FROM BEDSIDE TO BENCH: A TRANSLATIONAL METAMODEL FOR HOSPITAL INFORMATION SYSTEMS IN ACADEMIC MEDICAL CENTERS

by

Andreas Biesdorf

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Biomedical Informatics

The University of Utah

December 2007

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a thesis submitted by

Andreas Biesdorf

This thesis has been read by each member of the following supervisory committee and by majority vote has been found to be satisfactory.

Chair: J[illegible] F. Hurdle

Lewis J. Frey

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

FINAL READING APPROVAL

To the Graduate Council of the University of Utah:

I have read the thesis of Andreas Biesdorf in its final form and have found that (1) its format, citations, and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the supervisory committee and is ready for submission to The Graduate School.

Date

hair: Supervisory Committee

Approved for the Major Department

Joyce A. Mitchell
Chair/Dean

Approved for the Graduate Council

David S. Chapman
Dean of The Graduate School

ABSTRACT

To conduct clinical research, data are needed from the clinical routine. Due to the complexity of today's Healthcare Information Systems (HIS), clinical researchers oftentimes struggle finding relevant information. As part of the reengineering efforts of the clinical data warehouse towards a translational enterprise data warehouse, a general methodology is developed to describe the complexity of current HIS in all domains, to guide researchers with a clinical question to the data they need, and thus to increase accessibility of clinical data for research.

An information-requirements analysis was conducted with clinical researchers and data warehouse experts in the domain of pharmacy as groundwork for a new translational metamodel. The results of this analysis were used to adapt the strategic HIS management metamodel $3LGM^2$ and to implement it in the domain of pharmacy.

The new translational metamodel consists of a domain-, a logical tool- and a data description-layer with interlayer relationships. Two further layers, added in a perpendicular way, give information about access and quality. This metamodel is implemented as a Web-based solution, providing modeling and browsing functionality.

The presented translational metadata solution shows a promising approach to the problem of clinical data access in research. Further research is needed to prove its applicability and usefulness in the daily routine.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Today's hospitals are highly heterogeneous institutions and consist of a multitude of organizational units. They execute a broad spectrum of functions ranging from basic administrative tasks up to very complex medical procedures [1], [2]. A high level of interoperability has already been reached and requires intensive internal and external communication among organizational units and healthcare professionals [3]. As integrated care is the goal, the amount of teamwork, and thus communication, is still expected to increase [4].

This complexity on the business side is reflected as well on the technical side: hospitals are grounded on a network of heterogeneous, interconnected systems that are built to support the daily work of all organizational units. The spectrum of complexity ranges from basic administrative software tools up to highly specialized expert systems [2]. These systems are not static because information technology still moves and changes at a high pace. As technical interoperability is the goal, all of these systems are integrated and use complex data exchange standards to communicate [1], [5], [6].

Hospitals with education and research settings face additional hurdles, which results in a further increase in complexity and heterogeneity on both the clinical and the technical side [7]. On the clinical side, the healthcare enterprises face the triple-burden of education, research, and patient treatment. On the technical side, institutions have to keep up with providing cutting-edge technology to attract patients and researchers [4].

In clinical research, this situation leads to a paradox: massive amounts of data are collected and technically available in data warehouses, but the level of complexity, the legal constraints, and oftentimes just missing knowledge about the availability of certain data limit accessibility for clinical researchers [8-10]. Even if access would be possible, the complexity of the environment makes it impossible to identify all potential confounders, hence making it difficult to judge the usability of the data for a particular study. As a solution of this problem, Information Technology (IT) departments commit to time-consuming data retrieval processes to support the researchers. To make these processes successful, two mental models of "data collection" have to be reconciled first (i.e., the technical model and the data user model).

It has become apparent that research institutions, such as university hospitals, face a critical challenge due to the complexity of their system-architecture. It was shown that these systems do not support researchers in an optimal way, because system design does not consider the way researchers conduct their work [11]. Despite the importance of data for research, its accessibility is still considered a major problem.

Just recently, the clinical research environment has become on the focus of the National Institutes of Health (NIH). As one result of their ongoing communications with the research community, the NIH found that basic, translational, and clinical researchers complained "that their interactions became more remote and difficult, that clinical research was increasingly less attractive to new investigators, and that clinician-scientists were moving away from patient-oriented research" [10]. Among other factors, the complexity of the clinical research environment was named as one of the main causes for this problem. Consequently the "Re-engineering of the Clinical Research Enterprise" [10] towards a translational research environment was announced as one of three major themes for the upcoming years [8], [10].

Much work has already been done to simplify the "translation" of results from basic science into clinical application, in other words to translate results from bench to beside (e.g., [12-14]). In the beginning of translational research, most research was focused on one direction, trying to simplify the translation of results from basic to clinical science [15], [16]. Several translational researchers observed this imbalance and stated that future research in biomedicine would need a bidirectional approach [15-18]. A loop between basic and clinical research could create synergetic effects that might finally lead to improved patient-care. Obviously, the academic institutions play a key role in this development, as they have to provide the appropriate infrastructure to conduct translational research. This includes not only internal structures that transcend the laboratory [15], [17], but also an infrastructure that allows connecting multiple sites throughout the U.S. [19].

To join these efforts, the Health Sciences Campus of the University of Utah developed its own plan for the transformation of structures towards a translational research environment. One of the key elements of this plan for transformation was the integration of the Clinical Enterprise Data Warehouse (EDW), with its more than 200 feeder systems, into a new Translational-Research Enterprise Data Warehouse (trEDW). It was thought that due to the combination of the EDW with basic science databases like the Utah Population Database and genetic research- and clinical trials databases, the exchange of information between clinical and basic research, and finally the conduct of research itself, would be simplified. To manage the complexity of this new system and to enable communication among institutions, to protect and regulate access to the data, and to keep the overview about past and ongoing projects, three layers were put around this new architecture: a Data Metamodel Layer, an IRB Security Layer, and a Project View Layer.

The work presented in this thesis was initiated to develop, implement, and evaluate a part of the Data Metamodel Layer of this new translational architecture in the domain of Pharmacy. The initial goals of the project were (1) to obtain a description of the current IT-structure of the EDW, (2) to support both technical users and researchers to find the data they need, and (3) to support the transformation from the EDW to a future trEDW.

2 BACKGROUND AND PRELIMINARY WORK

As described in the introduction, the major goal of this project was to provide a metadata-based solution to increase accessibility of data stored in the clinical EDW. The current technical infrastructure is very complex as it integrates more than 200 systems. A development and evaluation strategy was chosen based on recommendations in the literature [20], [21]. Figure 1 shows the waterfall approach, which was used for the development. The graph also contains estimates of the time that was needed to complete each step.

The projected started with a preparation stage, in which problems were identified and the goals were defined. Following this initial stage, a user- and an information-requirements analysis were conducted. These results were then used to develop a general metamodel and to implement a metadata tool. As the time estimates show, the tool development took more time than all of the other stages together. After this tool was available, actual metadata were collected to prepare the usability assessment and the evaluation. The next sections will provide further background information about the single steps. After introducing the general concept of metadata and its benefits, each of the steps in the waterfall model will be put in their context and described in detail.

2.1 Metadata

This section will introduce the concept of metadata and explain why metadata is considered a key driver for health information systems in academic medical centers.

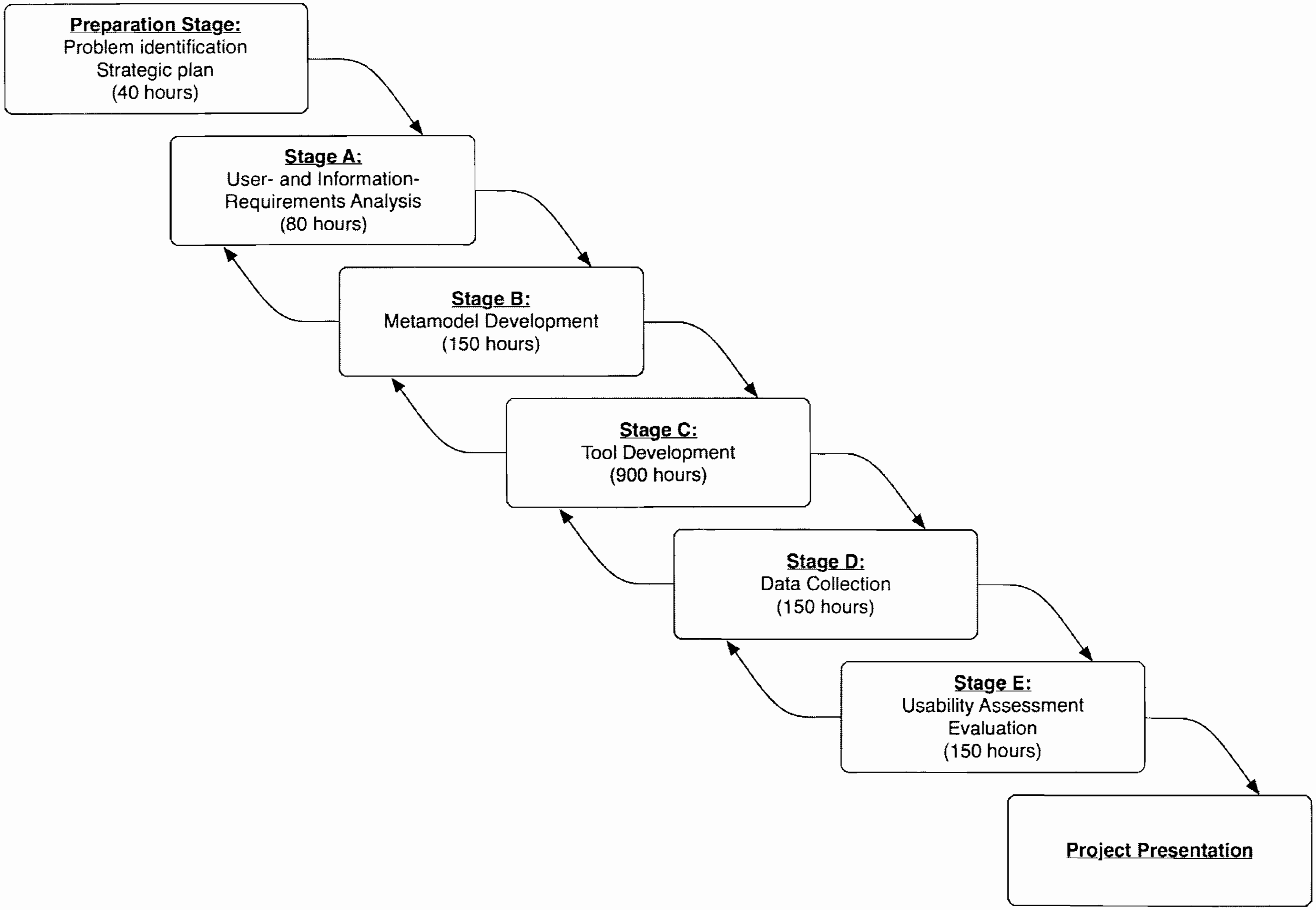

Figure 1: Waterfall model of the development and evaluation plan

Typical approaches, which might be suitable for the given problem in this project, are described and discussed.

2.1.1 Defining Metadata

The introduction of this thesis described the complexities that the clinical and technical parts of a hospital or research institution have to handle. As a consequence of this complexity, users struggle to find the information they need and to judge its quality, as they are not able to identify all data-affecting components of the system. If these systems were only complex but static, a comprehensive description, once compiled after the implementation of the system, could solve this problem. Unfortunately, the reality of these systems looks very different: The entire healthcare enterprise and all subsystems undergo rapid and continual change, and they cannot be described in a static form [22].

A very general definition of metadata is considered to be *data about data* [20], [21], [23]. This definition does not encapsulate the full scope of what metadata means in our context, so we need to use a much more comprehensive definition. Marco provides a suitable and comprehensive definition in the context of our problem:

> Metadata is all physical data (contained in software and other media) and knowledge (contained in employees and various media) from inside and outside an organization, including information about the physical data, technical and business processes, rules and constraints of the data, and structures of the data used by a corporation [20].

This definition includes the entire spectrum of metadata on the technical level. It is very important to note that not only technical metadata from software or databases are part of this description, but also knowledge of employees. We will use this definition as starting point for our modeling and development efforts.

2.1.2 The Need for Metadata in Healthcare

As the development of metadata solutions is a highly time-consuming and complex task, many people still question the "Return on Investment (ROI)" of this approach [21]. Looking at the metadata problem from a much broader scope, information about the structures of an HIS is important for several reasons.

First, today's systems are still considered inflexible and hard to integrate [20]. Keeping up a high pace of change in information technology and system architectures is considered key to persist in competition. Thus, hospitals are demanding more functionality and shorter development cycles from their IT departments to attain and maintain a competitive advantage in the healthcare system [4], [22]. System integration is a resource-intensive task with an effect in the long-term. From a short-term, naive perspective, it slows down innovation and uses valuable resources. As a consequence, many nonintegrated systems were and are still built. In the domain of healthcare, information exchange is very important, so information about the technical structure of such nonintegrated systems could help to understand them and to support the development of interfaces with other systems [4] .

Second, the growth of existing data warehouses and data marts is considered a major problem in today's healthcare IT systems. In larger healthcare institutions, much information about the current system, its size and speed of growth is required, even if only simple decisions concerning the IT infrastructure have to be made [20], [21].

Third, in times of integration of multiple disconnected systems across healthcare facilities, it is of high importance to know where information is found within the different systems and what this information means. In order to connect multiple systems and to build interfaces that exchange data, information about their location and meaning is a major requirement.

Fourth, successful research careers are usually marked by frequent institutional changes. Consequently, research institutions, like university hospitals, have a high employee-turnover, which increases the need for a comprehensive description of the institutional structures in place. Metadata can provide the semantic layer between the technical systems and the researcher in a way that it translates the technical terminology used in IT system into terms that are understandable to any user [11], [20], [21], [24].

Finally, many users experience a lack of trust of computerized data [20]. Oftentimes there is no information about their origin, how they were transformed, and about their quality. To develop technologies like decision support, metadata are needed, as the developer needs to know about available data sources and their quality. The quality of decision support can never be higher than the quality of the information it uses to make the decisions [20], [21], [24].

2.1.3 Metadata Solutions

During the last decades, several solutions for the metadata problem have been developed. The first approach to metadata, were the so-called data dictionaries. Data dictionaries are centralized repositories that contain information about data such as meaning, relational information, origin, or format [25]. For the most part, they were developed during the design-phase of the databases from and for developers. Data dictionaries were successful in environments that consisted of a single repository product, but they lost power as soon as several different repositories had to be integrated. Furthermore, as they lacked procedures to automatically create and update their information, they soon became out of date [21].

With the advent of Data Warehouses, the need for accurate documentation increased significantly. Many standalone metadata databases were built after it became apparent that the off-the-shelf metadata repositories did not provide the full spectrum of metadata documentation and as data warehouses made it easy to start collections. The data within these databases were still provided in a static way and contained data-dictionary-type of data with the addition of information, the developer thought to be of interest [21]. As the focus and scope of these standalone metadata stores remained on the technical level, and as these solutions were built and remained within IT departments, these data were still not accessible and usable to nontechnical users. In addition, those systems were usually maintained in a decentralized way through their developers [20].

