Modeling Hospital Information Systems (Part 1): The Revised Three-layer Graph-based Meta Model 3LGM²

A. Winter, B. Brigl, T. Wendt
Institute for Medical Informatics, Statistics and Epidemiology, University of Leipzig, Leipzig, Germany

Summary

Objectives: Not only architects but also information managers need models and modeling tools for their subject of work. Especially for supporting strategic information management in hospitals, the meta model 3LGM² is presented as an ontological basis for modeling the comprehensive information system of a hospital (HIS).

Methods: In a case study, requirements for modeling HIS have been deduced. Accordingly 3LGM² has been designed to describe HIS by concepts on three layers. The domain layer consists of enterprise functions and entity types, the logical tool layer focuses on application components and the physical tool layer describes physical data processing components. In contrast to other approaches a lot of inter-layer-relationships exist. 3LGM² is defined using the Unified Modeling Language (UML).

Results: Models of HIS can be created which comprise not only technical and semantic aspects but also computer-based and paper-based information processing. A software tool supporting the creation of 3LGM² compliant models in a graphical way has been developed. The tool supports in detecting those shortcomings at the logical or the physical tool layers which make it impossible to satisfy the information needs at the domain layer. 3LGM² can also be used as an ontology for describing HIS in natural language.

Conclusions: Strategic information management even in large hospitals should be and can be supported by dedicated methods and tools. Although there have been good experiences with 3LGM² concerning digital document archiving at the Leipzig University Hospital, which are presented in part 2, the benefit of the proposed method and tool has to be further evaluated.

Keywords

Hospital information systems, hospital information system management, information management, models for hospital information systems, ontology, organizational issues

Methods Inf Med 2003; 42: 544–51

Introduction

Heterogeneity is inherent to hospitals. A hospital consists of various organizational units with differing tasks for various types of healthcare professionals. Since integrated care should be the aim, a high degree of interoperability has to be reached. This requires intensive internal communication among organizational units and healthcare professionals as well as external communication (e.g. to insurance organizations, general practitioners, etc.).

The hospital is itself a system, precisely a sociotechnical system, in which human beings and machines carry out specific actions following established rules. In this context, it is not surprising, that constructing and managing communication and interoperation needs a sociotechnical approach [5]. Accordingly we can say that the hospital's system for communicating and processing information, i.e. the hospital information system (HIS), is that socio-technical subsystem of a hospital which allows this interoperability by presenting information at the right time, in the right place to the right people [1, 2]. Consequently the heterogeneity of a hospital is reflected by its HIS. Because of different requirements, its information processing tools range from pencils, paper charts, and human couriers to computer-based systems based on various hardware platforms and software products offered by several vendors [3]. So, modern computer-based information processing tools, legacy systems [4], and still paper-based records, forms etc. exist side by side as part of a HIS.

This situation remained unchanged since the mid nineties. Consequently, we still have "heterogeneous hospital information systems using a variety of software and hardware products […]. This leads […] to a stronger demand on approaches for integrating these products and, in general, for a dedicated methodology for the management (design, supervision) of computer-based hospital information systems in order to adequately support patient care and medical research" [5].

Management of HIS in this context does not only mean to react properly on given heterogeneity. Moreover it means to actively design and construct the information system like a (complex of) building(s) out of different and probably heterogeneous bricks and components. Like an architect the information manager needs a blueprint or model not only for planning the information system but also for directing and supervising it. Additionally a precise terminology for designating the components is needed both for the architect and the information manager. Otherwise it leads to communication problems e.g. with the craftsmen.

The aim of the paper is to provide a terminology or ontology for designating HIS and their components which also will serve as a meta model for modeling them.

Therefore we first examine information managers' requirements on models for HIS. Other modeling approaches for information systems and especially the former 3LGM [5] are checked against these requirements. As a consequence we propose 3LGM² as a new meta model for modeling HIS. Since preparing a model does not only need a meta model but also an appropriate tool – e.g. for computer aided design (CAD) – we will shortly present the

3LGM² tool; a more detailed description and some application experiences will follow in part 2 of this paper. Finally, the benefits as well as the needs for further development will be discussed.

Requirements

According to our definition, a HIS is not only a hospital's ADT system but the whole system of information processing in a hospital. It therefore comprises the paper-based patient records as well as for example the patient management system in the intensive care unit. This means that information managers in hospitals have to understand not only networks, computer hardware and software. Especially the hospital's enterprise functions [6, 7] themselves and the way they can be adequately supported by information and communication technology have to be understood.

