A Hierarchical Super Peer Network for Distributed Artifacts

Ludger Bischofs[1] and Wilhelm Hasselbring[1] and Jürgen Schlegelmilch[2] and Ulrike Steffens[2]

[1] University of Oldenburg, FK-2, Software Engineering Group, PO Box 2503, 26111 Oldenburg, Germany
{ludger.bischofs|hasselbring}@informatik.uni-oldenburg.de
[2] OFFIS, Escherweg 2, 26121 Oldenburg, Germany
{ulrike.steffens|juergen.schlegelmilch}@offis.de

Abstract. The transition from traditional paper libraries to digital libraries enables new strategies for the use and maintenance of artifact collections. Distributed software development can be regarded as a special case of digital library utilization, where developers or groups of developers are working on the same software geographically dispersed in time zones which might differ. We present a hierarchical super peer network which represents the organizational structures of distributed software development in a natural way and is able to integrate distributed resources like version control systems as well as local devices. This approach is then generalized to support the self-organization of widely distributed, loosely coupled, and autonomous digital library systems.

1 Introduction

The transition from traditional paper libraries to digital libraries enables new strategies for the use and maintenance of artifact collections. Collections are globally distributed and maintained by different organizations and even private persons. Digital binding techniques allow for the construction of project specific reference libraries by reorganizing existing library material and for the reintegration of project results [1]. As production, storage and classification of documents are now accomplished digitally, library support for intermediate and final results of collaborative writing can be achieved. Furthermore the collection and organization of assets other than documents, as for example works of art or services, is possible by referencing them from within the digital library.

Taking advantage of digital libraries in the described manner calls for a flexible support by a system architecture which enables the combination of collections against the background of different organizational, topical, and technical contexts, offering simple access to potential library users on the one hand and guaranteeing autonomy to library patrons on the other hand.

Distributed software development can be regarded as a special case of digital library utilization, where developers or groups of developers are working on the same software geographically dispersed in time zones which might differ. As the use of a central repository for shared artifacts might have substantial drawbacks like slower and less reliable network connections, software developers rely on distributed artifact collections, or in other words on distributed digital libraries of software engineering artifacts.

In this paper we introduce a hierarchical super peer network for software development as an example for a flexible distributed digital library architecture. Distributed teams, in particular for open source software projects, can be regarded as peer-to-peer systems themselves. The support granted to them by the super peer network begins with the formation of new developer groups and projects and enables the flexible self-organization of the involved organizational units and their respective relationships. Apart from artifacts, distributed resources like version control systems as well as local devices can be integrated, so that developers are able to access and use shared artifacts any time and anywhere within the network.

Beyond the context of software engineering, hierarchical super peer networks can be regarded as a general approach to achieve flexibility and autonomy for a loose coupling of digital library collections. For this purpose, the organizational units and their relationships in the area of software engineering have to be reconsidered and transformed to more general units applicable to digital libraries in general.

This paper is organized as follows. Section 2 describes peer types representing the four central organizational units in the area of software engineering and maps the structures of distributed software development projects into a peer-to-peer architecture. The integration of additional resources like version control systems is demonstrated in section 3. Section 4 shows how peers can be organized in a hierarchical peer-to-peer network and how an appropriate lookup service can be designed. Section 5 generalizes the presented approach and describes a super peer network to support cooperation between arbitrary distributed digital libraries and collections. After presenting related work in section 6 we conclude and outline some future work in section 7.

2 Organizational Units and their Relationships

Peer-to-peer architectures are often characterized as the opposite of Client/Server architectures. The most distinctive difference is that in peer-to-peer networks the peers are capable of acting as client and server at the same time. Furthermore, peers are accessible by other peers directly without passing intermediary entities [2].