Thus, standalone metadata stores were not very successful in the long run either, although they were built with a high amount of resources. Simple questions about the data inventory (e.g. “what data do we have?”) were difficult to answer with those tools [21]. As a consequence, vendors developed so-called “internal directories” with the ability to search for unstructured data within documents, source code, or other data sources. Again, these internal directories were developed as standalone systems without automatic updating functions, so that they were out of date soon. Finally, as many of those systems did not use standards, integration with other components or software products failed.

Still, there were also successful systems that provided access to metadata, which are still in use today. Key to their success were (1) standardized interfaces to allow integration with other metadata tools, (2) simple access for both technical and business users, and (3) automatic updating functions for at least a part of the data [20], [21], [24]. To allow access for the business users, initial solutions categorized their data and provided understandable user-interfaces, which were accessible from any place in the enterprise.

One recent improvement in the domain of metadata was the development of so-called "Metadata Solutions." These products were built based on the idea that metadata should be stored and kept where they were produced, instead of artificially moving them to a central place. Only a small subset of the entire metadata, called "common metadata," is collected in a central repository, which consists of some basic information to be used for searching, and which points to the real metadata sources. These metadata solutions are currently considered an efficient approach, as their metadata are accessible in a central place, but for the most part stored elsewhere. Still, metadata solutions require a considerable effort to create the general metadata model, to integrate the different sources, and to provide access for all user groups [21].

2.1.4 Healthcare-Specific Metadata Approaches

All the solutions described above were first developed in data warehousing. They were extended and adapted to many domains including business (e.g., [26]), geology (e.g., [27]), library information systems (e.g., [28]) and knowledge representation systems (e.g., [29]). Further research was initiated with the advent of ontologies and the semantic web (e.g., [30-32]).

For the domain of healthcare, no specific approach for a comprehensive and specialized Metadata Solution has been published yet. Still, there have been efforts in various other domains including the following:

- General clinical metamodels (e.g., [33-37])
- Organizational models (e.g., [38])
- HIS models (e.g., [1], [6], [39-42]).

There have also been many approaches for the development of architectural models (e.g. HISA [43], openEHR [44], or CorbaMed [45]), but none of them would be usable as basis for the development of a Metadata Solution. All of these models are highly specialized for a particular domain of application, but they are not usable without significant modification. Still, three approaches might be usable for the given problem. We will discuss these approaches more in detail:

- HL7 Reference Information Model (HL7 RIM) [46]
- ISO/IEC 11179 [47] and its application in caBIG [14], [48]
- The Three-Layered Graph-Based Metamodel (3LGM2) [1], [6]

2.1.5 HL7 Reference Information Model

The ultimate goal of the HL7 Reference Information Model is the sharing of consistent meaning beyond a local context [49]. This model evolved from a number of commercial and academic healthcare data models, and it accommodates the data elements defined in the current HL7 standard [4]. It is a collection of subject areas, scenarios, classes, attributes, use cases, actors, trigger events, and interactions that describe typical information and their interaction in healthcare. These "information packages" [49] and their descriptions can be exchanged using the HL7 information exchange standard.

The HL7 RIM is focused on "Entities" and their interactions: people, other living organisms, organizations, places, and manmade things. These entities can take on roles in particular contexts (e.g., patient, healthcare provider) and participate in actions. In these acts they take part as actors, targets, or resources to perform this act. Acts can be related to each other in a sequential and hierarchical way. The model is concerned with information interchange, not information storage. To allow information exchange, any entity is assigned a unique identifier and can be read, exchanged, and filed [46], [49].

Clearly, the HL7 RIM describes the information elements required to be understood between two information systems on a very high level. However, it does not describe all information recorded about these entities within a particular system. The standard does not include the description of information about the specific technical environment nor the specific data structures of any healthcare organization. As stated earlier, the HL7 RIM is built to describe information exchange, not information storage [49].

2.1.6 ISO/IEC 11179

The ISO/IEC 11179 standard describes the semantics of data, the representation of data, and the registration of their description. This standard provides a general model for the definition of data, which is independent of any domain. Its purpose is to promote the following [47]:

- Standardized description of data
- Common understanding of data across organizational elements and between organizations
- Reuse and standardization of data over time, space, and applications
- Harmonization and standardization of data within an organization and across

organizations

- Management of the components of data
- Reuse of the components of data.

As a consequence, the standard focuses on the technical description of the data. However, it explicitly excludes information about data models, application specifications, programming code, program plans, business plans, and business policies (see [47], page vi). Consequently, it does not contain information about the creation and modification of the data on the business level nor information about quality or access.

This standard has been implemented and extended in several projects. One example is the Cancer Biomedical Informatics Grid (caBIG) [48]. In this project, ISO/IEC 11179 has been used to support the semantic integration of data sources to simplify data exchange between several sites. The caBIG project uses and extends this standard for the definition of the so-called “Common Data Elements.” These data elements are used to ensure semantic interoperability between the different research sites. Common Data Elements consist of two elements: the data element concept and the value domain. The data element concept is based on the ISO/IEC 11179 standard, and provides general metadata for each element. The value domain describes how this value is represented in the particular setting. The data element concept is the same among all institutions that implement caBIG. As the implementations vary, the value domain might differ for every institution. Using the data element concept, information exchange among institutions is still possible. As a consequence, data remain interoperable among institutions, even if the concrete implementation of a particular data collection mechanism differs significantly.

2.1.7 The Three-Layer Graph-Based Metamodel

$3LGM^2$ is a specific metamodel for modeling hospital information systems. $3LGM^2$ stands for "three-layer graph-based metamodel" and was created to support the systematic management of HIS and to assess the quality of information processing in hospitals. It consists of a functional and a technical metamodel, which describe different aspects of the hospital information management on a Domain Layer, a Logical Tool Layer, and a Physical Tool Layer in UML notation.

2.1.7.1 Domain Layer

The Domain Layer describes the functions of a hospital. This is done independently of their concrete implementation in the setting. In order to apply a function, information about physical or virtual entities in the hospital is needed. This information is represented in entities. The information contained within those entities can be used or updated by a function. It is possible to (poly-) hierarchically structure functions and entities using the appropriate UML association. Organizational units execute functions. Figure 2 shows the domain layer metamodel, including functions, entities and a description of the type of access to the information contained within an entity. A typical instance of the domain layer can be the description of the treatment of a patient on a very general level as shown in Figure 3.

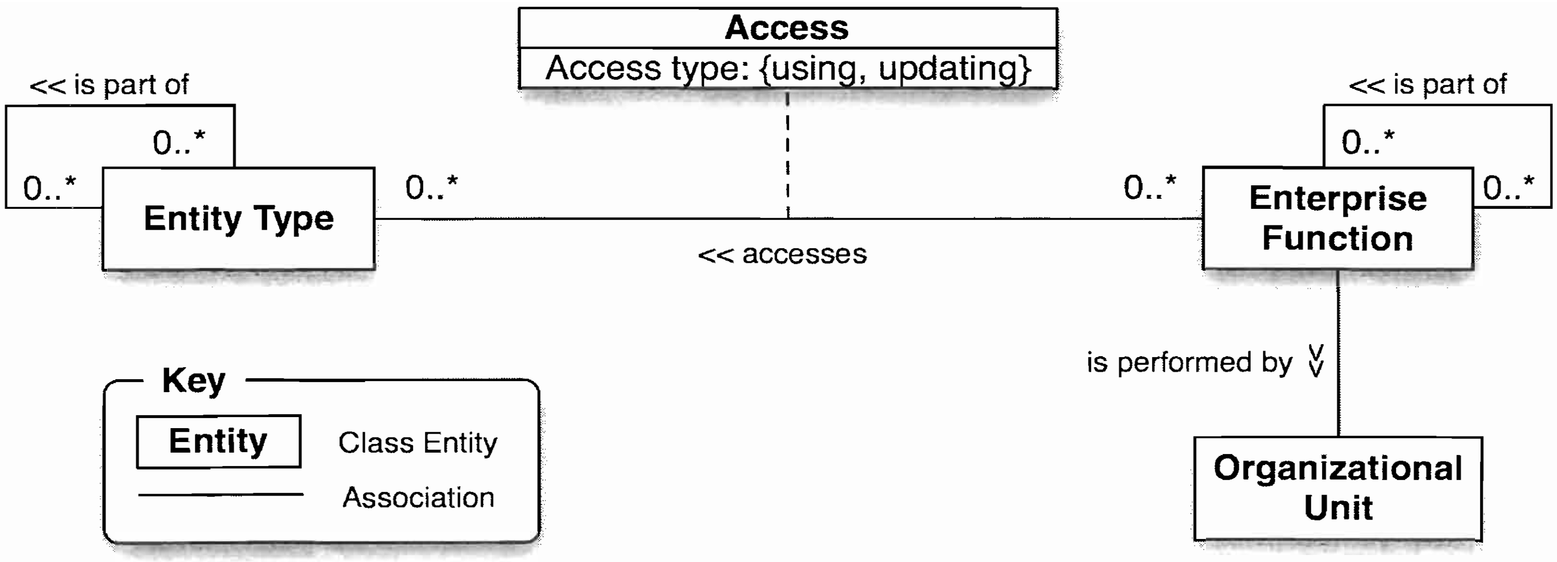

Figure 2: 3LGM^2 domain layer metamodel.
Reprinted by permission. Schattauer GmbH. Winter A, Brigl B, Wendt T. Modeling Hospital Information Systems (Part 1): The revised three-layer graph-based meta model 3LGM2. Methods Inf Med 2003; 42:544-51

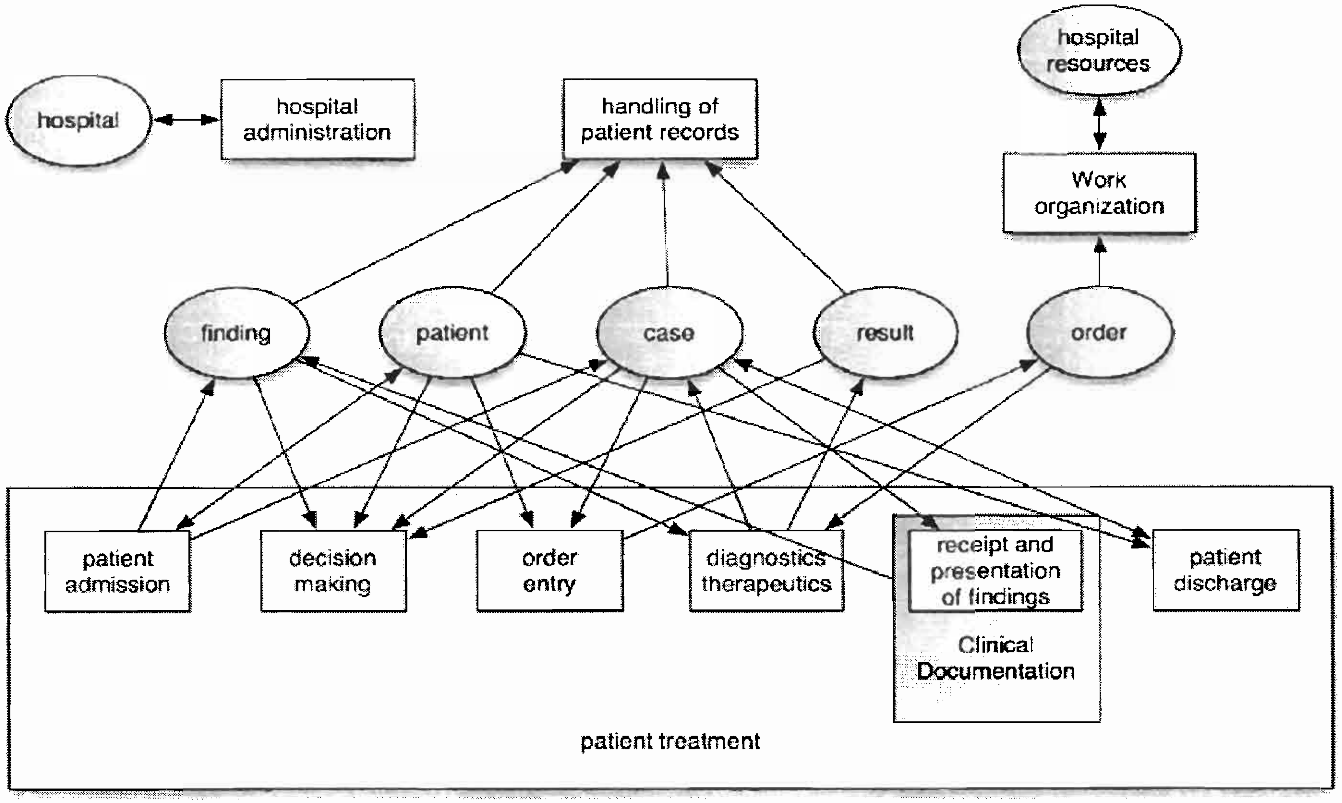

Figure 3: Instance of the domain layer.
Reprinted by permission. Schattauer GmbH. Winter A, Brigl B, Wendt T. Modeling Hospital Information Systems (Part 1): The revised three-layer graph-based meta model 3LGM². Methods Inf Med 2003; 42:544-51

2.1.7.2 Logical Tool Layer

The logical tool layer represents a metamodel of the application components used in a HIS. They support the functions on the domain layer and take over the task of processing, storage, and transportation of data. This layer integrates both the computer-based and paper-based application components. Adapted software products control computer-based application components, while paper-based application components are controlled by working plans. The computer-based application components are usually linked to a local database that is controlled by a database management system, while paper-based application components store their documents in document collections. The logical tool layer also includes communication interfaces to describe communication among components. It is important to note that those communication links are just virtual constructs whose physical representation will be done on the next layer. An instance of the logical tool layer is shown in Figure 4.

2.1.7.3 Physical Tool Layer

The physical tool layer represents the physical data processing components, which are used to realize the application components, described on the logical tool layer. Examples of physical data processing components are paper, telephones, and books but also all computer-based components like terminals or servers. This layer also represents the physical connection among those systems. Both physical and logical networks can be represented on the physical tool layer. An example of this layer is shown in Figure 5.

