A Federated Schema-Based Middleware Architecture for Hospital Information Systems

Wilhelm HASSELBRING
University of Dortmund, Dept. Computer Science, D-44221 Dortmund, Germany

Abstract. An important concern in distributed hospital information systems is the integration of information across heterogeneous subsystems. Consistent data replication is a central problem to be solved in this domain. The specific requirements and problems for integration of information within hospitals are discussed and a middleware architecture which has been designed according to these requirements is presented.

Communication servers are often deployed in hospitals to connect heterogeneous subsystems. This paper discusses some problems with this approach and presents a federated schema-based middleware architecture as an alternative solution for propagation of information updates across heterogeneous subsystems within hospitals.

1. Introduction

The purpose of a hospital information system (HIS) is to manage the information that health professionals need to perform their jobs effectively and efficiently [14]. Integrated systems which satisfy all requirements on information processing in hospitals are not available; even if some vendors promise this. Also, from an economical perspective, it is desirable to install a number of applications, which effectively support the specific needs of the individual organizational units of a hospital. Typical examples are systems for patient registration, admission, discharge and transfer, appointment scheduling, management of laboratory tests as well as decision support for medical treatment. This situation naturally leads to a collection of heterogeneous subsystems scattered across the hospital. To effectively support the work in hospitals, it is necessary to integrate these subsystems avoiding multiple entry of the same information and inconsistencies among information that is stored in different subsystems. A modular system of interoperable and cooperating subsystems, which retain their autonomy as far as reasonable, is required.

This paper discusses problems and solutions of propagation of information updates across heterogeneous subsystems within hospitals. The general structure of our presented architecture is based on the reference architecture for federated database systems [13] and adapted to the specific demands on integration of replicated information within hospital information systems. This architecture is the basis for algorithms that restore the integrity of replicated information when changes occur. A prototype implementation is outlined.

Section 2 discusses some standards for HISs that are relevant to the paper. The current state of the art in connecting subsystems within hospitals through communication servers is discussed in Section 3. Our federated middleware architecture for HISs is presented in Section 4. Section 5 sketches the implementation and Section 6 draws some conclusions.

Preprint of: W. Hasselbring: "A Federated Schema-Based Middleware Architecture for Hospital Information Systems". In: New Technologies in Hospital Information Systems (Proceedings Conference on Architectural Concepts for Hospital Information Systems), IOS Press, Amsterdam. pp. 24-29, 1997.

2. Standards for Hospital Information Systems

There exist several initiatives for standardizing healthcare systems. For instance, the Health Level 7 (HL7) protocol has been designed to standardize the data transfer within hospitals [6]. It is an application level protocol and so relates to level seven of the ISO/OSI-protocol hierarchy. HL7 covers various aspects of data exchange in HISs, e.g. admission, discharge and transfer of patients.

CORBA is the `Common Object Request Broker Architecture´ of the Object Management Group to standardize interoperability among heterogeneous hardware and software systems [9]. Simply stated, CORBA allows applications to communicate with one another no matter where they are located or who has designed them. CORBAmed is the Healthcare Special Interest Group for CORBA. In 1996, CORBAmed started the process of adopting standard interfaces for healthcare related objects by issuing a `request for information' which requested the healthcare and information technology industry to give the OMG guidance in its upcoming standardization efforts for CORBAmed.

It can be expected that the CORBAmed services within the CORBA framework will be an important standard for the interoperation among subsystems in hospitals [15]. However, in the current stage of standardization it is not clear what exactly CORBAmed will contribute to the challenge of integrating information within hospitals. The HL7 Special Interest Group for Object Brokering Technologies is mapping the forthcoming version 2.3 of HL7 to the Interface Definition Language of CORBA and version 3 of HL7 will be based on an object-oriented model of the underlying data [11]. So, we can expect a combination of CORBAmed and HL7 in the future.

Working Group 1 on `Healthcare Information Modelling and Medical Records' of the European Committee for Standardisation CEN TC 251 recently defined a general architecture which is intended as a basis both for the comparison, evolution and integration of existing systems as well as for the planning and high-level design of new open and modular systems of healthcare organizations [3]. The pre-standard mentions ``that, at a high level of abstraction, any healthcare organisation can be described by means of a federative model, as a set organisational components, mutually interacting for the effective delivery of services.'' [3]. Present projects under the umbrella of HANSA (an acronym for Healthcare Advanced Networked System Architecture) are demonstrating and transferring the technology of a series of European research projects such as the RICHE project [5]. The middleware in HANSA is the Distributed Healthcare Environment (DHE) which is built in accordance with the European standards.

