marseille, France (submitted).
- Thomas I. Petroligis, Jens Tambke & Renate Hagedorn, 2008: *Optimized Ensemble Forecast Systems applied to Wind Power Prediction*. Biennial German Wind Energy Conference (DEWEC), 26-27 November 2008, Bremen, Germany (submitted).
- Thomas I. Petroligis, Jan Ploski, Lüder von Bremen, Ulf Gräwe, Sönke Brummerloh & Detlev Heinemann, 2008: *Wind Power Prediction utilizing Weather Research & Forecasting Model with Lagged Average Forecasts*. International Supercomputer Conference, 17-20 June 2008, Dresden, Germany.
- Thomas I. Petroligis, Jan Ploski, Lüder von Bremen, Ulf Gräwe, Sonke Brummerloh, & Detlev Heinemann, 2008: *Wind Power Prediction utilizing LAFs (Lagged Average Forecasts) from the German D-Grid WRF (Weather Research & Forecasting) Model*. European Wind Energy Conference (EWEC), 31 March - 3 April 2008, Brussels, Belgium.
- Jan Ploski, Guido Scherp, Thomas I. Petroligis, Otto Büchner & Wilhelm Hasselbring, 2008: *Grid-based Deployment and Performance Measurement of the Weather Research & Forecasting (WRF) Model*. Future Generation Computer Systems (FGCS). The International Journal of Grid Computing: Theory, Methods and Applications.
- Thomas I. Petroligis, Lüder von Bremen & Detlev Heinemann, 2007: *Deployment and Validation of the Weather Research & Forecasting (WRF) Model on the German Grid for wind power prediction applications*. 8th ECAM (European Conference on Applications of Meteorology), 1-5 October 2007, El Escorial, Madrid, Spain.
- Jan Ploski, Thomas I. Petroligis & Wilhelm Hasselbring, 2007: *Implementing the Weather Research & Forecasting (WRF) model on the German Grid*. 8th ECAM (European Conference on Applications of Meteorology), 1-5 October 2007, El Escorial, Madrid, Spain.
- Jan Ploski, Thomas I. Petroligis, Detlev Heinemann, Thomas Scheidsteger & Wilhelm Hasselbring, 2007: *Grid-basierte Modellierung im Wissensnetz Energiemeteorologie (Grid-based Modelling in the Knowledge Net Energy Meteorology)*. DACH-Meteorologentagung - Conference of Meteorologists, 10-14 September 2007, Hamburg, Germany.

Parallele Simulation von Solarstromanlagen

Arbeitspaket 5.3

Jochen Rehwinkel
meteocontrol GmbH
Spicherer Straße 48
86157 Augsburg

Internet: <http://www.meteocontrol.de>

1 Einführung

1.1 Ausgangssituation

Die Simulation von Solarstromanlagen ist ein zentraler Bestandteil des Systems der meteocontrol GmbH. Mit Hilfe des Internetportals Safer'Sun werden die Solaranlagen der Kunden visualisiert und überwacht. Um mögliche Fehlkonfigurationen und Ausfälle bzw. Teilausfälle zu erkennen, wird jede Anlage täglich simuliert. Liegt der simulierte Ertrag über dem tatsächlich gemessenen, ist dies ein Hinweis auf einen möglichen Defekt. Der Kunde wird automatisch benachrichtigt (E-Mail, SMS) wenn der Verdacht auf Fehlfunktion vorliegt. Zusätzlich gibt es die Möglichkeit, das Simulationsergebnis mit den realen Daten im Safer'Sun graphisch zu vergleichen (siehe Abbildung 1). Eine möglichst genaue und zeitnahe Simulation von Solarerträgen ist also zentraler Bestandteil der Dienstleistungen der meteocontrol GmbH. Mittlerweile werden 7000 Solaranlagen mit Safer'Sun überwacht. Jede dieser Anlagen kann aus einer oder mehreren Teilanlagen bestehen. Jede dieser Teilanlagen, die sich in der Regel durch unterschiedliche Ausrichtung und Neigung auszeichnen, muss einzeln simuliert werden. Dies bedeutet eine erhebliche Rechenlast des Serverparks von meteocontrol durch die Programmprozesse der Simulation.

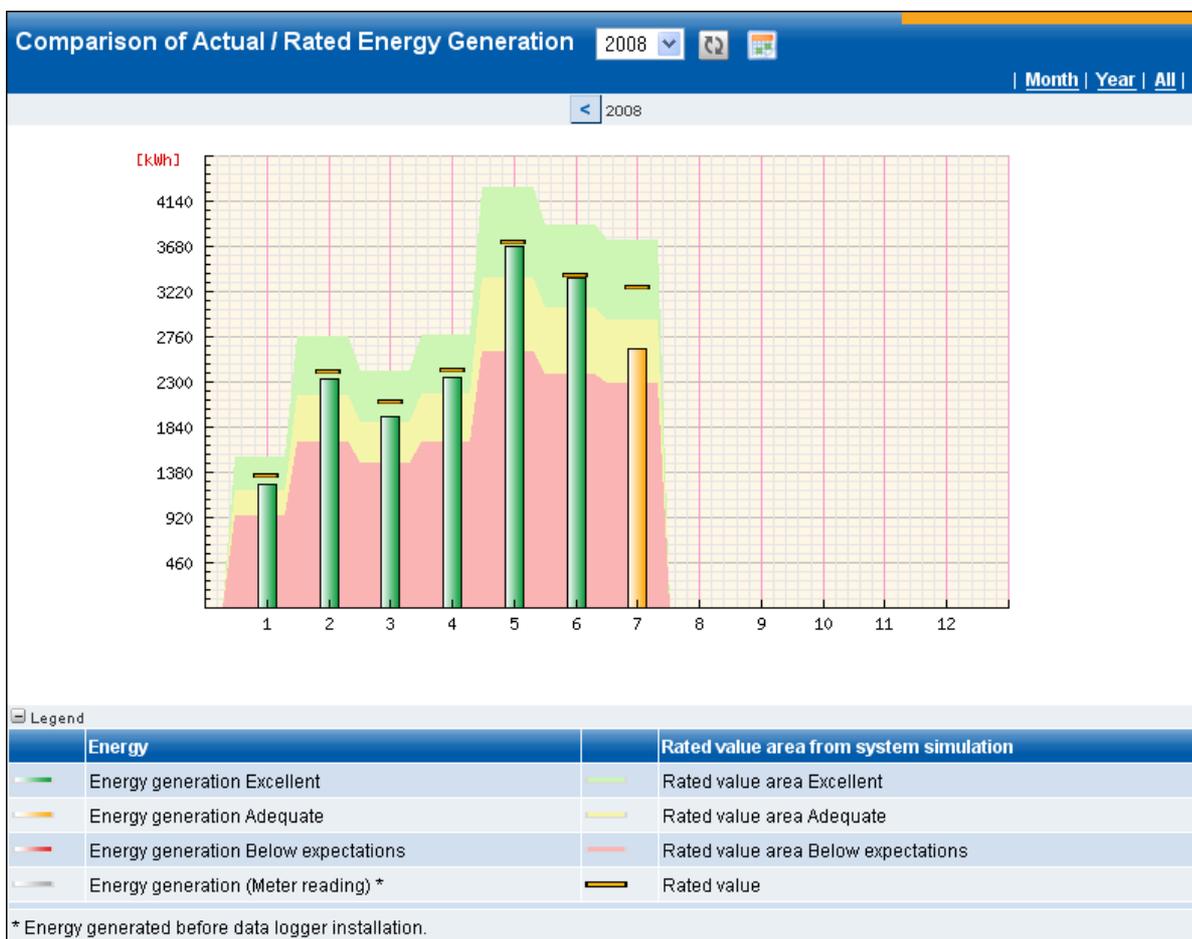


Abbildung 1 Safer'Sun: SOLL/IST Vergleich

Da die Anzahl der zu simulierenden Solaranlagen immer schneller wächst und zusätzlich immer komplexere Anlagenkonfigurationen berücksichtigt werden müssen, die immer längere Simulationszeiten nach sich ziehen, ist die Zielsetzung eine verteilte gridbasierte Simulation der Erträge um Kunden wieder zeitnah mit der notwendigen Information versorgen zu können.

1.1.1 SolEm

Zu Projektbeginn wurde bei meteocontrol die Simulationssoftware SolEm eingesetzt. SolEm wird an der FH-München entwickelt. SolEm ist ein Visual-Basic-Macro, das als Eingabe- und Ausgabeschicht Excel verwendet. Der Vorteil an SolEm ist somit die einfache Bedienung, da diese in der normalen Arbeitsumgebung des Users erfolgt. Daneben besitzt SolEm im Arbeitskontext von meteocontrol jedoch enorme Defizite:

- Die Automatisierung ist nur durch VBA-Macros möglich, die eine sehr aufwendige und komplizierte Methode darstellen, Prozesse zu automatisieren.
- Es ist sehr ineffizient bei der Ressourcennutzung, dadurch ergibt sich eine lange Laufzeit pro Simulation.
- Erweiterungen lassen sich nur sehr schwer in den unzureichend strukturierten Code einbinden.

Besonders gravierende Mängel sind Ein- und Ausgabe über Excel, die durch Verwendung von VBA unnötig viel Speicher- und Prozessorleistung beansprucht sowie die ineffizienten Berechnungsroutinen. Ferner ist die Software plattformgebunden (WINDOWS), wodurch die Möglichkeit ihrer Verteilung auf verfügbare Rechner nur schwer möglich ist. Der Endzeitpunkt der Simulation, bis alle Anlagen für den gestrigen Tag prozessiert sind, beträgt zu Projektbeginn 16:00h. Dies ist viel zu spät, da für die letzten simulierten Anlagen erst zu diesem Zeitpunkt ein eventueller Alarm ausgelöst werden kann.

1.1.2 Vorbereitende Tätigkeiten zu Projektbeginn

Zu Beginn des A.P 5.3 wurde es notwendig, die Prozesskette der Solarsimulation grundsätzlich neu zu planen und zu gestalten. Das alte System hatte das Defizit, dass die Anlagen einfach der Reihe nach simuliert wurden. Es wurde dabei aber keine Rücksicht darauf genommen ob alle zur Simulation benötigten Daten bereits im System vorhanden waren. Folgende Daten sind für eine erfolgreiche Simulation notwendig:

- Solaranlagenkonfiguration
 - Anzahl der Teilanlagen
 - Ausrichtung und Neigung der Teilanlagen
 - Standort
 - Solarmodultyp und Anzahl der Module
 - Wechselrichtertyp und Anzahl der Module
 - Verschaltung der Solarmodule und der Wechselrichter
- Messparameter
 - Globalstrahlung am Standort der Anlage (stündliche Auflösung)
 - Temperatur am Standort der Anlage (stündliche Auflösung)

Da die benötigten Daten für manche Solaranlagen zum Simulationszeitpunkt nicht vorhanden waren, kam es zu einer fehlerhaften Simulation. Dies konnte im schlimmsten Falle einen Fehlalarm oder einen nicht erzeugten Alarm trotz Fehlfunktion nach sich ziehen. Zusätzlich mussten fehlerhafte Simulationen nachträglich manuell erneut simuliert werden. Da bei einer gridbasierten Simulation von einem viel früheren Simulationszeitpunkt für alle Anlagen auszugehen ist und somit die Wahrscheinlichkeit für eine unkomplette Simulation bei vereinzelt Anlagen aufgrund von noch nicht gelieferten Wetterdaten steigt, war es notwendig eine Plausibilitätsprüfung in den Simulationsprozess

einzubauen. Da von Beginn an klar war, das SolEm im Laufe des Projektes ersetzt werden würde und noch keine neue Simulationssoftware zur Verfügung stand, in die die Plausibilitätsprüfung eingebaut werden konnte, wurde entschieden, die Plausibilitätsprüfung als eigenes Softwaremodul innerhalb der Simulationsprozessierung einzubauen. Hierzu wurde ein automatisierter Cronjob in PHP eingerichtet, der zu Beginn des Simulationsprozesses alle zu simulierenden Anlagen auf notwendige Simulationsparameter hin überprüft. Alle simulationsbereiten Anlagen werden dann in eine Datenbanktabelle eingetragen, die als Queue für den Simulationsprozess dient. Hier wurde im ersten Schritt ein serieller Simulationsverlauf implementiert.

←T→	ID_SAF_ANLAGE	STATUS	DONE	ERZEUGT	GESTARTET	BEENDET
<input type="checkbox"/>	8571	abgeschlossen	TFADCSM	2008-05-05 16:01:20	2008-05-05 16:01:55	2008-05-05 16:01:56
<input type="checkbox"/>	8493	abgeschlossen	TFADCSM	2008-05-05 16:01:12	2008-05-05 16:01:37	2008-05-05 16:01:52
<input type="checkbox"/>	8298	abgeschlossen	TFADCSM	2008-05-05 16:00:53	2008-05-05 16:01:18	2008-05-05 16:01:22
<input type="checkbox"/>	8148	abgeschlossen	TFADCSM	2008-05-05 16:00:48	2008-05-05 16:01:15	2008-05-05 16:01:16
<input type="checkbox"/>	8077	abgeschlossen	TFADCSM	2008-05-05 16:00:42	2008-05-05 16:00:59	2008-05-05 16:01:03
<input type="checkbox"/>	6678	abgeschlossen	TFADCSM	2008-05-05 16:00:33	2008-05-05 16:00:44	2008-05-05 16:00:47
<input type="checkbox"/>	8635	abgeschlossen	TFADCSM	2008-05-05 16:00:31	2008-05-05 16:02:37	2008-05-05 16:02:38
<input type="checkbox"/>	9030	abgeschlossen	TFADCSM	2008-05-05 16:00:29	2008-05-05 16:02:35	2008-05-05 16:02:36
<input type="checkbox"/>	6242	abgeschlossen	TFADCSM	2008-05-05 16:00:23	2008-05-05 16:00:29	2008-05-05 16:00:30
<input type="checkbox"/>	5056	abgeschlossen	TFADCSM	2008-05-05 16:00:12	2008-05-05 16:00:18	2008-05-05 16:00:21
<input type="checkbox"/>	2502	abgeschlossen	TFADCSM	2008-05-05 16:00:06	2008-05-05 16:00:09	2008-05-05 16:00:11
<input type="checkbox"/>	8827	abgeschlossen	TFADCSM	2008-04-01 00:00:02	2008-04-01 04:01:31	2008-04-01 04:01:33
<input type="checkbox"/>	1348	abgeschlossen	TFADCSM	2008-04-01 00:00:05	2008-04-01 00:02:18	2008-04-01 00:02:21
<input type="checkbox"/>	8706	abgeschlossen	TFADCSM	2008-04-01 00:00:14	2008-04-01 04:01:36	2008-04-01 04:01:37
<input type="checkbox"/>	8529	abgeschlossen	TFADCSM	2008-04-01 00:00:12	2008-04-01 00:02:22	2008-04-01 00:02:29

Abbildung 1 DB-Tabelle des Queueing-Mechanismus der Simulation (Auszug)

2 Die Simulationssoftware INSEL

Aufgrund der beschriebenen Nachteile von SolEm wurde es im Laufe des A.P 5.3 für meteocontrol notwendig, auf die auch vom Projektpartner Universität Oldenburg entwickelte Simulationssoftware INSEL umzurüsten. INSEL ist ein blockorientiertes Simulationsprogramm, d. h. verschiedene Berechnungsbausteine sind in Blöcken gekapselt. INSEL ist in Fortran geschrieben und arbeitet mit effizienten Routinen. Die Automatisierung erfolgt über Textdokumente. Wenn diese in ein Templatemodell eingebunden werden, lassen sich eigene Entwicklungen projektspezifisch einbinden. Die Berechnungszeit beträgt nur einen Bruchteil der Berechnungszeit mit SolEm. Zusätzlich lassen sich mit der grafischen Oberfläche HP VEE verschiedene Programm-Blöcke mit Modellen für Solarstrahlungsberechnungen, PV-Module, Wechselrichter, Batterien, Windkraftgeneratoren, Pumpsysteme und solarthermische Kraftwerke kombinieren.

Da die Anzahl der zu simulierenden Solaranlagen immer schneller wächst und zusätzlich immer komplexere Anlagenkonfigurationen berücksichtigt werden müssen, die immer längere Simulationszeiten nach sich ziehen, wurde die Zielsetzung eine verteilte gridbasierte Simulation der Erträge mit INSEL auf mehreren Systemen.

2.1 Evaluation von INSEL

Nach genauer Analyse und Erstellung von Simulationsmodulen mit INSEL wurden die Schwächen der Software deutlich. Es ist nicht möglich, neueste Anlagen mit Multistringtechnologie zu simulieren. Hinzu kam ein schlechter Linux-Support von Seiten des Herstellers. Ein Update der auf eine neue Version von INSEL war nicht möglich, da es auf absehbare Zeit keinen Linux-Support gegeben hätte. Aus diesen Gründen wurde beschlossen, eine eigene Simulationssoftware zu implementieren, die von Beginn an auf die Anforderungen einer späteren Parallelisierung angepasst werden kann. Diese Entscheidung bedeutete einen großen Schritt in der Produktentwicklung von meteocontrol, da nun keine Abhängigkeit mehr zu Anbietern von Simulationssoftware besteht. Die Simulation kann an die Bedürfnisse von meteocontrol angepasst werden und auf neue technische Entwicklungen (z.B. Multistringwechselrichter) kann zeitnah eingegangen werden. Die Realisierung einer gridbasierten Simulation wurde durch eine Eigenentwicklung stark vereinfacht.

3 MC-Simulation

3.1 Releaseplanung

Um das Ziel einer parallelen Simulation zu erreichen und bereits vorhandene Anforderungen an das System in Einklang zu bringen, wurde eine Releaseplanung in fünf Schritten gemacht. Die Software wurde so entworfen, dass der Rechenkern der Simulation für alle weiteren Versionen verwendet werden kann.

- Basisversion: Rechenkern, der als Basis für alle weiteren Produkte dient
- Version 1: Prognose des Jahresertrages einer geplanten Solaranlage mit zusätzlicher Web-Service Schnittstelle
- Version 2: Simulation aller mit Safer'Sun überwachten Solaranlagen und Ersetzung von SolEm
- Version 3: Gridbasierte parallele Simulation auf dem OFFIS-Cluster basierend auf Version 2

3.2 Basisversion Rechenkern

Der erste Schritt für die Entwicklung einer eigenen Simulation, im Folgenden MC-Simulation genannt, war die Evaluierung und Festlegung des zu verwendeten Simulationsmodells. Es wurde entschieden das Hans-Georg Beyer Modell zu verwenden.

Der Rechenkern für die Simulation stellt das Kernstück der Entwicklung dar. Zur Umsetzung wurde die Programmiersprache PHP verwendet. Der Rechenkern ist modular aufgebaut. Sämtliche Anwendungen, die Simulationsergebnisse benötigen, greifen in Zukunft auf dieses System zu. Der Rechenkern wurde im Rahmen des Projekts vollständig umgesetzt. Der Kern wurde grob in drei Kernmodule aufgeteilt:

- **Modul 1:** Container für den Input von Wetterdaten. Für die Simulation sind Temperaturen und Globalstrahlung nötig.
- **Modul 2:** Solarmeteorologie: Berechnung der tatsächlichen Einstrahlung auf den Modulen. Die gemessene Globalstrahlung wird umgerechnet auf die tatsächliche Einstrahlung auf den Modulen.
- **Modul 3:** Technischer Teil: Eigentliche Simulation des Ertrages, abhängig von konfigurierten Solarmodulen und Wechselrichtern.

```

84 // Fläche nicht 0, da Divisor
85 if (!is_numeric($iFlaeche) || $iFlaeche <= 0.000000001) {
86     trigger_error("Parameter ungültig (iFlaeche: $iFlaeche)");
87     return 0;
88 }
89
90 if (!is_numeric($fTemperatur) || !is_numeric($iHgb1) || !is_numeric($iHgb2) ||
91     trigger_error("Parameter ungültig (fTemperatur: $fTemperatur, iHgb1: $iHgb1,
92     return 0;
93 }
94
95 // Temperaturabhängigkeit iAlpha
96 if (!is_numeric($iAlpha) || ($iAlpha > -0.003) || ($iAlpha < -0.01)) {
97     trigger_error("Parameter (iAlpha: $iAlpha) auf Default gesetzt");
98     $iAlpha = -0.004;
99 }
100
101
102 //
103 // Berechnung der DC - Energie
104 //
105
106 // Verlustkorrektur der Einstrahlung
107 $iEinstrahlungKor = $fGesamtRadGen * (100 - $iVerschmutzung) / 100;
108
109 // Wirkungsgrad bei aktueller Leistung
110 $iEtaG = $iHgb1 + $iHgb2 * $iEinstrahlungKor + $iHgb3 * log($iEinstrahlungKor);
111
112 // Wirkungsgrad bei Nennleistung
113 $iEtaNenn = $iLeistung / $iFlaeche;
114
115 // Begrenzung Nennwirkungsgrad
116 if ($iEtaNenn < 0.0 || $iEtaNenn > 0.2){
117     // $iEtaG = 0.12; // EteG macht keinen Sinn
118     $iEtaNenn = 0.12;

```

Abbildung 3: Quellcode elektrischer Teil (Auszug aus ElektrischeBerechnung.class.php)

3.3 Facilities

Zu den Tätigkeiten am Rechenkern gehörte auch die Entwicklung eines Schnittstellen-Containers, der es ermöglicht, unterschiedliche Facilities zu definieren. Eine Facility ist ein Container, der es ermöglicht, die Basismodule der Simulation anzusprechen. So ist es möglich, für verschiedene Simulationsanwendungen (z.B. Solarplansimulation Version 1, Anlagensimulation für Safer'Sun Version 2, Solarstromprognose) aufrufende Container zu erstellen, die den Simulationsrechenkern benützen.

3.4 Version 1: Prognose des Jahresertrages

Die Erstellung der Version 1 beinhaltet eine Simulation, die den jährlichen erwarteten Ertrag einer noch nicht gebauten Solaranlage errechnet. Es wurde ein Softwaremodul entwickelt, das aufgrund einer vorgegebenen Solaranlagenkonfiguration (Wechselrichter, Module, Verschaltung, etc.) und der meteocontrol-Wetterdaten (Einstrahlung, Temperatur) des geplanten Standorts die erwartete erzeugte Energie simuliert. Die Version 1 ist vor allem eine Dienstleistung für Firmen, die selbst Solaranlagen installieren und potentiellen Kunden eine schnelle Möglichkeit bieten wollen, den möglichen Ertrag ihrer geplanten Solaranlage zu erfahren.