Depending on the information management tasks to be fulfilled, the models may be varying. It is therefore useful to differentiate between strategic information management as an integrative part of strategic business planning [7] on the one hand and tactical and operational information management on the other hand. Strategic information management deals with the hospital's information processing as a whole. It depends strictly on the hospital's mission and the depending strategic goals and has to translate these into a well fitting information strategy as part of a strategic information management plan [7, 8].

In a case study we explored information managers' requirements for a meta model and/or tool for modeling and assessing HIS. Nine persons responsible for information management in small-sized or medium-sized hospitals and two healthcare IT consultants were interviewed concerning strategic information management questions. A questionnaire comprising questions about 'organizational structures of information management', 'information management tasks', 'information about the hospital information system', 'quality of hospital information systems' and 'methods and tools for information management' was used. The questionnaire was not standardized in order to allow the interviewees to express their open opinion. For the oral interviews it served as guideline for the interviewers, who were allowed to deviate from the questionnaire and to inquire more deeply if necessary. From these information managers' needs we deduced the following requirements on a meta model for HIS and their management:

(r1) It should be possible to model enterprise functions and the tools supporting them. An *enterprise function* [6] is considered to be a directive for human or machine action. It doesn't have a beginning or an end. It may be understood as duty arising from the enterprise's mission and goals. For example, PATIENT ADMISSION, ORDER ENTRY, or CLINICAL DOCUMENTATION may be enterprise functions. Tools may be computers, applications, and paper-based tools as well.

(r2) Information has to be modeled as entity types used by enterprise functions. An *entity type* is the abstraction of physical or virtual things (of the hospital), which share same properties. For example, PATIENT, CASE, or RESULT may be entity types, whereas e.g. the patient John Doe is an entity.

(r3) The representation of entity types in form of datasets, forms and messages etc. have to be part of a model.

(r4) Relationships between both the enterprise functions and the tools in (r1) should be able to be modeled. This would explain, what the tools are used for and how functions are supported.

(r5) Interworking of functions as well as communication between the tools in (r1) should be able to be modeled whereby the relationships of (r4) have to be taken into account. This is to show how e.g. ORDER ENTRY depends on PATIENT ADMISSION and how data representing the jointly used entity type PATIENT is communicated from and to the respective applications.

(r6) If an entity type is stored in one application but the function which needs it is supported by another, communication is needed. It should be possible to model the corresponding communication sequences. This would explain, how e.g. data representing a PATIENT and stored in a remote database can be transported to the application supporting an enterprise function like ORDER ENTRY.

These results correspond to the characteristics of enterprise architecture planning as outlined in [7].

Approaches

There are a lot of information system modeling approaches available, which are or can be used for enterprise information systems in general and for hospital information systems in particular (e.g. [9-12]). Designed for business process optimization, they do not meet the above mentioned requirements for several reasons:

ad (r1) Most of the examined approaches just concentrate on the domain layer, considering information processing tools as resources that don't have to be specified any more (see e. g. [9, 10]). Other approaches take these tools into account, but just describe what layers and views have to be modeled, yet not how (see e. g. [11, 12]).

ad (r2) None of the examined approaches concentrate on the representation of enterprise functions as concepts which primarily access and update information.

ad (r3) None of the examined approaches distinguishes explicitly between processing, storage, and communication of data representing entities.

ad (r4) As a consequence of (r1) the relationship between enterprise functions and information processing tools can not be modeled adequately.

ad (r5) Through the concentration on the domain layer the dependencies between enterprise functions can be modeled in detail as business processes, but not the interaction be-

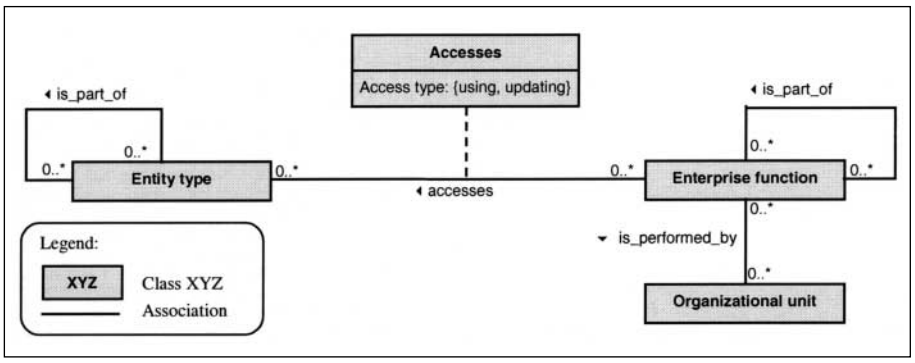

Fig. 1 The UML class diagram for 3LGM² domain layers (Symbols as in the UML Notation Guide [19]).

tween information processing tools to support enterprise functions.

ad (r6) Most of the examined approaches do not specify processes between the information processing tools. This would reflect what must happen e.g. between applications and computers to satisfy the information needs at the domain layer.