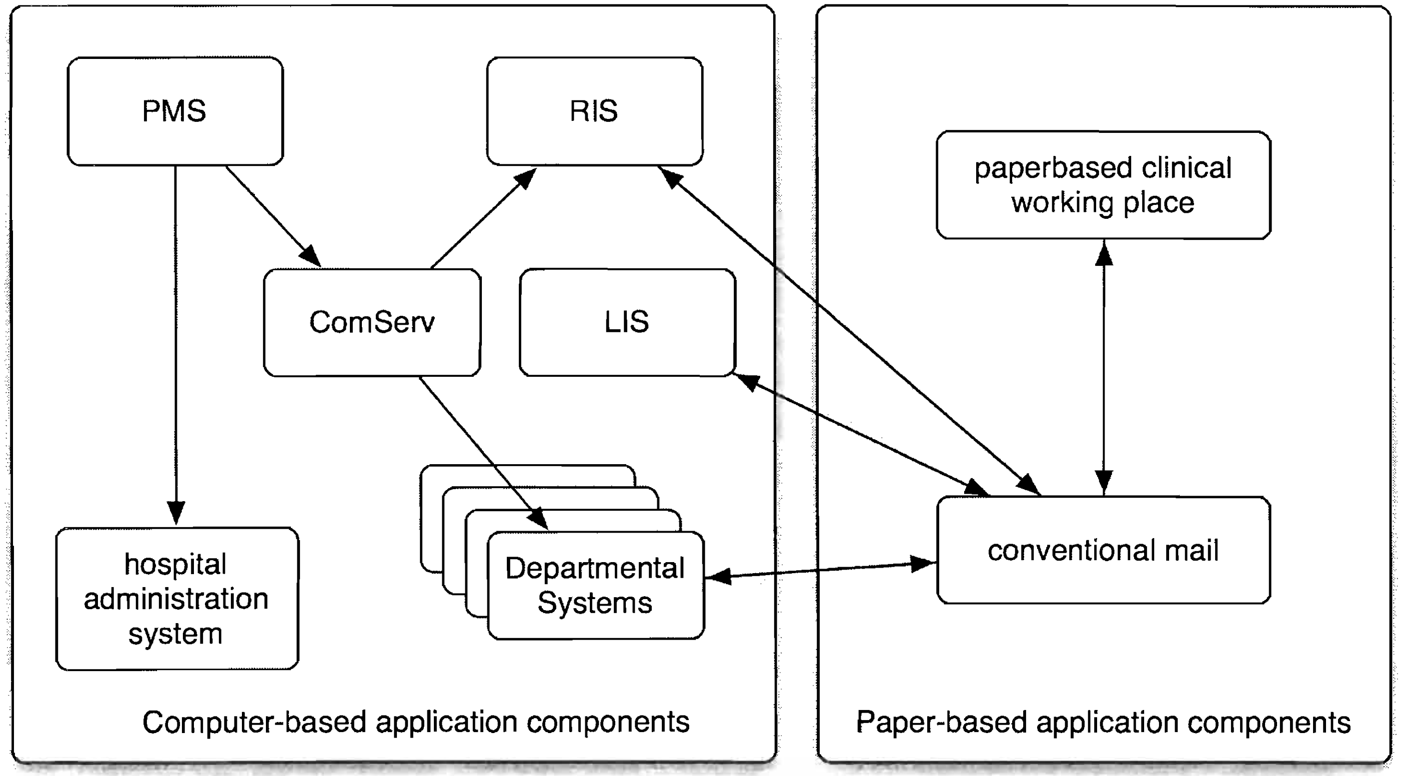

Figure 4: Instance of the logical tool layer
Reprinted by permission. Schattauer GmbH. Winter A, Brigl B, Wendt T. Modeling Hospital Information Systems (Part 1): The revised three-layer graph-based meta model $3LGM^2$. Methods Inf Med 2003; 42:544-51

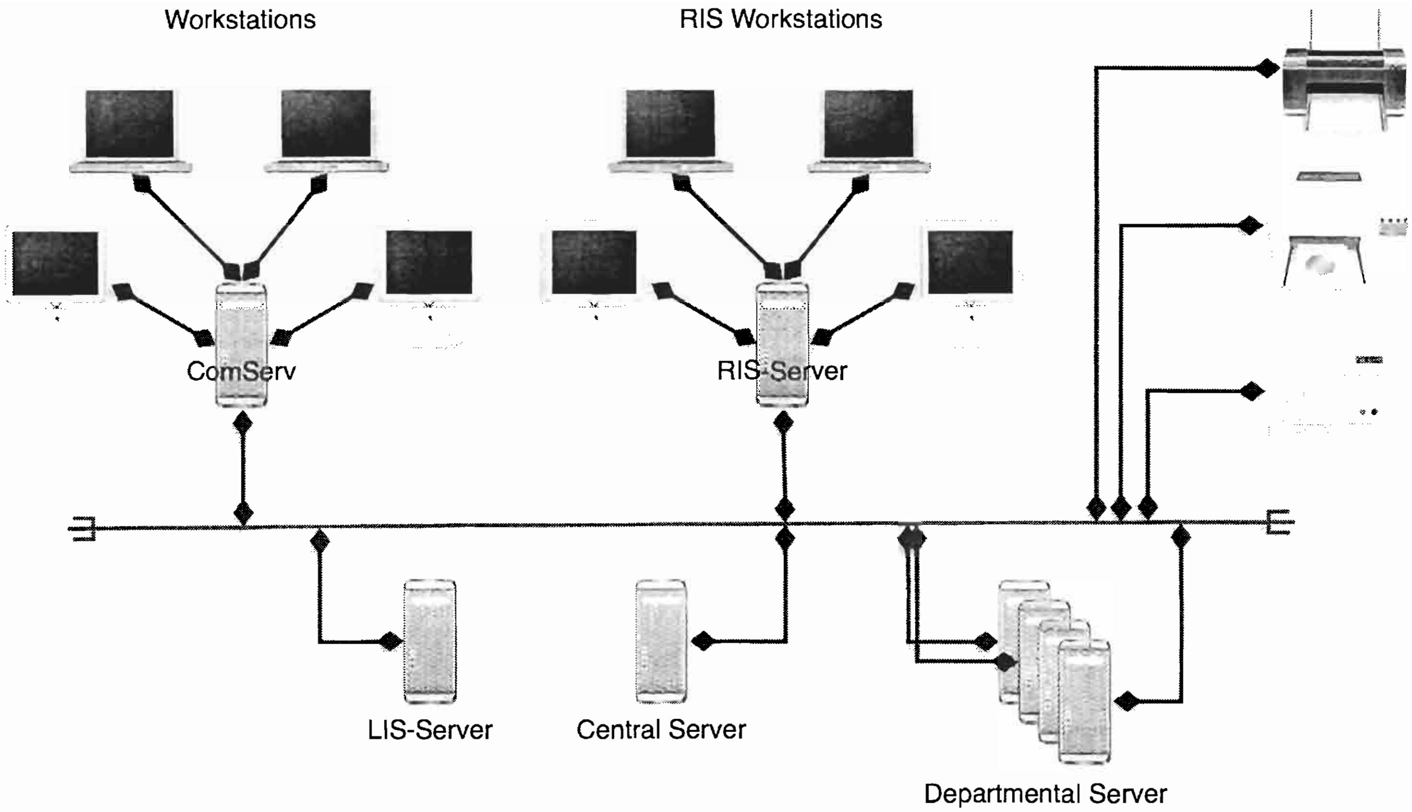

Figure 5: Instance of the physical tool layer

2.1.7.4 Interlayer Relationships

The concepts of the three layers are connected among each other. So-called inter-layer-relationships exist between the domain and logical tool layer as well between the logical tool and the physical tool layer. The central connection between the domain and logical tool layer is the relationship between functions and application components. Any function can be supported by one or several application components. Between the logical and the physical tool layer there exists a relation between application components and physical data components. This means that any application component may be installed on one or several data processing components. These relationships are important because they can be used to analyze and optimize the given structure. This model is able to show the impact of changes on one layer to other layers.

2.2 User- and Information-Requirements Analyses

A user and an information-requirements analysis were performed to get a clear description of the users, their information-needs, and a description of the context of the Metadata Solution to be developed. User studies need to be conducted as a first step towards building an understanding of users and their activities [50]. Typically, they are based on interviews, observations, or questionnaires and try to determine typical user groups, their background, knowledge, and mental models [51]. It is important to conduct those studies prior to designing a new system, as an inappropriate design might make users refuse the new product [52]. Involving the user during the development process improves the match between application functions and user requirements. In this project, the results of user and information-requirements analyses were important choosing and designing an appropriate metamodel, and to collect the requirements for a future tool.

2.2.1 The User Analysis

To identify potential users to include in the analysis, a modified version of the five-step procedure of Hackos and Redish was used [53]. First of all, a user-analysis team was assembled, composed of IT specialists analysts, who regularly interacted with users. The team consisted of one manager and two staff who were involved in the process of data and metadata collection on demand for clinical researchers. All three persons were aware of typical metadata users as they directly interacted with them. In a brainstorming session, the team created a list of potential users and user groups. All three members of the team considered themselves as users of the final Metadata Solution and were included. Five user-groups were defined and a list of fifteen representative users was assembled. A user/information-need matrix was created to derive an initial model of the user-community. Those characteristics were based on typical recent metadata requests and from the literature (e.g., [20], [21]). The assumptions of the user community and its characteristics were generalized based on their roles and profession and were tested during the interviews with the users as part of the information-requirements analysis.

The total list consisted of 15 users, with different backgrounds, roles, and positions within the University hospital. Based on assumptions about the users, they were assigned to at least one of five roles in the healthcare environment. As Table 1 shows, all users were classified as researcher, developer, student, database administrator (DBA), or member of the management. It was assumed that the characteristics of those five roles differed significantly.

Using this classification in five different user groups, characteristics and typical information needs of each group were discussed. Seven characterizing information needs

Table 1: Users of the Metadata Solution and Their Roles

#	Researcher	Developer	Student	DBA	Management
1	X				
2	X				
3	X				
4	X	X	X		
5	X				
6	X				
7		X			
8					X
9	X				X
10		X			
11					X
12				X	
13				X	
14	X				
15	X	X	X		

were considered important by the team. The list of metadata needs was compiled based on typical inquiries of users and information available from the literature.

As this matrix shows (see Table 2), nearly all users require a description of the data and information about the data processing. Information about data quality, business processes, and data lineage were important to only few user groups. Information needs about data access rights and metadata history were highly specific to single roles. In order to test the assumptions about the users, and in order to find out about typical tasks, thirteen persons from the initial list of 15 persons were selected for an interview. At least one interview was conducted for each role.

2.2.2 Information Requirements Analysis

Information requirements were analyzed in semi structured interviews by one interviewer. Thirteen interviews were conducted with users representing the user groups

Table 2: User/Information-Requirements Matrix

Information Requirement	Researcher	Developer	Student	DBA	Management
Data Quality	X				X
Data Processing Information	X	X	X	X	X
Data Description	X	X	X	X	
Business Process Information	X		X		X
Lineage of Data	X	X			
Data Access Rights				X	
Metadata History	X				

previously identified in the user analysis. Interviews started with an analysis of the technical knowledge and expertise. Using a modified version of the task identification technique from Vallacher and Wegner [54], the interviewer tried to get an idea of typical tasks to be accomplished during a normal working day. Using these tasks, information requirements were derived together with the interviewee. Finally, the interviews closed with an analysis of potential application fields of metadata in their work.

As the interviews showed, the user/information-need matrix was an appropriate representation of the information requirements of the different roles. Due to the small number of interviewees for each role, the variability of responses was high, but the classification of the user/information-need matrix still provided a good overall impression of their information requirements.

The information-requirements analysis showed that the information needs of the five groups were similar, but their way of asking for information differed significantly depending on their background. Users with a technical background asked for very detailed technical information of particular objects, while users without a technical background asked very general questions, starting on the business level, stepwise increasing the level of technical complexity in an iterative approach. It became apparent that these two groups had a different mental model of the concept "data" and a different approach to the task of data collection. This difference in the mental model was not linked to a specific role, but to the technical knowledge and expertise.

2.2.3 Applicable Results

As a consequence of the results of the user- and information-requirements analyses, it was decided to focus on two user-groups, which were separated based on the level of technical expertise. Two personae were developed to serve as exemplary users for the development:

2.2.3.1 Persona 1: The Clinical Researcher

Clinical researchers are the users with the highest metadata information requirements. For their research, they need information about which data are available about which business processes, and how they are entered and transformed before they are finally stored in the database. Furthermore, they need to know which data are available for a specific research question and where they are stored. Finally, it is of high value for them to know about the quality of data and possible problems. Their mental representation of the Metadata Solution is based on the real world and so are their initial questions. Thus, they try to start working with the Metadata Solution on the level of business processes. They expect to access the data with the help of a simple user interface instead of retrieving them based on a SQL query.

2.2.3.2 Persona 2: The Data Warehouse Analyst

Data warehouse analysts need very detailed information for the daily work. They are mostly interested in the description of data and their processing. Furthermore they need information about how the data are transformed through different processes and how they get to the fields where they are finally stored. It is also important for them to have a clear description of the access rights of the various users of the system. Although they have to react on problems related to data quality, they do not need this information for their daily work. Their mental model of data access is based on databases and they clearly understand where to find which information or how to figure out which information is available. A developer wants to have high control on the Metadata Solution. They are interested in a user interface that represents metadata in the way it is stored in the database. In some cases they also want to query the data directly with the help of SQL.

2.2.3.3 Final Requirements

These personae were built to support the final design of the metadata tool. To develop the metadata model, a list of typical metadata questions was compiled from the interviews and grouped into seven categories. Table 3 served as starting point to the metadata model development.

2.2.3.4 Consequences for the Metamodel Development

Based on this user- and information-requirements analysis, the Metadata Solutions and metamodels presented in section 2.1 were analyzed to see if they would be able to represent the required amount of information. It was found that none of the presented

Table 3: Typical Metadata-Related Questions

Category	Detailed Questions
Data availability	Which data are available?
	Where can they be found?
Data description and meaning	Meaning of a certain entity (e.g., table, column)
	Detailed description of certain entities
	Relations among entities (e.g., columns of a table)
	Data Coding (e.g., Cerner Multum or ICD-9)
Data lineage	Business related source of data entry in database
	Technical source (e.g., mapped from column)
Quality	Typical data quality measures (e.g., completeness, accuracy, timeliness)
	Data-entering role (e.g., physician or nurse)
	Known problems with data entering
	Data-entering context (e.g., patient visit)
	Processes affecting data-entering
	Software tool used for data-entering
	Noticeable problems in one of the following domains: • Business Processes (e.g., time pressure, overwhelmed by number of patients, maybe evaluation on process) • Tools (good/bad software) • Database-Environment (e.g., DB-crash, data loss, known legacy DBs with low quality)
History	Software tool or process changed? (e.g., new interface, new software, different process of patient visit)
	Impact of this change on the data in the DB? Better/worse data entering?
	In the DB: Has the vocabulary been changed? (e.g., ICD-9 to ICD-10)
Data transformation	Humans: who can access/change the data?
	System: which procedures access/change the data?
Data access	Can I access/change the data?
	Who can give me the data?

approaches would provide the full amount of information required, but that every approach could represent a subset of the required information. Still, there was consensus that the future model should not be built from scratch. After analyzing the given models, it was found that the "Three-Layer Graph-Based Metamodel [1], [6], [40], [41]", which combines a functional and technical metamodel, seemed to be applicable to the given problem. There were several reasons for this choice:

- **Simplicity:** Compared to the other approaches, the 3LGM2-approach is relatively simple to understand. As this project tries to simplify data access for users without a technical background, a model should be chosen that is understandable to all potential users.
- **Metamodel:** Unlike the other approaches, 3LGM2 is a metamodel that is built to describe the complexity of the clinical domain on all layers. Its main focus is to describe the complexity in a hospital information system to support strategic IT-management, hence it includes information about the technical and the operational domain. The goal of this project is to describe the complexity of a hospital and its information processing, and not to exchange between institutions.
- **Domain specificity:** The 3LGM2-approach is a healthcare-specific metamodeling approach. Unlike ISO/IEC 11179, it is clearly focused on the requirements of healthcare.
- **Different mental models:** As the user and information-requirements analysis showed, technical and nontechnical users have a significantly different mental representation of data access. The 3LGM2 metamodel has not been built to account for this difference, but it obviously separates the description of the technical and

business domain, which will be helpful for the development of an appropriate metamodel.