3.4.1 Web-Service Anbindung

Bei Version 1 ist die Schnittstelle zum Web-Service unbedingt notwendig, ansonsten gibt es keine Verwendung für diese Simulationsart. In der Regel sind Verkäufer von Solaranlagen direkt vor Ort beim Kunden und erstellen live im Internet ein spezielles, auf die Kundenbedürfnisse angepasstes Angebot. Mit einer lauffähigen Version 1 ist es nun möglich, das Angebot um einen erwarteten Jahresertrag der geplanten Solaranlage zu erweitern. Die Tätigkeiten am Web-Service waren Bestandteil von Arbeitspaket 3.3.

Über den Web-Service-Client wird eine Anfrage über das Internet an den Web-Service-Server im meteocontrol-Serverpark verschickt. Die notwendigen Konfigurationsdaten der

Solaranlage werden übermittelt. Die Simulation wird gestartet und greift dabei auf die Datenbank von meteocontrol zu. Es werden Wetterdaten sowie technische Informationen über verwendete Wechselrichter und Solarmodule ausgelesen. Der erwartete Jahresertrag wird abschließend dem Web-Service-Client übermittelt. Es wurde bereits ein zahlender Kunde (S.A.G.) für die Version 1 gewonnen.

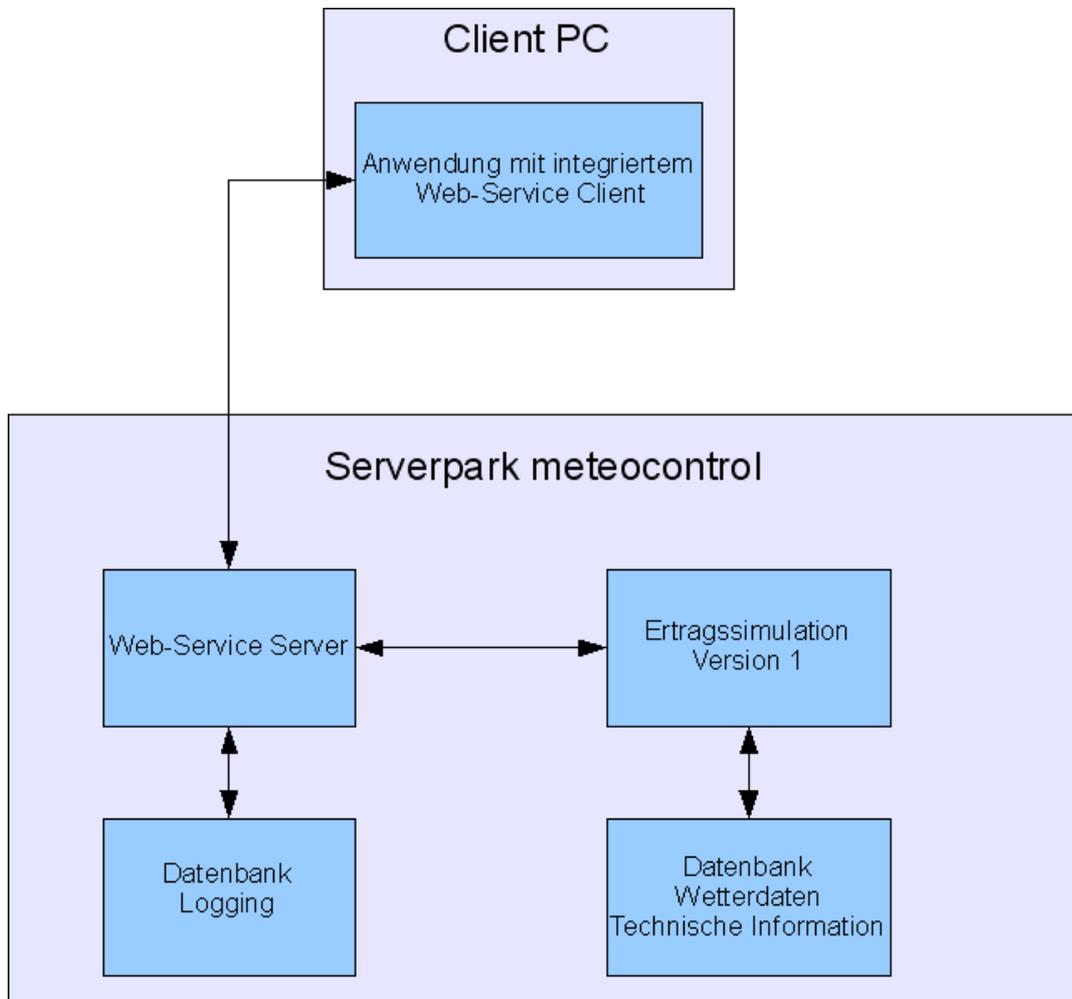


Abbildung 4: Schematischer Verlauf der Ertragssimulation

Abbildung 4 zeigt den schematischen Verlauf der Ertragssimulation Version 1. Endergebnis ist der zu erwartende Jahresertrag einer vom Kunden geplanten Solaranlage.

3.4.2 Logik der Ertragssimulation Version 1

Im Folgenden wird die innere Logik der Ertragssimulation Version 1 dargestellt. Es handelt sich hierbei um den in PHP programmierten Rechenkern der dann vom Web-Service Server über entsprechende Programmschnittstellen angetriggert werden kann. Die Module „Berechnung der Einstrahlung“ und „Technischer Teil“ werden so auch in allen anderen Versionen verwendet.

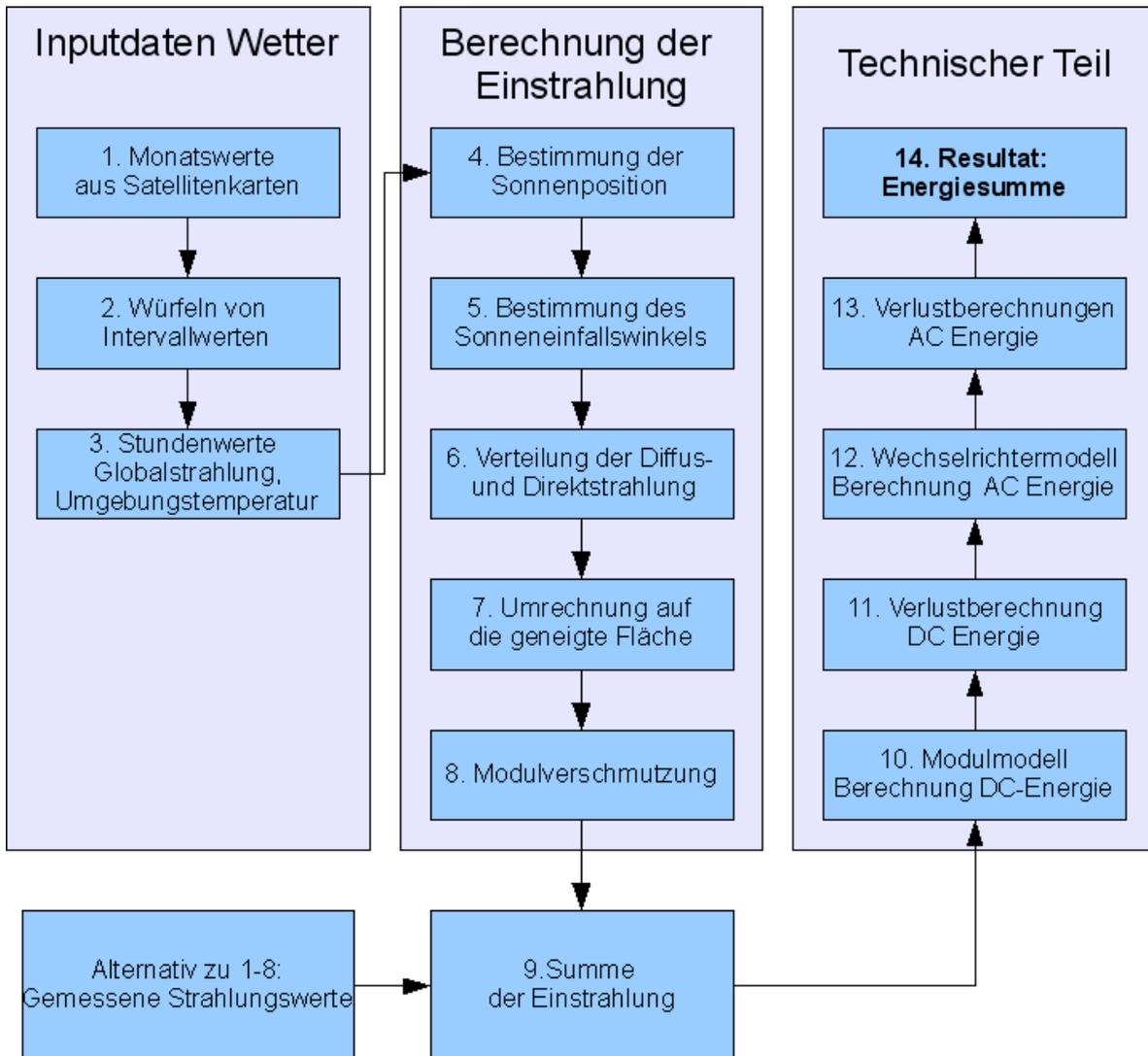


Abbildung 5: Schematische Darstellung der Programmlogik von Version 1

Die in Abbildung 5 dargestellten notwendigen Schritte zur Ertragssimulation werden im Folgenden näher erläutert. Ein Durchlauf von Schritt 1 bis 14 wird immer für einen Kalendertag durchgeführt. Soll ein ganzes Jahr berechnet werden, so wird die Simulation 365 mal durchgeführt. Es ist also generell möglich, den erwarteten Ertrag für einen beliebig großen Zeitraum zu berechnen.

Schritt1: Monatswerte aus Satellitenkarten: Es werden aus vorhandenen digitalen Satellitenkarten Wetterdaten für den Standort der geplanten Solaranlage ausgelesen. Es handelt sich hierbei um Globalstrahlung und Umgebungstemperatur. Es gibt nur einen Wert pro Monat

Schritt 2: Würfeln von Intervallwerten: Für die Simulation werden Intervallwerte im Stundenrhythmus benötigt. Aus diesem Grund wird eine Tagesverteilung der Satellitendaten per Zufallszahlengenerator erwürfelt.

Schritt 3: Das Ergebnis sind Stundenwerte für die Globalstrahlung und die Umgebungstemperatur.

Schritt 4: Bestimmung der Sonnenposition für den zu berechnenden Tag

Schritt 5: Bestimmung des Sonneneinfallswinkels aufgrund des Standorts der Solaranlage

Schritt 6: Die synthetisierte Einstrahlung wird aufgeteilt in Diffus- und Direktstrahlung.

Schritt 7: Die berechnete Einstrahlung wird umgerechnet auf die Neigung der Solaranlagenmodule. Die Neigung ist ein Parameter, der dem Web-Service Server übergeben wird.

Schritt 8: Eine simulierte Modulverschmutzung verringert die Einstrahlung auf die Solarmodule.

Schritt 1-8 alternativ: Bei vor Ort gemessener Einstrahlung kann sofort mit Schritt 9 begonnen werden. Diese Alternative ist vor allem in Version 2 und 3 möglich. Es können hier die an bereits gebauten Solaranlagen befestigten Einstrahlungssensoren verwendet werden. Ist eine Anlage noch in Planung, besteht diese Möglichkeit nicht.

Schritt 9: Die erwartete Summe der Einstrahlung, die auf die Solarmodule trifft, wird auf Stundenbasis berechnet.

Schritt 10: Es wird die erwartete Energie auf der Gleichstromseite der Solaranlage berechnet. Hierzu werden die technischen Daten der verwendeten Solarmodule benötigt. Außerdem ist es erforderlich zu wissen, wie die Module miteinander verbunden sind (Strings). Es wird der modulabhängige Faktor pro Wechselrichtereingang bestimmt. In diesem Schritt fließen auch die ermittelten Temperaturwerte mit ein.

Schritt 11: Es werden Verluste (z.B. Kabelverluste) auf der Gleichstromseite berechnet.

Schritt 12: Es wird die erwartete Energie auf der Wechselstromseite berechnet. Hierzu wird das Wechselrichtermodell verwendet. Es werden technische Daten zum ausgewählten Wechselrichterhersteller aus der Datenbank ausgelesen.

Schritt 13: Es werden Verluste (z.B. Kabelverluste) auf der Wechselstromseite berechnet.

Schritt 14: Es wird die simulierte Energiesumme des Tages und letztendlich des Gesamtzeitraumes berechnet

Version 1 wird im laufenden Betrieb verwendet. Durch zusätzliche Kundenwünsche ergeben sich immer wieder leichte Anpassungen und Erweiterungen. Version 1 wurde erfolgreich in die Produktpalette von meteocontrol aufgenommen und wird auch nach Ablauf des Projektes weiterentwickelt werden.

3.5 Version 2: Simulation für Safer'Sun

Die Simulation in Version 2 greift direkt auf Solaranlagenkonfigurationen im System von meteocontrol zu. Es handelt sich hierbei um die ca. 7000 Solaranlagen, die momentan überwacht werden. Ziel ist hier eine tägliche Simulation des erwarteten Ertrags des Vortages jeder Anlage im System, dies wurde bis jetzt von SolEm erledigt. Im meteocontrol Internetportal Safer'Sun kann dann die Simulation mit den tatsächlich gemessenen Erträgen verglichen werden. Weicht die Simulation stark vom tatsächlichen Ertrag ab, so ist ein Defekt bei der Solaranlage zu vermuten, und es wird ein Alarm ausgelöst. Die Version 2 setzt auf den Tätigkeiten für Version 1 auf.

Folgende Module wurden für Version 2 entwickelt:

- **Modul 1:** Ablaufsteuerung: Dieses Programm ist für die komplette Ablaufsteuerung der Simulation verantwortlich und ruft die anderen Module der Reihe nach auf.
- **Modul 2:** Wetterdatenbeschaffung. Es werden Globalstrahlungsdaten und Temperaturdaten aus der Wetterdatenbank von meteocontrol ausgelesen. Hierbei werden u.a. die vom Projektpartner Universität Oldenburg bereitgestellten koordinatenbasierten Daten verwendet. Die Einstrahlung und die Temperatur

können von lokal an der Solarstromanlage installierten Sensoren stammen. Ist vor Ort keine Sensorik vorhanden, wird auf die meteosat-Globalstrahlung der Universität Oldenburg und die Temperaturen des meteomedia-Wetterstationsnetzes zurückgegriffen.

- **Modul 3:** Berechnung der Einstrahlung (**siehe Version 1**)
- **Modul 4:** Technischer Teil (**siehe Version 1**)
- **Modul 5:** Speicherung: Abspeichern der Daten in der Datenbank und Logging

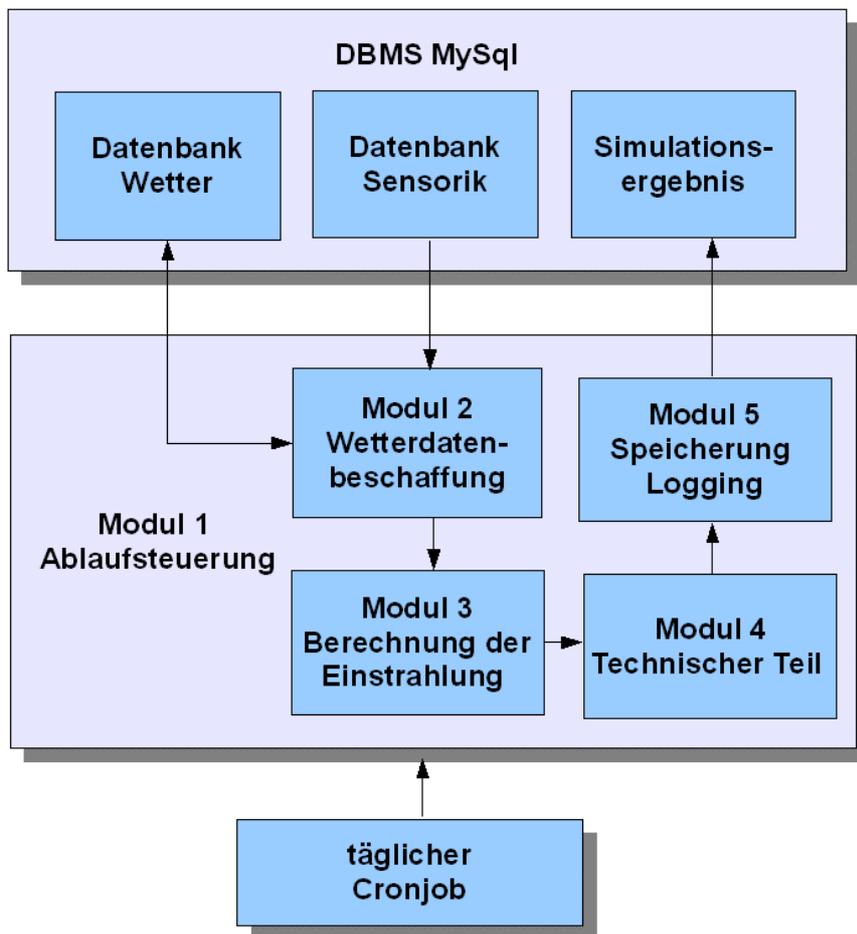


Abbildung 6: Ablaufdiagramm Version 2

Die Tätigkeiten zu Version 2 sind vollständig abgeschlossen. Eine Anbindung an das Live-System von meteocontrol hat aber noch nicht stattgefunden. Nach Ablauf von WISENT wird dies die erste Tätigkeit von meteocontrol im Bereich Solarsimulation sein.

3.6 Version 3 Parallele Simulation von Solarstromanlagen

3.6.1 Anbindung an den WISENT-Cluster

Um eine gridbasierte und parallelisierte Simulation zu realisieren, war eine technische Anbindung an den WISENT-Cluster des Projektpartners OFFIS notwendig. Folgende Schritte wurden hierfür durchgeführt.

Serverpark meteocontrol

- Bereitstellung eines Servers im meteocontrol-Serverpark
- Installation von Debian-Linux, JDK, ANT, Postgres
- Installation von Globus Toolkit 4
- Installation von Condor
- Installation von GridFTP-Server auf xinetd Basis
- Austausch von Zertifikaten mit OFFIS über Gridproxy
- Anpassen der Firewall, um Kontakt zum OFFIS zu ermöglichen

WISENT-Cluster (Tätigkeiten von OFFIS)

- Installation von PHP

3.6.2 Parallelisierung

Um die Simulation zu parallelisieren ist es notwendig, den PHP-Programmcode vollständig auf dem WISENT-Cluster zu installieren. Hier können dann mehrere Simulationen von Solaranlagen gleichzeitig durchgeführt werden. Ein automatischer Cronjob im meteocontrol-Serverpark bereitet die Simulation vor. Es werden die Konfigurations- und Messdaten aller zu simulierenden Solaranlagen ausgelesen und in einer Transferdatei gespeichert. Die Datei wird via Condor und GridFTP zum OFFIS-Cluster übertragen.

Die parallele Simulation beinhaltet folgende Arbeitsschritte:

- Schritt 1: Auslesen der Simulationsqueue und Bestimmung der zu simulierenden Solaranlagen
- Schritt 2: Auslesen der Konfiguration der Solaranlagen (z.B. Wechselrichtertyp, Modultyp, Verschaltung)
- Schritt 3: Auslesen der notwendigen Wetterdaten (Temperatur, Globalstrahlung)
- Schritt 4: Ablegen der ausgelesenen Informationen in ein Transferfile
- Schritt 5: Übertragung des Transferfiles via Condor und GridFTP zum WISENT-Cluster
- Schritt 6: Starten der parallelen Simulation auf dem WISENT-Cluster
- Schritt 7: Speichern der Ergebnisse via Condor und GridFTP im meteocontrol-Serverpark
- Schritt 8: Speichern der Simulationsergebnisse in der meteocontrol-Datenbank

Die Arbeiten für die parallele Simulation sind noch nicht abgeschlossen. Die Kommunikation via Condor mit dem WISENT-Cluster ist bereits etabliert. Im Moment wird noch an der Fertigstellung des PHP-Containers für die parallele Simulation gearbeitet. Das Ziel einer gridbasierten parallelen Simulation wird bis Projektende erreicht werden.

4 Positive Seiteneffekte

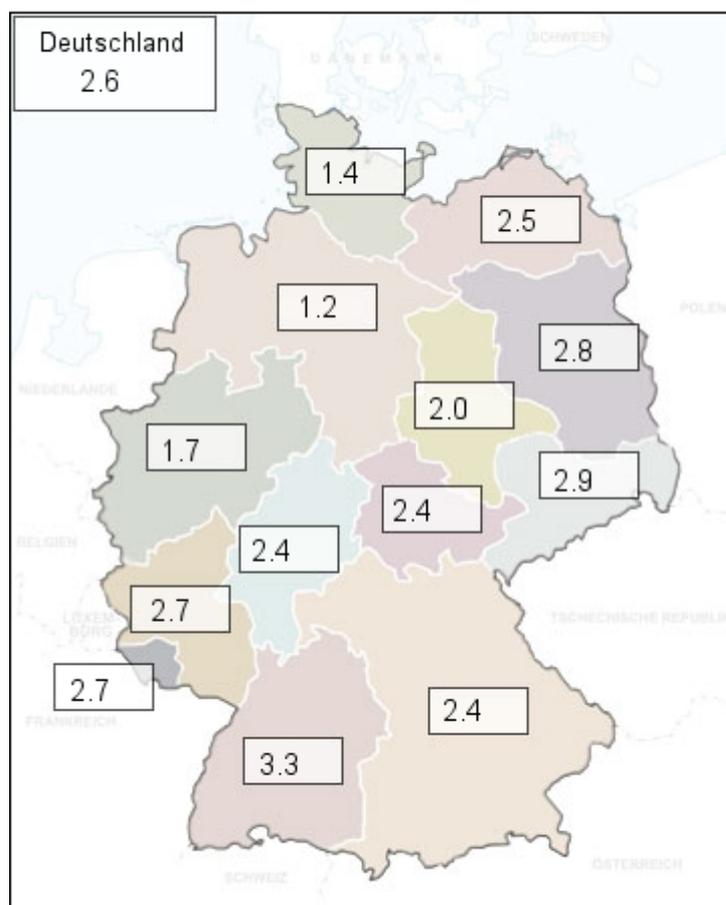
Durch die Tätigkeiten bei WISENT gab es positive Seiteneffekte auf zwei weitere Projekte bei meteocontrol.