In addition to these restrictions, none of the examined approaches presents a comprehensive meta model guiding the modeler in building a usable model of the information system.

For HIS there are plenty of models published, but there are no specialized meta models or modeling techniques. Even architectures like HISA [13, 14] or CORBAmed [15] are to be considered as reference models [16] but not as meta models.

The 3LGM in [5] concentrated on the HIS domain and intended to overcome the problems mentioned before by proposing three interconnected layers for modeling HIS.

A procedure layer describes the information procedures of a HIS (and their information interchange) and thus, the HIS's functionality. An *information procedure*, or briefly procedure, is a set of methods, procedures, and tools and of rules prescribing the way in which they are to be applied in information processing. PATIENT ADMISSION may be an example of a procedure. In contrast to enterprise functions in 3LGM² the procedures may carry out an *information interchange* and may have access points. Hence, this concept often had been confused with the application systems of the logical tool layer. An application system had been defined as having a memory of its own and being autonomous. These are severe restrictions conflicting with modern component based information system architectures. The physical tool layer consists of physical subsystems (e.g. computer systems) and data transmission lines, but lacks concepts for modeling modern networks.

Thus, this approach looks similar to the three integration levels for modeling "interworking" in a "software factory"[17]. 3LGM had been used for supporting strategic information management by different working groups.

3LGM, however, was also unable to satisfy the described requirements:

ad (r1) Instead of functions 'procedures' had been defined at a 'procedure level'. But it was not clear, how procedures had to be distinguished from 'application systems'.
The very restrictive concept of 'application systems' instead of 'application components' turned out to be too inflexible to model modern component based architectures.
More detailed concepts are needed to describe hardware components and networks connecting them.

ad (r2) Entity types could not be modeled.

ad (r3) The representation of entity types as datasets, forms and messages and the situation of redundant data storage could not be modeled.

ad (r4) The meta model was described in first-order predicate logic and was hardly understandable.

ad (r5) There is a lack of concepts for describing event driven and standardized communication.

ad (r6) There has been no means for modeling sequences of information flows between procedures respectively of message flows between applications.

3LGM²: a revised meta model for modeling hospital information systems[1]

The redesign of 3LGM as 3LGM² intended to keep the benefits of 3LGM but also to meet the new requirements. 3LGM² combines a functional meta model with technical meta models and is represented using the Unified Modeling Language (UML) [19]. 3LGM², similar to its predecessor 3LGM, distinguishes between three different layers within an information system.

Domain Layer

According to [7] the *domain layer* (see Fig. 1) replaces the former 'procedure level' and describes a hospital independently of its implementation by its enterprise functions. Enterprise functions need information of a certain type about physical or virtual things of the hospital. These types of information are represented as *entity types*. The access of a function to an entity type can be in a using or an updating manner which is expressed by the attribute `access type` of the association class `access`. Enterprise functions and entity types can

[1] An early version has been presented at the MEDINFO 2001 conference [18].

be (poly-)hierarchically structured using the respective 'is_part_of' associations. Functions are performed by *organizational units*.

An exemplary instance of the domain layer is shown in Figure 2. The association between the function PATIENT ADMISSION and the entity type PATIENT denotes that PATIENT ADMISSION uses information about patients, but also updates information (expressed by the bi-directional arrow). The 'is_part_of' relations between enterprise functions are represented by putting one rectangle inside another one.

Which entity types and which functions are modeled depends on the hospital being considered. Reference models, like the requirements index for information processing in hospitals [20] and the Reference Information Model (RIM) of the HL7 standard may offer recommendations about important entity types and functions.

Note that the meta model of the domain layer up to now just considers the static view of a hospital. Thus, there are no associations between functions which represent processes.