Although the 3LGM2 approach was found appropriate for the current problem, it still requires several changes to be completely sufficient to represent all information-requirements from the users. Consequently, the $3LGM^2$ metamodel will be remodeled and extended based on the user- and information-requirements analyses.

2.2.4 Discussion of $3LGM^2$

In general, it is important to recognize that the $3LGM^2$ metamodel combines a functional and a technical metamodel. Its main purpose is the management of health information systems.

The $3LGM^2$ metamodel provides information about all the functions that a hospital can perform. Although the basic definition of the domain layer focuses on functions related to information processing, descriptions can get very detailed and even be extended to functions not directly related to the information management [55], [56]. In addition, the model provides a view on how applications support those functions. Using interlayer relationships, it describes clearly how applications interact and how they affect the functions at the domain layer. This is particularly interesting if an application component stops working or is to be replaced, as the model provides a view of the consequences of this problem. On its third layer, the meta-model describes those components of the HIS that are finally used to enter data and which other physical tools are used to support the computer- and paper-based application components. Furthermore, this model allows judging the quality of the information processing [1]. Overall, this metamodel is useful to support the management and analysis of the HIS.

The standard $3LGM^2$ can be used to analyze whether the processing of information is done adequately. It does not provide information about the result of the processing, which means it does not includes measures for quality of the information processed. The model gives an overview of the overall system and where redundancies exist, but it does not point to concrete entities if a general function is executed with low speed or quality. If, for example, the function of "order entry" takes a lot of time to complete, it remains unclear whether the software is designed poorly, if the data quality is so poor that it cannot be used, or if the server is just slow. The only information $3LGM^2$ provides is that there is a problem related to the function of "order entry" and the tools that support this function, but no details about the underlying reason. It would be very useful to include this notion of quality in the tool. The interlayer relationships would allow one to analyze quality problems using a much broader view. Also, although $3LGM^2$ is able to provide information about whether whole components (e.g., a database) are redundant in a HIS, it is not able to determine whether the internal structure of those systems (e.g., a particular table in two databases or a particular column in two tables) is redundant and should be removed. In addition, this metamodel does not provide information about the details of how and where data is finally stored in the databases. It remains unclear which data are available for research or development.

As a consequence, the capabilities of the given model are limited for research, for database management, and for transformations on the data-layer (e.g. the transformation of the clinical EWD into a trEWD), but it seems to be adaptable with a reasonable amount of work to enable it answering those questions. In the following paper (section 3) such an extension is described.

3 FROM BEDSIDE TO BENCH: A TRANSLATIONAL METAMODEL FOR HOSPITAL INFORMATION SYSTEMS IN ACADEMIC MEDICAL CENTERS

To conduct clinical research, data are needed from the clinical routine. Due to the complexity of today's healthcare information systems, clinical researchers oftentimes struggle finding relevant information. As part of the reengineering efforts of a clinical data warehouse towards a translational enterprise data warehouse, a general methodology to describe the complexity of current HIS in all domains is developed, to guide researchers with a clinical question to the data they need, and thus to increase accessibility of clinical data for research. An information-requirements analysis was conducted with clinical researchers and data warehouse experts in the domain of pharmacy as groundwork for a new translational metamodel. The results of this analysis were used to adapt the strategic HIS management metamodel $3LGM^2$, and to implement it in the domain of pharmacy. The new translational metamodel consists of a domain-, a logical tool- and a data description-layer with interlayer relationships. Two further layers, added in a perpendicular way, give information about access and measure quality. This metamodel technology is implemented as a Web-based solution, providing modeling and browsing functionality. The presented translational metadata solution shows a promising approach to the problem of clinical data access in research. Further research is needed to prove its applicability and usefulness in the daily routine.

3.1 Introduction

Today's hospitals are highly heterogeneous institutions and consist of a multitude of organizational units. They execute a broad spectrum of functions ranging from basic administrative tasks up to very complex medical procedures [2]. A high level of interoperability has already been reached and requires intensive internal and external communication among organizational units and healthcare professionals [3]. As integrated care is the goal, the amount of teamwork, and thus communication, is still expected to increase [4].

This complexity on the business side is reflected as well on the technical side: hospitals are grounded on a network of heterogeneous, interconnected systems that are built to support the daily work of all organizational units. The spectrum of complexity ranges from basic administrative software tools up to highly specialized expert systems [2]. These systems are not static because information technology still moves and changes at a high pace. As technical interoperability is the goal, all of these systems are integrated and use complex data exchange standards to communicate [1], [5], [6].

Hospitals with education and research settings face additional hurdles, which results in a further increase in complexity and heterogeneity on both the clinical and the technical side [7]. On the clinical side, the healthcare enterprises face the triple-burden of education, research, and patient treatment. On the technical side, institutions have to keep up with providing cutting-edge technology to attract patients and researchers [4].

In clinical research, this situation leads to a paradox: massive amounts of data are collected and technically available in data warehouses, but the level of complexity, the legal constraints, and oftentimes just missing knowledge about the availability of certain data limits accessibility for clinical researchers [8-10]. Even if access would be possible, the complexity of the environment makes it impossible to identify all potential confounders, hence making it difficult to judge the usability of the data for a particular study. As a solution of this problem, IT departments commit to time-consuming data retrieval processes to support the researchers. To make these processes successful, two mental models of "data collection" have to be reconciled first (i.e., the technical model and the data user model).

It has become apparent that research institutions, such as university hospitals, face a critical challenge due to the complexity of their system-architecture. It was shown that these systems do not support researchers in an optimal way, because system design does not consider the way researchers conduct their work [11]. Despite the importance of data for research, their accessibility is still considered a major problem.

Just recently, the clinical research environment has become on the focus of the National Institutes of Health (NIH). As one result of their ongoing communications with the research community, the NIH found that basic, translational, and clinical researchers complained "that their interactions became more remote and difficult, that clinical research was increasingly less attractive to new investigators, and that clinician-scientists were moving away from patient-oriented research." Among other factors, the complexity of the clinical research environment was named as one of the main causes for this problem. Consequently the "Re-engineering of the Clinical Research Enterprise" towards a translational research environment was announced as one of three major themes for the upcoming years [8], [10].

The aim of this paper is to present a general methodology to describe the complexity of current health information systems (HIS) in all domains, to guide researchers with a clinical question down to the data they need, and thus to increase accessibility of clinical data.

3.2 Requirements

A user- and an information-requirements-analysis were conducted at the beginning of the project. For the user-analysis, a modified version of the five-step procedure of Hackos and Redish [53] was used and identified two major user groups, which could be discriminated by their level of technical expertise. Thirteen representative users were chosen and interviewed to identify typical tasks of their daily work, using a modified version of the task identification technique from Vallacher and Wegner [54]. These results were used to derive their information requirements. All information requirements were grouped into seven general themes: (1) data availability in relation to real-world processes and entities, (2) data description and meaning, (3) data lineage, (4) quality, including data quality, data processing quality and quality of the business processes related to data processing, (5) change-tracking for any HIS related change directly or indirectly affecting data or data processing, (6) data access, and (7) data transformation.

Summarizing the above, a highly comprehensive description of the data itself, its producing, processing, and storing environment, as well as relational information is needed to fulfill all information requirements. In other words, metadata about the business processes, the processing tools and the data itself is requested. Consequently, a translational metamodel needs to be developed that brings bedside data to the bench.

3.3 Approaches

For the domain of healthcare, no specific metamodeling-approach has been published yet to the extent described above. Still, there have been several approaches in the area of general healthcare metadata (e.g., [33-37]), organizational modeling (e.g., [38]), and HIS modeling (e.g., [1], [6], [39-42]). Other efforts, like HISA [43], openEHR [44], or CorbaMed [45] try to provide reference models, but no typical metamodels which would be usable to develop a Metadata Solution. HL7 RIM [46], [49], ISO/IEC 11179 [47], and 3LGM2 are approaches which could fulfil parts of the requirements. Based on the user- and information-requirements analysis, the presented Metadata Solutions and metamodels were analyzed if they would be able to represent the required amount of information. It was found that none of these approaches would provide the full amount of information that is required, but that every approach could represent a subset of the required information. Still, there was consensus that the future model should not be built from scratch. Due to reasons of simplicity, domain-specificity, and the architecture, it was decided to choose the 3LGM2 approach as starting point of further modeling efforts.

$3LGM^2$ considers the HIS as a socio-technical subsystem of a hospital [3], [22], and describes business and communication processes, logical tools and the technical subsystem on three different interconnected layers (see Figure 6): (1) The domain layer describes the hospital in terms of his enterprise functions and business processes. (2) A logical tool layer describes paper based and computer based application components and their communication, which support the enterprise functions on the domain layer. (3) Finally, the physical tool layer gives information all physical data processing

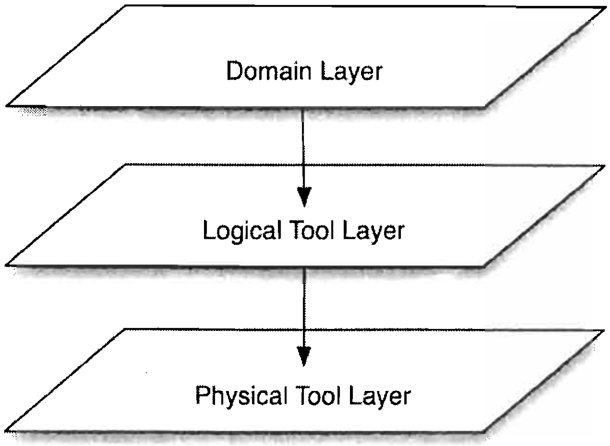

Figure 6: Architecture 3LGM^2

components, including computers, servers, switches and routers and how they are connected. The concepts of the three layers are connected to each other using interlayer relationships. The following section presents the modified 3LGM^2 metamodel.

3.4 Modification of 3LGM^2

For the new translational 3LGM^2 metamodel, it is important to keep in mind that unlike for the original model, the main goal is to increase accessibility of clinical data for researchers. Like the original metamodel, it is composed of three interconnected layers: a domain layer, a logical tool layer, and a data description layer. Furthermore, two layers are added in a perpendicular way describing quality and access. Figure 7 illustrates the general structure of the architecture and the positioning of the different layers.

3.4.1 Meta-Metamodel

To start the development of the final model, a higher level model was defined. This model was used as "language" to describe the actual metamodel. Such a model is

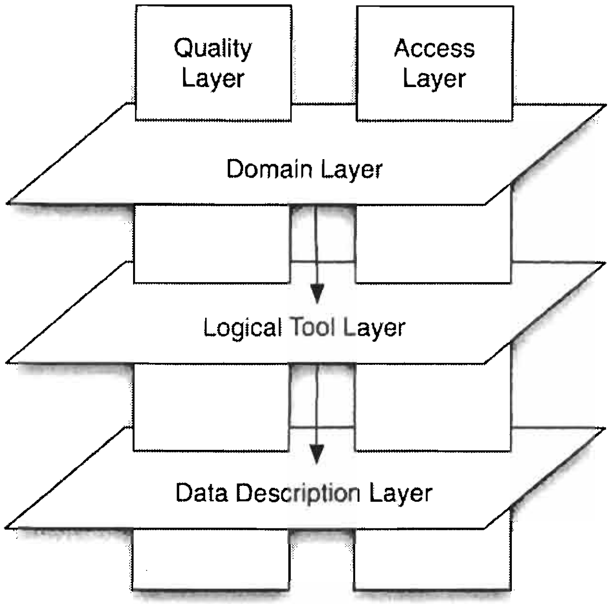

Figure 7: Architecture new metamodel

called meta-metamodel and allows accounting for a subset of the information requirements. Figure 8 shows the meta-metamodel, defines general attributes of any entity, including type and aliases, relationships among entities and historical relations.

3.4.2 Domain Layer

The Domain Layer describes the business processes and enterprise functions of a hospital, specific to the setting they are implemented in. The model does not need to be highly detailed, but should give a good understanding of the functions in order to provide an entry point for researchers in the domain to be described.

Enterprise functions are the centerpiece of this layer and can be organized in a sequential way, to represent an entire business process in several single steps. They are performed by *organizational units* and *access* information about entity types. Access to

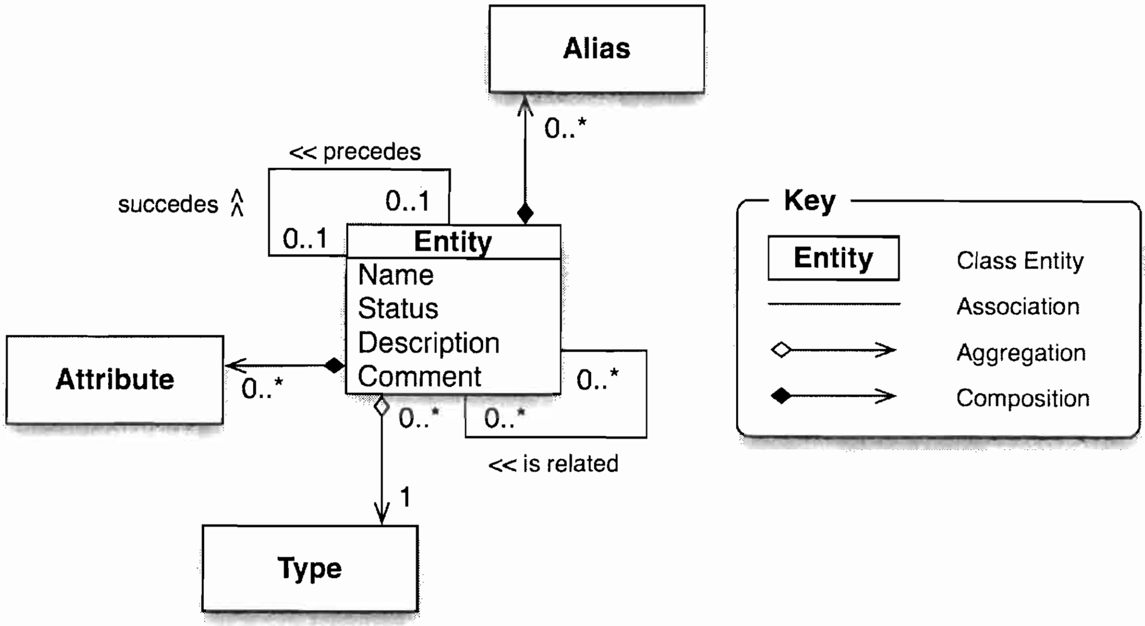

Figure 8: Meta-metamodel

an entity can either be using or updating. Entity types represent physical or virtual "things" in a hospital and can be structured in a hierarchical way. Figure 9 shows the class structure of the domain layer metamodel using the Unified Modeling Language (UML).