4.1 Solarstromprognose

Die von meteocontrol entwickelte Solarstromprognose dient zur Berechnung der erwarteten, durch deutsche Solarstromanlagen erzeugte, Energie. Sie ist somit ein Werkzeug zur Prognostizierung der eingespeisten Energie ins Stromnetz. Diese Information ist vor allem für die deutschen Netzbetreiber immer wichtiger, da der Anteil der Solarenergie beim Energiemix immer höher wird. Die Solarstromprognose diente als Einstieg für die Programmierung der parallelen Simulation von Solaranlagen. Momentan können Prognosewerte für heute und morgen berechnet werden. Die Auswertung findet auf Bundeslandebene statt, zusätzlich wird ein Wert für Gesamtdeutschland geliefert. Es handelt sich hierbei um die relative erzeugte Leistung für jedes Gebiet. Die Solarstromprognose befindet sich schon im kommerziellen Einsatz, es liegen weitere Kundennachfragen für eine andere räumliche und zeitliche Auflösung vor. Erst die Tätigkeiten für die parallele Simulation machten die Erstellung der Solarstromprognose möglich. Der entwickelte Rechenkern wird für die Solarstromprognose verwendet. Die Solarstromprognose wurde veröffentlicht, und auf Kundenwunsch in die Portale von meteocontrol grafisch eingebunden.

Solarstromprognose Deutschland für den

Tag: 18 Monat: 7 Jahr: 2008



[Werte in kWh/kWp berechnet aus Ertrag und Nennleistung]

Abbildung 7: Eingespeiste Solarenergie relativ zur installierten Leistung

In der in Abbildung 7 dargestellten Karte wird die mittlere eingespeiste Energie relativ zur installierten Leistung pro Tag und Bundesland bzw. ganz Deutschland dargestellt.

4.2 Automation von Solargutachten

Durch die Version 1 der Simulation wurde es ermöglicht, automatisierte Solargutachten für Kleinanlagen zu realisieren. Es wurden bereits 130 Gutachten für Kunden erstellt.

5 Globalstrahlungsdaten Universität Oldenburg

Der Projektpartner Universität Oldenburg stellte der meteocontrol meteosat-Globalstrahlungsdaten zur Verfügung. Es wurde eine bidirektionale Datenübertragung eingerichtet. Die Globalstrahlungsdaten sind koordinatenbasiert, d.h. für jeden Standort einer europäischen Solaranlage im Safer'Sun Portal liefert die Universität Oldenburg täglich die Globalstrahlungsdaten des Standorts in stündlicher Auflösung.

20080730														
000066	48.3667	10.9000	0	0	0	0	39	144	335	511	632	714	801	831
000081	49.3833	9.1500	0	0	0	0	41	160	316	476	620	724	785	763
000087	48.7000	8.9500	0	0	0	0	43	154	300	471	600	702	781	789
000089	48.1333	9.0833	0	0	0	0	38	156	328	492	633	754	833	838
000090	48.3167	10.8333	0	0	0	0	36	146	342	510	628	731	811	839
000095	48.1167	11.5833	0	0	0	0	55	124	262	524	670	758	814	846
000110	48.5500	11.4833	0	0	0	0	50	137	273	387	636	768	788	816
000116	48.1667	8.6667	0	0	0	0	31	149	326	486	633	750	817	663
000118	47.5500	7.6833	0	0	0	0	19	129	299	453	617	716	718	337
000121	48.3667	10.9000	0	0	0	0	39	144	335	511	632	714	801	831
000123	48.0167	7.8167	0	0	0	0	28	130	298	468	611	741	826	846
000128	48.0000	7.8500	0	0	0	0	28	125	300	469	610	739	814	838
000129	52.5167	13.3000	0	0	0	4	36	83	144	162	265	273	238	370
000130	48.5000	9.1167	0	0	0	0	43	161	320	489	640	727	794	809
000133	48.5500	11.4833	0	0	0	0	50	137	273	387	636	768	788	816
000134	48.5500	11.4833	0	0	0	0	50	137	273	387	636	768	788	816
000135	48.5500	11.4833	0	0	0	0	50	137	273	387	636	768	788	816
000136	48.5500	11.4833	0	0	0	0	50	137	273	387	636	768	788	816
000137	47.5500	7.6833	0	0	0	0	19	129	299	453	617	716	718	337
000138	48.4333	8.8167	0	0	0	0	40	151	318	485	624	720	807	803
000140	48.8833	9.4167	0	0	0	0	46	166	325	476	602	703	788	800
000141	48.0000	7.8500	0	0	0	0	28	125	300	469	610	739	814	838
000144	48.5167	8.0667	0	0	0	0	27	145	297	455	595	734	808	832
000145	53.1167	8.2667	0	0	0	1	37	108	186	301	581	704	773	797

Abbildung 8. An meteocontrol gelieferte Globalstrahlung der Universität Oldenburg

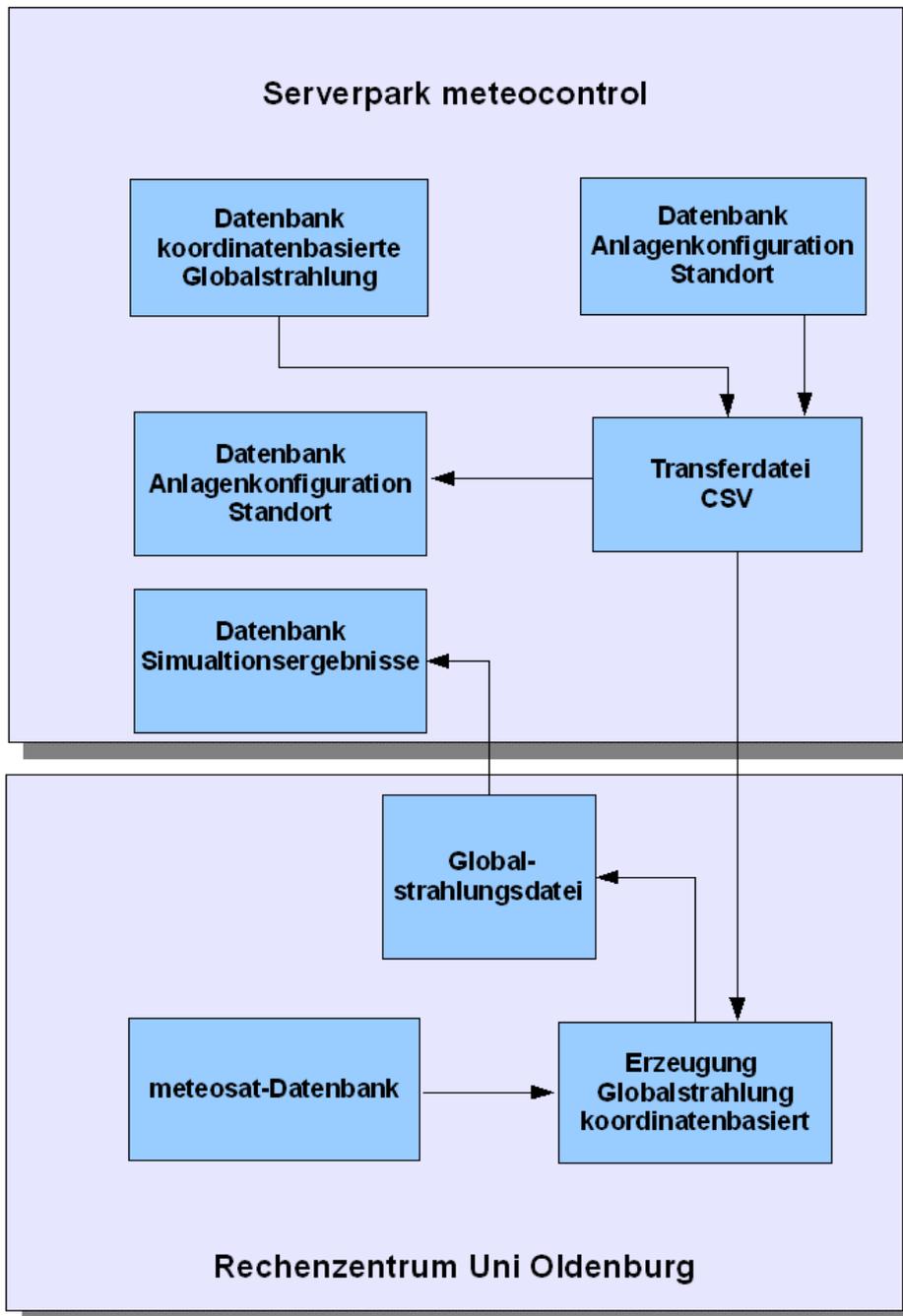


Abbildung 9: Globalstrahlung Universität Oldenburg -> meteocontrol



DLR

**Deutsches Zentrum
für Luft- und Raumfahrt e.V.**
in der Helmholtz-Gemeinschaft



OFFIS
INSTITUT FÜR INFORMATIK

Verarbeitung energiemeteorologischer Daten in Geographischen Informationssystemen

Arbeitspaket 5.4

Sönke Brummerloh
OFFIS e.V.
Escherweg 2
26121 Oldenburg

Internet: <http://www.offis.de>

Carsten Hoyer-Klick
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Technische Thermodynamik
Pfaffenwaldring 38
D-70569 Stuttgart

Internet: <http://www.dlr.de/tt/>

1 Einleitung

Bisherige energiemeteorologische Analysen bei DLR-TT sind durch relativ lange Rechenzeiträume gekennzeichnet, z.B. rechnet eine 10-jährige Solarstrahlungskarte für ein mittelgroßes Land wie Spanien durchaus 2-3 Monate. Einer vergleichenden Analyse über mehrere Konfigurationen werden dadurch enge zeitliche Grenzen gesetzt. Gerade im Kontext der Standortsuche bei der Projektplanung für Energieanlagen und bei der Erstellung von politikberatenden Studien sind jedoch statistisch abgesicherte, umfangreiche Entscheidungsgrundlagen erforderlich, die auf der Analyse von breiten Optionsräumen beruhen. Diese Tatsache wird umso wichtiger, wenn nicht nur eine Stromerzeugungstechnologie (z.B. Solarkraftwerke), sondern mehrere Technologien und ihre Wechselwirkungen analysiert und aus wirtschaftlicher Sicht verglichen werden sollen.

Einer der Schwerpunkte bei der Verarbeitung energiemeteorologischer Daten bei DLR-TT ist die Prozessierung von Fernerkundungsdaten, um Karten der solaren Einstrahlung zu erstellen. Die Rohdaten werden im DIMS-Archiv (DIMS = Data Information and Management System) des DFD geholt und bei DLR-TT verarbeitet. Dabei geht es oft auch um große Regionen, wie z.B. den Mittelmeerraum. Auf einem einzelnen Rechner würde die Verarbeitung der Daten mehrere Jahre dauern. Ein Vorteil der genutzten Daten ist, dass sie sich gut in Teilrechnungen aufteilen lassen. Eine Gesamtrechnung lässt sich in unabhängige Regionen oder Zeitabschnitte (z.B. eine 10-Jahresrechnung in die einzelnen Jahre oder Monate) aufteilen. Eine manuelle Parallelisierung in mehrere Läufe von Teilbildern oder Teilzeiträumen ist personell aufwändig und fehleranfällig.

Die oben genannten Anforderungen lassen sich durch die Durchführung von vielen unabhängigen, nebenläufigen Simulationen erfüllen, die mit verschiedenen Randbedingungen und Parametervariationen der Ausgangskonfiguration angestoßen werden. Exemplarisch wurde in diesem Arbeitspaket der Strahlungsprozessor als zu parallelisierende Komponente ausgewählt. Die entwickelte Lösung lässt sich aber auch leicht auf andere rechenintensive Komponenten übertragen.

Im Folgenden wird zunächst die Funktionsweise des Strahlungsprozessors beschrieben, bevor auf die implementierte Parallelisierungslösung eingegangen wird. Anschließend werden Erfahrungen mit der neuen Lösung beschrieben und ein Ausblick auf weitere Entwicklungsmöglichkeiten gegeben.

2 Strahlungsprozessor

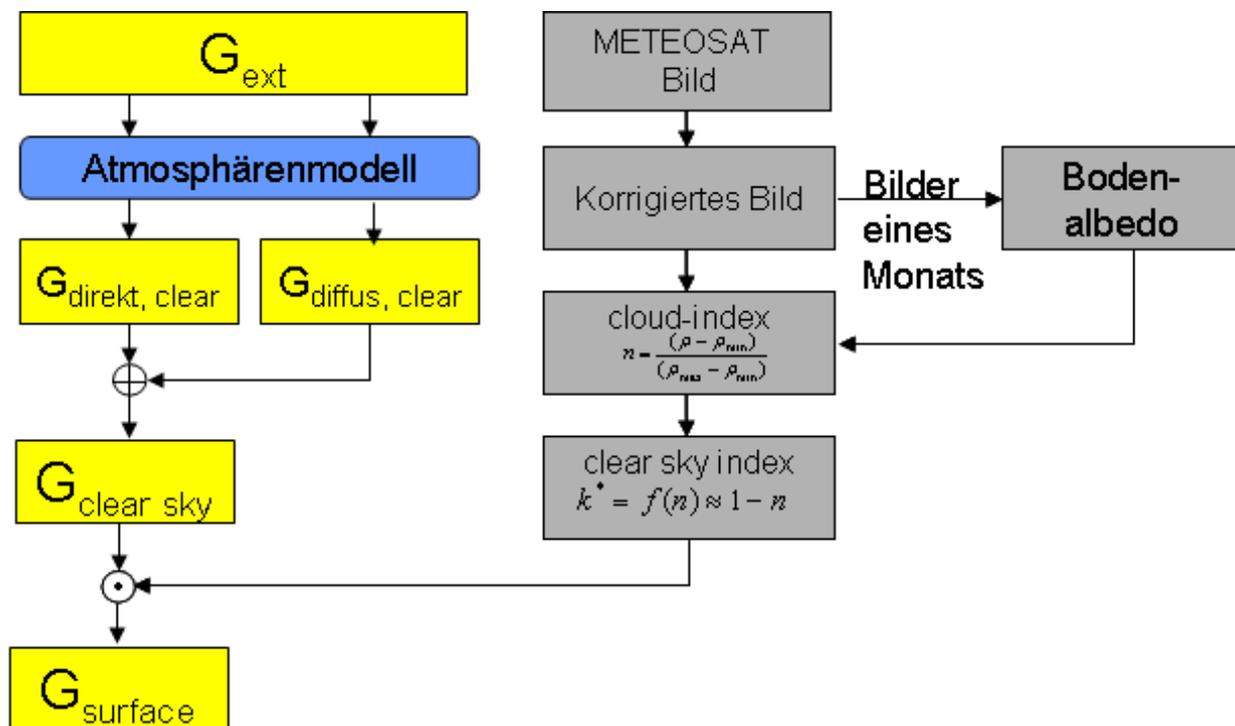


Abbildung 1: Modell der Berechnung der solaren Einstrahlung.

Abbildung 1 zeigt den schematischen Ablauf der Berechnung. Die Prozessierung gliedert sich in zwei Schritte: In die Berechnung des Wolkenindex (rechter Teil) und in die Berechnung der solaren Einstrahlung (linker Teil).

Als erstes wird der Wolkenindex aus den halbstündlichen Meteosat-Bildern berechnet. Der Wolkenindex ist ein Indikator für die im Meteosat-Bild sichtbare Bewölkung, er läuft von 0 (keine Bewölkung) bis 1 (vollständig bewölkt). Die Berechnung erfolgt monatsweise für jeden Bildslot. Die Satelliten der ersten Meteosat-Generation erstellen jede halbe Stunde ein Bild, bei dem die Erde innerhalb dieser Zeit zeilenweise von Süden nach Norden abgescannt wird. Ein solches halbstündiges Bildintervall wird als Bildslot bezeichnet. Das Programm für die Wolkenindexerstellung nimmt alle Bilder eines Monats für einen bestimmten Slot und bestimmt daraus den Wolkenindex. Dazu werden innerhalb dieses Monats die dunkelsten Bildpixel bestimmt, um daraus ein Referenzbild der wolkenfreien Bodenreflektivität zu bestimmen. Aus einer Differenzbildung von dem aktuellen Bild und dem Referenzbild kann der Wolkenindex bestimmt werden. Da die Berechnungen für die einzelnen Monate und Slots unabhängig sind, lassen sich diese Rechnungen gut parallelisieren.

Im zweiten Schritt wird zunächst die solare Einstrahlung für den wolkenfreien Fall berechnet. Aus dem Wolkenindex kann der clear sky index bestimmt werden. Der clear sky index ist das Verhältnis von wirklicher Einstrahlung zum wolkenfreien Fall. Im wolkenfreien Fall ist er eins und im völlig bewölkten Fall nahe null. Mit diesem clear sky index kann die stündliche Einstrahlung für jedes Bildpixel berechnet werden. Für diese Berechnung wird eine räumliche Mittlung über typischer Weise 5x5 Pixel und ein gewichtetes Mittel aus drei halbstündigen Satellitenbildern genutzt. Deswegen sind diese

Rechnungen auf der Zeitskala nur tageweise unabhängig. Da bei Karten aber Tagessummen weniger interessant als Monatssummen sind und die Ausgabe auch monatsweise erfolgt, bietet sich eine Aufteilung in Monatsblöcke an.

Die Programme für die eigentliche Prozessierung der Satellitendaten sind in ANSI-C implementiert. Die notwendigen Skripte, z.B. für die Monats- und Slotweise Berechnung des Wolkenindex, sind in Perl implementiert.

3 Parallelisierung

Es wurde nicht direkt der C-Code des Strahlungsprozessors parallelisiert, sondern eine flexible Möglichkeit geschaffen, die bestehenden Strahlungsprozessorberechnungen mit unterschiedlichen Parametern aufzurufen. Zur Parallelisierung werden diese Aufrufe als Jobs an die High-Throughput Computing Software Condor¹ geschickt.

Beispiel:

Statt die Berechnung von Strahlungskarten jeweils mit einem Zeitraum von einem Jahr durchzuführen, kann die Berechnung jetzt monatsweise parallel durchgeführt und zum Schluss die Ergebnisse integriert werden. Bei einer monatsweisen Berechnung für ein Jahr kann die Laufzeit bei genügend Rechenressourcen auf ein zwölftel der nicht parallelisierten Laufzeit reduziert werden.

Im Folgenden wird zunächst der grobe Aufbau der entwickelten Lösung beschrieben, um darauf aufbauend die Realisierung mit Konfigurationsdatei und Template-Dateien zu erläutern.

3.1 Vorbereitungen

Die bestehende Software für die Prozessierung der Solarstrahlungsdaten war eine „single user“ Einzellösung. D.h. der C-Quellcode war bei einem Nutzer in einer alten Borland (C-BuilderX) Entwicklungsumgebung integriert. Aus diesem Grund gab es kein zentrales Repository und keine verteilte Entwicklung. Als vorbereitender Schritt ist der bestehende Quellcode in ein Eclipse-Projekt überführt worden, dass in einem Subversion-Repository verwaltet wird. Dies war zum einen Voraussetzung, um mit DLR-TT und OFFIS gemeinsam an der Parallelisierung der Prozesskette zu arbeiten, zum anderen ermöglicht es dem DLR in Zukunft im Team (z.B. aus DLR-TT und dem DFD) an der Weiterentwicklung der Prozessoren zu arbeiten.

3.2 Aufbau

Ziel war es, eine Lösung zu entwickeln mit der auf möglichst einfache Weise eine Berechnung parallelisiert werden kann. Das Aufteilen der Berechnung auf mehrere Jobs sollte möglichst automatisch geschehen.

Für eine möglichst flexible und einfach zu handhabende Lösung wurden die Template-Fähigkeiten² der Programmiersprache Groovy³ genutzt. Diese ermöglichen es, in Template-Dateien Variablen zu verwenden, die in einer Groovy-Anwendung mit Werten gefüllt werden können. Auf diese Weise können aus einer Grunddatei mit dynamischen Inhalten neue konkrete Dateien generiert werden. Die veränderlichen Inhalte werden zur leichten Handhabung in einer Konfigurationsdatei angegeben, bei der es sich um eine

¹ <http://www.cs.wisc.edu/condor/>

² <http://groovy.codehaus.org/Groovy+Templates>

³ <http://groovy.codehaus.org/>

Java-Properties-Datei⁴ handelt. In Abbildung 2 ist dieser Workflow grafisch veranschaulicht.

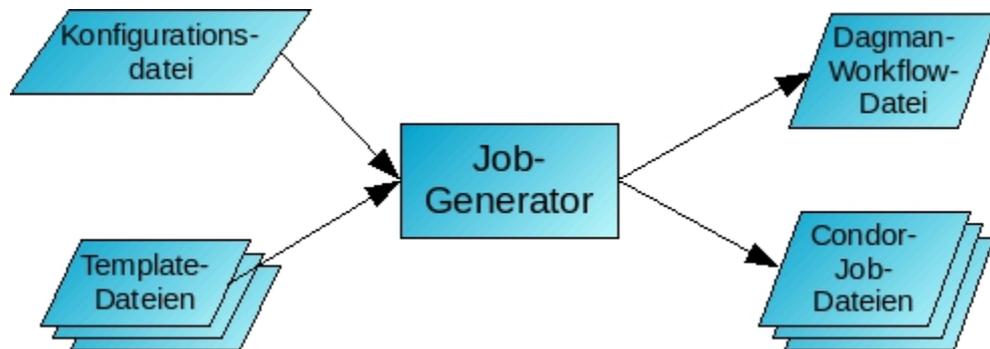


Abbildung 2: Workflow mit Job-Generator-Artefakten

Die grundsätzliche Struktur beim Ausführen der Jobs wird in Abbildung 3 dargestellt: Zunächst kann optional vor den eigentlichen Berechnungen ein initialer Job ausgeführt werden, um die Laufzeitumgebung auf die Berechnungen vorzubereiten. Danach werden die Berechnungen parallel über mehrere Jobs ausgeführt. Wenn die Berechnungen abgeschlossen sind, kann optional noch ein finaler Job ausgeführt werden, um Aufräumarbeiten durchzuführen und Ergebnisse zusammenzusammeln.

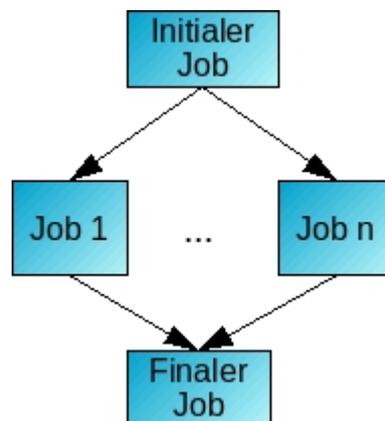


Abbildung 3: Jobs, die an einer Berechnung beteiligt sein können.