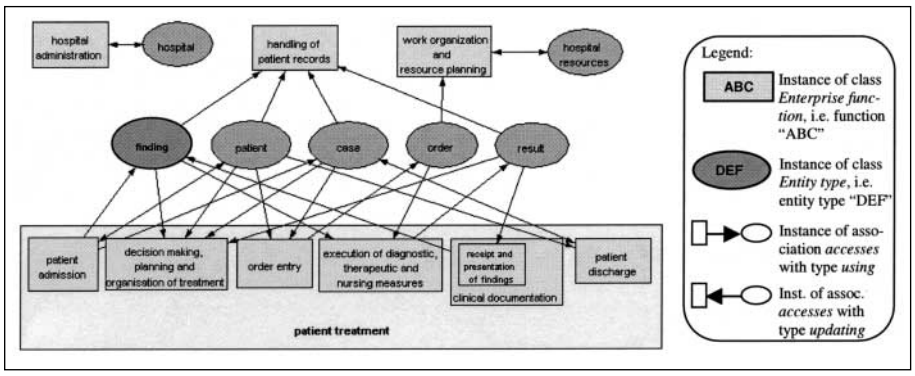

Fig. 2 An examplary instance of the 3LGM² domain layer

Logical Tool Layer

At the *logical tool layer* (see Fig. 3), *application components* are at the center of interest. Application components support enterprise functions and are responsible for the processing, storage and transportation of data representing entity types. Computer-based application components are controlled by *application programs*, which are adapted *software products* (this is what we can buy), paper-based application components are controlled by *working plans*. Computer-based application components may have a local *database*, which is controlled by a *database management system*, to store data. Paper-based application components may file their documents in a *document collection*. *Communication interfaces* ensure the communication among application components *(component interfaces)*, but also between an application component and a user *(user interfaces)*. A component interface bases on a message exchange

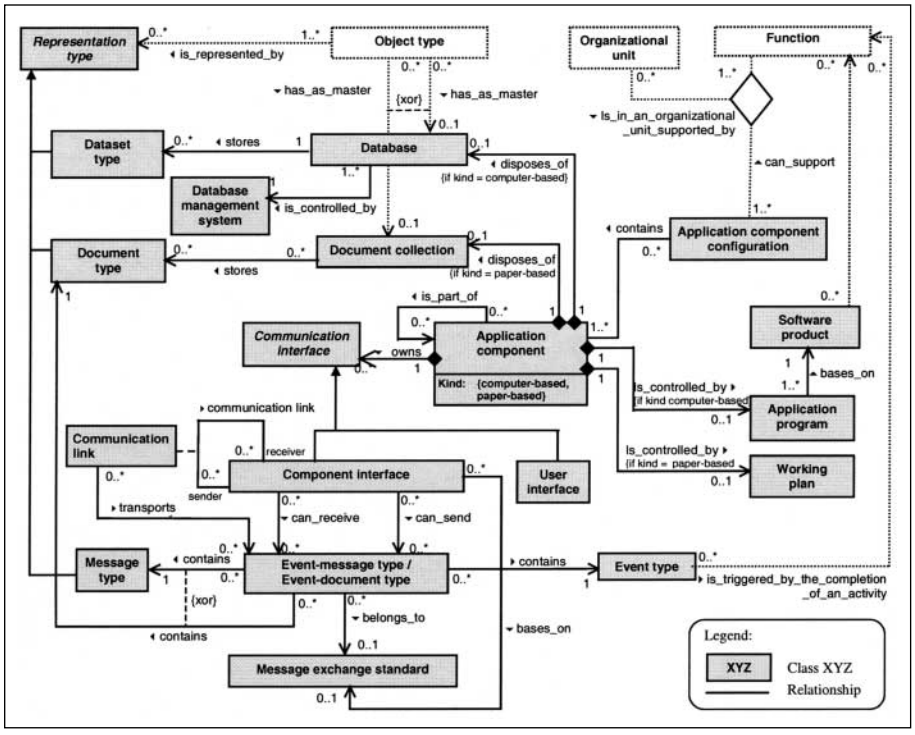

Fig. 3 The UML class diagram for 3LGM² logical tool layers. Dotted lines and symbols denote interlayer relationships (Symbols as in the UML Notation Guide [19])