A simple instance of the domain layer is the enterprise function "order drug." The *organizational unit* "Physician" performs the *enterprise function* "order drug" by using information about the *entity type* "Patient" and by updating the *entity type* "Order."

3.4.3 Logical Tool Layer

On the logical tool layer, several simplifications have been made compared to the original 3LGM^2. It represents a metamodel of the *application components,* which process and store data about *entity types*. *Application components* support *enterprise functions* on

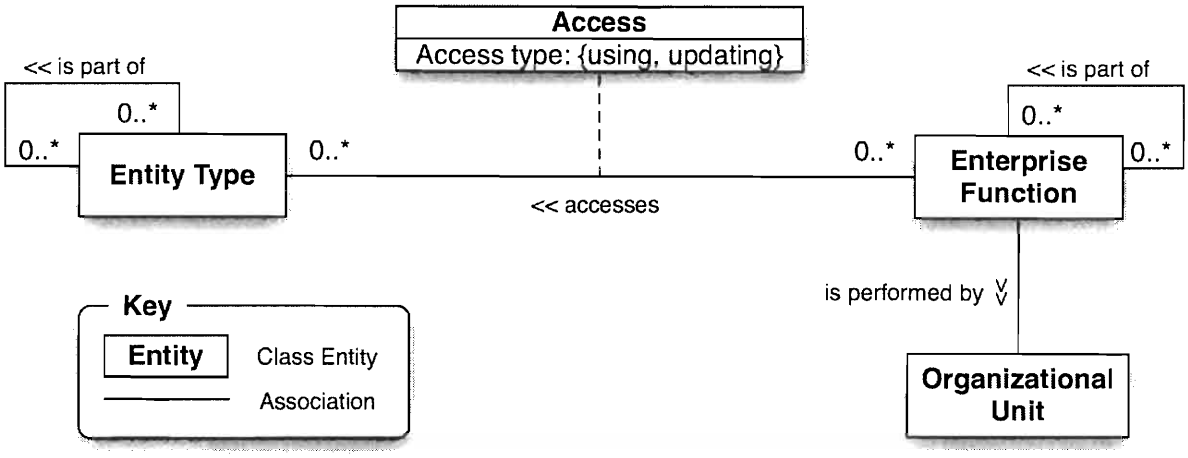

Figure 9: Class diagram for the domain layer (as in [1])
Reprinted by permission. Schattauer GmbH. Winter A, Brigl B, Wendt T. Modeling Hospital Information Systems (Part 1): The revised three-layer graph-based meta model 3LGM². Methods Inf Med 2003; 42:544-51

the domain layer and can either be computer based *software tools* or paper based *working plans*. *Software tools* can be connected to local or central *databases*, in which they store information about *entity types* in form of *dataset types*. These *dataset types* can consist of several *dataset elements*. *Dataset types* can be stored in one or multiple *tables*; *dataset elements* can be stored in *columns* of these *tables*. *Working Plans* can use several d*ocument types* to support an *enterprise function.* Figure 10 shows the UML class diagram of this layer. An example for a *software tool* could be "Cerner PharmNet," which stores information about the *entity type* "Drug Order" in form of a *dataset type* "Order" in the central *database* "PHARM_DB". This *dataset type* "Order" is stored in the table "ORDER" and consists of the *dataset elements* "Order Time," "Ordering Physician" and "Drug Name," which are stored in the appropriate columns.

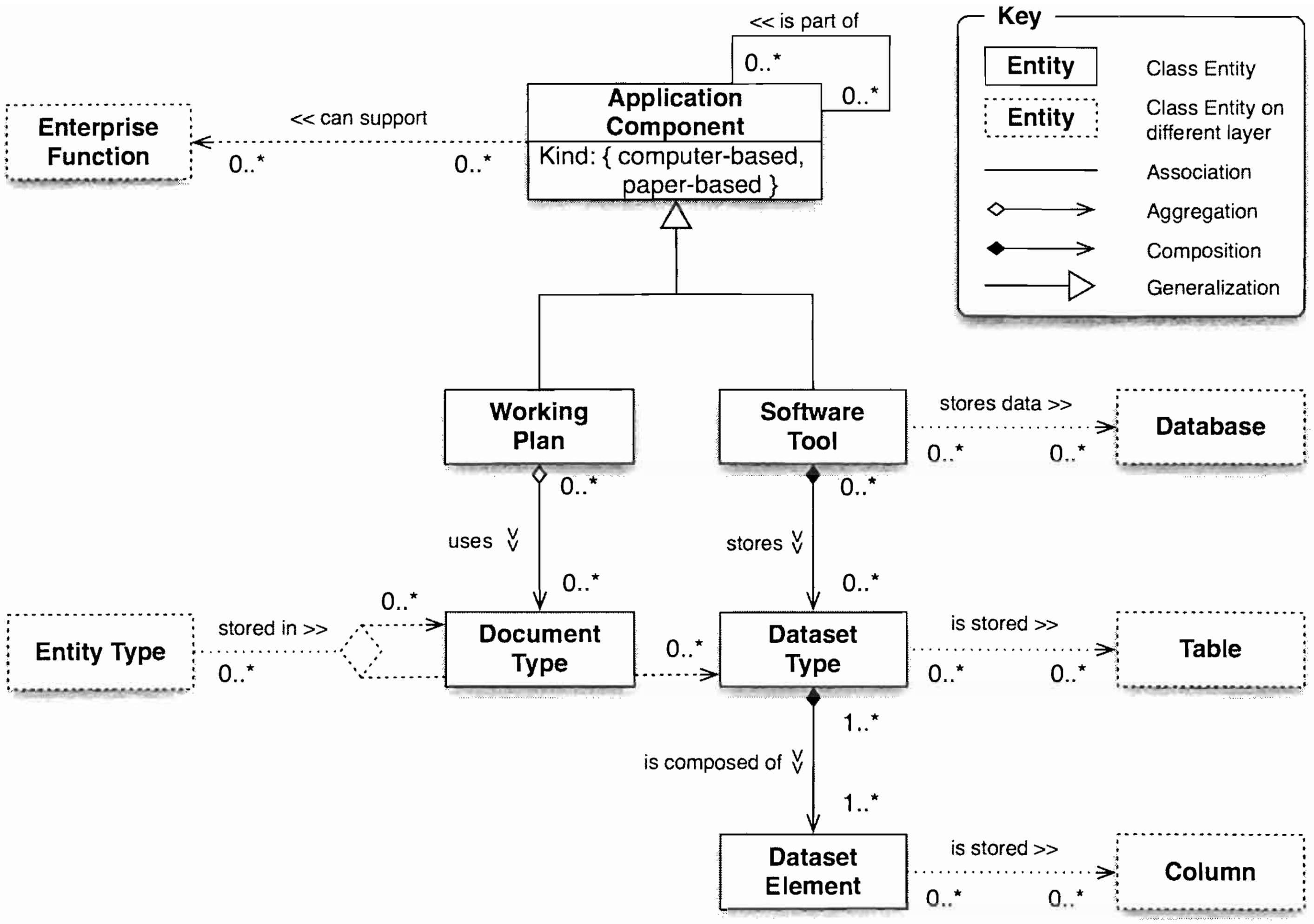

Figure 10: Class diagram of the logical tool layer

3.4.4 Data Description Layer

The data description layer contains information about where and how data about the entity types is finally stored. As mentioned before, *software tools* are connected to *databases,* which are controlled by *a database management system. Databases* contain *tables* and *views,* which can be organized in schemas. Tables have one or several *indices*, which consist of one or multiple *columns*. Both *tables* and *views* contain *columns*. A *vocabulary* can be assigned to a *column*. Updates, inserts and deletes on databases, schemas, tables and columns can fire *triggers*, which can start one or several *procedures*. *Procedures* can call other *procedures* and access *tables*, *views* and *columns*. Figure 11 shows the UML class diagram of this layer.

3.4.5 Quality and Access Layers

The quality and access layers cross the other layers in a perpendicular way. On the quality layer, *rules* can be assigned to *entities* described on any of the other layers to measure their *quality*. On the access layer, *users* can receive *privileges* to access *entities*.

The term "Quality" is to be used very loosely as it has different meanings on the different layers. On the domain and logical tool layer, quality is measured in form of evaluations of enterprise functions and application components. On these layers, the quality measurement tries to give a general impression about the quality of the data processing. On the data description layer, the term quality is related to typical data quality measures, e.g., accuracy, completeness, or timeliness.

The term "Access" is to be considered in a general way, too. On the domain layer, access describes which user group is allowed to execute certain enterprise functions. This allows the discrimination between user groups who have the right to order a drug

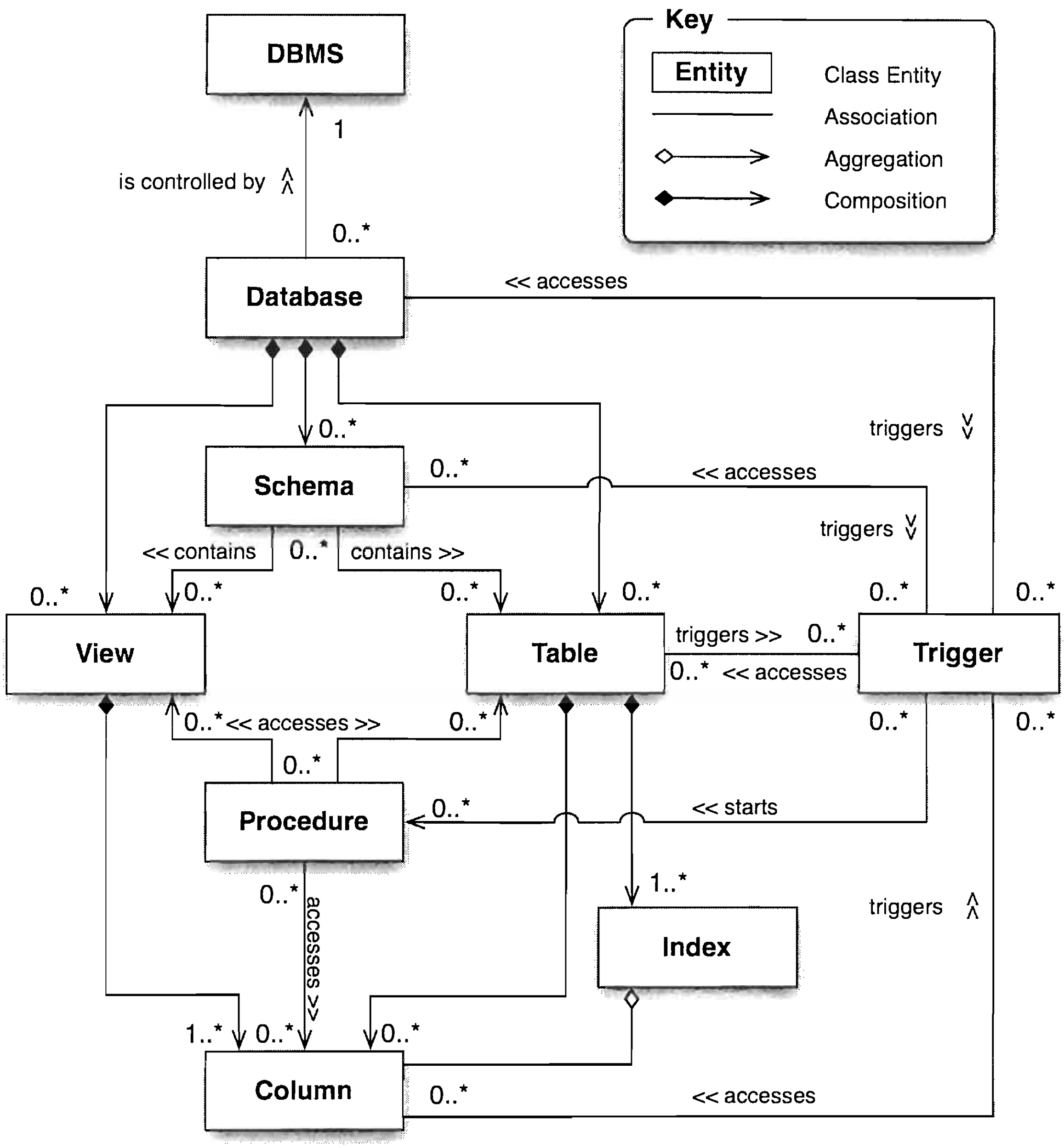

Figure 11: Class diagram of the data description layer

(e.g., physician) and those who do not (e.g., technician). The same concept of access applies to the logical tool layer, where it describes which user groups are allowed to access a particular tool. An example for this concept is that a physician is allowed to access the electronic medical record, whereas a technician would not be allowed to access all parts. On the data description layer, access is regulated as in normal database access. Figure 12 describes the metamodels of the quality and access layers.

3.4.6 Interlayer-Relationships

Entities on the different layers are interconnected through so-called interlayer-relationships. These relationships are important to represent the complexity of the healthcare environment and to allow bidirectional navigation. The domain layer and the

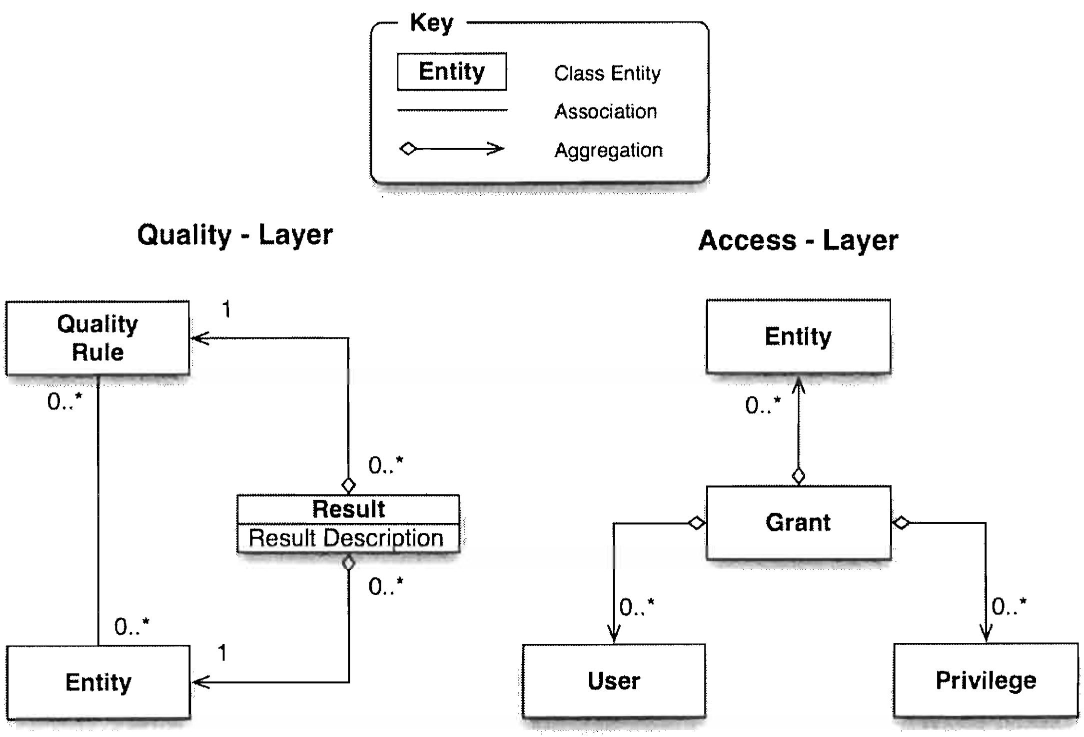

Figure 12: Class diagrams of the access- and quality-layers

logical tool layer are connected via two entities. *Enterprise functions* on this layer can be supported by one or several *application components*. To execute these *enterprise functions*, data from *entity types* is required, which can be stored in either a *document type* or a *dataset type* on the logical tool layer. The logical tool layer is connected to the data description layer by three entities. A *software tool* handles information *about entity types* internally in the form of *dataset types*. These *dataset types* are composed of *dataset elements*. *Software tools* are connected to *databases* where they store *dataset types* in *tables*. These *tables* are composed of several *columns*, which can store information about *dataset elements*.