3.3 Konfigurationsdatei und Template-Dateien

In der implementierten Lösung wird ein Java-basiertes Programm mit einer Konfigurationsdatei als Parameter aufgerufen. Die Konfigurationsdatei ist relativ einfach aufgebaut und besteht lediglich aus einzelnen Wertzuweisungen in folgender Form:

```
base_dir = /home/condoruser/strahlungsprozessor
```

Diese Konfigurationsdatei ist in der Regel das Einzige, das von einem Nutzer angepasst werden muss.

⁴ <http://java.sun.com/docs/books/tutorial/essential/environment/properties.html>

In der Konfigurationsdatei werden das zu durchlaufende Intervall, sowie die Schrittgröße mit der über das Intervall iteriert wird, spezifiziert. Dabei können mehrere Schrittgrößen angegeben werden, die in ihrer Reihenfolge für die Iteration verwendet werden. Auf diese Weise ist es beispielsweise möglich, die Tage eines Jahres zu durchlaufen und dabei als Schrittlänge die Tage des jeweiligen Monats zu verwenden.

Beispiel:

Für ein gewöhnliches Jahr mit 365 Tagen würde sich das Intervall von 1 bis 365 erstrecken. Die Schritte wären dann 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30 und 31. Auf diese Weise würde die Iteration mit den Werten 1, 32, 60, 91, 121 usw. arbeiten, was dem jeweils ersten Tag eines Monats, bezogen auf die Tage im Jahr, entspricht.

Statt einzelner Zahlen können auch Zeitangaben verwendet werden (siehe unten). Zeitliche Besonderheiten, wie z.B. unterschiedlich viele Tage pro Monat oder Schaltjahre, werden dann automatisch vom Job-Generator berücksichtigt.

Für den initialen, die zentralen und den finalen Job werden in der Konfigurationsdatei die jeweilige ausführbare Datei, deren Parameter und eine Template-Datei spezifiziert. Aus den Template-Dateien werden die Job-Dateien für Condor generiert. Normalerweise brauchen diese durch die Nutzer nicht modifiziert zu werden. In speziellen Fällen, wenn noch nicht berücksichtigte Berechnungen parallelisiert werden sollen, können hier Anpassungen vorgenommen werden. Als ausführbare Datei können sowohl binäre ausführbare Dateien, als auch ein Java-Interpreter oder Interpreter von Skriptsprachen wie z.B. Bash und Perl angegeben werden.

Die Template-Datei für die Jobs könnte beispielsweise wie folgt aussehen:

```
notification = error
executable = ${executable}
arguments = ${arguments}
transfer_executable = false
whentotransferoutput = ON_EXIT
universe = vanilla
error = ${job.name}.err
output = ${job.name}.out
log = ${job.name}.log
queue
```

Die Variablen `executable` und `arguments` stehen hier für die aktuell verwendeten Werte der ausführbaren Datei und der zugehörigen Parameter. Für den initialen Job, die Rechenjobs und den finalen Job werden jeweils eine eigene ausführbare Datei und Parameter angegeben. Diese werden in jeweils unterschiedlichen Variablen gespeichert (`pre_executable` und `pre_arguments`, `main_executable` und `main_arguments`, sowie `post_executable` und `post_arguments`). Um eine Template-Datei für alle Job-Arten verwenden zu können, können die beiden Variablen `executable` und `arguments` verwendet werden. Diese stellen Synonyme für die jeweils verwendete ausführbare Datei und deren Parameter dar.

Eine mit der obigen Template-Datei generierten Job-Datei könnte dann wie folgt aussehen:

```
notification = error
```

```
executable =
/data/condoruser/strahlungsprozessor/cloudindex2Gmap_v6.14
arguments = -sy 2005 -sd 1 -ey 2005 -ed 31 -o
/home/condoruser/strahlungsprozessor/SE_Spain1/SE_Spain
transfer_executable = false
whentotransferoutput = ON_EXIT
universe = vanilla
error = SE_Spain1.err
output = SE_Spain1.out
log = SE_Spain1.log
queue
```

Wie in Template-Dateien können in der Parameterangabe für die ausführbare Datei ebenfalls Variablen eingefügt werden, die für den jeweiligen Job durch konkrete Werte ersetzt werden. Für jeden Iterationsschritt im spezifizierten Intervall wird eine Berechnungs-Job-Datei erstellt. Die Werte der aktuellen Iteration können verwendet werden, um die Parameter der Berechnung zu variieren. Wenn mit dem Intervall und den Schritten aus dem obigen Beispiel gearbeitet wird, können für den Starttag der jeweiligen Berechnung die Variable `loop_current_value` und für den Endtag der Ausdruck `loop_next_value-1` eingesetzt werden. Auf diese Weise würden immer der erste und der letzte Tag eines Monats angegeben. Die Variable `loop_current_value` steht für den aktuellen Wert der Iteration während die Variable `loop_next_value` den Wert der nächsten Iteration enthält. Es wäre daher in der Konfigurationsdatei eine Parameterangabe wie

```
main_arguments =-sy 2005 -sd ${loop_current_value} -ey 2005
-ed ${loop_next_value-1} -o ${job.dir}/${job_name_prefix}
```

möglich. Falls als Start- und Endwerte der Iteration Zeitpunkte angegeben wurden, liefern `loop_current_value` und `loop_next_value` ein Datum bzw. eine Uhrzeit. Das Format kann über `output_format` festgelegt werden. Falls ganz gezielt mit einzelnen Zeitelementen gearbeitet werden muss, können zusätzlich die Variablen `loop_current_value_object` und `loop_next_value_object` verwendet werden. Hierbei handelt es sich um Java-Calendar-Objekte⁵ von denen einzelne Methoden aufgerufen werden können, um gezielt Werte auszulesen. Die obige Zeile könnte daher auch wie folgt lauten:

```
main_arguments = -sy $
{loop_current_value_object.get(Calendar.YEAR)}
-sd ${loop_current_value} -ey $
{loop_next_value_object.get(Calendar.YEAR)}
-ed ${loop_next_value} -o ${job.dir}/${job_name_prefix}
```

Hierbei gilt die Annahme, dass folgende Werte gesetzt sind:

```
from = 01-01-2005
to = 31-12-2005
```

⁵ <http://java.sun.com/j2se/1.5.0/docs/api/java/util/Calendar.html>

```
input_format = dd-MM-yyyy
output_format = DDD
steps = 1M
next_value_modifier = -1d
```

Die Variablen `from` und `to` sind gesetzt auf den ersten bzw. letzten Tag des Jahres 2005. Das Format der Datumsangabe wird mit `input_format` angegeben. Mit `output_format` wird das Ausgabeformat von `loop_current_value` und `loop_next_value` auf den jeweiligen Tag im Jahr festgelegt. Jahresangaben und ähnliches fallen bei der Ausgabe also weg. Das mit `input_format` und `output_format` spezifizierte Format entspricht den Angaben, die für die Java-Klasse `SimpleDateFormat`⁶ möglich sind. Die Variable `steps` legt die Schrittlänge der Iterationen auf einen Monat fest. Auf diese Weise braucht der Nutzer nicht mehr zu berücksichtigen, wie lang die einzelnen Monate sind. Die hier verwendete Variable `next_value_modifier` gibt dabei einen Wert an, der zum `loop_next_value` bei der Ausgabe addiert wird. Diese Modifikation wirkt sich nicht auf das Durchlaufen des Intervalls aus, sondern nur auf die Ausgabe. (Ebenso steht die Variable `current_value_modifier` zur Verfügung, die den gleichen Effekt auf `loop_current_value` hat.) Um Werte zu subtrahieren, kann, wie im Beispiel gezeigt, ein negativer Wert angegeben werden. Mit Hilfe dieser Modifikatoren ist es möglich, die Werte einer Iteration für die Ausgabe so zu Ändern, dass wie im Beispiel gezeigt, einmal der letzte Tag eines Monats, durch `loop_next_value`, modifiziert um `next_value_modifier`, ausgegeben wird und in der nächsten Iteration durch `loop_current_value` der erste Tag des (nächsten) Monats. Mögliche Einheiten der Werte von `steps`, `current_value_modifier` und `next_value_modifier` für Zeitangaben sind folgende:

Einheit	Bedeutung
y	Steht für Jahre
M	Steht für Monate
w	Steht für Wochen
d	Steht für Tage
H	Steht für Stunden
m	Steht für Minuten
s	Steht für Sekunden

Die Variablen, die sich in jeder Iteration ändern sind:

Variable	Bedeutung
<code>loop_current_value</code>	Der aktuelle Wert des Intervalls, das durchlaufen wird (wenn das Intervall einen Zeitraum beschreibt, ist dieser Wert nach dem <code>output_format</code>

⁶ <http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html>

	formatiert)
loop_next_value	Der Wert der folgenden Iteration (wenn das Intervall einen Zeitraum beschreibt, ist dieser Wert nach dem output_format formatiert)
loop_current_value_object	Der aktuelle Wert der Iteration als Java-Calendar-Objekt (steht nur zur Verfügung, wenn das Intervall einen Zeitraum beschreibt)
loop_next_value_object	Der Wert der folgenden Iteration als Java-Calendar-Objekt (steht nur zur Verfügung, wenn das Intervall einen Zeitraum beschreibt)
job.name	Der Name des aktuellen Jobs
job.dir	Das Verzeichnis, in dem die aktuellen Job-Dateien liegen
job.cmd	Der Pfad der Condor-cmd-Datei des aktuellen Jobs

Falls über ein Zeitintervall iteriert wird, stehen neben diesen Variablen die beiden unveränderlichen Variablen `from_object` und `to_object` zur Verfügung. Diese repräsentieren das Start- und das Enddatum in Form eines Java-Calendar-Objekts. Über diese Objekte können die einzelnen Zeitelemente ausgelesen werden:

```
post_arguments = spain_copy.pl ${base_dir} ${job_name_prefix} $
${from_object.get(Calendar.YEAR)} ${from_object.get(Calendar.MONTH)+1} $
${to_object.get(Calendar.YEAR)} ${to_object.get(Calendar.MONTH)+1}
```

Hierbei fällt eine ungewohnte Konvention des Java-Calendar-Objekts auf: Monate werden von 0 bis 11 gezählt. Wenn die gewohnte Zählweise von 1 bis 12 benötigt wird, muss also der Wert 1 auf den Monat addiert werden. Dies gilt auch für die weiter oben vorgestellten Variablen `loop_current_value_object` und `loop_next_value_object`.

Weitere Angaben innerhalb der Konfigurationsdatei sind, ob pro Job ein Verzeichnis angelegt werden soll, sowie das Arbeitsverzeichnis, in dem neue Dateien angelegt werden, der Jobname und eine Template-Datei für die Dagman-Workflow-Datei. Dagman⁷ ist eine Anwendung von Condor mit der voneinander abhängige Jobs nacheinander ausgeführt werden können. Auf diese Weise ist es möglich, erst nach dem initialen Job die Berechnungsjobs zu starten und den finalen Job erst auszuführen, wenn alle Berechnungsjobs erfolgreich beendet wurden.

Da die Dagman-Datei je nach Job-Anzahl unterschiedlich viele Einträge enthalten kann, müssen für die Template-Datei in der Regel weiterführende Template-Konzepte von Groovy verwendet werden. Die Dagman-Template-Datei könnte z.B. so aussehen:

```
<%
    if (pre_job != null) {
        out.println "JOB ${pre_job.name} ${pre_job.cmd} DIR $
{pre_job.dir}"
        out.print "PARENT $pre_job.name CHILD "
        main_jobs.each {job -> out.print "${job.name} "}
    }
```

7 http://www.cs.wisc.edu/condor/manual/v7.0/2_10DAGMan_Applications.html

```

        out.println ""
    }

    main_jobs.each {job -> out.println "JOB ${job.name} ${job.cmd} DIR
    ${job.dir}"}

    if (post_job != null) {
        out.println "JOB ${post_job.name} ${post_job.cmd} DIR $
        {post_job.dir}"
        out.print "PARENT "
        main_jobs.each {job -> out.print "${job.name} "}
        out.print "CHILD $post_job.name"
    }
}
%>

```

Wie zu sehen ist, stehen für die Dagman-Datei noch zusätzlich das Array `main_jobs` mit den Berechnungsjobs, sowie die Variablen `pre_job` und `post_job` mit dem initialen und dem finalen Job zur Verfügung. Da die Dagman-Datei immer nach dem gleichen Prinzip aufgebaut ist, ist kein Änderungsbedarf der Dagman-Template-Datei durch den Nutzer gegeben.

Variablen mit besonderer Bedeutung, die in einer Konfigurationsdatei angegeben werden können, sind:

Variable	Bedeutung	Beispiel
from	Setzt den Startwert des zu durchlaufenden Intervalls. Alternativ kann auch ein Datum oder eine Uhrzeit angegeben werden. In diesem Fall müssen die Variablen <code>input_format</code> und <code>output_format</code> gesetzt werden, um das Datumsformat zu spezifizieren.	from = 1 oder from = 01-01-2005
to	Setzt den Endwert des zu durchlaufenden Intervalls. Alternativ kann auch ein Datum oder eine Uhrzeit angegeben werden. In diesem Fall müssen die Variablen <code>input_format</code> und <code>output_format</code> gesetzt werden, um das Datumsformat zu spezifizieren.	to = 365 oder to = 31-12-2005
steps	Setzt die Schritte mit denen durch das Intervall iteriert wird. Wenn <code>from</code> und <code>to</code> Zeitangaben darstellen, muss <code>steps</code> ebenfalls Zeitangaben enthalten. Diese werden angegeben durch eine Zahl mit direkt angehängtem Buchstaben, der die Zeiteinheit	steps = 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 oder steps = 1d, 2w, 6M, 12H, -4d

Verarbeitung energiemeteorologischer Daten in GIS (AP 5.4)

	spezifiziert (siehe oben).	
input_format	Legt das Format der Zeitangabe für <code>from</code> und <code>to</code> fest.	<code>input_format = dd-MM-yyyy</code>
output_format	Legt das Zeitformat der Ausgabe von <code>loop_current_value</code> und <code>loop_next_value</code> fest.	<code>output_format = dd-MM-yyyy</code>
current_value_modifier	Legt den Wert fest, um den die Ausgabe von <code>loop_current_value</code> modifiziert wird.	<code>current_value_modifier = 2w</code>
next_value_modifier	Legt den Wert fest, um den die Ausgabe von <code>loop_next_value</code> modifiziert wird.	<code>next_value_modifier = -1d</code>
create_a_dir_for_each_job	Mögliche Werte sind <code>true</code> und <code>false</code> . Bei <code>true</code> , wird ein neues Verzeichnis für jeden Job angelegt (empfohlen)	<code>create_a_dir_for_each_job = true</code>
job_name_prefix	Setzt den Namen, mit dem die Jobs beginnen.	<code>job_name_prefix = SE_Spain</code>
dagman_template	Setzt die Template-Datei, aus der die Dagman-Workflow-Datei für Condor generiert wird.	<code>dagman_template = template.dag</code>
base_dir	Setzt das Basisverzeichnis, in dem Jobs und Job-Verzeichnisse angelegt werden.	<code>base_dir = /home/condoruser/strahlungsprozessor</code>
pre_executable	Setzt die ausführbare Datei für den den Berechnungsjobs vorgelagerten initialen Job.	<code>pre_executable = /usr/bin/perl</code>
pre_arguments	Setzt die Parameter für die <code>pre_executable</code> .	<code>pre_arguments = copy_files.pl</code>
pre_template	Setzt die Template-Datei, aus der der initiale Job generiert wird.	<code>pre_template = template.cmd</code>
main_executable	Setzt die ausführbare Datei für die Berechnungsjobs, von denen mehrere ausgeführt werden.	<code>main_executable = /data/condoruser/strahlungsprozessor/cloudindex2Gmap_v6.14</code>
main_arguments	Setzt die Argumente, mit denen die <code>main_executable</code> aufgerufen wird.	<code>main_arguments = -sy 2005 -sd \$ {loop_current_value} -ey 2005 -ed \$ {loop_next_value-1} -o \${job.dir}/\$ {job_name_prefix}</code>
main_template	Setzt die Template-Datei, aus der die Berechnungsjobdateien generiert werden.	<code>main_template = template.cmd</code>

post_executable	Setzt die ausführbare Datei für den den Berechnungsjob nachgelagerten finalen Job.	post_executable = /usr/lib64/jvm/java-1_5_0-ibm-1.5.0_sr7/jre/bin/java
post_arguments	Setzt die Parameter für die post_executable.	post_arguments = -cp .:java-net-commons-1.3.11.jar:groovy-all-1.5.6.jar spain_copy \${base_dir} \${job_name_prefix} 12
post_template	Setzt die Template-Datei, aus der der finale Job generiert wird.	post_template = template_post.cmd

Neben den Variablen mit besonderer Bedeutung können auch beliebige andere Variablen in der Konfigurationsdatei gesetzt und in den Template-Dateien verwendet werden. Auf diese Weise können sehr flexible Konfigurationen geschaffen werden.

Das Java-basierte Programm erstellt auf Basis der Konfigurationsdatei und der dort angegebenen Template-Dateien die Job-Dateien und die Dagman-Datei für Condor. Die Dagman-Datei wird dann als Job an Condor geschickt.

3.4 Wolkenindex Berechnung

Die Parallelisierung der Wolkenindexberechnung konnte mit einem anderen Verfahren erreicht werden. Die bisherige Berechnung ist bereits stark granular, d.h. das Programm wird bereits monatsweise und für jeden Slot einzeln aufgerufen (siehe Beschreibung Kapitel 2). Dieses Skript musste in der Form umgeschrieben werden, dass es statt des sequenziellen Aufrufs des Prozessors entsprechende Condor-Jobs erstellt, die dann von Condor mit allen zur Verfügung stehenden Prozessoren abgearbeitet werden.

4 Erfahrungen & Ausblick

Dagman erwartet, dass eine ausführbare Datei mit einem Rückgabewert von 0 beendet wird, wenn kein Fehler aufgetreten ist. Bei anderen Rückgabewerten wird angenommen, dass ein Fehler aufgetreten ist. Der Job wird dann mit dem entsprechenden Rückgabewert beendet und gilt als fehlerhaft. Das führt dazu, dass von dem fehlerhaften Job abhängige Jobs nicht ausgeführt werden. Die ausführbare Datei des Strahlungsprozessors gab zunächst standardmäßig nicht 0, sondern 11 als Rückgabewert zurück. Das führte dazu, dass die Jobs mit dem Wert 11 beendet wurden. Daher ging Dagman davon aus, dass die Berechnungen nicht korrekt beendet wurden und ein Fehler aufgetreten war. Dadurch wurde der finale Job, der die Ergebnisse einsammeln sollte, nicht ausgeführt. Um das Problem zu lösen, war es notwendig, den C-Code des Strahlungsprozessors zu modifizieren und neu zu kompilieren.

Die ausführlichen Log-Dateien von Condor waren sehr hilfreich, um Fehler bei der Job-Ausführung zu finden. Der Fehler mit dem fehlerhaften Rückgabewert von 11 konnte mit Hilfe der Log-Dateien z.B. in kurzer Zeit identifiziert werden.

Die Mächtigkeit der Template-Funktionalität von Groovy zusammen mit der einfach strukturierten Konfigurationsdatei, ermöglichten es, eine Lösung zu implementieren, die auch von programmieretechnisch unerfahrenen Nutzen leicht angepasst werden kann.

Während eine Strahlungsprozessorberechnung von einem Jahr für ein bestimmtes Beispielszenario ohne Parallelisierung ca. 2,5 Tage benötigt, läuft ein Job in der gleichen Umgebung nur ca. 4,5 Stunden. Auf diese Weise kann, bei ausreichend vielen Rechenressourcen, die es ermöglichen 12 Jobs parallel zu berechnen, eine Berechnung für ein Jahr in unter 5 Stunden durchgeführt werden. Ein feineres Aufspalten der Berechnung von Strahlungskarten ist zur Zeit nicht möglich, da die Routine immer monatsweise die Berechnung durchführt. Um bei mehr als 12 Rechenressourcen die Laufzeit einer einjährigen Strahlungskartenberechnung weiter zu verringern, müsste die eigentliche Routine im C-Code überarbeitet werden. Eine Möglichkeit um eine feinere Aufspaltung zu ermöglichen, wäre es, Tage als kleinste Zeiteinheit der Strahlungsprozessorberechnung einzuführen.



WISSENT-Verwertungsplan

Stand: 28.09.2007

GEFÖRDERT VOM



1 Marktübersicht

Das Volumen des Solarenergiemarktes betrug 2005 ca. 3 Milliarden € in Deutschland und 2 Milliarden € in der restlichen EU. Bis zum Jahre 2010 wird ein anhaltendes deutliches Wachstum des weltweiten Solarenergiemarktes prognostiziert, das wie bisher vor allem auf Fördergeldern und vergünstigten Tarifen basiert. In der Übergangsphase des Solarenergiemarktes sollte sich ein Wandel vom geförderten Markt hin zu einem kapitalgetriebenen Markt vollziehen. Um diese Entwicklung zu unterstützen, ist Investitionssicherheit erforderlich, die vor allem durch Dienstleistungen in den Phasen Investitionsentscheidung und Anlagenmanagement unterstützt wird. In beiden Phasen bietet die räumliche Verteilung von Fernerkundungsdaten einen Vorteil gegenüber Bodenmessdaten. Neben den Marktsegmenten, die mit den beiden genannten Phasen zusammenhängen, ist die Last- und Erzeugungsvorhersage für Betriebsmittel ein weiteres vielversprechendes Marktsegment. Aufgrund liberalisierter Märkte und einer hohen Durchdringung erneuerbarer Energien, ist eine präzise Leistungsvorhersage erforderlich, um gute Preise an den Energiemärkten zu erzielen. Für dieses Marktsegment sind Temperatur und Solarstrahlung unverzichtbare Kenngrößen bei der Vorhersage der Energieerzeugung. Die für derartige Vorhersagen benötigten Zeitreihen können auf Grundlage von Fernerkundungsdaten bereitgestellt werden, ohne die ortsbezogenen Beschränkungen von Bodenmessdaten in Kauf nehmen zu müssen.

Die Marktentwicklung im Solarenergie-Sektor ist deutlich rasanter als es die anfänglichen Erwartungen vermuten ließen. Auf der anderen Seite wurden die Erwartungen im Betriebsmittel-Sektor nicht erfüllt.