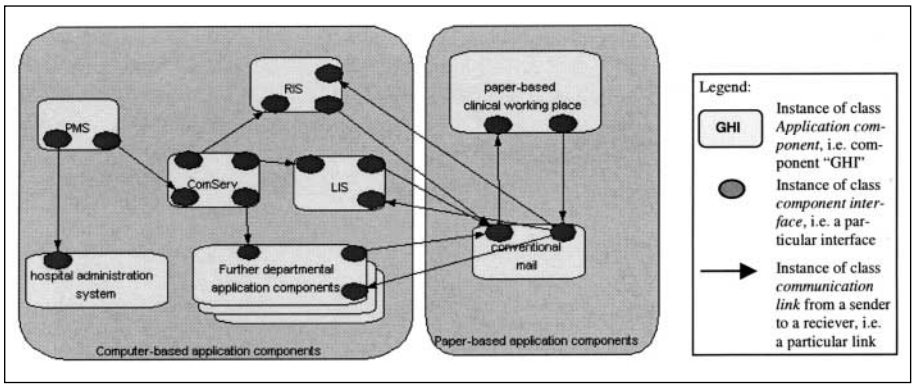

Fig. 4 An examplary instance of the 3LGM² logical tool layer

standard and can receive or send messages of a certain event-message type or – in case of paper-based communication – documents of a certain event-document type. Event-message types are composed of an event type and the message type or document type caused to be communicated by that event type. For the communication among application components *communication links* can be defined as relations between two communication interfaces. Note that communication links are pure virtual constructs, there is no physical representation for them. Each communication link is described by the event-message types which in fact are communicated. Application components may be refined. The concept of *application component configuration* will be explained in detail later on.

Figure 4 shows an exemplary instance of the logical tool layer. In this example, we just look at the application components and the relationships between them via communication interfaces.

The left part of the example shows computer-based application components: a patient management system (PMS), a radiological information system (RIS), a laboratory information system (LIS), a communication server (ComServ), a hospital administration system, and some further departmental application components (not otherwise specified). The right part shows paper-based application components: a paper-based mailing system and a paper-based clinical working place. This example is simplified and fictive, but reflects a typical situation. Whereas hospital administration and all service units are supported by computer-based application components, typical clinical functions are often just supported by a paper-based application component.

For the communication within the computer-based part there is a communication server available, but not all application components use this communication service. As a consequence, a lot of proprietary interfaces are implemented. The fact, that some typical clinical functions are not supported by computer-based application components results in a lot of communication links between the computer-based and the paper-based part of the HIS. This might cause some problems due to the transfer of data from one storage device to another ("media cracks"). Those digital-analog interfaces are, e.g., realized through printers and form readers and the corresponding software.

Physical Tool Layer

The *physical tool layer* (Fig. 5) is a set of physical data processing components which are used to realize the computer-based and the paper-based application components. Physical data processing components can be human actors (such as the person delivering mail), components like paper, telephones, books, patient records, which are to support the paper-based application components, or computer-based components (such as terminals, servers, personal computers). Components can be physically connected via so-called data transmission connections (e.g. data wires). The constellation of these connections between computer-based components leads to physical networks, which are based on network protocols. Arbitrary subnets can be defined as projections of the entire network. Note that physical as well as logical networks can be represented on the physical tool layer.

Figure 6 shows an exemplary instance of the physical tool layer. In this example, we see a server for each departmental application component (a LIS server, a RIS server, server for further departmental application components) and a central server where the PMS and the hospital administration system is installed. Each server is connected to a set of personal computers. The black dots represent network components. The data transmission from and to the paper-based part of the HIS is denoted only for the LIS data processing components. An order may come from the outbox of the clinical working place (CWP outbox) to the LIS inbox and be read by a form reader. A result is printed by the LIS printer and transfers via the LIS outbox to the CWP inbox. Data transmission connections are depicted as lines. In this example, there are no subnets specified. Information about network type, network protocol, and location is not represented here.

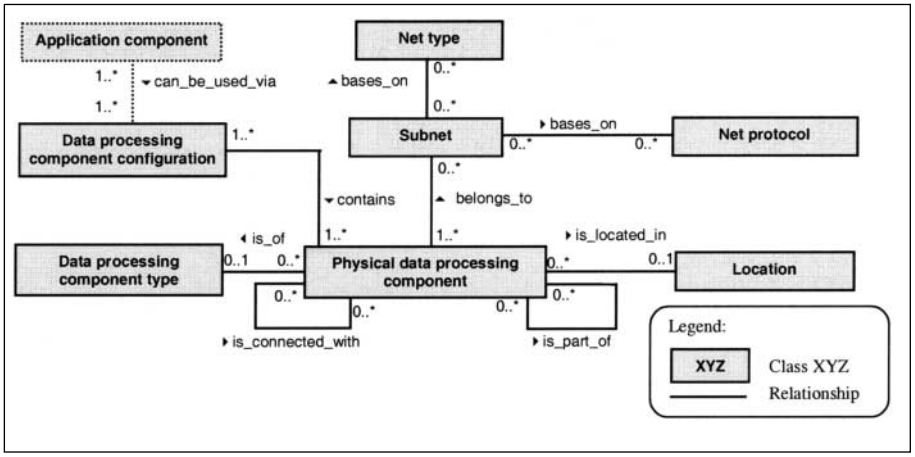

Fig. 5 The UML class diagram for 3LGM² physical tool layer. Dotted lines denote interlayer relationships. (Symbols as in the UML Notation Guide [19])