3.5 Implementation

This theoretical model was implemented in the domain of pharmacy as prototypical Web-based metadata browsing and modeling tool. Real data were collected in the domain of pharmacy and loaded into the tool. The software also offers a data request module, which may improve communication among researchers and the IT department during the data request process. Technically, the tool is able to present information related to all information requirements (1) - (7), but initial usability studies have already shown that further work needs to be done to improve navigation and visualization. In addition, the metadata loading process is cumbersome and should be simplified.

3.6 Discussion

This paper presented a general metamodel, which is able to describe the complexity of current HIS in all domains and which provides the opportunity to guide researchers with a clinical question to the data they need. This model incorporates all requirements found in early user- and information-requirement-analyses and serves the overall goal to increase accessibility of clinical data. In addition, the metamodel could be implemented in form of a prototypical Web-based tool and loaded with data from the domain of pharmacy.

In order to apply the new metamodel in other domains than pharmacy, additional user-studies have to be conducted. As much research in pharmacy is connected to other clinical domains, it can be expected that the needs of a researcher in pharmacy are similar to those of a researcher in related domains. Still, this assumption should be analyzed scientifically to prove generalizability.

In addition to the work on the underlying metamodel, the metadata tool requires further effort to be usable in the daily routine. Besides improvements on the user interface, much work will still be required on the technical side to simplify the automated metadata collection process. The data collection process for the initial solution has already been complex and time-consuming, although only a small subset of all available metadata was loaded.

Although there is still much further research to be conducted to extend and implement it in other domains besides pharmacy, the current solution provides a good starting point for a comprehensive, translational Metadata Solution in healthcare. Once this solution has been implemented in the daily routine of research, it has the potential to create benefits in both short- and long-term. Considering the short-term perspective, the model might allow to speed up research, as data accessibility will be increased. In addition, the quality of research in general can be improved, as more information about project-related data is available (e.g., data quality).

From a long-term perspective, such a metadata solution might have the potential to shape translational research structures in academic medical centers. Due to the combination of a description of the infrastructure with measurements of its quality, metamodels become comparable if implemented in different settings. Assuming that several research institutions would implement the metamodel, best-practice structures or general recommendations for “good” translational research structures could be derived and published as translational reference metamodels.

4 IMPLEMENTATION AND EVALUATION

The presented metamodel shows a promising approach for the description of the complexities in healthcare with the ultimate goal of simplifying access to the clinical data. As part of the project, the tool was implemented and evaluated.

4.1 Metadata Browser Development

In a first step, the general requirements towards a Metadata Solution were discussed. First and foremost, the tool has to be highly accessible to all potential users in the institution. Second, it should be very flexible to be adaptable to the different needs of the users. It should allow future modification towards a user-specific tool, which provides different views on the metadata, or different sets of metadata to be accessed based on the roles or access-rights. Data need to be collected in a central database, but a connection to other data sources in other databases should be possible. Collected metadata should be stored in a secure form and only be accessible for users of the institution. If possible, only open-source software should be used for the development of the tool in order to allow a maximum of flexibility.

As a consequence of these requirements, the tool could have been developed either as a standalone application, which users would install locally on their personal computers, or as a Web-based tool. The Web-based solution was preferred, as it offered the highest level of accessibility and flexibility.

As a next step, the requirements for the Web-technologies to be used were analyzed. The tool has to present dynamic information, which might possibly change in real-time. Most of the tool's content will be text-based data, but some information might also be presented in form of graphs. Furthermore, users should not be required to install any software on their computer to use the tool. Finally, it was clear that data needed to be entered into and loaded from a database. For the basic user interface, clearly a combination of HTML, JavaScript (JS) and Cascading Style Sheets (CSS) was mandated. These three Web-programming languages will allow an appropriate design of the metadata tool. For the dynamic parts of the tool, it was decided to use a client-server approach based on Java Server Pages (JSP), JavaScript, and Ajax to allow asynchronous communication between local clients and the server. Work-intensive tasks are executed on the server-side, while all presentation-related tasks are performed on the client side. This separation was required, as the tool needs to conduct complicated searches that have to be done on a server. For the graphical presentation of the model, the graphviz-toolkit was chosen to be embedded in the application. The graphviz-toolkit is a visualization-toolkit for graphs, which will be used to create graphical models of the data at runtime.

In order to connect the Metadata-tool to the database, the object-relational mapping (ORM) solution hibernate was chosen. Hibernate simplifies database access by representing database entities as normal Java-objects, to simplify integration within the application. Using plain SQL-queries, it would have been hardly possible to complete the implementation of the tool, as the final database model consisted of about 88 interrelated tables. For the initial testing phase, MySQL was chosen as underlying database. As hibernate allows the connection to all common databases without significant changes in the code, this system can easily be connected to a commercial, and maybe faster database. For the Web-server it was decided to install the application within an Apache Tomcat environment, as it is a high-performance, open-source Web-server solution.

Ajax especially is a very elaborate technology, which requires substantial effort to be used within an application. In recent years, there have been several development projects to simplify the implementation of complex web-based applications. These "Ajax-frameworks" are based on a high-level programming language as Java, and (in theory) allow developing the application entirely within this language. Although these frameworks are not perfect, they still simplify the work to a significant extent. For this project, the "Google Web Toolkit" (GWT) was chosen as development framework. Since this toolkit is supported by a large company, it is very likely that the development on this open-source framework will continue in future and be still available for use in this project. Figure 13 shows the general architecture of the metadata tool.

The initial development focused on the implementation of four central functions: (1) metadata modeling, (2) metadata presentation, (3) metadata navigation, and (4) metadata searching. Those functions were chosen because they represent core-functionalities of typical metadata solutions (see [20], [21], [24]), and because they were required for the evaluation. After the first usability studies, a data requesting function was added in a prototypical way.

4.1.1.1 Metadata Modeling

This functionality allows creating, modifying and deleting of metadata entities. After entities have been created, they can be related to other entities within the limitations of the underlying metamodel.

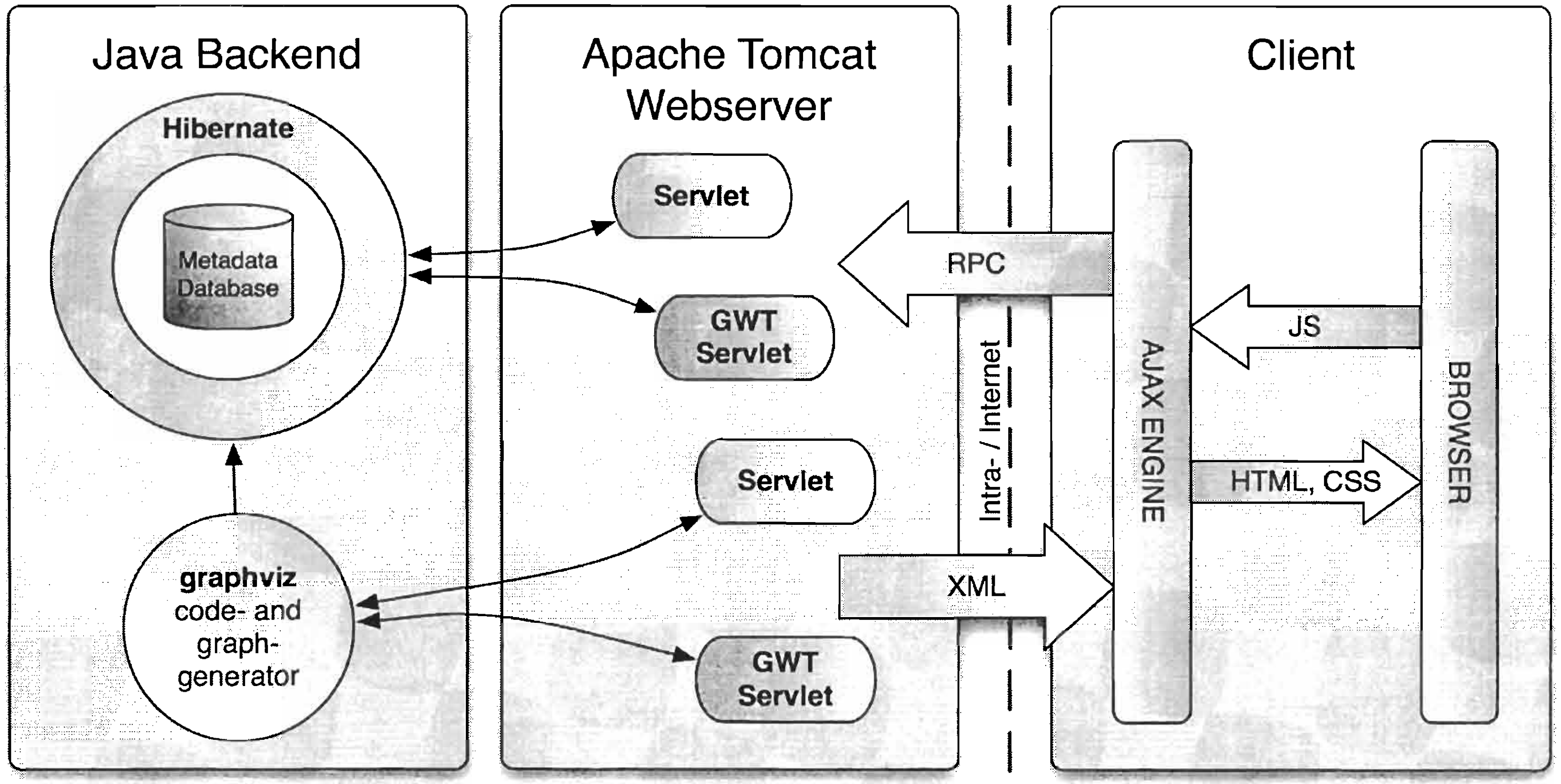

Figure 13: General architecture of the metadata tool

4.1.1.2 Metadata Presentation

In general, the metadata presentation is done in a text-based form. To illustrate relations, an additional graph-based presentation was implemented. These graphs are created at runtime, so that they always show the most recent state of the modeling activities.

4.1.1.3 Metadata Navigation

The metadata tool can be used as metadata browser. Like in other Web-based applications, the user can navigate from one entity to a related entity just by clicking on them. Furthermore, a navigation-bar and a navigation-tree were included.

4.1.1.4 Metadata Searching

The tool offers two search functions, which differ in their level of complexity. As many users requested a "Google-type" search-interface, it was decided to keep one interface very simple. The second interface allows more complex searches.

4.1.1.5 Metadata-Request Creation

The tool offers the opportunity to create data requests. If a certain dataset was found, it can be added in a simple way to a data request list, which can be submitted to a data warehouse analyst at the end of a metadata browsing session. This functionality is still a prototype.

4.2 Data Collection

As for most metadata solutions, the data collection process for the given tool needs a combination of manual data entry and automated data collection. For the data description layer, a highly automated data collection process could be installed. On the domain- or logical-tool layer, manual data collection processes were required. This data collection was not intended to be complete, but had to be comprehensive enough for presentation and evaluation. Figure 14 shows the data flow diagram of the current data collection processes.

For the domain- and logical tool-layer, a limited business analysis has been conducted to get an idea of the processes, entities and tools used in the domain of pharmacy. This manual data collection process transforms the collected data in a structured file that can be parsed by the metadata tool. A simple grammar was defined for the structure of those files. The simplified grammar is shown in Figure 15.

For the data description layer, several data collection procedures needed to be installed. The database structure was provided from the IT-department in form of the "Data Description Language (DDL)". A DDL-file contains information about the structure, procedures, triggers and functions of a database. As part of the tool, a parser for these files was developed that is able to load the structure and to filter information about triggers, procedures and functions. A more complex problem was the parsing of the code of the procedures, triggers and functions. As the metamodel allows showing the relations between these "active" (e.g. procedures, functions) and "static" entities (e.g. tables, columns), an analyzer had to be developed to derive these relations from the code.