Die WISENT-Projektpartner sind mit ihren angebotenen Produkten und Diensten in vier Marktsegmenten aktiv:

1. Investitionsentscheidungen (für solarthermische Kraftwerke; große und kleine Photovoltaik-Anlagen)
2. Anlagen-Management (große und kleine Photovoltaik-Anlagen)
3. Betriebsmittel: Lastvorhersage, Leistungsvorhersage von Solaranlagen
4. Wissenschaft und Beratung: Zeitreihen meteorologischer Parameter

Der Zielmarkt kann anhand der Art des Auftraggebers klassifiziert werden, wobei die folgenden Klassen unterschieden werden:

- Gewerbliche Investoren, Versicherungen, Banken, Projektentwickler und größere Anlagenbetreiber. Dieser Marktsektor wird im Folgenden als Großanlagen-Solarmarkt bezeichnet.
- Private Investoren, kleine und mittlere Anlagenbetreiber (Kleinanlagen-Solarmarkt).
- Dienstleistungen im Bereich der Energieversorgung (Netzmanagement, Vorhersage der Erzeugung, Lastvorhersage u.a.)

2 Anforderungen der Community Energiemeteorologie

Für das Treffen von Investitionsentscheidungen werden zwei Arten von Dienstleistungen benötigt:

1. Bereitstellung räumlich aufgelöster Daten, die den Investor unterstützen, einen geeigneten Standort auszuwählen

2. Standortprüfungen, welche Investoren dabei unterstützen, eine detaillierte Standortbewertung vorzunehmen

Für die Optimierung des finanziellen Ertrags ist die Bewertung von Standorten für Investoren äußerst bedeutend. Die Bewertung wird dabei durch auf GIS (Geographical Information System) basierenden Dienstleistungen ermöglicht. Die Kosten für ein Solaranlagenprojekt hängen z.B. von der Landabdeckung, der Entfernung zur Infrastruktur (Transportwege, Straßen, Wasserversorgung etc.) ab. Der finanzielle Ertrag ist zudem vom verfügbaren „Brennstoff“, der solaren Strahlungsdichte, abhängig. Auf diese Weise ist es möglich, die besten Standorte für Solaranlagen zu finden, die eine hohe Strahlungsdichte aufweisen und geringe Investitionskosten verursachen. Bei wachsenden Anlagengrößen und höheren Investitionen wird die Standortoptimierung für Investoren immer wichtiger. Standortgutachten werden für den Entwurf der Anlagen und für die genaue finanzielle Vorausschau des Anlagenbetriebs genutzt. Die Genauigkeit der Informationen sollte deshalb hoch sein, weshalb aufgrund der jährlichen Schwankungen Langzeit-Zeitreihen erforderlich sind. Eine Zeitreihe über 10 Jahre ermöglicht z.B. die Reduzierung von Ungenauigkeiten des Langzeit-Durchschnitts auf $\pm 5\%$. Des Weiteren ist zu berücksichtigen, dass die Finanzierung von Solaranlagen zu 50 bis 75% auf Krediten beruht. Deshalb verlangen Banken und Versicherungen entsprechende Gutachten, bevor sie den Kredit für einen Investor bewilligen. Für große Anlagen werden sogar zwei Gutachten gefordert: ein oder zwei Gutachten werden vom Investor geliefert und eine von der Bank selbst, um die grundlegenden Zahlen der Investitionsentscheidung prüfen zu können. Auch für kleine Anlagen ist eine Abschätzung des erwarteten Ertrags erforderlich, um das Risiko des privaten Investors zu minimieren. Auf diese Weise können Fehlentscheidungen vermieden werden und Angebote unterschiedlicher Anbieter miteinander verglichen werden. Da ein Investor in dieser Stufe des Verkaufsprozesses in der Regel nichts für die Ertragsschätzung bezahlt, sollte die Ertragsschätzung sehr preisgünstig sein. Aufgrund einer hohen Anzahl von Angeboten ist ein schneller und einfacher Zugriff auf diese Art von Ertragsschätzungen wünschenswert.

Dienstleistungen für Investitionsentscheidungen werden ebenso benötigt, um den Export von Solaranlagentechnologie zu sichern. Beispiele dafür sind die Export-Initiative des Bundesumweltministeriums für Solarthermische Stromerzeugung. Die Technologie braucht Standorte mit hoher direkter Einstrahlung, welche in Deutschland nicht verfügbar sind. Das Ministerium unterstützt deshalb auch Studien und Einstrahlungs-Kartierungen außerhalb Deutschlands, um die Exportchancen der deutschen Industrie zu fördern, die viele Komponenten für Anlagen nach Südeuropa und Nordafrika liefert. Für die Sicherstellung des Ertrags bereits installierter Solaranlagen ist in jedem Falle eine Überwachung erforderlich. Die Größe der Systeme bestimmt dabei den Umfang der Dienstleistungen: Kleine Solaranlagen benötigen eine kostengünstige Überwachung (Leistungsdiagnose), während große Anlagen eine detaillierte Überwachung mit automatischen Fehlererkennungsroutinen erfordern. In beiden Fällen wird eine Reduktion der Ausfallzeiten erreicht.

Die Nachfrage nach Zeitreihen kommt von wissenschaftlichen und/oder beratend tätigen Einrichtungen wie Universitäten und Forschungseinrichtungen, Beratern in Systemtechnik, Beratern in Finanzentwicklungen, lokalen Behörden und Umweltbehörden. Von der weltweiten Nachfrage stammt 80% aus Europa. Zeitreihen werden genutzt, um den Ertrag von Solaranlagen zu berechnen. Des Weiteren können sie genutzt werden, um im Architekturbereich Beleuchtungs- und Wärmebedarf in Gebäuden abzuschätzen. Zeitreihen sind nicht anwendungsspezifische Daten, die i. d. R. für energietechnische Anwendungen entsprechend weiter verarbeitet werden („value adding“). Im Allgemeinen sind Planer, Architekten und Wissenschaftler die Hauptkunden in diesem Sektor.

3 Grid-Infrastruktur als Grundlage für neue Dienste

Die im Rahmen des WISENT-Projektes realisierte Grid-Infrastruktur ist in Abbildung 1 dargestellt. Alle an WISENT beteiligten Projektpartner (meteocontrol, Universität Oldenburg, OFFIS und DLR) sind bereits in diese Infrastruktur integriert und konnten dementsprechend Erfahrungen mit Grid-Technologien sammeln. Zudem erfolgte bereits eine Integration in die D-Grid-Infrastruktur. Die wertvollen Erfahrungen des Einsatzes von Grid-Technologie wurden bzw. werden in dem Community-spezifischen Wiki-System TikiWiki dokumentiert (siehe Abbildung 2), welches eine Art zentrale Wissensbasis der Community darstellt. Die Nachhaltigkeit des TikiWiki-Systems ist inhaltlich durch die Fortführung bzw. Erweiterung der Arbeiten im virtuellen Institut für Energiemeteorologie

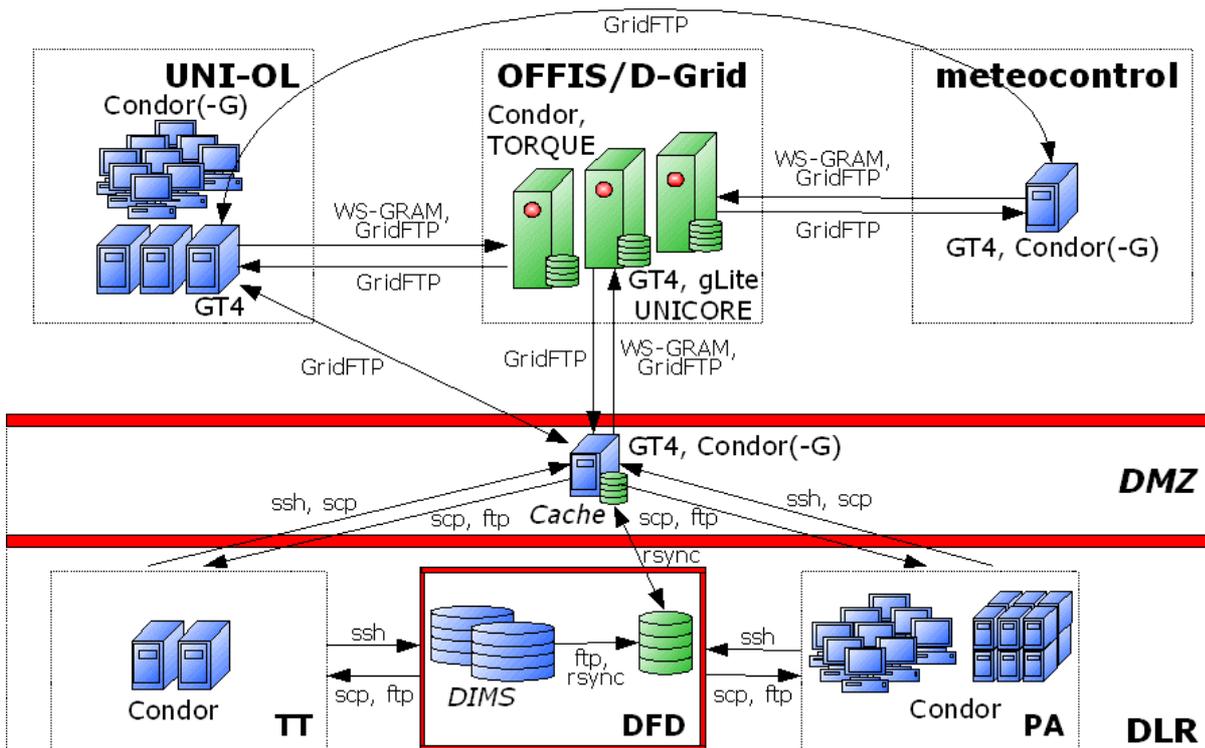


Abbildung 1: WISENT-Grid-Infrastruktur

(viEM) gegeben. Die technische Nachhaltigkeit wird erreicht, indem sich die beteiligten Institutionen über die Fortführung des Hostings und die Pflege des Systems einigen. Aufgrund der intensiven Nutzung des TikiWiki-Systems wurde bereits von Seiten der beteiligten Partner ein großes Interesse an der Fortführung bekundet. Damit bleiben auch die im Projektverlauf gesammelten Erfahrungen im Umgang mit Grid-Technologien erhalten. Die Community-spezifischen Erweiterungen des TikiWiki-Systems sind bereits teilweise in das offizielle Release des Open-Source-Projekts zurückgefließen und stehen damit auch anderen Projekten zur Verfügung.

Über die interne Verwendung der Erfahrungen hinaus wurden die im Projektverlauf gewonnenen Erkenntnisse im Umgang mit Grid-Technologien auch in Zeitschriften, auf Workshops und auf Konferenzen veröffentlicht.

4 Perspektive DLR-DFD (Deutsches Fernerkundungsdatenzentrum)

Der Zugang zu Strahlungsmessungen am Boden ist generell sehr schwierig, da diese Informationen jeweils vom nationalen Wetterdienst jeden Landes angefordert bzw. gekauft werden müssen. Die räumliche Entfernung und Sprachbarrieren hemmen diesen Austausch erheblich. Leider sind die meteorologischen Datentransferwege der WMO (World Meteorological Organisation) für die Allgemeinheit oder interessierte Firmen nicht zugänglich.

In diesem Umfeld bieten satellitenbasierte Informationen erhebliche Vorteile, da sie bei einem einzigen nationalen Partner für große geographische Regionen erhältlich sind. Der Einsatz weniger Satellitensensoren mit einem einheitlichen Messstandard vereinfacht die Nutzung und erhöht die Repräsentativität der Information. Daher ist der einfache Datenzugang zu satellitenbasierten Strahlungsmessungen und der notwendigen Zusatzdaten aus der Atmosphärenforschung für die Community der Energiemeteorologie sehr wichtig.

Das Deutsche Fernerkundungsdatenzentrum (DFD) forscht auf dem Gebiet der Fernerkundung und entwickelt wissenschaftliche Auswertemethoden und

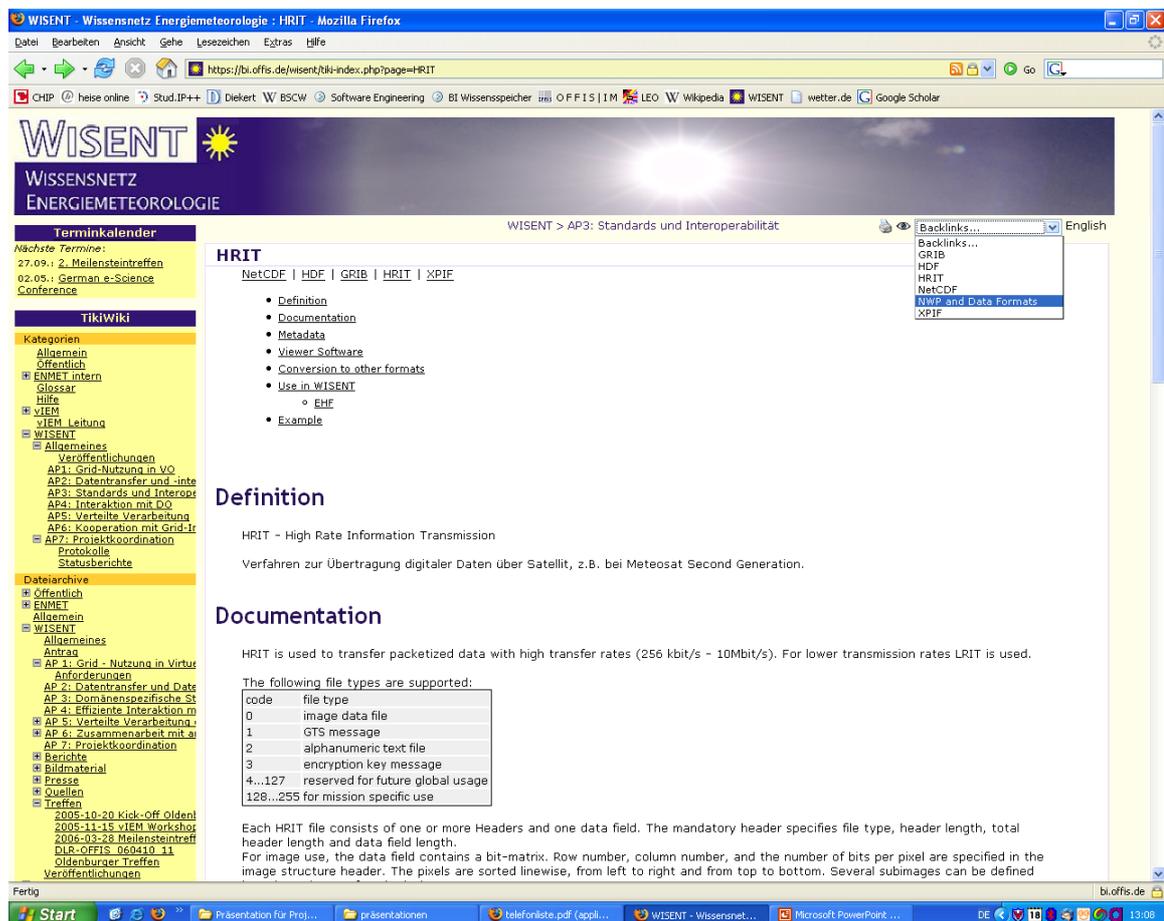


Abbildung 2: TikiWiki-System der Community Energiemeteorologie

anwendungsorientierte Lösungen zu Umwelt und Klima, Mobilität und Planung, zivile Sicherheit sowie zur Prävention und Management bei Naturkatastrophen. Die Energiemeteorologie ist eine der Nutzergruppen des DFD. Kern seiner Mission ist der einfache Datenzugang und die Verteilung von Informationsprodukten in der

Nutzergemeinschaft z.B. über das Weltdatenzentrum für Fernerkundung der Atmosphäre (WDC-RSAT) oder das EOWEB (User interface Earth Observation on the Web).

Angesichts der neuen Satellitengenerationen (z.B. Meteosat Second Generation, ENVISAT und METOP) nehmen jedoch die benötigten Datenmengen erheblich zu. So liefert z.B. die neue Meteosat Second Generation (MSG) 1 TB Rohdaten pro Monat. Die ausgewerteten Produkte erreichen eine ähnliche Größenordnung.

Sowohl für das WDC als auch das EOWEB gelten derzeit folgende Einschränkungen: Es handelt sich um statische Datenbestände, d.h. nur was hinterlegt ist kann auch heruntergeladen werden und die Nutzeroberflächen sind auf einzelne Satellitenüberflüge oder fest spezifizierte Komposite optimiert.

Für die neuen Satellitendaten und auch dreidimensionale Modelldaten ist es jedoch notwendig, beliebige raum-zeitliche Ausschnitte von 3D-Daten (x,y,t) oder 4D-Daten (x,y,z,t) zu extrahieren. Im Rahmen von WISENT wurden wolkenphysikalische Parameter und atmosphärischer Wasserdampf in Form eines dreidimensionalen Datensatzes (x,y,t) als Testfall ausgewählt. Die erarbeiteten Methoden der Datenextraktion sollen in Zukunft für weitere Produkte wie z.B. die HELIOSAT-4-Strahlungsprodukte, die SOLEMI-Strahlungsdatenbank und Zusatzinformationen wie z.B. die Schneebedeckung, 2m-Temperaturen und Landoberflächen-Parameter eingesetzt werden. Ebenfalls können sonstige WDC-Nutzer von solchen Datenextraktionsmethoden für Spurengase (4D-Datensatz) in der Atmosphärenchemie und Luftqualitätsforschung profitieren.

DLR-intern wurde als Reaktion auf die gestiegenen Anforderungen bereits eine Umgestaltung des METEOSAT-Archivbestands in eine so genannte Cube-Struktur vorgenommen. Als erstes Produkt werden METEOSAT-Rohdaten auf diese Weise abgespeichert und extrahiert. Es bietet sich nun an, einen ähnlichen Ansatz für andere 3D- und 4D-Datensätze zu realisieren und diesen v.a. auch für externe Nutzer bereitzustellen.

Dazu werden z.B. Webservices benötigt, die den Nutzer im Internet mit dem Infrastruktur- und auch dem Produktionsnetz des DFD durch zwei verschiedene Firewall-Systeme hinweg verbindet. Ein Beispielfall unter Nutzung der neugeschaffenen Demilitarized Zone (DMZ) des DLR wird derzeit in WISENT aufgebaut.

Das DLR verspricht sich durch diese Projektergebnisse eine erhebliche Reduktion der Kosten der Datenbereitstellung. Derzeit benötigt die interaktive Bereitstellung pro Nutzeranfrage einen Aufwand von einigen Arbeitsstunden bis hin zu -tagen. Eine automatische Lösung verschafft hier Vorteile und wird letztendlich erlauben, mehr Produkte und Datensätze als bisher verfügbar zu machen.

Da das Weltdatenzentrum für Fernerkundung der Atmosphäre eine langfristige Perspektive innerhalb des DLR hat, ist ein nachhaltiger Einsatz der WISENT-Projektergebnisse gewährleistet, wenn sich die Erwartungen an einen vereinfachten Datenzugang bzw. eine kostengünstigere Datenbereitstellung erfüllen.

Neben der Datenextraktion wird im DLR v.a. der Einsatz von Wiki-Systemen und Web-Konferenz-Werkzeugen nach Projektende von WISENT nachhaltig erhalten bleiben. Aufgrund der positiven WISENT-Erfahrungen werden auf Initiative des DLR derzeit Wiki-Systeme in der Energy Community of Practise der Global Earth Observation System of Systems (GEOSS) Initiative, dem EU-Projekt MESOR und der Task 35 innerhalb des SolarPaces-Programms der Internationalen Energieagentur genutzt. Damit wird die Zusammenarbeit in solchen internationalen Gremien deutlich vereinfacht.

5 Perspektive DLR-IPA (Institut für Physik der Atmosphäre)

In der Abteilung für Fernerkundung der Atmosphäre bei DLR-IPA werden unterschiedliche Aspekte der Wolken-Strahlungs-Wechselwirkung untersucht. Ein wichtiger Aspekt in der Energiemeteorologie und innerhalb des vIEM-Konsortium im Speziellen ist die Entwicklung, Bewertung und Validierung von Verfahren zur Bestimmung von Solarstrahlung am Erdboden aus Satellitenbeobachtungen. Wolken weisen die höchste zeitliche und räumliche Variabilität aller Einflussfaktoren auf die Solarstrahlung am Boden auf.

Detaillierte Strahlungstransfersimulationen mithilfe des Softwarepakets libRadtran (library for radiative transfer) bilden eine wichtige Basis der Verfahrensentwicklung. Beispielsweise werden Wolken- und Atmosphärendaten aus Wettervorhersagemodellen und Zusatzdaten zur Beschreibung des Bodenreflexionsvermögens als Input für die Simulation von Satellitendaten und entsprechend einfallender Solarstrahlung am Boden genutzt. Auf diese Weise entsteht eine Testumgebung, in der die Solarstrahlungsalgorithmen auf die simulierten Satellitendaten angewandt werden. Im Vergleich mit den ebenfalls berechneten Solarstrahlungsdaten wird die Bestimmung der Genauigkeit bestehender Retrievalverfahren, deren Weiterentwicklung oder die Neuentwicklung von Verfahren ermöglicht.

Allerdings sind Simulationen dieser Art sehr rechenzeitintensiv. Zur Demonstrationsaufgabe innerhalb des WISENT-Projekts wurde daher die Einbindung bestehender Rechenkapazitäten bei DLR-IPA und der zur Strahlungssimulation verwendeten Algorithmen in eine Grid-Lösung. Bisher wurden solche Simulationen auf einzelnen Linux/UNIX Arbeitsplatz-PCs oder auf einem 32 Prozessor-Cluster durchgeführt. Außerhalb täglicher Kernarbeitszeiten wenig genutzte Arbeitsplatzrechner wurden in manueller Steuerung für verteiltes Rechnen genutzt. Eine automatische technische Lösung zur Vernetzung der einzelnen Knoten des Clusters von 30 Arbeitsplatz-PCs mit Hilfe der Condor-Software zur Verwaltung verteilter Prozessierungs-Aufgaben wurde im Rahmen von WISENT aufgebaut und erprobt.

Das Condor-Paket wurde bei DLR-IPA erfolgreich installiert und nach kleineren Anpassungen der verwendeten Strahlungstransport-Software auch erfolgreich getestet. Dabei wurden im Betrieb in einem sehr heterogenen Arbeitsplatz-PC-Netzwerk zwar Schwächen erkannt, trotzdem wird auf der Basis der im Rahmen von WISENT gemachten Erfahrungen und vorgenommenen Software-Installation ein Ausbau der automatischen Vernetzungsmöglichkeiten angestrebt. Im Betrieb des DLR-PA-Cluster erwies sich die Condor-Software zum Management verteilter Prozessierung als sehr geeignet und hat dort daher die bisher verwendete Software bereits abgelöst.