Inter-Layer-Relationships

A variety of dependencies, called *inter-layer-relationships* exist among concepts of different layers. These inter-layer-relationships may be used to detect shortcomings at the logical or the physical tool layers which make it impossible to satisfy the information needs at the domain layer.

Considering the domain layer and the logical tool layer, the most important relationship is between enterprise functions and application components. An enterprise

function performed by a specific organizational unit may be supported by several application components together, by a single application component, or by combinations of the two. An *application component configuration* contains all application components which are needed in collaboration with each other to support a function. If we remove any application component from this configuration, the function will no longer be supported by this configuration. An enterprise function may be supported by more than one application component configuration. If we remove such a configuration the function is still supported by one of the remaining configurations. So there is a hint about redundant and possibly unnecessary application components.

There are some other important inter-layer-relationships between concepts of the domain and the logical tool layer which we will describe in the following:

- The 'has_as_master' relation between entity type (e.g. PATIENT) and database (e.g. database of the application component PMS) respectively document collection describes which database or document collection is responsible for the storage of a certain entity type. If data e.g. about PATIENT is stored redundantly in several databases, updates should only be made in the 'master' database (e.g. PMS).
- A software product, what can be bought from software vendors, 'can' support enterprise functions. This relationship does not express which functions are really supported by an application component based on that software product, but rather describes the potential a software product offers.
- The completion of an activity of a certain enterprise function may lead to an event of a certain event type, which in turn triggers the sending of a message of a certain message type. So it can be modeled that after the PATIENT ADMISSION of a PATIENT the HL7 event A01 occurs which will cause sending an appropriate message by the PMS.
- Entity types can logically be represented by a dataset type, a document type, or a message type. Dataset types describe how data is stored in a database system. This may be done e.g. by a relation schema. Document types may e.g. be names of paper-based forms; they describe either how information is stored in a document collection or how it is transported e.g. by a human courier. A message type describes how information is transported by a communication link between two computer-based application components. The HL7 message type "ORU" may for example be used to transport data about a RESULT.

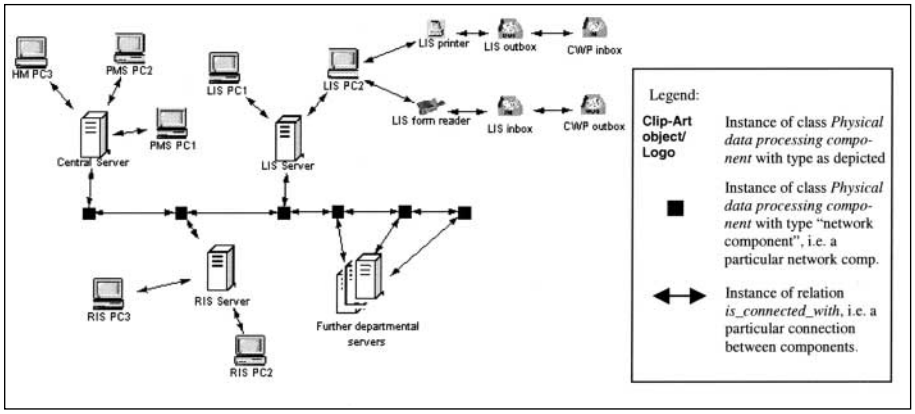

Fig. 6 An examplary instance of the 3LGM² physical tool layer

Between the logical tool layer and the physical tool layer, there exists a relation between application components and physical data processing components which is represented by a so-called *data processing component configuration*. It states, that an application component may be installed either on several data processing components together (e.g. typical client-sever-installations), on a single data processing component (typical stand-alone-application components), or through combinations of these two. A data processing component configuration contains all physical data processing components which are needed in collaboration with each other to install an application component completely. If we remove any physical data processing component from this configuration, the application component will no longer work. An application component may be utilized by more than one data processing component configuration. If we remove such a configuration, the application component still works through one of the remaining configurations, but may suffer from loss of quality. So there is again a hint about redundancy; in this case there may be redundant and possibly unnecessary physical data processing components on the physical tool layer.