For the data description layer, several data collection procedures needed to be

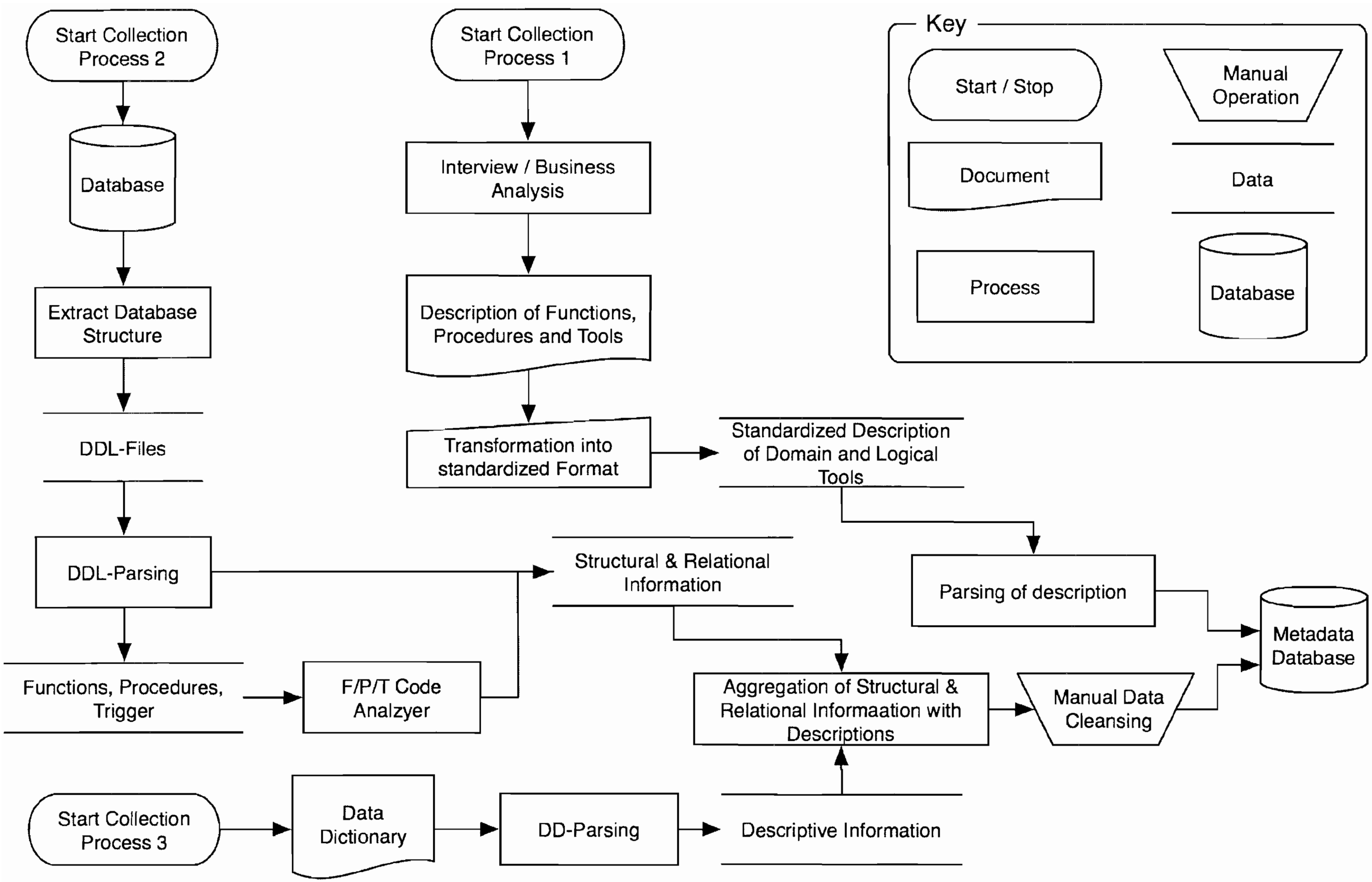

Figure 14: Current data collection process

```
<syntax> ::= <rule> | <rule> <syntax>
<rule> ::= <entitytype> <name> | <entitytype> <name> <rule> |
           <entitytype> <name> { "{" <description> { <entitytype>
            "{" <rule> "}" } "}" }
<description> ::= <description-name> "{" <description-text> "}"
<entitytype> ::= {"ENTERPRISE FUNCTION", "APPLICATION",
                    "ORGANIZATIONAL UNIT", "ENTITY TYPE"}

Notes:
------
<name> and <description-text> are just placeholders/labels for text
Other elements in <entitytype> have to be added for full model
```

Figure 15: Grammar in Exended Backus Naur Form

installed. The database structure was provided from the IT-department in form of the "Data Description Language (DDL)." A DDL-file contains information about the structure, procedures, triggers and functions of a database. As part of the tool, a parser for these files was developed that is able to load the structure and to filter information about triggers, procedures and functions. A more complex problem was the parsing of the code of the procedures, triggers and functions. As the metamodel allows showing the relations between these "active" (e.g., procedures, functions) and "static" entities (e.g., tables, columns), an analyzer had to be developed to derive these relations from the code.

All data loaded until now were static, and represented only single entities and the relations among them. In order to describe these entities more in detail, information from two different data dictionaries was loaded to enrich the current data in the metadata database. As all data dictionaries were only available in proprietary formats, additional parser needed to be built. Despite all these programming efforts, manual data cleansing was still needed to transform the data in a usable form. Finally, the database consisted of approximately 10,000 described and related metadata entries.

4.3 Usability Study

A simple usability study was conducted to analyze whether the tool is usable for the different user groups and to determine those areas where further development needs to be done. As this part of the project is still going on at the time of this writing, only preliminary results are presented.

4.3.1 Study Details

A representative group of users was selected. Literature recommends using three users per user group to be able finding 85% of the usability problems [57]. Consequently, three users from both user groups (three researchers and three users with technical background) were chosen to be included in the usability assessment. It was planned that participant receive a 15-minute one-on-one training to use the functions of the tool, but it turned out that this training usually took longer as users started to explore the tool and its functions. In addition, the general metamodel was explained to provide background about the general concepts.

For usability testing, users were asked to execute ten simple tasks, which were derived from the requirements in Table 3. These tasks were representative for typical problems users have to solve in their daily work. In addition, some more complicated questions were asked, which forced the users to navigate the entire tool and to use all functions. The developer tried to correct significant and obvious problems after each usability test.

4.3.2 Preliminary Usability Results

In general, users appreciated the idea of this translational metamodel and saw the potential of a comprehensive implementation. In addition, they liked the general structure and functionality of the Metadata browsing tool and realized the potential application fields in their daily work. Still, several problems were found during these first usability assessments.

Users considered the navigation as the biggest problem of the tool. It was very difficult for them to find particular information and to figure out how to get there just by browsing the structure of the model. As a consequence, a second navigation bar was introduced in the tool, which shows information about the navigation path. Still, navigation remained a problem, which apparently had to do with the general complexity of the metamodel.

The way of structuring information within this metamodel seems to be uncommon to the users. Although they got an initial introduction to it, they had difficulties to find the information they were searching for. Participants complained about the complexity of the model, as it was unclear to them on which layer they had to search for particular information. A solution for this problem might be further improvements in navigation, a better introduction to the model, or improvements in the naming of the concepts of the model.

Another problem for users was that they were not able to mark or save information, which would allow them to step back after further searches. This problem has already been solved by the introduction of a data-request panel and by the new navigation-bar. The data-request panel allows adding metadata entities, to which the user wants to have access. The navigation bar shows the single navigation steps of the user, which allows them to step back.

Furthermore, difficulties became apparent due to the naming of entities. There was a discrepancy between the understanding of terms that were used by the developer and those of the final users. Not surprisingly, this problem was much more critical for researchers than for technical users of the tool. Consequently, it is recommended to analyze the terms that especially researchers use to name particular entities. There might even be a need for two completely different naming schemes matching the two different mental models and names of technical users and researchers.

Another design-problem was that some buttons were not visible enough for the end-users. Even if those buttons were shown and explained to them earlier, users forgot about them and were not able to execute particular functions. This problem could be fixed by changing the color-scheme of the user interface, or by adding other design elements to focus the attention on these buttons.

Finally, the most critical issue was the problem of data availability. Metadata Solutions are known to be very data-intensive tools, and that the data collection process is a very time-consuming task. In this project, over a week was devoted to collect enough data from the domain of pharmacy to be able to test the tool. It turned out that the limitation on one single domain was a problem, as users exceeded the scope as soon as they tried to answer a question which was not directly related to pharmacy. As long as users had very clear assignments to search for data, they were able to browse the tool and find these data under the limitations described above. However, as soon as they were allowed to continue the search on their own, they were limited by the scope of the data collection (e.g., most users wanted to access medical record metadata, which was not part of Pharmacy metadata).

To solve this problem, additional data sources, like the epidemiology research databases, were loaded in the tool. In this particular case it turned out that data dictionaries were missing, so that a manual metadata entry would have been needed. In addition, different users required different additional data sources. Considering that the implementation of automated data loading processes is still a very complex and time-consuming task (see section 4.2), it was found unrealistic to collect enough data to allow a complete evaluation. This data-availability problem is a very strong and critical, but also a very typical limitation of the current tool.

4.4 Evaluation

Originally, it was planned to conduct a quantitative study, which tests if the metadata tool improves the quality of data-requests from researchers that are sent to the IT-department. To conduct such a study, the tool needs to offer large amounts of data from the domain of pharmacy and from all related domains that users could potentially need for their request. The usability study showed that the tool offers only limited amounts of data; consequently this evaluation could not be conducted as planned. As a consequence, it was decided to change this quantitative study into a more qualitative evaluation, using the think-aloud protocol and a usability questionnaire. In this study, the following questions need to be answered:

Q1: Does the tool enable users to answer their information requirements as listed in Table 3 on their own?

Q2: Do participants perceive the benefits of the metamodel during their searching

5 DISCUSSION AND FUTURE WORK

This research clearly demonstrates that data access in the domain of healthcare is a significant problem, and that there is a need for a metadata-supported approach. The user- and information-requirements analyses could identify the needs and help to shape the metamodel. The translational metamodel as a result of these analyses, together with the implementation as Metadata Solution, appears to be a promising approach, especially as it is the first systematic approach for this problem in the domain of healthcare. A successful implementation of this model in the daily routine might be able to create benefits both in the short- and in the long-term. Still, much more research will be required to turn this tool in a usable "solution." Since the evaluation of the tool is still going on at the time of this writing, the presented results are only a part of the eventual results that will accrue.

Besides the creation of reference-models, this new metamodel can also be used to simplify or optimize given structures in a systematical way. Using general network-analysis algorithms (e.g., Floyd Warshall Algorithm), this model might be able to reveal redundant structures in the clinical and technical area. For the former $3LGM^2$ metamodel, several algorithms have already been adapted and used for optimization of business processes and communication paths [42], [55], [56]. It might be possible to use these results within the new metamodel.

One of the central problems of this research might be the lack of generalizability. During the initial information-requirements analysis, the study was focused on activities?

Q3: After trying to answer these questions, how do they score the usability of the new system on the "System Usability Scale" (SUS)?

In order to answer Q1, participants receive a set of 20 questions, based on their requirements listed in Table 3. For every question it is noted whether the participant is able to answer the question or not, or if only a partial answer is given. To be able to find out whether the user understands and uses the benefits of the metamodel during these activities (Q2), participants are asked to explain what they are thinking during the process of navigating the tool and answering the questions. The results of these think-aloud sessions (see [50], [52], [53]) will provide insights in the way participants apply the metamodel in their activities, but also show further problems that might be caused by the design of the tool. As a side product, a list of terms can be compiled that are used by the participants, which might be used to improve the usability of both the model and the tool. Finally, these interviews will point out those data sources which are needed to turn the tool into a fully operational solution for pharmacists. In order to answer Q3, the 10-item "System Usability Scale" (SUS) [58] is used to get an impression of the overall usability of the tool. The results of this evaluation will be presented as part of the defense as the evaluation is currently still going on.

researchers from the domain of pharmacy. A real "translational" model would require the involvement of several domains, as cooperation among disciplines is one of the central goals. As a consequence, more research in domains other than pharmacy will be required.

In addition, the number of people interviewed at the beginning of the study would have been high enough to derive the metadata requirements of one homogeneous user group. Unfortunately, the group of interviewees was composed of two very distinct user-groups, so that the number of interviewees per group might have been too low. Still, as only IT-specialists have been interviewed that were involved in the metadata retrieval process, it can be assumed that the results are still representative. Clearly, it is recommended to increase the number of interviewees per group for further studies.

Initially, the project started with the goal to support the transformation of the clinical enterprise data warehouse into a translational research data warehouse and to increase accessibility of clinical data for research. Consequently, users with clinical and technical backgrounds were interviewed as groundwork for the metamodel. After these interviews were finished, the goals of the project shifted more towards a translational metamodel, strongly focusing on researchers as final users. If this tool is still used to support users in the IT-department, it might be reasonable to reevaluate the interface-design-decisions under consideration of the particular needs of technical users. If technical users required different or additional ways of access to the metadata, further changes in the tool or a complete redevelopment of the interface should be considered.

At the end of the tool-development process, several additional problems became apparent. As for most metadata-approaches, the data collection process was the central bottleneck of this Metadata Solution. The data collection was very time-consuming, although only few data were collected. If this data collection process is not improved significantly, the potential of this tool might be limited. This research provides several examples of how the data collection process could be automated and simplified on all levels. Still, this collection process requires a significant amount of additional research to be conducted before the tool can be installed in a real-life environment. In the current version, the tool requires the development of a separate parser for every data source, which obviously is not optimal. For the technical description, more research in the field of automated data collection is needed. The same applies for those parts of the model that need manual data collection, e.g., the domain layer. Here, significant efforts need to be done to collect all data related to processes and software tools.

Besides the problems mentioned above, the tool still offers much room for additional development. Work should be done to improve integration of the tool in the daily work of all users. As one example, the data requesting process could be fully integrated within this tool. Currently, the data are still requested using a paper-based form, which is faxed to the IT-department and manually entered in a data-request database. Usually, as most researchers are rather unclear about the concrete information they need, this request is improved in several iterations before the data are released. Already now, the tool provides a prototypical function to select requested data elements during a metadata session. More work in this domain could replace and maybe simplify the current data request process completely.

As Metadata Solutions on this comprehensive level will always need a combination of manual and automated data collection, it might be reasonable to create an institutional metadata steering committee, which is composed of members from the technical and the clinical domain. This steering committee needs to organize the efforts to keep the metadata up-to-date, and should coordinate the work of metadata user-groups on a domain-specific level (e.g., pharmacy). Domain-specific metadata user-groups need to be composed of members with technical, clinical, and scientific background to find out which data needs to be collected and kept up to date. The general structure is illustrated in Figure 16.

Both the interviews and the current efforts of the NIH show that in times of exponentially growing data volumes, metadata are considered a key factor for conducting "good" biomedical research. If it is true that metadata plays such a key-role in the daily work of a researcher, biomedical scientists might learn some lessons from software engineers: In computer science, documentation is a central component for any successful development effort. Consequently, software engineers deliver large development projects with up to three different types of documentation: one documentation for the end-users, a separate documentation for the developers, and a third documentation for maintenance. These different types of documentation are nothing else than metadata, and their creation is considered part of the development process. This amount of documentation and the associated costs are required in order to deal with the internal complexities of these products, to allow communication with other products, and to enable developers to keep this tool up-to-date in an environment that changes at a high pace.