In case of distributed software development each developer can be considered a peer. A *developer peer* can offer and access artifacts within the peer-to-peer network. Developers are often organized in groups which are managed by special *group peers*. Beyond, developers and groups of developers are organized in projects to reach a common goal. A *project peer* offers the needed project management services. Organizations (e.g. an enterprise or institution) consist of projects, groups and developers and are managed by *organization peers*. The entirety of the described organizational units constitutes a hierarchical structure. Figure 1 depicts a logical view of a possible structure of peers which does not necessarily reflect the physical structure of the involved computers. P_2 and G_3 for example could physically reside on the same computer.

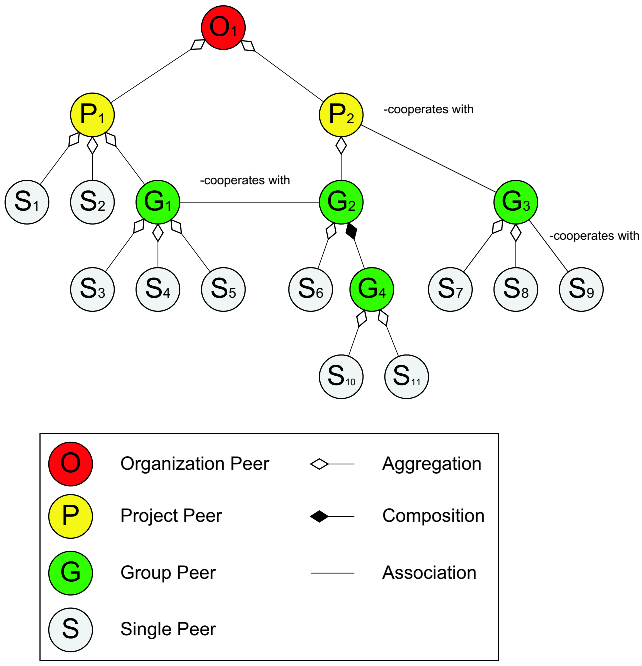

Fig. 1. Hierarchical structure of organizational units

In order to model the relationships between the different peer types the UML notation for aggregation, composition and association is used (cf. figure 1):

– **Aggregation** and **composition** describe a close cooperation between organizational units or peers, respectively, and can also be used to describe their hierarchical order. Typically, to establish an aggregation or composition relationship a peer registers at a superior peer. The superior peer has the special ability to control its registered peers. In figure 1 for example group peer G_1 has control over the single peers S_3 to S_5. The group peers G_2 and G_4 are connected by a composition relationship. This means that peer G_4 cannot exist without group peer G_2 whereas in an aggregation relationship the partners of the aggregation can exist without each other.
– The **association** describes a loosely coupled cooperation with widely autonomous partners. No clear hierarchical structure can be extracted from an association relationship. The groups G_1 and G_2 in figure 1 cooperate. The same is true for the developers within these groups so that access to resources

of the respective other group can be granted to them. The developer at single peer S_9 is associated to group peer G_3 which means that he cooperates with the group G_3. Since an association signifies a loose connection only, a developer can be associated to more than one group at a time.

3 Resource Integration

For cooperative software development developer groups typically use a number of tools like version control systems (e.g. CVS or subversion) and CSCW systems. In the following such tools, storages, or services are understood as *resources*. In the context of a tight cooperation organizational units share resources among each other. A project peer for example usually shares its resources with the involved groups, i.e. the group peers are granted access to these resources. Group peers typically share their resources with single peers, i.e. all developers in a project can use the resources which are connected to the appropriate project peer.