3LGM² Tool for Modeling Hospital Information Systems

The 3LGM² tool software supports information managers in constructing graphical models of HIS, which correspond exactly to the meta model described above. 3LGM² tool runs on all platforms supporting Java 1.3.1. A convenient setup tool is currently available for MS Windows.

3LGM² tool is designed like a lot of other modeling tools. It contains a hierarchical model browser, a panel for drawing model elements and specific dialogs for details of the model elements. Button bars, menus and a search dialog are provided to make modeling easy (see Fig. 7).

It differs from other tools for drawing and modeling in the following aspects:

1. Model elements and their links can (only) be created on the base of the 3LGM². Thus, the tool itself is a strict guide to create 3LGM² compliant models of information systems.
2. All three layers of a model can be viewed and edited separately but also in a multilayer view. This view provides an overview of a model and shows links

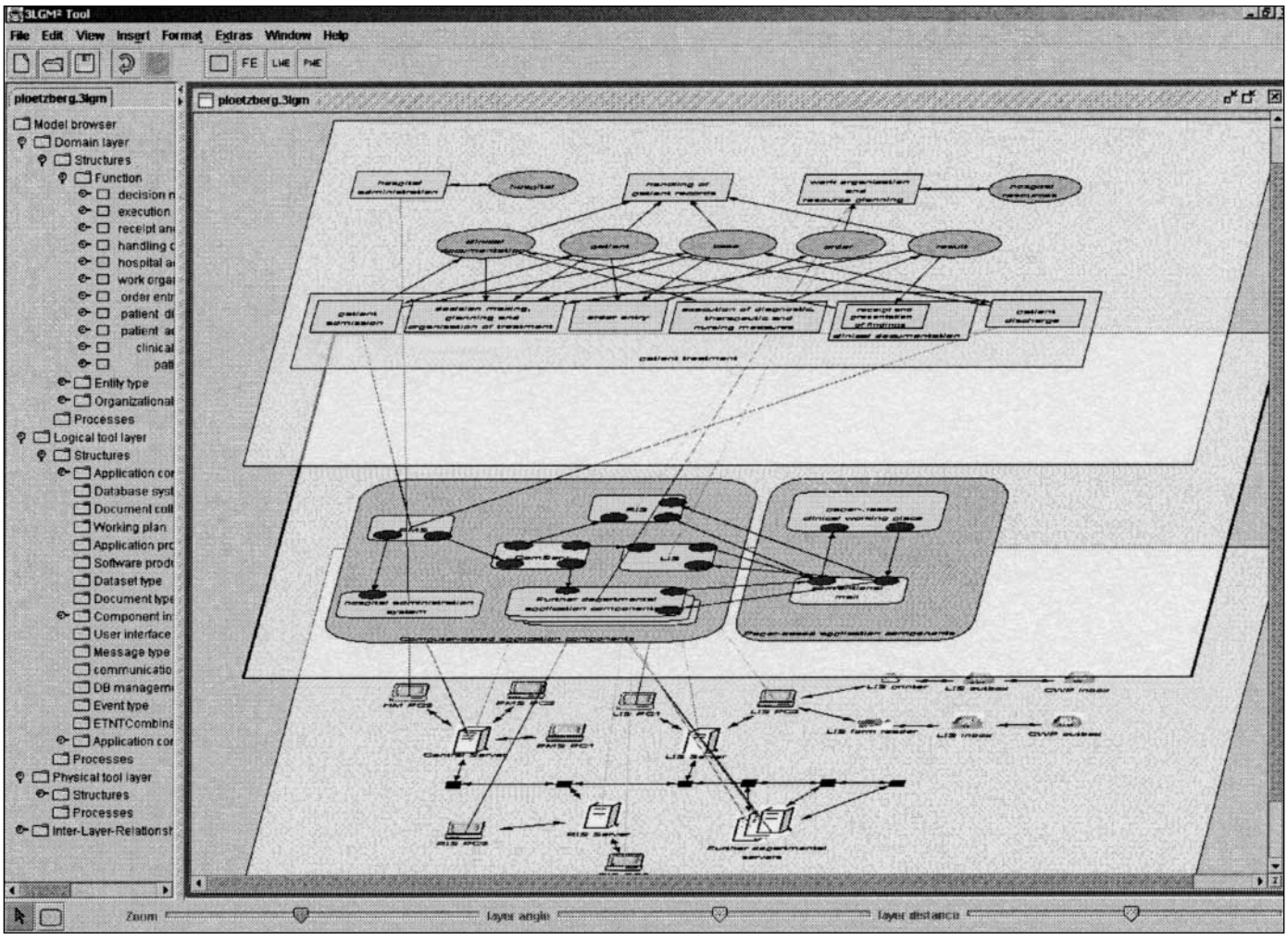

Fig. 7 3LGM² tool

between layers, e.g. links from a function PATIENT ADMISSION to all application components supporting it (Fig. 7).