3. Current State of the Art: Communication Servers

To connect heterogeneous subsystems in hospitals, central communication servers are often deployed [8]. The central communication server enables the subsystems to send messages to each other. Each subsystem is connected to the communication server and sends messages only to this server. The communication server determines the receivers and forwards the messages. Hospital communication servers usually support standard protocols such as HL7 (see Section 2) and the translation across different protocols when forwarding messages.

The requirement for building complex systems that combine heterogeneous subsystems can be met at the low level of *interconnectivity* or at the higher level of *interoperability* [10]. Interconnectivity simply supports system *communication*, while interoperability additionally supports systems to cooperate in the joint execution of tasks. A communication server primarily supports interconnectivity: the subsystems need to know where to send which messages, must take the initiative to update replicas and send messages for this purpose, and must be aware to receive messages from other systems and store the message contents appropriately in their local data bases.

With an integration that is based on a communication server, it is *not* known at the integration level at which sites data actually is *stored*. It is only known that data is *exchanged*. A communication server does not know whether data is replicated or just needed temporarily by a client for answering a user query.

4. A Schema-Based Middleware Architecture for HISs

This section presents our federated schema-based middleware architecture which has been designed according to the specific requirements of integrating replicated information among heterogeneous HISs. Below, we present an extended schema architecture and the associated algorithms that restore the integrity of replicated information when changes occur.

A database system (DBS) consists of a database management system and one or more databases that it manages. A federated DBS is an integration of autonomous database systems [13]. For federated DBSs, the traditional three-level schema architecture [4] is extended to support the dimensions of distribution, heterogeneity, and autonomy. The generally accepted reference architecture for schemas in federated DBSs is presented in [13]. It is rather obvious that this reference schema architecture has been designed primarily to support global access to the component DBSs, only secondarily to support integrity control. Therefore, we extended the reference schema architecture of [13] with import, export and import/export distinction for *public schemas* to adequately support the algorithms for changing replicated information [7].

Figure 1 illustrates an example scenario for changing replicas with our architecture. To explain Figure 1: A *local schema* is the conceptual schema of a component DBS which is expressed in the (native) data model of that component DBS. A *component schema* is a local schema transformed into the (common) data model of the federation layer. An *export* or *import* schema is derived from a component schema and defines an interface to the local data that is made available to the federation. A *federated schema* is the result of the integration of multiple public schemas, and thus provides a uniform interface for global applications. An *external schema* is a specific view on a federated schema or on a local schema which serves as a specific interface for applications (local or global). In Figure 1, no external schemas are shown.

The European Committee for Standardisation (see Section 2) proposes to structure the `Healthcare Information System Architecture' for HISs in three layers [3]. External schemas relate to the *application layer*. The schemas shown in Figure 1 relate to the *middleware layer*, which is responsible for supporting the cooperation of the different applications. As the basic technological infrastructure, the *bitways layer* is not related to the schema architecture. However, the bitways layer is *used* by the federation layer to carry out the integration of the subsystems, but this is not relevant at the schema level.

Specifying an import schema in our architecture is a subscription to change notifications for the corresponding data. Export schemas specify data to be exported to other systems. The schema architecture is the basis for algorithms that restore the integrity of replicated information when changes occur. For generality, we use the term *change* for insertion, deletion and update of data.

Figure 1 illustrates the passing of information up and down in the schema architecture by means of a small example. On the lower left corner in Figure 1, the local relational schema of a simple patient administration system consisting of two tables is indicated. This local schema is mapped to the corresponding component schema in the object-oriented canonical data model of the federation layer. Here, the foreign key relation between Patient and Invoice is transformed to an object-oriented relationship. The patient administration system exports the basic patient data through an export schema via the federated schema to the object-oriented research system as indicated by the dashed arrows. On the other side, the object-oriented research system exports information about materials to the relational patient administration system. Here, the research system stores *used materials*, which are transformed via the federated schema to *materials on invoices* for the administration system. When transmitting material, the patient identification is attached to identify the corresponding patient. For simplicity, we assume that a patient is registered in the administration system before the treatment starts for which materials are entered into the research system. We can only present an overview of our middleware architecture in this short paper. For a more detailed description refer to [7].

5. A Prototype Implementation

In a prototype implementation based on the middleware architecture, which has been presented in Section 4, a system that controls the results of therapies in angiopathy [16] is integrated with a system that manages patient data. The therapy control system has been implemented in cooperation of the University of Dortmund and the University Hospital Wuppertal [16] with the object-oriented database system O2 [1]. The patient data management system has been implemented with the relational database system Oracle [2].