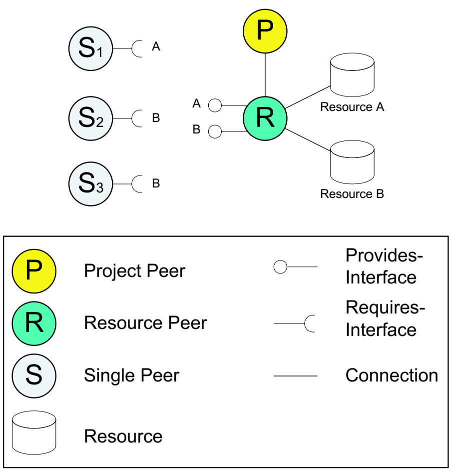

Fig. 2. Resource peer

Resources can directly be integrated into a peer-to-peer network as can be seen in figure 2. The resource peer R connects to the resources A and B in an application specific way. Externally another interface (e.g. in form of a web service) is offered to the rest of the peers. The peers connect to a resource through a special adapter which uses the corresponding external interface of the resource peer. The resource peer either manages access control lists of the peers which have access to the resources or queries another peer, like the project peer P in figure 2, whether to grant access to the requesting peer. All in all, the following steps are necessary to connect to a resource which is integrated into a peer-to-peer network:

1. The requesting peer (e.g. S_2) asks the resource peer for access to a resource.
2. The resource peer asks the responsible peer for access rights or consults its access control list.
3. Afterwards, access is granted or denied to the requesting peer.
4. If access is granted, the resource can be accessed.

By using the interfaces of a resource peer other peers can for example access documents which are stored on a CVS server or a local device. Another example for the shared use of resources is the registration of group members for a forum or a groupware system carried out by their group peer via a resource peer.

4 Multi-tier Look Up Service

The lookup service is a central requirement for peer-to-peer systems. It assigns and locates resources and artifacts among peers [3]. Distributed "flat" peer-to-peer lookup services are e.g. Chord [4], CAN [5], Pastry [6] and Tapestry [7]. The approach presented in this paper is the introduction of a hierarchical multi-tier lookup service where peers are organized in disjoint clusters as it is depicted in figure 3. Super peers route the messages along clusters to the destination cluster. Within the clusters the messages move through the hierarchical structure of the peers. The hierarchical peer structures in combination with their super peers form a hierarchical super peer network [8]. The super peers hold a common metadata index of available artifacts which are distributed over the different organizational units or peer types, respectively. They are able to answer simple queries. Detailed queries additionally pass through the hierarchical structure of the peers. The exchange of the located artifacts takes place directly from peer to peer.

Super peer networks have some advantages in comparison to pure peer-to-peer networks. They combine the efficiency of the centralized client-server model with the autonomy, load balancing, and robustness of distributed search. They also take advantage of the heterogeneity of capabilities across peers. Parameters like cluster size and super peer redundancy have to be considered by designing a super peer network. Redundancy decreases individual super peer load significantly whereas the cluster size affects the behavior of the network in an even more complex manner [9, 10].

The most important benefits of the approach discussed in this paper are scalability and administrative autonomy. A super peer can independently route messages within its cluster. Similarly, organization, project, and group peers can route the messages to subordinated peers using their own strategy. Queries to selected organizational units do not flood the entire network, but can be routed directly.

4.1 Metadata Index

The super peers are connected to each other and share a common metadata index of the available artifacts. Physically, any peer which has enough computing