The work of a software engineer and the work of a biomedical researcher are clearly very different, but apparently, some characteristics of their fields are similar. If a hospital was considered similar to a complex software project, the amount and quality of documentation would surely be higher. If there was a better documentation, conducting

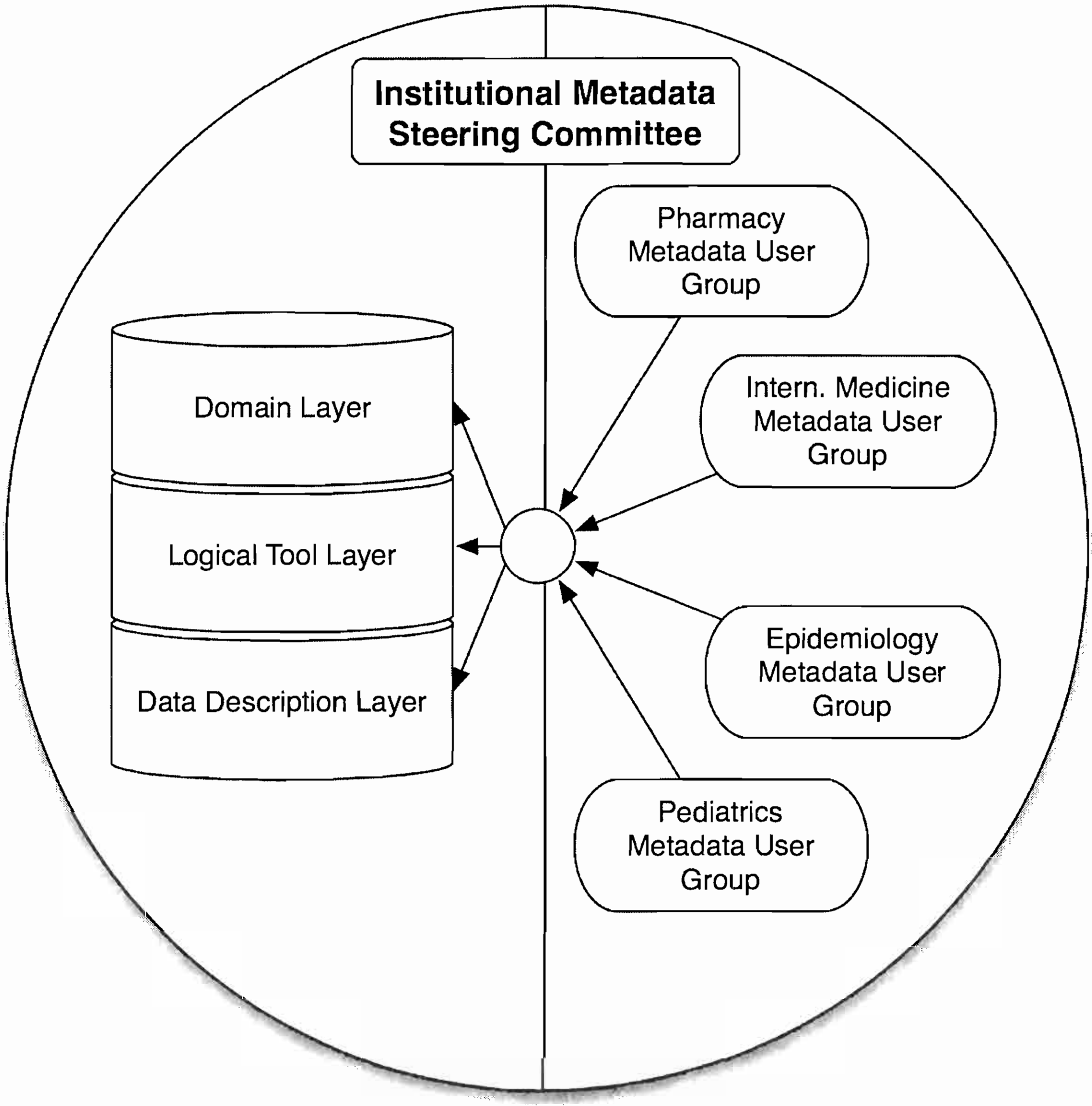

Figure 16: Institutional Metadata Steering Committee

translational research would not be as cumbersome as it is right now. As the level of complexity in healthcare and the related problems continue to grow in all domains, documentation in form of metadata will become a central component for successful change. This project does not claim to provide this documentation in healthcare on such a comprehensive level. Still, it might be considered a promising start in this direction.

REFERENCES

[1] A. Winter, B. Brigl, and T. Wendt, "Modeling hospital information systems. Part 1: The revised three-layer graph-based meta model 3LGM2," *Methods Inf Med,* vol. 42, pp. 544-51, 2003.

[2] A. Winter, "Health information systems," in *Yearbook of Medical Informatics*, R. Haux and C. Kullikowski, Eds. Stuttgart: Schattauer, 2004, pp. 359-368.

[3] R. L. Wears and M. Berg, "Computer technology and clinical work: still waiting for Godot," *JAMA,* vol. 293, pp. 1261-3, Mar 9 2005.

[4] E. H. Shortliffe and J. J. Cimino, *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*. New York, NY: Springer; 3rd ed. edition, 2006.

[5] D. Blumenthal and J. P. Glaser, "Information technology comes to medicine," *N Engl J Med,* vol. 356, pp. 2527-34, Jun 14 2007.

[6] A. Winter and R. Haux, "A three-level graph-based model for the management of hospital information systems," *Methods Inf Med,* vol. 34, pp. 378-96, Sep 1995.

[7] K. A. Kuhn and D. A. Giuse, "From hospital information systems to health information systems. Problems, challenges, perspectives," *Methods Inf Med,* vol. 40, pp. 275-87, 2001.

[8] V. L. Contie, "Clinical and Translational Science - Speeding the translation of medical discovery into enhanced patient care. ," *NCRR Reporter,* pp. 3-5, 2006.

[9] J. L. Goldstein and M. S. Brown, "The clinical investigator: bewitched, bothered, and bewildered--but still beloved," *J Clin Invest,* vol. 99, pp. 2803-12, Jun 15 1997.

[10] E. A. Zerhouni, "Translational and clinical science--time for a new vision," *N Engl J Med,* vol. 353, pp. 1621-3, Oct 13 2005.

[11] S. Deutsch, "Tomorrow's successful research organizations face a critical challenge," *Scientific Computing,* pp. 31-33, 2007.

[12] J. Sakamoto and S. Morita, "From translational research to a large randomized clinical trial: a long and streanuous way from bench to bedside," *Nagoya J Med Sci,* vol. 69, pp. 9-16, Jan 2007.

[13] M. Zaninotto, M. M. Mion, E. Novello, S. Altinier, and M. Plebani, "New biochemical markers: from bench to bedside," *Clin Chim Acta,* vol. 381, pp. 14-20, May 2007.

[14] X. Zhang, L. Li, D. Wei, Y. Yap, and F. Chen, "Moving cancer diagnostics from bench to bedside," *Trends Biotechnol,* vol. 25, pp. 166-73, Apr 2007.

[15] S. P. Mankoff, C. Brander, S. Ferrone, and F. M. Marincola, "Lost in translation: obstacles to translational medicine," *J Transl Med,* vol. 2, p. 14, May 18 2004.

[16] K. C. Sonntag, "Implementations of translational medicine," *J Transl Med,* vol. 3, p. 33, Aug 30 2005.

[17] H. Hoerig, E. Marincola, and F. M. Marincola, "Obstacles and opportunities in translational research," *Nature Medicine,* vol. 11, pp. 705-708, 2005.

[18] J. M. Westfall, J. Mold, and L. Fagnan, "Practice-based research--"Blue Highways" on the NIH roadmap," *JAMA,* vol. 297, pp. 403-6, Jan 24 2007.

[19] E. A. Zerhouni, "US biomedical research: basic, translational, and clinical sciences," *JAMA,* vol. 294, pp. 1352-8, Sep 21 2005.

[20] D. Marco, *Building and Managing the Meta Data Repository: A Full Lifecycle Guide*. New York, NY: John Wiley & Sons, 2000.

[21] A. Tannenbaum, *Metadata Solutions. Using Metamodels, Repositories, XML, and Enterprise Portals to Generate Information on Demand.* Upper Saddle River, NJ: Pearson Education Corporate Sales Division, 2002.

[22] J. Aarts, H. Doorewaard, and M. Berg, "Understanding implementation: the case of a computerized physician order entry system in a large Dutch university medical center," *J Am Med Inform Assoc,* vol. 11, pp. 207-16, May-Jun 2004.

[23] R. Haux, A. Winter, E. Ammenwerth, and B. Brigl, *Strategic Information Management in Hospitals: An Introduction to Hospital Information Systems*. New York, NY: Springer, 2004.

[24] G. Shankaranarayanan and A. Even, "The metadata enigma," *Communications of the ACM,* vol. 49, pp. 88-94, 2006.

[25] P. P. Uhrowczik, "Data dictionaries," *IBM Systems Journal,* vol. 12, pp. 332-350, 1973.

[26] G. Shankaranarayanan, A. Even, and S. Watts, "Enhancing decision making with process metadata: theoretical framework, research tool and exploratory examination," in *Proceedings of the 39th Hawaii International Conference on System Sciences*, Kauai, 2006, pp. 209a - 209a.

[27] R. Lake, "The application of geographymarkup language (GML) to the geological sciences," *Computers & Geoscience,* vol. 31, pp. 1081-1094, 2005.

[28] E.-P. Lim, D. H.-L. Goh, Z. Liu, W.-K. Ng, C. S.-G. Khoo, and S. E. Higgins, "G-Portal: a map-based digital library for distributed geospatial and georeferenced resources," *International Conference on Digital Libraries. Proceedings of the 2st ACM/IEEE-CS joint conference on Digital libraries,* 2002.

[29] J. Broekstraa, M. Klein, S. Deckerc, D. Fenselb, F. v. Harmelenb, and I. Horrocksd, "Enabling knowledge representation on the Web by extending RDF Schema," *Computer Networks* vol. 39, pp. 609-634, 2002.

[30] O. Casher and H. S. Rzepa, "SemanticEye: a semantic web application to rationalize and enhance chemical electronic publishing," *J Chem Inf Model,* vol. 46, pp. 2396-411, Nov-Dec 2006.

[31] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs, "Swoogle: a search and metadata engine for the semantic web," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management* Washington, D.C., USA ACM Press, 2004, pp. 652-659

[32] J. Kohler and S. Schulze-Kremer, "The semantic metadatabase (SEMEDA): ontology based integration of federated molecular biological data sources," *In Silico Biol,* vol. 2, pp. 219-31, 2002.

[33] B. Thirion, G. Loosli, M. Douyere, and S. J. Darmoni, "Metadata element set in a quality-controlled subject gateway: a step to a health semantic Web," *Stud Health Technol Inform,* vol. 95, pp. 707-12, 2003.

[34] Y. Denekamp, A. A. Boxwala, G. Kuperman, B. Middelton, and R. A. Greenes, "A meta-data model for knowledge in decision support systems," *AMIA Annu Symp Proc,* p. 826, 2003.

[35] G. Malet, F. Munoz, R. Appleyard, and W. Hersh, "A model for enhancing Internet medical document retrieval with "medical core metadata"," *J Am Med Inform Assoc,* vol. 6, pp. 163-72, Mar-Apr 1999.

[36] A. L. Rector, J. Rogers, and A. Taweel, "Models and inference methods for clinical systems: a principled approach," *Medinfo,* vol. 11, pp. 79-83, 2004.

[37] P. Merialdo and G. Sindoni, "MrBrAQue: a multimedia medical report management system," *1997 International Conference on Multimedia Computing and Systems (ICMCS'97),* pp. 638 - 639, 1997.

[38] A. Vasconcelos, R. Mendes, and J. Tribolet, "Using organizational modeling to evaluate healthcare IS/IT Projects," *Proceedings of the 37th Hawaii International Conference on System Sciences,* 2004.

[39] T. Wendt, A. Haber, B. Brigl, and A. Winter, "Modeling Hospital Information Systems (Part 2): using the 3LGM2 tool for modeling patient record management," *Methods Inf Med,* vol. 43, pp. 256-67, 2004.

[40] A. Winter, B. Brigl, G. Funkat, A. Haber, O. Heller, and T. Wendt, "3LGM(2)-modelling to support management of health information systems," *Stud Health Technol Inform,* vol. 116, pp. 491-6, 2005.

[41] A. Winter, B. Brigl, G. Funkat, A. Haber, O. Heller, and T. Wendt, "3LGM2-modeling to support management of health information systems," *Int J Med Inform,* vol. 76, pp. 145-50, Feb-Mar 2007.

[42] A. Winter, B. Brigl, O. Heller, U. Mueller, A. Struebing, and T. Wendt, "Supporting information management for regional health information systems by models with communication path analysis," *IDEAS Workshop on Medical Information Systems: The Digital Hospital, 2004. IDEAS '04-DH. Proceedings.,* pp. 139- 146, 2004.

[43] F. Ferrara, "Healthcare information systems architecture," in *New Technologies in Hospital Information Systems*, J. Dudeck, B. Blobel, W. Lordieck, and T. Buerkle, Eds. Amsterdam: IOS Press, 1997, pp. 1-10.

[44] T. Beale and S. Heard, "openEHR Architecture Overview," *The openEHR Foundation*, April 2007. [Online]. Available: http://openehr.org/. [Accessed: Jul. 28, 2007]

[45] C. Marietti, "Building the empire. CORBAmed focuses on goals of connectivity and interoperability," *Healthc Inform,* vol. 15, p. 39, Feb 1998.

[46] W. E. Hammond, "HL7--more than a communications standard," *Stud Health Technol Inform,* vol. 96, pp. 266-71, 2003.

[47] International Standardization Organization, "International Standard ISO/IEC 11179, Information Technology - Metadata registries (MDR)," *International Standardization Organization*, September 2004. [Online]. Available: http://isotc.iso.org/. [Accessed: Jul. 28, 2007]

[48] P. A. Covitz, F. Hartel, C. Schaefer, S. De Coronado, G. Fragoso, H. Sahni, S. Gustafson, and K. H. Buetow, "caCORE: a common infrastructure for cancer informatics," *Bioinformatics,* vol. 19, pp. 2404-12, Dec 12 2003.

[49] P. George Beeler Jr, J. Curry, and A.-M. Shakir, "HL7 Reference Information Model," *Health Level Seven*, February 2002. [Online]. Available: http://www.miforum.net/distillate/rim/RIM0112_body.htm. [Accessed: Aug. 15, 2007]

[50] W. M. Newman and M. G. Lamming, *Interactive System Design*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[51] L. Spinhof and L. Calvi, "User and task analysis in a home care environment," in *Proceedings of the 20th International Symposium on Human Factors in Telecommunication.* Sofia Antipolis, France, 2006, pp. 1-7.

[52] A. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction* vol. 3rd. Upper Saddle River, NJ: Prentice Hall, Inc., 2003.

[53] J. Hackos and J. Redish, *User and Task Analysis for Interface Design*. New York, NY: John Wiley & Sons, 1998.

[54] R. Vallacher and D. Wegner, "What do people think they are doing? Action identification and human behavior.," *Psych Rev,* pp. 3-15, 1987.

[55] B. Brigl, A. Strubing, T. Wendt, and A. Winter, "Modeling interdependencies between information processes and communication paths in hospitals," *Methods Inf Med,* vol. 45, pp. 216-24, 2006.

[56] B. Brigl, T. Wendt, and A. Winter, "Modeling interdependencies between business and communication processes in hospitals," *Stud Health Technol Inform,* vol. 95, pp. 863-8, 2003.

[57] J. Nielsen, "Why you only need to test with 5 users," March 2000. [Online]. Available: http://www.useit.com/. [Accessed: Jul. 28, 2007]

[58] J. Brooke, "SUS - A quick and dirty usability scale," in *Usability Evaluation In Industry*, P. W. Jordan, B. Thomas, I. L. McClelland, and B. A. Weerdmeester, Eds. London: Taylor & Francis, 1996, pp. 189-194.