As the infrastructure to overcome distribution and heterogeneity on the operating system level, we use a CORBA implementation [9]. Figure 2 illustrates the layers in the general architecture of the prototype implementation which is based on an Object Request Broker according to the CORBA architecture [9]. At the bottom, the meta data of the federation layer (federation graph for the schema architecture etc.) are stored in O2 [1]. Communication interfaces encapsulate CORBA services to send and receive operations which restore the integrity of replicated information when changes occur in a component DBS. For a discussion of using CORBA for implementing a federated DBS refer to [12].

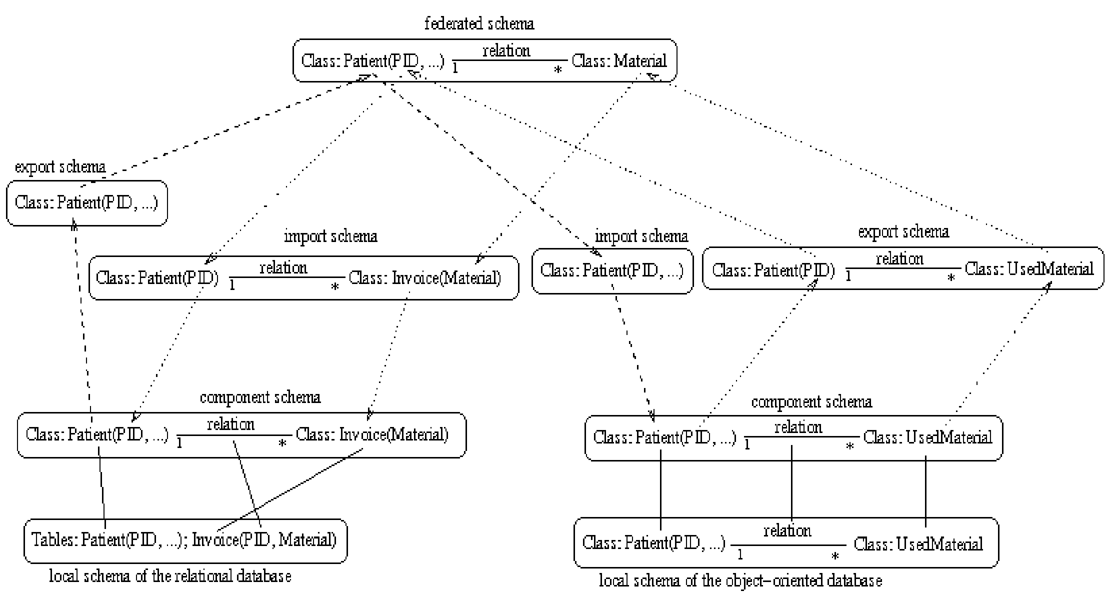

Figure 1: An example for exporting basic patient data by a relational patient administration system to an object-oriented research system (dashed arrows) and for exporting information about used materials by an object-oriented research system to the relational patient administration system (dotted arrows).

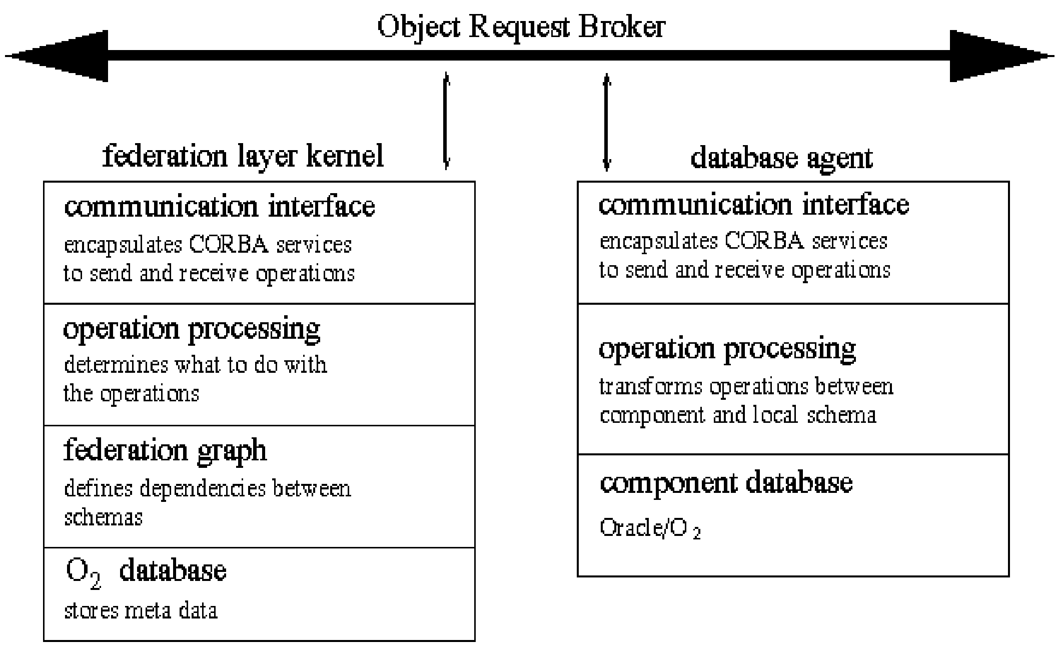

Figure 2: The layers in the CORBA-based architecture of our prototype implementation.