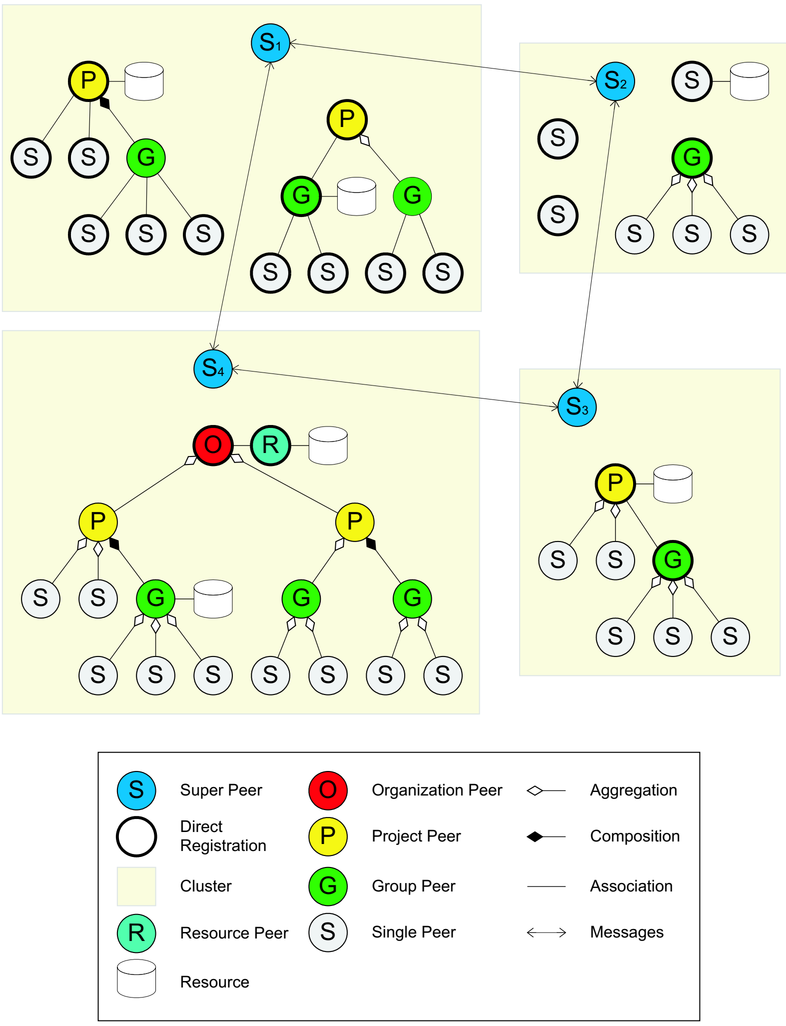

Fig. 3. Multi-tier lookup service in a hierarchical super peer network

power, storage capacity, and an adequate network connection can be chosen to be a super peer. In figure 3, a ring of super peers is shown for reasons of simplicity. There are other techniques like HyperCuP [11] that are able to increase the scalability and reduce the lookup time drastically. Peers which are registered at a super peer make up a cluster as depicted in figure 3.

There are different ways in which super peers can collect metadata. Typically peers which are linked to other peers by associations only register at the associated peers and send their metadata directly to the super peer whereas peers which are in an aggregation or composition relationship register at the superior peer and send their metadata there. Thus, a superior peer has extensive knowledge of the subordinated peers. The superior peers send the available metadata to higher peers in the hierarchical order until the super peer is reached. One advantage of this approach is that not every peer in a network needs an internet connection to make its metadata available. Furthermore, the superior peers have control over the metadata which is sent to a super peer. Hence, superior peers have the ability to decide whether metadata of subordinated peers is made available or unavailable to superior peers or the global index, respectively.

Within the metadata index additional information on the organizational units like group members, project goals and capabilities of developers is stored. Moreover, metadata can also be extracted from resources which are connected to a resource peer. In figure 3 a simplified view of resource connections is illustrated in which the resource peer is not explicitly visible. In this case the associated peers are assumed to have the capabilities of a resource peer.

5 A General Super Peer Network for Digital Libraries

The hierarchical super peer network for distributed software development described above can be generalized to support the flexibility and self-organization of widely distributed, loosely coupled, and autonomous digital library systems. The architecture allows for the search over collections of arbitrary artifacts as for example traditional documents, on-line books, digital images, and videos, which is a basic service requirement for digital libraries [12]. Beyond, the network enables library users to also store, administer, and classify their own artifacts. Thus, it supports scenarios like the construction of personal or group reference libraries and collaborative authoring.

Figure 4 depicts a hierarchical super peer network for digital libraries. The organizational units and their respective peer types are adapted to the situation within a general digital library. Persons are able to search for artifacts and offer artifacts on their own. They are therefore supplied with *person peers.* On the next organizational level, the artifacts are grouped within collections managed by *collection peers.* Collection peers offer functionality relating to collection organization as for example the provision of a common classification scheme. A digital library can combine a number of different collections and is associated with a *digital library peer.* A digital library peer supports the integration of different collections, for example by offering merging services for different classification schemes [13]. Furthermore, it manages access to the digital library artifacts, for example by ensuring a certain mode of payment [14]. Person peers and collection peers can also exist independently from a superior peer and autonomously offer artifacts.