Aufgrund der erfolgreichen Erfahrungen mit Condor ist DLR-IPA für die Zukunft sehr daran interessiert, Aufgabenstellungen wie rechenintensive Strahlungstransportrechnungen auch auf D-Grid-Rechenressourcen durchführen zu können. Einerseits könnten damit herkömmliche Aufgaben weit schneller bearbeitet werden, andererseits werden auch besonders rechenintensive höchst-realistische drei-dimensionale Strahlungstransportsimulationen im großen Stil möglich, die bisher aufgrund ihrer Rechenintensivität nur für sehr eingeschränkte Fälle möglich waren.

6 Perspektive DLR-TT (Institut für Technische Thermodynamik)

Entscheidungen in der Energiewirtschaft, Energiepolitik und Energieforschung haben stets weit reichende und lang wirkende Folgen. Durch vorausschauendes aktives Handeln können Chancen neuer Technologien rechtzeitig erkannt und mögliche negative

Auswirkungen unseres heutigen Handelns auf Umwelt und Gesellschaft minimiert werden. Voraussetzung für ein solches Handeln ist systemanalytisches Wissen, welches Entscheidungsträgern erlaubt, eine Balance zwischen den heutigen technischen und ökonomischen Möglichkeiten und den zu erwartenden langfristigen Entwicklungsperspektiven zu finden.

DLR-TT-STB (Systemanalyse und Technikbewertung) entwickelt und nutzt Methoden und Werkzeuge, die auf die spezifischen Fragestellungen der Systemanalyse und Technikbewertung im Bereich der Energieversorgung zugeschnitten sind. Es wird außerdem Wissen bereitgestellt, das Entscheidungsträgern aus Wissenschaft, Wirtschaft und Politik als Orientierung zur Entwicklung langfristiger Forschungsschwerpunkte und zur Gestaltung von energie-, umwelt- und forschungspolitischer Rahmenbedingungen dient.

Die Arbeiten von TT-STB kombinieren energiewirtschaftliche Top-down-Analysen des gesamten Energiesystems auf regionaler, nationaler und europäischer Ebene mit technologieorientierten Bottom-up-Studien in relevanten Teilbereichen wie z.B. Lebenszyklusbilanzen, Potenzial- und Wirtschaftlichkeitsanalysen. Konzeptionelle Arbeiten zur Nachhaltigkeitsproblematik, zum technischen Lernen oder zur Markteinführung neuer Technologien ergänzen das Arbeitsgebiet.

Die Tätigkeiten lassen sich in vier Schlüsselbereiche einordnen:

1. Entwicklung von Konzepten einer nachhaltigen Energieversorgung auf der Basis von Szenarioanalysen
2. Technisch-ökonomisch-ökologische Einsatzanalysen und forschungsbegleitende Systemanalysen für einzelne Technologielinien
3. Ermittlung zweckmäßiger Markteinführungsstrategien für neue Technologien und Strukturen im Energiebereich
4. Methodische Weiterentwicklung systemanalytischer Instrumente insbesondere im Hinblick auf die Eigenschaften erneuerbarer Energien und dezentraler Versorgungsstrukturen

In den Bereichen 3 und 4 werden bei DLR-TT eine Vielzahl geographischer und meteorologischer Daten verarbeitet. Die Menge der Daten ist in den letzten Jahren deutlich gewachsen, so dass eine manuelle Verwaltung der Daten ineffizient wird. Die im Rahmen von WISENT entwickelten Werkzeuge werden die Nutzung der bestehenden Daten- und Informationsbestände leichter nutzbar machen und dauerhafter Bestandteil der in der Arbeitsgruppe entwickelten und genutzten Werkzeuge werden.

7 Perspektive meteocontrol

Für meteocontrol betrachten wir zunächst die Einführung neuer Produkte und die Erweiterung bestehender Produkte, bevor Aspekte der Verbesserung der meteocontrol-Infrastruktur zusammengefasst werden.

7.1 Einführung neuer Produkte

Als Weltmarktführer bei der Fernüberwachung von Solaranlagen ist es für meteocontrol wichtig, durch die Einführung von neuen Produkten die Marktposition zu festigen. Durch Tätigkeiten im Zuge von WISENT war es möglich, neue Produkte und Produktideen zu generieren.

- Solarstromprognose: meteocontrol berechnet mit Hilfe der Solaranlagen- und Daten sowie prognostizierter Globalstrahlung eine Solarstromprognose für Deutschland 48

Stunden in die Zukunft. Die prognostizierten Werte geben die mittlere eingespeiste Energie relativ zur installierten Leistung in kWh/kWp in stündlicher Auflösung wieder, bezogen auf alle von meteocontrol überwachten Anlagen. Zudem wird der prozentuale Anteil der einzelnen Bundesländer berechnet. Auf besonderen Kundenwunsch können auch einzelne Postleitzahlengebiete berechnet werden. Durch die graphische Aufbereitung der Solarstromprognose im meteocontrol-Wetterportal kann auf einen Blick der zu erwartende Ertrag, also die eingespeiste Energie des aktuellen Tages, sowohl mit dem gestrigen als auch dem morgigen Wert verglichen werden.

- Solarertragsrechner: Durch die Tätigkeiten innerhalb von WISENT zur parallelen Berechnung der Solarsimulation konnte schon ein zusätzliches Produkt in diesem Geschäftsbereich erstellt werden. Der Solarertragsrechner ist eine als Webservice realisierte Software, die aus einer vorgegebenen Solaranlagenkonfiguration (Wechselrichter, Module, Verschaltung, etc.) und den meteocontrol-Wetterdaten (Einstrahlung, Temperatur) des geplanten Standorts einer Solaranlage die erwartete erzeugte Energie simuliert. Der Solarertragsrechner wurde bereits an einen Kunden verkauft.
- Datentransfer Solar: Die im Zuge von WISENT entstandene Webservice-Schnittstelle ist dazu geeignet, Konfiguration und Messdaten von Solaranlagen direkt zum Kunden zu übertragen. Bis jetzt ist es dem Kunden nur möglich, diese Daten über die Web-Portale von meteocontrol abzurufen. Der direkte Abruf über Webservices ermöglicht den Kunden völlig neue Umgangsmöglichkeiten mit den Daten. Es können eigene Web-Portale unabhängig von meteocontrol entwickelt werden und auch eigenes Data-Mining und Fehlersuche wird möglich. Das Potential in diesem Markt ist sehr groß und es muss noch ein genaues Geschäftsmodell erarbeitet werden, bevor Umsatzschätzungen abgegeben werden können. Konkrete Kundenanfragen zu diesem Thema liegen bereits vor.
- Datentransfer Wetter: meteocontrol verarbeitet Daten von weltweit verteilten Wetterstationen unterschiedlicher Betreiber, um u.a. Leistungssimulationen von Solarstromanlagen für seine Auftraggeber durchzuführen. Die Daten werden über das Wettermessnetz von Meteomedia sowie über das Synop-Messnetz in regelmäßigen zeitlichen Abständen (z. B. stündlich) geliefert, bei meteocontrol dekodiert und in lokale Datenbestände importiert. Durch die in WISENT entstandene Webservice-Schnittstelle ist es nun möglich, sehr zeitnah Kunden diese Messdaten bereitzustellen. Zielgruppe dieses Services sind unter anderem Firmen mit Gebäudeleitsystem zur Steuerung und Regelung von Heizungs- und Lüftungssystemen. Ein erster Auftrag wurde bereits abgewickelt.
- Data-Mining: Aufgrund der ca. 5000 durch meteocontrol überwachten Solaranlagen ist eine große Datenbank mit Solaranlagenkonfigurationen und Messdaten entstanden. Mit Hilfe von Data-Mining und Data-Warehousing können hieraus neue Erkenntnisse über die verwendeten Technologien (Wechselrichter, Solarmodule, etc) erarbeitet werden. Ein erstes Projekt zu diesem Thema wurde in Zusammenarbeit mit der Fachhochschule Ulm gestartet. Mit der innerhalb von WISENT entstandenen Webservice-Schnittstelle ist es möglich, gezielt große Datenmengen aus der meteocontrol-Datenbank ins Data-Mining-System der Fachhochschule Ulm zu übertragen. Hieraus entstehende Erkenntnisse können dann weiter vermarktet werden. Die finanziellen Aspekte dieses Projektes lassen sich noch nicht abschätzen

7.2 Erweiterung bestehender Produkte

Die Tätigkeiten bei WISENT trugen zur Verbesserung bereits bestehender Produkte von meteocontrol bei. Erzielte höhere Qualität und größerer Funktionsumfang trugen zu einer stärkeren Kundenbindung sowie zur Gewinnung von Neukunden bei. Diese Zugewinne für das meteocontrol-System lassen sich nicht monetär ausdrücken. Im Folgenden wird eine Aufstellung der verbesserten Produkte geliefert.

- Solarsimulation: Die bestehende Simulation der Solaranlagen wird innerhalb von WISENT neu programmiert. Hieraus ergibt sich eine höhere Qualität der Simulationsergebnisse und die Möglichkeit, Solaranlagen mit Multistringwechselrichtern zu simulieren. Zusätzlich ist die Simulation performanter, d.h. die Ergebnisse können dem Kunden früher präsentiert werden. Grundsätzlich wird die Qualität des meteocontrol-Web-Portals Safer'Sun erhöht. Die früher gegebene Abhängigkeit von Drittanbietern wird aufgelöst.
- Qualitätskennzahlen: Die eingeführten Qualitätskennzahlen und Metadaten können zur Qualitätssteigerung der Datenprodukte innerhalb von Safer'Sun und der Solargutachten beitragen. Die Auswahl von Wetterstationen für Kunden kann verbessert werden.

7.3 Nachhaltige Verbesserung der internen Infrastruktur

Im Laufe von WISENT konnte bei meteocontrol eine nachhaltige Verbesserung der internen Infrastruktur erzielt werden. Die Erweiterung des internen Systems wurde notwendig, um den Einsatz von Grid-Technologien zu ermöglichen und damit eine Grid-basierte Kommunikation und einen zuverlässigen Datenaustausch mit Partnern und Kunden zu realisieren. Sämtliche Änderungen und Erweiterungen werden nach WISENT erhalten bleiben und in den täglichen Arbeitsprozess integriert sein. Da die neu eingeführten Werkzeuge und Technologien organisch in das meteocontrol-System eingeflochten wurden, wird es auch nach WISENT zu einer Weiterentwicklung in den einzelnen Teilbereichen kommen. Die verbesserte Infrastruktur führt generell zu optimierten internen und externen Arbeitsabläufen, höherer Performance der meteocontrol-Web-Portale und Datenlieferungen, einem sichereren und stabilerem Grundsystem und einem Qualitätsgewinn der Zwischen- und Endprodukte. Im Folgenden wird auf die einzelnen Maßnahmen zur Verbesserung der Infrastruktur eingegangen.

- Dokumentation: Um konkrete Abstimmungen mit den Projektpartnern vornehmen zu können wurde die interne Dokumentation des Programmcodes, der Serverparkstruktur und des Datenbanksystems verbessert. Dies hat zur Folge, dass auch in Zukunft intern wie extern schneller und zuverlässiger Informationen ausgetauscht werden können. Zusätzlich ist die Nachhaltigkeit des Systems gewährleistet, falls es z.B. zu Personalveränderungen kommen sollte.
- Testsystem: Es wurde ein neues völlig eigenständiges Testsystem im Serverpark aufgebaut. Das alte System war noch sehr stark an Teile des Livesystems gekoppelt. Ein sinnvolles Testen der bei WISENT eingesetzten Technologien wäre so nicht möglich gewesen. Das neue Testsystem wird ständig erweitert und verbessert. Auch nach Projektende wird das Testsystem zentraler Bestandteil der Entwicklung bleiben und es wird somit ein sicherer, schneller und qualitativ hochwertiger Entwicklungsprozess auch in Zukunft gewährleistet.
- Virtuelle Maschinen: Um Erweiterungen im Serverpark u.a. für das Testsystem vornehmen zu können wurden virtuelle Maschinen (XEN) installiert. Diese Technologie wurde neu bei meteocontrol eingeführt. Aufgrund des Einsatzes von

virtuellen Maschinen wurden Kosten eingespart, da weniger neue Hardware angeschafft werden musste. Weiterhin ist es jetzt einfacher, Erweiterungen an der Serverparkstruktur vorzunehmen. Die Virtualisierung des meteocontrol-Serverparks wird auch in Zukunft auf diese Weise vorangetrieben werden.

- **Monitoring mit Nagios:** Der komplette Serverpark wurde im Laufe von WISENT an die Überwachungssoftware Nagios gekoppelt. Es werden die Hardware, das Betriebssystem, sowie einzelne Dienste und Datenimporte überwacht. Nagios hat sich als sehr zuverlässiges Überwachungssystem erwiesen. Es ermöglicht schnellere Reaktionszeiten bei Systemausfall, das Verhindern von Systemausfällen und eine schnellere Kundenbenachrichtigung, falls sich Datenlieferungen verzögern. Die benötigte Arbeitszeit für manuelle Überwachung ist stark zurückgegangen, was sich direkt auf die Produktivität der Entwicklung auswirkt. Nagios wird auch in Zukunft zentraler Bestandteil der Überwachung bei meteocontrol sein.
- **Neuer Datenbankserver:** Um die Systemanforderungen innerhalb von WISENT erfüllen zu können wurde ein neuer Datenbankserver angeschafft, der die Serverparkstruktur ergänzt hat. Der Server sorgt für höhere Performance und wird auch weiterhin im System von meteocontrol integriert bleiben.
- **Versionsverwaltung Subversion:** Im Zuge der Einrichtung des neuen Testsystems wurde das Software-System zur Versionsverwaltung Subversion eingeführt. Subversion bietet eine professionelle Möglichkeit, die Verwaltung von Quellcode zu organisieren. Subversion wird auch nach Ablauf von WISENT im Einsatz bleiben.
- **VNC:** Mit Hilfe von VNC ist es möglich, Fernadministration von Servern zu betreiben. VNC ist ein weiterer Baustein, der im Laufe von WISENT nachhaltig in die Wartung des Serverparks integriert wurde.
- **Grid-Technologie:** Es wurde die Hard- und Softwarestruktur geschaffen, um Grid-Technologie wie z.B. Globus Toolkit und GridFTP zu nutzen.

8 Perspektive OFFIS

Als Forschungsinstitut sieht sich OFFIS vor allem als Brücke zwischen Forschung und Anwendung. Entsprechend sind die Tätigkeiten im Projekt und auch die Verwertungsperspektiven an die Idee gekoppelt, die Ergebnisse des Projektes aus der Forschung herauszutragen, insbesondere in die Anwendung in der Wirtschaft. OFFIS ist durch das WISENT-Projekt in der Lage, sein Profil in den Bereichen Grid-Technologien und Wissensvernetzung zu schärfen und so neue Einnahmequellen im Bereich der Auftragsforschung zu erschließen. OFFIS beabsichtigt, das Know-How durch unterstützte Dissertationen und weitergehende Auftragsprojekte zu vertiefen.

Eine weitergehende Verwertung der Projektergebnisse erfolgt über Beiträge bzw. Veröffentlichungen in der wissenschaftlichen Community und Fachwelt. Dies sind u. a. Ergebnisse, die sich aus der wissenschaftlichen Begleitung des Gesamtvorhabens ergeben und Erfahrungen beinhalten, die im Rahmen des Projektes gewonnen werden und allen Unternehmen, die den Einsatz dieser Methoden und Techniken zum Ziel haben, insbesondere den Projektpartnern, zur Verfügung gestellt werden.

OFFIS ist als Technologietransfer-Institut immer sehr daran interessiert, dass die Ergebnisse aus Kooperationsprojekten mit Industriepartnern nachhaltig tragfähig sind. Daher ist es im OFFIS üblich, dass auch nach Abschluss von Projektförderungen mit den Industriepartnern der Kontakt gepflegt wird, um so eine Unterstützung bei der nachhaltigen Umsetzung des Technologietransfers zu bieten. In WISENT wird OFFIS auch über die

Laufzeit der Förderung durch das BMBF hinaus mit den Projektpartnern in Kontakt bleiben, um so eine direkte Unterstützung bei der nachhaltigen Umsetzung des Projektergebnisses zu bieten.

Folgende Aktivitäten tragen dazu bei, dass die Nachhaltigkeit der Projektergebnisse gesichert ist:

- Durch das Projekt wird das vorhandene Grid-Know-How im OFFIS gestärkt. Bereits im Projektverlauf wird die interne Infrastrukturabteilung Teile des Betriebs des Clusters aus den BMBF-Sonderinvestitionen und der Grid-Middleware übernehmen, so dass auch über das Projektende hinaus das Wissen in der Organisation OFFIS verbleibt.
- OFFIS wird nach Projektende die Ergebnisse verfügbar machen. So wird weiteren potentiellen Partnern der Weg in die D-Grid-Community Energiemeteorologie eröffnet. Jeder Partner verfügt allein über exzellente Firmenkontakte, die sich von den Ansprechpartnern und der regionalen Aufteilung gut ergänzen. So können über Auftragsforschungs- und Kooperationsprojekte weitere Unternehmen von den Ergebnissen profitieren.
- Im Projektverlauf, insbesondere aber nach Vorliegen der Ergebnisse, werden aktiv Partner gesucht, für die das Thema Grid-Providing bzw. -Computing relevant ist. Hierzu gehören vor allem Unternehmen der Energiebranche (Windprognosen, Lastmanagement).

9 Perspektive Universität Oldenburg

Die Arbeitsgruppe Energiemeteorologie an der Universität Oldenburg (AG EnMet) betreibt seit langem sowohl Forschung zur Solarstrahlungsberechnung aus Satellitendaten als auch operationelle Dienste für kommerzielle Partner, insbes meteocontrol GmbH, bei denen entweder tägliche Datenprodukte bereitgestellt werden oder auf Anfrage Auswertungen aufgrund einer langjährigen Forschungsdatenbasis erstellt werden. Die durch die Teilnahme am WISENT-Projekt entstandenen nachhaltigen Fortschritte in beiden Bereichen werden im Folgenden dargestellt und anschließend auch eine durch WISENT erst ermöglichte neue Anwendung im Bereich numerischer Modellierung vorgestellt.

9.1 Methodenentwicklung zur Solarstrahlungsberechnung

9.1.1 IST-Zustand vor dem WISENT-Projekt

Im EU-Projekt HELIOSAT-3 wurde unter maßgeblicher Beteiligung der AG EnMet die sogenannte Heliosat-Methode zur Solarstrahlungsberechnung verbessert und ist immer noch Gegenstand intensiver Arbeiten zur Methodenentwicklung. Dabei muss man sich wegen der notwendigen Rechenzeiten auf kleinere Ausschnitte der Satellitendaten (bspw. Deutschland) beschränken, was die Qualität der statistischen Berechnungen einschränkt.

9.1.2 Neue Anforderungen

Eine Ausweitung der Datenbasis für Tests im Rahmen der Methodenentwicklung würde die Statistik wesentlich verbessern:

- geografisch: Übergang von Deutschland-Ausschnitten der hochaufgelösten Satellitenbilder auf Europa (und die "Welt")
- zeitlich: längere Zeiträume (mindestens 10 Jahre)

- Einbeziehen von vielen Standorten für die Berechnung von Zeitreihen

Außerdem werden ausführliche Sensitivitätsstudien ermöglicht, bei denen relevante Parameter graduell verändert werden.

9.1.3 Fortschritt durch WISENT

- Durch die parallele Nutzung der verteilten Rechen- und Speicherressourcen im D-Grid und die benutzerfreundliche Bedienung mit Hilfe der Condor-Scheduler-Installationen an der Universität Oldenburg und am OFFIS-Rechen-Cluster lässt sich die Rekalkulation der Datenbasis mit veränderten Methoden in vertretbarer Zeit und damit auch in kürzeren Abständen verwirklichen, so dass die Qualität der der weiteren Forschung zugrunde liegenden Daten erhöht wird.
- Die beteiligten Wissenschaftler erzielen in der gleichen Arbeitszeit also wesentlich bessere Ergebnisse bei der Methodenentwicklung, was wiederum die Ausgangslage für die Akquisition zusätzlicher Mittel sowohl aus der öffentlichen Hand als auch aus der Wirtschaft verbessern wird.
- Außerdem erübrigt sich z. T. die Anschaffung (und Betreuung) teurer eigener Compute- und Speicherressourcen.

9.2 Solarstrahlungsberechnung für zurückliegende Zeiträume

9.2.1 IST-Zustand vor dem WISENT-Projekt

Sowohl für die Forschung in der AG EnMet als auch im Rahmen eines Auftrages von meteocontrol GmbH wurde in langwieriger Arbeit aus den Rohdaten verschiedener Satelliten der Meteosat-Reihe mit Hilfe der Algorithmen der Heliosat-Methode eine langjährige Datenbasis von Bewölkungskarten aufgebaut, die eine schnelle Erzeugung von langjährigen Strahlungszeitreihen für beliebige Standorte im verwendeten Europa-Ausschnitt erlaubt.

9.2.2 Neue Anforderungen

- Durch die aktive Forschung in der Methodenentwicklung gibt es Verbesserungen in den Algorithmen, deren Anwendung auf historische Rohdaten die Qualität der Ergebnisse verbessern würde. Die Rekalkulation hat jedoch einen hohen Bedarf an Rechenzeit (und damit auch Wartezeit auf die Ergebnisse) sowie temporärem Speicherplatz, der durch die aktuelle lokale IT-Infrastruktur nicht abgedeckt werden kann.
- Auch für andere räumliche Ausschnitte der Satellitenbeobachtung sind solche langjährigen Datenbasen erforderlich.

9.2.3 Fortschritt durch WISENT

- Durch die parallele Nutzung der verteilten Rechen- und Speicherressourcen im D-Grid und die benutzerfreundliche Bedienung mit Hilfe der Condor-Scheduler-Installationen an der Universität Oldenburg und am OFFIS-Rechen-Cluster lässt sich die Rekalkulation der Datenbasis mit verbesserten Methoden in vertretbarer Zeit und damit auch in kürzeren Abständen verwirklichen.
- Dadurch stehen Industriepartnern wie meteocontrol eine bessere Datenbasis für Standortgutachten von Photovoltaik-Anlagen zur Verfügung, was die Marktposition dieses Partners stärkt und die Kooperation mit der Universität Oldenburg verstetigen hilft.

- Aktuell (Herbst 2007) ist die AG EnMet in der Lage, durch die Nutzung der D-Grid-Ressourcen weitere langjährige Datenbasen für andere Gebiete in kurzer Zeit aufzubauen und so den Bedarf von Industriepartnern (z. B. Kundenanfragen an meteocontrol) zu befriedigen.