6. Conclusions

A federated schema-based middleware architecture which is based on the reference architecture for federated database systems [13] and adapted to the specific demands on integration of replicated information is presented. This architecture is the basis for associated algorithms that restore the integrity of replicated information when changes occur. The change algorithms are based on the schema architecture. This approach keeps these algorithms simple and the analysis of the dependencies within the schema architecture can be used to detect possibly infinite loops of change propagation or deadlocks. A prototype implementation which uses CORBA as the infrastructure to overcome distribution and heterogeneity is outlined.

Replication of information is the problem to be solved and not the suggested solution for the problems in todays HISs. We do not suggest replicating as much data as possible, but to integrate the replicated data across autonomous (legacy) systems which are already in use in the hospital. Here, you cannot avoid the replication of information without re-engineering the local systems.

In contrast to the current state of the art in connecting subsystems within hospitals through communication servers, a federated middleware whose data integration is on the basis of schema integration is capable of supporting subsystems to *interoperate*. Instead of enabling the subsystems to send messages for information exchange, the exchange of information is accomplished through updates of replicated data in subsystems by the federation layer, which *knows* the dependencies among replicas. This approach allows to analyze and optimize the data flow within hospitals on the middleware level.

References

[1] F. Bancilhon, C. Delobel, and P. Kanellakis. Building an Object-Oriented Database System: The Story of O2. Morgan Kaufman, 1992.

[2] M. Bronzite. Introduction to Oracle. McGraw-Hill, London, 1989.

[3] CEN TC/251 Working Group 1. Healthcare information system architecture part 1 (HISA): healthcare middleware layer. European prestandard, CEN European Committee for Standardisation, January 1997.

[4] C.J. Date. An introduction to database systems. Addison-Wesley, 6th edition, 1995.

[5] B. Frandji, J. Schot, M. Joubert, I. Soady, and A. Kilsdonk. The RICHE reference architecture. Medical Informatics, 19(1):1-11, 1994.

[6] W.E. Hammond. Health Level 7: A protocol for the interchange of healthcare data. In G.J.E. De Moor, C.J. McDonald, and J.N. van Goor, editors, Progress in Standardization in Health Care Informatics, pages 144-148. IOS Press, 1993.

[7] W. Hasselbring. Federated Integration of Replicated Information within Hospitals, International Journal on Digital Libraries, Springer-Verlag, 1997 (in press).

[8] M. Lange, N. Osada, H.-U. Prokosch, and W. Hasselbring. An approach to classify and compare communication servers. In Abstracts of the 41. GMDS-Jahrestagung, Bonn, Germany, September 1996. (in German).

[9] T.J. Mowbray and R. Zahavi. The Essential CORBA: Systems Integration Using Distributed Objects. Wiley, 1995.

[10] E. Pitoura, O. Bukhres, and A. Elmagarmid. Object orientation in multidatabase systems. ACM Computing Surveys, 27(2):141-195, June 1995.

[11] W. Rishel and J. Quinn. Software components, the clinical workstation and healthcare networks: How HL7 is helping you get there. In Proc. Healthcare Information and Management Systems Society's Annual Conference, Atlanta, Georgia, March 1996.

[12] S. Sander and W. Hasselbring. Realizing a federated database system based on the standards CORBA and ODMG-93. In Abstracts 2. Workshop ``Föderierte Datenbanken'', pages 45-50. University of Dortmund, December 1996. (in German).

[13] A. Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys, 22(3):183-236, 1990.

[14] E.H. Shortliffe, L.E. Perreault, L.M. Fagan, and G. Wiederhold, editors. Medical informatics: computer applications in health care. Addison-Wesley, 1990.

[15] R. Sokolowski. Integration services for clinical data exchange. OMG/CORBAmed DTF White Paper 97-01-01, Kurzweil Applied Intelligence, January 1997.

[16] I. Ullrich, W. Hasselbring, T. Jahnke, A. Röser, and A. Christmann. An object-oriented system for therapy control in angiopathy. In Abstracts of the 41. GMDS-Jahrestagung, Bonn, Germany, September 1996. (in German).