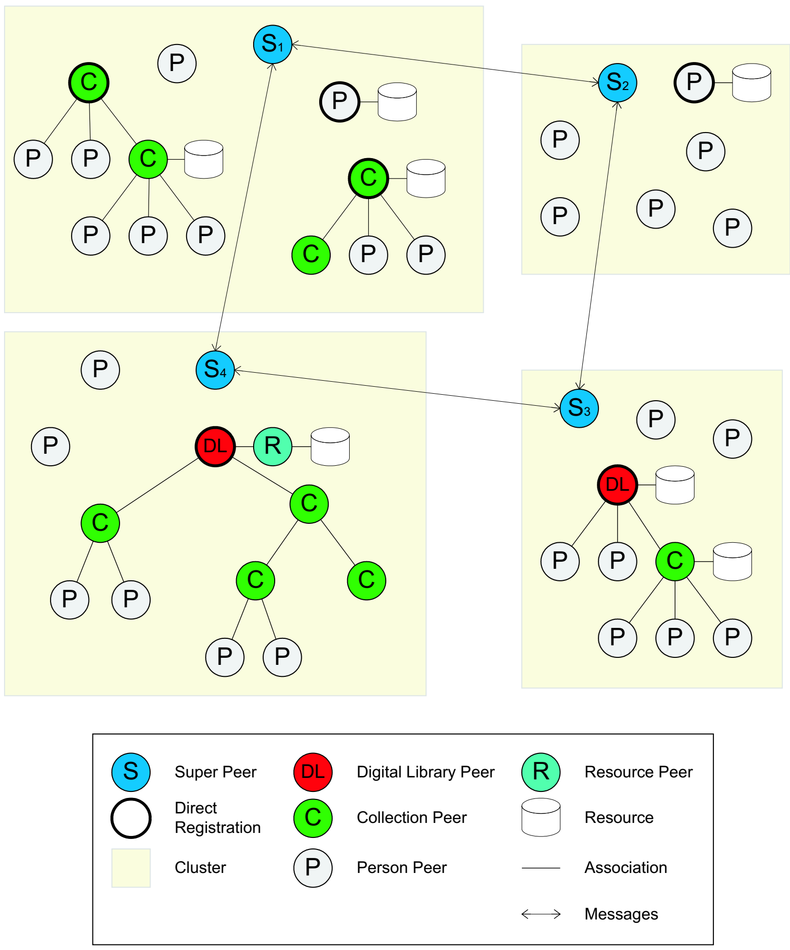

Fig. 4. A hierarchical super peer network for digital libraries

Person peers, collection peers, and digital library peers, as in the approach for distributed software development presented above, are clustered. The clusters again are connected via super peers in the described manner (cf. sect. 4) and searched via super peers and superior peers.

The described network represents a first step in order to capitalize on the advantages of peer-to-peer technology for digital libraries. For personal or project reference libraries most of the upcoming traffic will remain within subareas of the network where co-workers cooperate intensely. For specialized collections which focus on special topics or special media types queries can be routed directly to selected collections or even library experts without flooding the entire network. Precision and query performance can hence be improved. Additionally, a self-organization of collections and libraries is possible. Scalability and administrative autonomy are also ensured.

6 Related Work

Enabling interoperability among heterogeneous, widely distributed, autonomous digital library services is also the purpose of some other projects as described for example in [12]. The goal of establishing a manageable system of personal and project reference libraries as pursued in [1] also calls for a flexibility which can be achieved by the use of a super peer network.

The design of a super peer network is described in [9]. The costs and benefits of a new hybrid approach called structured super peers is explored in [10]. It partially distributes lookup information among a dynamically adjusted set of high-capacity super peers to provide constant-time lookup. Super peer based routing strategies for RDF-based peer-to-peer networks are described in [15].