9.3 Near-Real-Time-Dienste zur Solarstrahlungsberechnung

9.3.1 IST-Zustand vor dem WISENT-Projekt

Schon vor und unabhängig von WISENT bestand eine Kooperation mit meteocontrol zur täglichen Berechnung und Bereitstellung von Globalstrahlungsdaten als tabellenbasierten Dateien (CSV) für ca. 7000 Standorte und alle deutschen PLZ-Gebiete.

9.3.2 Neue Anforderungen

- Bei meteocontrol entstand der Bedarf nach umfassenderen Qualitätsinformationen bzgl. der Globalstrahlungsdaten.
- Die Datenhaltung sollte vereinfacht und die Datenprüfung auf Integrität und Validität vor einer weiteren Prozessierung ermöglicht werden.

9.3.3 Fortschritt durch WISENT

- Die erweiterten Anforderungen wurden erhoben und spezifiziert.
- Ein neu entwickeltes XML-basiertes Format hat die CSV-Dateien abgelöst und ermöglicht nun die leichte Prüfung der Daten mit Hilfe von Standard-XML-Software-Tools und -Bibliotheken.
- Die zusätzlichen Qualitätsinformationen in Form von Konfidenzintervallen stellen für meteocontrol als Anbieter von energiemeteorologischen Informationen ein Alleinstellungsmerkmal dar.

9.3.4 IST-Zustand Produkt/Dienstleistung vor dem WISENT-Projekt

- Schon vor und unabhängig von WISENT bestand eine Kooperation mit meteocontrol zur täglichen Simulation der erwarteten Leistung von einigen Hundert PV-Anlagen mit Hilfe der an der Universität Oldenburg berechneten Globalstrahlungsdaten sowie der bei meteocontrol vorliegenden Daten bzgl. PV-Anlagenkonfigurationen und Umgebungstemperaturen.
- Letztere beide Datenarten werden bei meteocontrol über einen SOAP-Webservice angeboten, der einen Abruf der entsprechenden lokalen Datenbank vermittelt.

9.3.5 Neue Anforderungen

- Modulare Dienstinfrastruktur
- Höhere Ausfallsicherheit und Datenintegrität

9.3.6 Fortschritt durch WISENT

- Umsetzung der Berechnungen an Universität Oldenburg als Module, die über Webservices gekapselt werden
- Übermittlung via GridFTP (Reliable File Transfer)

9.4 Meteorologische Modellrechnungen

9.4.1 IST-Zustand vor dem WISENT-Projekt

Die weltweit viel verwendete Open-Source-Software WRF (Weather Research and Forecast Model) wurde am Zentrum für Windenergieforschung ForWind an der Universität Oldenburg zunächst nur auf einzelnen Arbeitsplatzrechnern installiert und konfiguriert. Berechnungen haben einen hohen Bedarf an Rechenzeit sowie Massenspeicher für Input und Output, so dass entweder nur Testrechnungen durchgeführt wurden oder aber lange Wartezeiten für verwertbare Ergebnisse in Kauf genommen werden mussten, die die Forschungsarbeit sehr bremsten.

9.4.2 Fortschritt durch WISENT

- Es wurde viel Aufwand für die Konfiguration und Implementation von WRF auf dem OFFIS-Cluster sowie die Entwicklung von Schnittstellen, insbes. GUIs, betrieben. Auf diese Weise ist WRF jetzt für reale Anwendungen in Forschung und operationellem Betrieb in angemessener Zeit nutzbar.
- Erhöht wird diese Nutzbarkeit sogar noch durch die geplante Nutzung anderer D-Grid-Ressourcen.
- Bei ForWind wird zunehmend an mesoskaliger Modellierung geforscht und ein operationelles Windleistungsvorhersagesystem (HUGIN) betrieben, das WRF-Ergebnisse als Input nutzt. Die nachhaltige Nutzung des WRF-Grid-Prozessors ist also gesichert.
- Durch das Angebot von DV-Ressourcen, aber auch von Experten-Know-How zur sinnvollen Konfiguration an die Forschungs-Community Energiemeteorologie soll eine nachhaltige Nutzergemeinde aufgebaut werden.
- Durch kommerzielle Anwendungen soll der Betrieb des WRF-Grid-Prozessors auch eigenständige Einnahmen generieren. Die Einzelheiten sind allerdings im Rahmen des D-Grid-Geschäftsmodelles zu klären.
- Die Attraktivität des WRF-Grid-Prozessors wird durch Bestrebungen, ein einheitliches Frontend für die Ausführung von Numerischen Wettermodellen zu schaffen (*NWP Plug-in for Eclipse*), weiter erhöht. Dies ist als Open-Source-Projekt mit einer auch für die kommerzielle Verwertung freundlichen Lizenz geplant. Seine Nachhaltigkeit wird vor allem durch die Gewinnung einer breiten Nutzerbasis (auch außerhalb von WISENT) gewährleistet.

10 D-Grid-Geschäftsmodell der Community Energiemeteorologie

Die BMBF-Sonderinvestitionen im OFFIS und am DLR haben für eine beschleunigte Integration der WISENT-Partner in die D-Grid-Infrastruktur gesorgt. Insbesondere der High-Performance-Computing-Cluster im OFFIS konnte in WISENT bereits für die Durchführung von Experimenten auf Basis komplexer Simulationsmodelle genutzt werden und verdeutlicht damit die Potenziale, die sich durch die Verfügbarkeit derartiger Hardware-Ressourcen ergeben. Für den dauerhaften Betrieb der Sonderinvestitionen ist allerdings ein kostendeckendes Geschäftsmodell erforderlich. Die Community Energiemeteorologie sieht sich dabei als Teil eines Ganzen (dem D-Grid), wobei die Randbedingungen, die sich durch das D-Grid ergeben, bei der Erstellung eines Community-spezifischen Geschäftsmodells berücksichtigt werden müssen. Ein detailliertes D-Grid-Geschäftsmodell ist zum jetzigen Zeitpunkt noch nicht verfügbar. Im Folgenden wird auf Basis des aktuellen Kenntnisstandes beschrieben, wie sich die

Community Energiemeteorologie in das D-Grid integrieren könnte, um eine nachhaltige Community als Teil des D-Grids zu werden.

In der Community Energiemeteorologie gibt es zahlreiche Anwendungsfelder, in denen Grid-Ressourcen sinnvoll eingesetzt werden können. Dies sind z.B. die Solarstrahlungsberechnung, meteorologische Modellrechnungen (z. B. auf Basis des Weather Research and Forecast Model) und die Strahlungsvorhersage. Aufgrund der im Vergleich zu anderen Communitys noch geringen Größe der jungen und im Wachstum befindlichen Community Energiemeteorologie ist es sinnvoll, eine enge Zusammenarbeit mit dem Kern-D-Grid anzustreben und nicht zu stark von den angebotenen Dienstleistungen und Lösungen abzuweichen. Voraussetzung dafür ist allerdings, dass im D-Grid auch die Anforderungen des WISENT-Projektes berücksichtigt werden, was u.a. bereits durch die Präsentation der WISENT-Anforderungen auf zwei Nachhaltigkeits-Workshops des D-Grid angestoßen wurde. WISENT stützt sich damit auf die Arbeiten/Technologien des D-Grid im Bereich der Grid-Middleware (Globus Toolkit), des VO-Management (VOMRS), der Security und des Accounting und Billings. Es wurden bereits zahlreiche Überlegungen angestellt, die zu einer Deckung der Kosten zum Betrieb einer ins D-Grid eingebetteten WISENT-Grid-Infrastruktur führen können:

- Einsparungspotenzial bei Anschaffung und Betrieb eigener Compute-Server (z.B. Auslagerung der Solaranlagen simulation bei meteocontrol)
- Erschließung neuer (rechenintensiver) Geschäftsfelder
- Effizientere Arbeit von Wissenschaftlern (durch schnellere und bessere Ergebnisse)
- Verbesserung der Datenqualität (z.B. aufgrund von Methodenverbesserungen)
- Schnellere Reaktionszeiten bei der Anforderung aufwändiger Datenprodukte
- Vermietung von temporärem oder permanentem Speicherplatz

Die genannten Punkte bieten Einsparpotenziale, wodurch eingesparte Gelder der Grid-Infrastruktur zugeführt werden können. Zusätzlich können durch die Erschließung neuer Geschäftsfelder bzw. die Bereitstellung verbesserter Dienste (z.B. Verbesserung der Datenqualität oder der Reaktionszeit) höhere Einnahmen erzielt werden, die zum Teil in die Grid-Infrastruktur fließen können.

Die Erstellung eines detaillierten Geschäftsmodells für die Community Energiemeteorologie ist aufgrund des Wunsches der dauerhaften Integration in das D-Grid in Planung. Bis Anfang Oktober 2007 wird eine erste Version des Geschäftsmodells entstehen. Das weitere Vorgehen wird in enger Absprache mit dem D-Grid erfolgen.



Geschäftsmodell der Community Energiemeteorologie

Stand: 19.06.2008

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

1 Ausgangssituation

1.1 Marktübersicht

Das Volumen des Solarenergiemarktes betrug 2007 ca. sieben Milliarden € in Deutschland und vier Milliarden € in der restlichen EU. Bis zum Jahre 2010 wird eine Verdoppelung der Umsätze im deutschen Solarenergiemarkt prognostiziert, weltweit wird ein Faktor drei erwartet. Die deutsche Solarwirtschaft wird bis 2010 über sieben Milliarden Euro in den Ausbau modernster Solarfabriken und rund 700 Millionen Euro in Forschung und Entwicklung investieren. Der Markt, zur Zeit noch deutlich auf Fördergeldern und vergünstigten Tarifen basierend, wandelt sich zunehmend zu einem kapitalgetriebenen Markt. Um diese Entwicklung zu unterstützen, ist Investitionssicherheit erforderlich, die vor allem durch Dienstleistungen in den Phasen Investitionsentscheidung und Anlagenmanagement unterstützt wird.

In beiden Phasen ist die Verfügbarkeit von hochwertigen Daten zum Solarstrahlungsangebot eine essentielle Voraussetzung für entsprechende Dienstleistungen. Hierfür haben sich Fernerkundungsdaten wegen ihrer hohen räumlichen Auflösung im Vergleich zu Bodenmessdaten sehr bewährt. Neben den Marktsegmenten, die mit den beiden genannten Phasen zusammenhängen, ist die Last- und Erzeugungsvorhersage für Betriebsmittel ein weiteres vielversprechendes Marktsegment. Aufgrund liberalisierter Märkte und einer hohen Durchdringung erneuerbarer Energien, ist eine präzise Leistungsvorhersage erforderlich, um gute Preise an den Energiemärkten zu erzielen. Für dieses Marktsegment sind Temperatur und Solarstrahlung wesentliche Kenngrößen bei der Vorhersage der Energieerzeugung. Die für derartige Vorhersagen benötigten Zeitreihen können auf der Grundlage von Fernerkundungsdaten bereitgestellt werden, ohne die ortsbezogenen Beschränkungen von Bodenmessdaten in Kauf nehmen zu müssen.

Die Entwicklung des Solarenergie-Marktes ist deutlich stürmischer verlaufen als erwartet. Umsatz-Prognosen mussten regelmässig nach oben korrigiert werden. Auf der anderen Seite wurden die Erwartungen im Betriebsmittel-Sektor nicht erfüllt.

Die WISENT-Projektpartner sind mit ihren angebotenen Produkten und Diensten in vier Marktsegmenten aktiv:

1. Investitionsentscheidungen (für solarthermische Kraftwerke; große und kleine Photovoltaik-Anlagen)
2. Anlagen-Management (große und kleine Photovoltaik-Anlagen)
3. Betriebsmittel: Lastvorhersage, Leistungsvorhersage von Solaranlagen
4. Wissenschaft und Beratung: Zeitreihen meteorologischer Parameter

Der Zielmarkt kann anhand der Art des Auftraggebers klassifiziert werden, wobei die folgenden Klassen unterschieden werden:

1. Gewerbliche Investoren, Versicherungen, Banken, Projektentwickler und größere Anlagenbetreiber. Dieser Marktsektor wird im Folgenden als Großanlagen-Solarmarkt bezeichnet.
2. Private Investoren, kleine und mittlere Anlagenbetreiber (Kleinanlagen-Solarmarkt).
3. Dienstleistungen im Bereich der Energieversorgung (Netzmanagement, Vorhersage der Erzeugung, Lastvorhersage u.a.)

1.2 Anforderungen der Community Energiemeteorologie

Für das Treffen von Investitionsentscheidungen werden zwei Arten von Dienstleistungen benötigt:

1. Bereitstellung räumlich aufgelöster Daten, die den Investor unterstützen, einen geeigneten Standort auszuwählen
2. Standortprüfungen, welche Investoren dabei unterstützen, eine detaillierte Standortbewertung vorzunehmen

Für die Optimierung des finanziellen Ertrags ist die Bewertung von Standorten für Investoren äußerst bedeutend. Die Bewertung wird dabei durch auf GIS (Geographical Information System) basierenden Dienstleistungen ermöglicht. Die Kosten für ein Solaranlagenprojekt hängen z.B. von der Landabdeckung und der Entfernung zur Infrastruktur (Transportwege, Straßen, Wasserversorgung etc.) ab. Der finanzielle Ertrag ist zudem vom verfügbaren „Brennstoff“, der solaren Strahlungsdichte, abhängig. Unter Berücksichtigung dieser Einflussfaktoren ist es möglich, die besten Standorte für Solaranlagen zu finden, die eine hohe Strahlungsdichte aufweisen und geringe Investitionskosten verursachen. Bei wachsenden Anlagengrößen und höheren Investitionen wird die Standortoptimierung für Investoren immer wichtiger. Standortgutachten werden für den Entwurf der Anlagen und für die genaue finanzielle Vorausschau des Anlagenbetriebs genutzt. Die Genauigkeit der Informationen sollte deshalb hoch sein, weshalb aufgrund der jährlichen Schwankungen Langzeit-Zeitreihen erforderlich sind. Eine Zeitreihe über 10 Jahre ermöglicht z.B. die Reduzierung von Ungenauigkeiten des Langzeit-Durchschnitts auf $\pm 5\%$. Des Weiteren ist zu berücksichtigen, dass die Finanzierung von Solaranlagen zu 50 bis 75% auf Krediten beruht. Deshalb verlangen Banken und Versicherungen entsprechende Gutachten, bevor sie den Kredit für einen Investor bewilligen. Für große Anlagen werden sogar zwei Gutachten gefordert: ein oder zwei Gutachten werden vom Investor geliefert und eine von der Bank selbst, um die grundlegenden Zahlen der Investitionsentscheidung prüfen zu können. Auch für kleine Anlagen ist eine Abschätzung des erwarteten Ertrags erforderlich, um das Risiko des privaten Investors zu minimieren. Auf diese Weise können Fehlentscheidungen vermieden werden und Angebote unterschiedlicher Anbieter miteinander verglichen werden. Da ein Investor in dieser Stufe des Verkaufsprozesses in der Regel nichts für die Ertragsschätzung bezahlt, sollte die Ertragsschätzung sehr preisgünstig sein. Aufgrund einer hohen Anzahl von Angeboten ist ein schneller und einfacher Zugriff auf diese Art von Ertragsschätzungen wünschenswert.

Dienstleistungen für Investitionsentscheidungen werden ebenso benötigt, um den Export von Solaragententechnologie zu sichern. Beispiele dafür sind die Export-Initiative des Bundesumweltministeriums für Solarthermische Stromerzeugung. Die Technologie braucht Standorte mit hoher direkter Einstrahlung, welche in Deutschland nicht verfügbar sind. Das Ministerium unterstützt deshalb auch Studien und Einstrahlungs-Kartierungen außerhalb Deutschlands, um die Exportchancen der deutschen Industrie zu fördern, die viele Komponenten für Anlagen nach Südeuropa und Nordafrika liefert. Für die Sicherstellung des Ertrags bereits installierter Solaranlagen ist in jedem Falle eine Überwachung erforderlich. Die Größe der Systeme bestimmt dabei den Umfang der Dienstleistungen: Kleine Solaranlagen benötigen eine kostengünstige Überwachung (Leistungsdiagnose), während große Anlagen eine detaillierte Überwachung mit automatischen Fehlererkennungsroutinen erfordern. In beiden Fällen wird eine Reduktion der Ausfallzeiten erreicht.

Die Nachfrage nach Zeitreihen kommt von wissenschaftlichen und/oder beratend tätigen Einrichtungen wie Universitäten und Forschungseinrichtungen, Beratern in Systemtechnik,

Beratern in Finanzentwicklungen, lokalen Behörden und Umweltbehörden. Von der weltweiten Nachfrage stammt 80% aus Europa. Zeitreihen werden genutzt, um den Ertrag von Solaranlagen zu berechnen. Des Weiteren können sie genutzt werden, um im Architekturbereich Beleuchtungs- und Wärmebedarf in Gebäuden abzuschätzen. Zeitreihen sind nicht anwendungsspezifische Daten, die i. d. R. für energietechnische Anwendungen entsprechend weiter verarbeitet werden („value adding“). Im Allgemeinen sind Planer, Architekten und Wissenschaftler die Hauptkunden in diesem Sektor.

1.3 Grid-Infrastruktur als Grundlage für neue Dienste

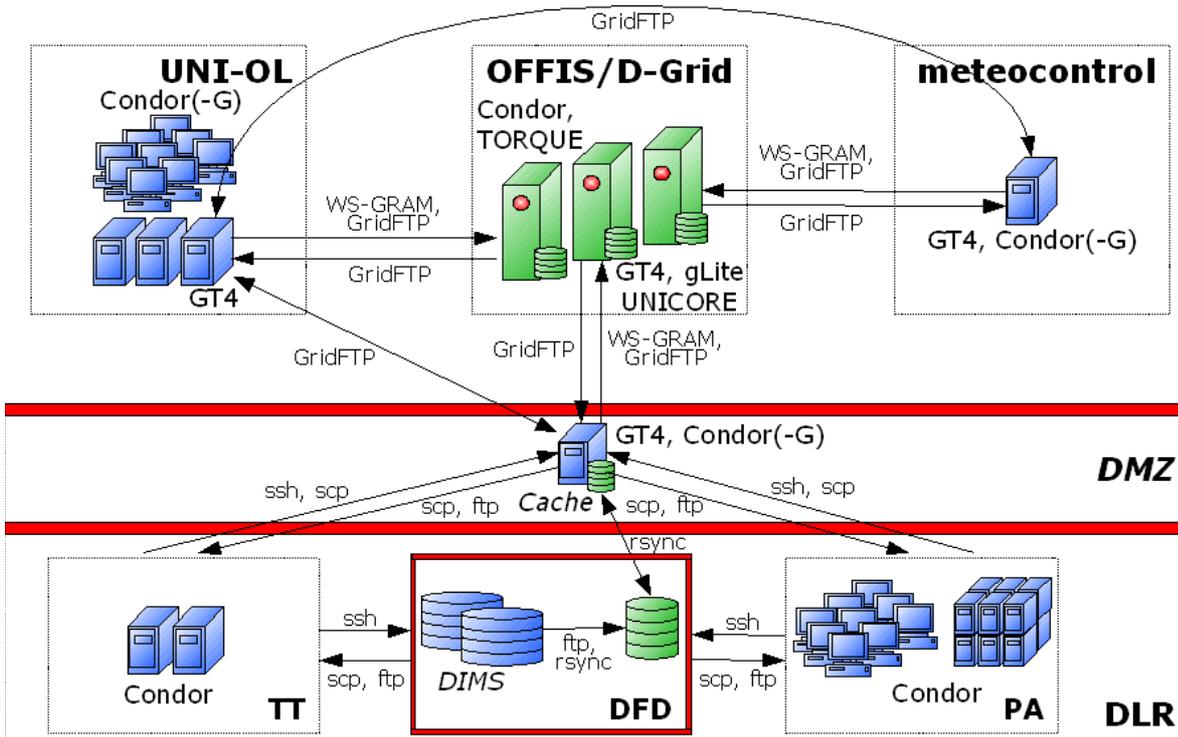


Abbildung 1: WISENT-Grid-Infrastruktur

Die im Rahmen des WISENT-Projektes realisierte Grid-Infrastruktur (siehe Abbildung 1) konnte bereits als Grundlage für die Erstellung neuer Dienste genutzt werden. Alle an WISENT beteiligten Projektpartner (meteocontrol, Universität Oldenburg, OFFIS und DLR) sind bereits in diese Infrastruktur integriert und konnten dementsprechend Erfahrungen mit Grid-Technologien sammeln. Zudem erfolgte bereits eine Integration in die D-Grid-Infrastruktur. Die wertvollen Erfahrungen des Einsatzes von Grid-Technologie wurden bzw. werden in dem Community-spezifischen Wiki-System TikiWiki (siehe Abbildung 2) dokumentiert, welches eine Art zentrale Wissensbasis der Community darstellt. Die Nachhaltigkeit des TikiWiki-Systems ist inhaltlich durch die Fortführung bzw. Erweiterung der Arbeiten in der Community Energiemeteorologie gegeben. Die technische Nachhaltigkeit ist durch die Fortführung des Hostings und die Pflege des Systems an der Universität Oldenburg über die WISENT-Projektlaufzeit hinaus gegeben. Damit bleiben

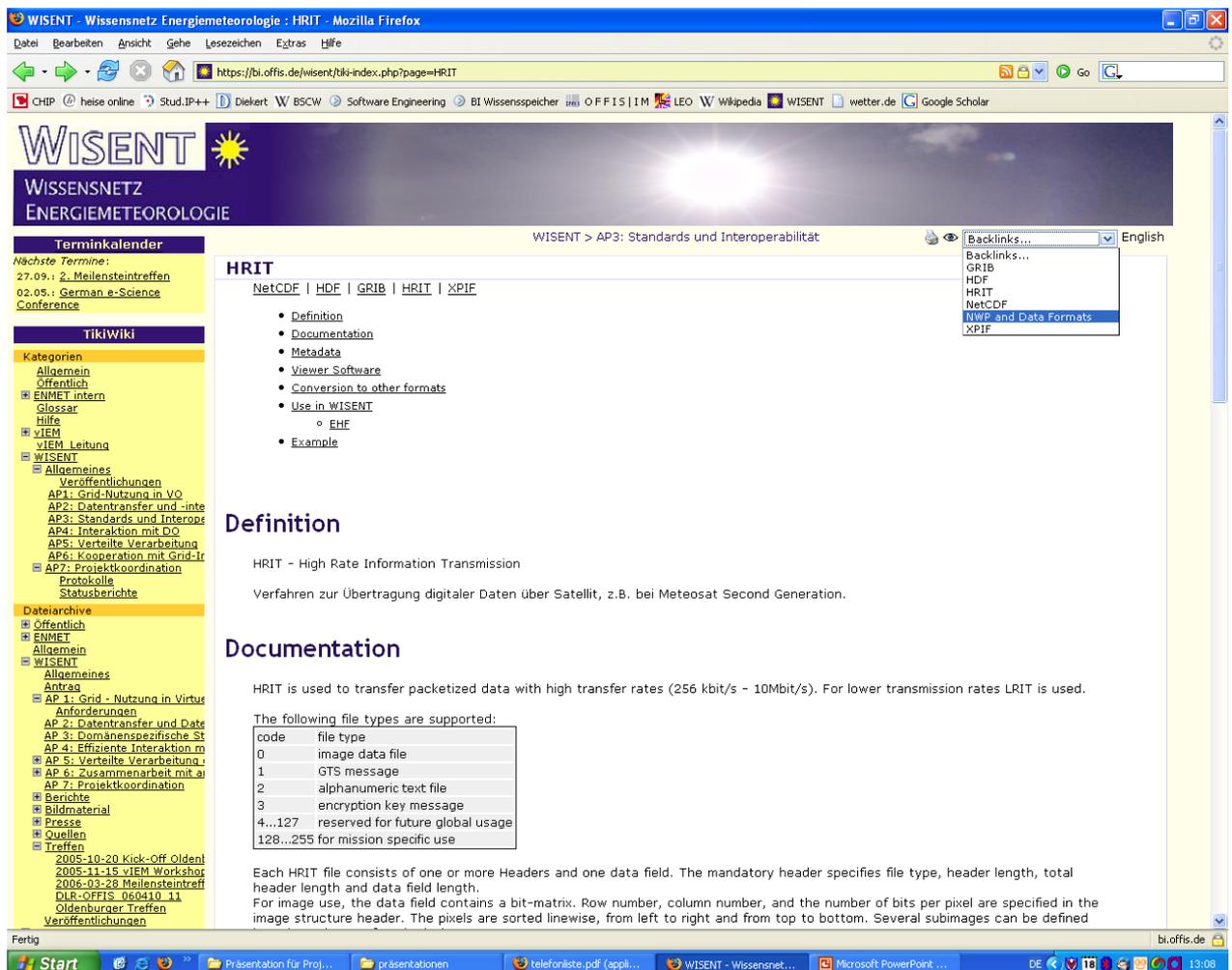


Abbildung 2: TikiWiki-System der Community Energiemeteorologie

auch die im Projektverlauf gesammelten Erfahrungen im Umgang mit Grid-Technologien innerhalb des TikiWiki-Systems erhalten.