In hierarchical peer-to-peer systems, peers are organized into groups, and each group has its autonomous intra-group overlay network and lookup service. A general framework and scalable hierarchical overlay management is provided in [8].

7 Conclusions and Future Work

This paper has presented a super peer network approach for autonomous and self-organizing digital libraries and artifact collections and substantiated it by describing a network instance for special libraries dedicated to software development tasks.

One future challenge with regard to hierarchical super peer networks is to analyze the dynamic behavior of the network, particularly if peers fail. Enhancing the availability of artifacts and peer services by replication seems to be one promising approach to solve this problem.

Another issue is to gain further understanding on how the presented approach can be refined against the background of reference libraries. The use of project peers as they have already been introduced for distributed software development could be an option. Yet, the project peers have to be adequately fit into the overall hierarchical structure of the network.

References

1. Schmidt, J.W., Schröder, G., Niederée, C., Matthes, F.: Linguistic and Architectural Requirements for Personalized Digital Libraries. International Journal of Digital Libraries **1** (1997)

2. Schollmeier, R.: A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. In: 1st International Conference on Peer-to-Peer Computing (P2P 2001), Linköping, Sweden, IEEE Computer Society (2001)
3. Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R., Stoica, I.: Looking Up Data in P2P Systems. Communications of the ACM **46** (2003) 43–48
4. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord - A Scalable Peer-to-peer Lookup Service for Internet Applications. In: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM Press (2001) 149–160
5. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Schenker, S.: A Scalable Content-Addressable Network. In: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM Press (2001) 161–172
6. Rowstron, A., Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In: IFIP/ACM International Conference on Distributed Systems Platforms (Middleware). (2001) 329–350
7. Zhao, B.Y., Huang, L., Rhea, S.C., Stribling, J., Joseph, A.D., Kubiatowicz, J.D.: Tapestry: A Resilient Global-Scale Overlay for Service Deployment. IEEE J-SAC **22** (2004) 41–53
8. Garces-Erice, L., Biersack, E.W., Ross, K.W., Felber, P.A., Urvoy-Keller, G.: Hierarchical Peer-to-peer Systems. In: Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par), Klagenfurt, Austria (2003)
9. Yang, B., Garcia-Molina, H.: Designing a Super-Peer Network. In: IEEE International Conference on Data Engineering, 2003, San Jose, California (2003)
10. Mýzrak, A.T., Cheng, Y., Kumar, V., Savage, S.: Structured Superpeers: Leveraging Heterogeneity to Provide Constant-Time Lookup. In: The Third IEEE Workshop on Internet Applications, San Jose, California (2003)
11. Schlosser, M.T., Sintek, M., Decker, S., Nejdl, W.: HyperCuP - Hypercubes, Ontologies, and Efficient Search on Peer-to-Peer Networks. In Moro, G., Koubarakis, M., eds.: Agents and Peer-to-Peer Computing, First International Workshop, AP2PC 2002, Bologna, Italy, July, 2002, Revised and Invited Papers, Springer (2002) 112–124
12. Paepcke, A., Chang, C.K., Gravano, L., Baldonado, M.: The Stanford Digital Library Metadata Architecture. (1997)
13. Matthes, F., Niederée, C., Steffens, U.: C-Merge: A Tool for Policy-Based Merging of Resource Classifications. In: Research and Advanced Technology for Digital Libraries, Proceedings of the 5th European Conference, ECDL2001, Darmstadt, Germany. (2001)
14. Weber, R.: Chablis - Market Analysis of Digital Payment Systems. Institutsbericht, Technische Universitaet Muenchen, Institut fuer Informatik (1998)
15. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Löser, A.: Super-Peer-Based Routing Strategies for RDF-Based Peer-to-Peer Networks. Web Semantics: Science, Services and Agents on the World Wide Web **1 Issue 2** (2004) 177–186