Für den Erhalt der Grid-Infrastruktur der Community Energiemeteorologie sind die beteiligten Partner verantwortlich, deren Zusammenhalt durch die gemeinsame inhaltliche Arbeit und gemeinsame Projekte gegeben ist. Bei einer weiterhin steigenden Nutzung der D-Grid-Infrastruktur durch die Community Energiemeteorologie macht auf lange Sicht die

Bildung einer Community-spezifischen Einrichtung Sinn, deren Aufgabe darin besteht, den Erhalt der Community-spezifischen Grid-Infrastruktur sicherzustellen und neue Anforderungen aus der Community in die Fortentwicklung der Grid-Infrastruktur einfließen zu lassen. Solange keine eigene Einrichtung für den Community-spezifischen Support existiert, muss jeder Partner auf eigene Ressourcen oder auf die vom D-Grid angebotenen Dienstleistungen zurückgreifen. Im Folgenden wird das Nutzungsversprechen einer Einrichtung beschrieben, die Support für die Community-spezifische Grid-Infrastruktur leisten könnte. Diese Einrichtung wird im Folgenden **EnMetSE** (Energiemeteorologie-Support-Einheit) genannt. Das Nutzungsversprechen ist bewusst an das Nutzungsversprechen der DGSE (D-Grid Support-Einrichtung) angelehnt, berücksichtigt jedoch verstärkt Community-spezifische Anforderungen. Die Rolle der EnMetSE wird im WISENT-Projekt verstärkt durch den Projektpartner OFFIS wahrgenommen, der diese Rolle nach Projektende jedoch aufgrund der auslaufenden Förderung nicht mehr in gleicher Weise wahrnehmen kann.

2 Nutzungsversprechen

Das Nutzungsversprechen der Community Energiemeteorologie schließt in Anlehnung an das Geschäftsmodell der DGSE (D-Grid Support-Einrichtung) folgende Kundengruppen für die EnMetSE mit ein:

1. **Betreiber (KB)** der Gridinfrastruktur der Community Energiemeteorologie
2. **Anbieter (KA)** von Community-spezifischen Gridressourcen. Dabei kann es sich um Hardwareressourcen oder andere Dienste handeln.
3. **Entwickler (KE)** Community-spezifischer Softwarekomponenten für das Grid

Das Nutzungsversprechen basiert im Wesentlichen auf dem Portfolio der von der D-Grid-Kern-Infrastruktur unterstützten Softwarekomponenten und weiteren Community-spezifischen Softwarekomponenten.

1. **Wissensbasis:** Der Betrieb eines Systems, das der Vernetzung des Community-spezifischen Wissens und dem Erfahrungsaustausch dient, wird sichergestellt und allen Kunden (KB, KA, KE) zur Verfügung gestellt. Momentan handelt es sich dabei um ein TikiWiki-System. Die Community-spezifischen Erweiterungen des TikiWiki-Systems fließen in das offizielle Release des Open-Source-Projekts zurück, um auch anderen Projekten zur Verfügung zu stehen.
2. **Portfolioanpassungen:** Anpassungsanforderungen des Portfolios der unterstützten Softwarekomponenten von Seiten der Kunden (KB, KA) werden berücksichtigt.
3. **Support:** Für die Community-spezifischen Softwarekomponenten des Portfolios wird Support geleistet. Der genaue Leistungsumfang muss bestimmt werden, insbesondere Überschneidungen mit der DGSE sind zu berücksichtigen. (KB, KA)
4. **Information:** Alle Kunden werden über das Internet und gegebenenfalls über Präsenzveranstaltungen über grundsätzlich verfügbare Dienste und neue Entwicklungen im Community-Grid informiert. (KB, KA, KE)
5. **Einbindung:** Anbieter von Diensten und Ressourcen werden bei der Integration in die Community-Infrastruktur beraten. (KA)
6. **Anpassungsentwicklungen:** Bei Bedarf werden Community-spezifische Anpassungsarbeiten durchgeführt. (KB, KA)

7. **Entwicklerberatung:** Entwickler neuer Softwarekomponenten werden bezüglich der Einbindung ihrer Komponenten in das Community-Grid beraten. Dies erfolgt ggf. in enger Zusammenarbeit mit der DGSE. (KE)
8. **Teststellung:** Entwicklern von Softwarekomponenten wird die Möglichkeit gegeben, ihre entwickelten Community-spezifischen Komponenten mit Unterstützung der EnMetSE zu testen. (KE)
9. **Betrieb:** Der Betrieb der Community-Grid-Infrastruktur wird auf der Basis von vereinbarten Betriebskonzepten durchgeführt. Die D-Grid-Betriebskonzepte werden berücksichtigt. (KB)
10. **Betriebskonzept:** Die Betriebskonzepte werden (ggf. in Abstimmung mit der DGSE) auf Grundlage des jeweiligen Entwicklungsstandes der Softwarekomponenten entsprechend der Bedürfnisse der Kunden angepasst. (KB)
11. **Aufbau:** Betreiber Community-spezifischer Grid-Infrastrukturen werden bzgl. Aufbau und Betriebskonzept beraten. (KB)
12. **Nichttechnische Fragestellungen:** Die Behandlung nichttechnischer Fragestellungen erfolgt nur, wenn sie für eine Mehrzahl von Kunden Relevanz hat. (KB)

3 Architektur

Die Community Energiemeteorologie sieht sich als Teil eines Ganzen (dem D-Grid), wobei die Randbedingungen, die sich durch das D-Grid ergeben, bei der Erstellung einer Community-spezifischen Grid-Infrastruktur berücksichtigt werden müssen. Aufgrund der im Vergleich zu anderen Communities noch geringen Größe der jungen und im Wachstum befindlichen Community Energiemeteorologie ist es sinnvoll, eine enge Zusammenarbeit mit dem Kern-D-Grid anzustreben und nicht zu stark von den angebotenen Dienstleistungen und Lösungen abzuweichen. Voraussetzung dafür ist allerdings, dass im D-Grid die Anforderungen der Community Energiemeteorologie berücksichtigt werden. Es ist daher Aufgabe der Community, dem D-Grid geänderte Anforderungen aktiv mitzuteilen. Die Community stützt sich auf die Arbeiten/Technologien des D-Grid im Bereich der Grid-Middleware, des VO-Management, der Security und des Accounting und Billings. Community-spezifische Lösungen sind u.a. Condor für die Verbindung von Desktop-Ressourcen (siehe Abbildung 1) und in Zukunft möglicherweise Shibboleth für das VO-Management. Eine Orientierung an dem Portfolio der DGSE hilft der Community, den Aufwand für Eigenentwicklungen gering zu halten. Bevor Eigenentwicklungen angestrebt werden, sollte die Machbarkeit mit den bestehenden Lösungen analysiert werden.

Bei der EnMetSE handelt es sich um eine Einrichtung, deren organisatorische Rahmenbedingungen noch ausgearbeitet werden müssen. Sinnvoll wäre eine Zusammensetzung aus Mitarbeitern an der Community beteiligter Institutionen. Entscheidend für den Aufbau einer solchen Support-Einrichtung ist die Nachfrage, die momentan noch nicht ausreicht, um sie in einer gesonderten Organisationsform auszulagern. Momentan wird der Community-spezifische Support verstärkt durch den Projektpartner OFFIS geleistet. Nach Projektende wird für Support-Anfragen verstärkt auf D-Grid-Dienstleistungen zurückgegriffen werden müssen, wenn diese nicht intern geklärt werden können.

Es ist beabsichtigt, die Community Energiemeteorologie in eine größere Community Energie einzubetten, die den Fokus verstärkt auf die Energiewirtschaft legt. Eine Tendenz in diese Richtung ist bereits bei Betrachtung des Ertragsmodells im folgenden Abschnitt zu erkennen. Das entstehende Energie-Grid soll eine deutlich größere Community abdecken

und aufgrund der verstärkt kommerziellen Ausrichtung den dauerhaften Erhalt der Community-spezifischen Grid-Infrastruktur sicherstellen.

4 Ertragsmodell

Für eine Deckung der Kosten zum Betrieb einer Community-spezifischen Grid-Infrastruktur ist eine Abschätzung von Einsparpotenzialen und höheren Einnahmen durch die Nutzung dieser Infrastruktur erforderlich. Dazu ist es sinnvoll, die Wertschöpfungsketten der von der Community Energiemeteorologie erzeugten Produkte bzw. angebotenen Dienstleistungen zu untersuchen. Die Abbildungen 3 bis 5 zeigen beispielhaft drei Wertschöpfungsketten für die Produkte Kurzzeitstrahlungsvorhersage, Photovoltaik-Ertragsüberwachung und STEPS, an denen mehrere Organisationen beteiligt sind. Für die Kurzzeitstrahlungsvorhersage liefert meteocontrol beispielsweise Daten von

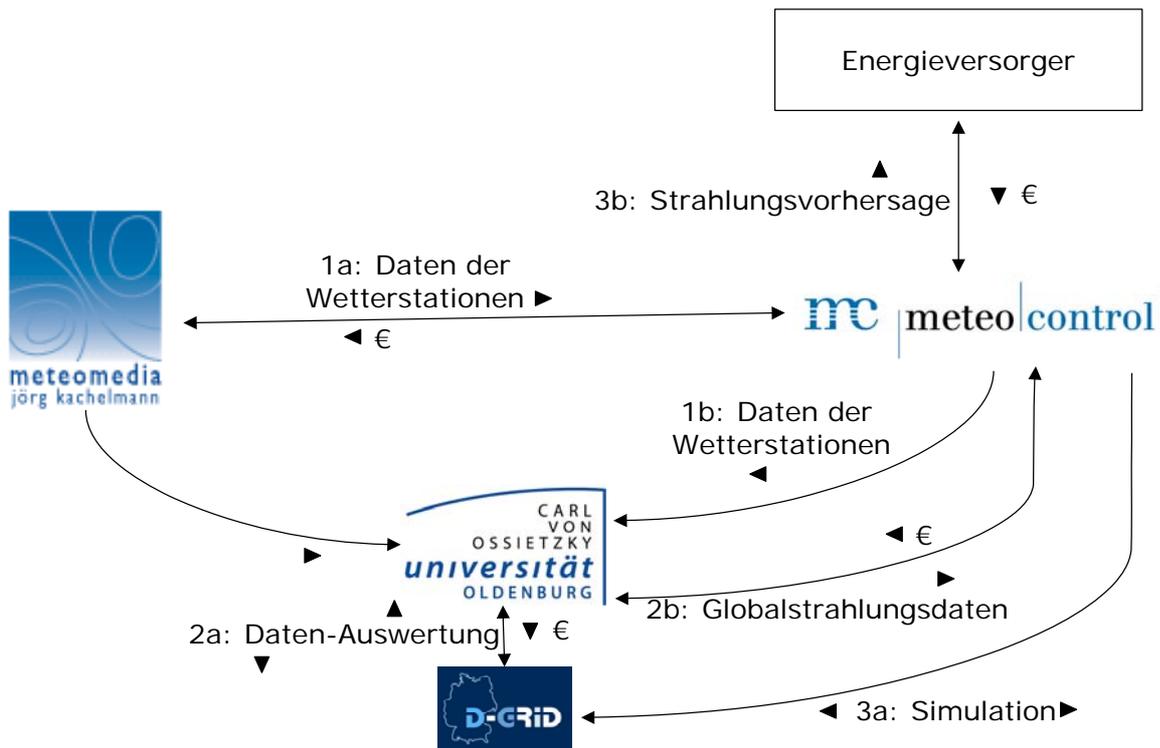


Abbildung 3: Wertschöpfungskette zur Kurzzeitstrahlungsvorhersage

Wetterstationen, die zuvor von meteomedia bezogen wurden, an die Universität Oldenburg. Die Universität Oldenburg liefert auf Basis der Wetterstationsdaten und zusätzlicher METEOSAT-Daten ein (Mehrwert-)Produkt an meteocontrol zurück. Meteocontrol erstellt aus den gelieferten Daten wiederum eine Strahlungsvorhersage für den „Verbraucher“ meteomedia. Es ist dabei dargestellt, an welcher Stelle kostenpflichtige D-Grid-Ressourcen genutzt werden könnten, deren Bezahlung durch die Wertsteigerung der prozessierten Daten ermöglicht wird. In Abbildung 4 wird eine Wertschöpfungskette zur Photovoltaik-Ertragsüberwachung gezeigt, in der zusätzlich noch die Betreiber von Solaranlagen hinzukommen, die als Verbraucher für das Überwachungsprodukt bezahlen. Abbildung 5 zeigt die Durchführung einer auf STEPS (Expert System for Solar Thermal Power Stations) basierenden Standort-Analyse für Investoren mit Hilfe von D-Grid-Ressourcen. In allen Abbildungen entstehen Wertsteigerungen des „Ursprungs“-Produktes durch bestimmte, von der jeweiligen Organisation durchgeführte Prozessierungen.

Auf Basis der Wertschöpfungsketten und weiterer Überlegungen können folgende Punkte aufgeführt werden, die zu einer Deckung der Kosten zum Betrieb einer ins D-Grid eingebetteten Community-spezifischen Grid-Infrastruktur führen können:

- Verbesserung der Datenqualität (z.B. aufgrund von Methodenverbesserungen)
- Schnellere Reaktionszeiten bei der Anforderung aufwändiger Datenprodukte
- Einsparungspotenzial bei Anschaffung und Betrieb eigener Compute-Server (z.B. Auslagerung der Solaranlagen-simulation bei meteocontrol)
- Erschließung neuer (rechenintensiver) Geschäftsfelder
- Effizientere Arbeit von Wissenschaftlern (durch schnellere und bessere Ergebnisse)
- Vermietung von temporärem oder permanentem Speicherplatz

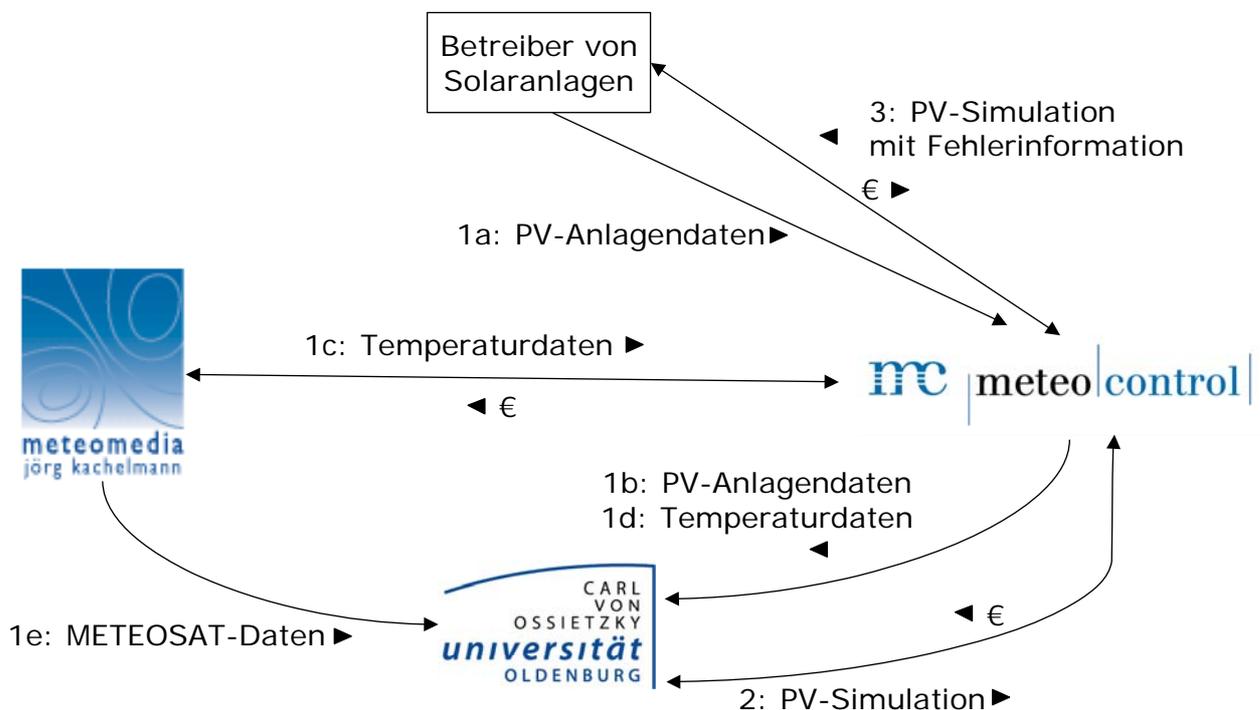


Abbildung 4: Wertschöpfungskette zur Photovoltaik-Ertragsüberwachung

Die genannten Punkte bieten Einsparpotenziale, wodurch eingesparte Gelder der Grid-Infrastruktur zugeführt werden können. Zusätzlich können durch die Erschließung neuer Geschäftsfelder bzw. die Bereitstellung verbesserter Dienste (z.B. Verbesserung der Datenqualität oder der Reaktionszeit) höhere Einnahmen erzielt werden, die zum Teil in die Grid-Infrastruktur fließen können.

Die Kosten einer EnMetSE werden durch Personalkosten dominiert, wie es auch bei der DGSE der Fall ist. Hinzu kommen ggf. Sachmittelkosten für die Anschaffung und den Betrieb von Testservern sowie die Arbeitsplatzausstattung der Mitarbeiter. Die Personalmittel hängen vor allem davon ab, inwieweit sich der Community-spezifische Support vom Support der DGSE abgrenzt und wie intensiv die Community-spezifischen Supportleistungen in Anspruch genommen werden.

Aufgrund der unterschiedlichen Anforderungen der Kunden an die EnMetSE und die unterschiedliche Ressourcen-Nutzung ist ein Mischkonzept ähnlich dem der DGSE sinnvoll:

- Basis: Eine oder mehrere Institutionen bilden die Basis der EnMetSE und sichern die Grundfinanzierung.
- Mitglieder: Kunden, die langfristig von dem Nutzungsversprechen Gebrauch machen, finanzieren die EnMetSE durch ihre Mitgliedsbeiträge.
- Teilnehmer: Teilnehmer sind Kunden, die einzelne Angebote (Nutzungsversprechen) der EnMetSE wahrnehmen und dafür zahlen. Die Preise für diese Angebote und ihre Gültigkeitsspanne werden von der EnMetSE festgelegt, wofür entsprechende Prozesse zu vereinbaren sind.

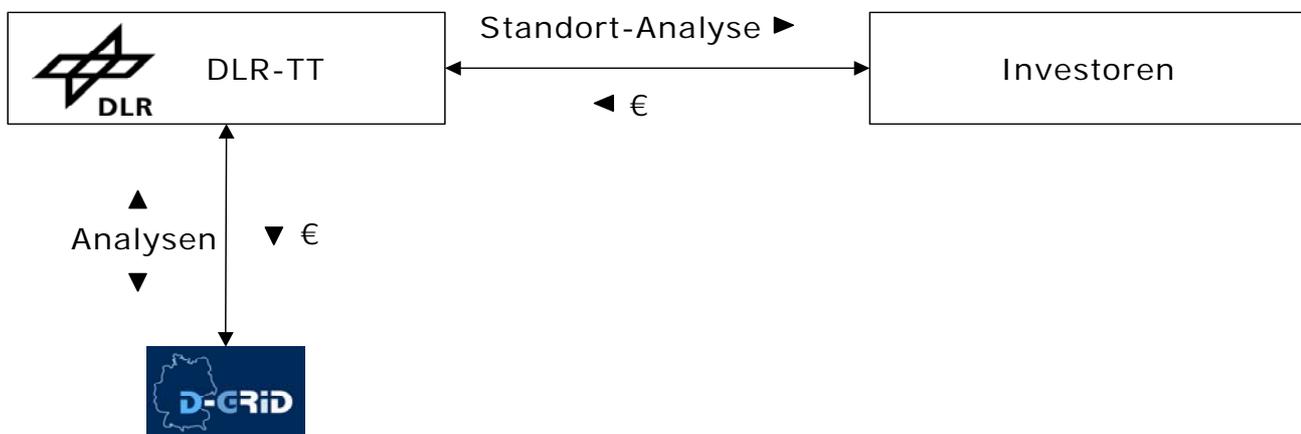


Abbildung 5: Wertschöpfungskette zu STEPS