3. Models of real environments are very complex. It is often necessary for an information manager to focus on certain aspects like a particular application component or enterprise function. The tool supports this by sub models which can be extracted from a complex model.

Menu elements and dialogs adapt automatically to the current modeling context. The tool performs background consistency checking to avoid formal errors.

3LGM² models are saved in XML format to make them importable for other applications. Additionally, the tool contains a bitmap export function. In bitmap formats like GIF or TIFF model graphics can be embedded into presentations or text documents.

Current development activities focus on functions to analyze information systems. An example for an analyze function is the checking of communication paths on the logical tool layer according to the relations between functions and entity types on the domain layer. The tool will also answer questions like "Which functions are missing support by application components when a specific server-machine does not work properly?".

Discussion

We presented a meta model, which has been used as specification for the graphical 3LGM² tool. Meta model and tool at least roughly meet the requirements (r1)-(r5), which were found by a case study with information managers. These positive results do not mean we have proven the usefulness of this approach to routine work of information managers in hospitals in general. Actually there are some problems left:

- *3LGM and 3LGM² tool do not meet the requirement (r6) concerning process modeling*. We are now implementing algorithms, which map sequences of functions to communication flows of messages between application components.
- *Modeling even a small (sub-)information system from scratch requires considerable effort.* We are therefore preparing a reference model for the domain layer (including typical functions and

entity types), templates for describing typical architectural styles of HIS (e.g. communication server based architecture, CORBA-based, etc.) as well as catalogues for commonly used items (e.g. message- and event-type combinations of HL7, document types of the Clinical Document Architecture CDA, database management systems, etc.)

- *We don't have experiences concerning the maintenance of large models.* We constructed different models of parts of the Leipzig University Hospital Information System with the 3LGM² tool to support tasks of strategic information management and we will present these models in part 2 of this paper. But further evaluations on different sites have to show whether it is possible not only to build large models but to keep them up to date.

Up to now we described how to use the 3LGM² in the sense of a meta model, i.e. as a specification for modeling. Additionally 3LGM² can be understood as a semantic net of concepts or as a terminology or ontology. It can be used for describing HIS even in everyday-professional-life using natural language. We use this terminology for academic teaching of HIS and it has been the terminological basis for two textbooks about HIS [21, 22].

Acknowledgements
This work is part of the research project 'Integrative modeling of structures and processes in hospital information systems' sponsored by the Deutsche Forschungsgemeinschaft (DFG), grant no. WI 1605/2-1.

We thank Reinhold Haux (Innsbruck) for his contribution to the fundamentals in [5] and fruitful discussions. Thanks to Sigmar Gerber (Leipzig) for supporting effectively the critical reflection of 3LGM.

References

1. Winter AF, Ammenwerth E, Bott OJ, Brigl B, Buchauer A, Gräber S, et al. Strategic Information Management Plans: The Basis for systematic Information Management in Hospitals. International Journal of Medical Informatics 2001; 64 (2-3): 99-109.
2. Berg M. Patient care information systems and health care work: A sociotechnical approach. International Journal of Medical Informatics 1999; 55: 87-101.
3. Kuhn KA, Giuse DA. From Hospital Information Systems to Health Information Systems. Methods of Information in Medicine 2001; 40: 275-87.
4. Brodie ML, Stonebraker M. Migrating legacy systems. San Francisco: Morgan Kaufmann; 1995.
5. Winter A, Haux R. A Three-Level Graph-Based Model for the Management of Hospital Information Systems. Methods of Information in Medicine 1995; 34 (4): 378-96.
6. Martin J. Information Engineering, Book II: Planning & Analysis. Englewood Cliffs: Prentice Hall; 1990.
7. Spewak SH, Hill SC. Enterprise Architecture Planning: Developing a blueprint for Data, Applications and Technology. New York: John Wiley & Sons; 1992.
8. Winter A, Brigl B, Buchauer A, Dujat C, Gräber S, Hasselbring W, et al. Purpose and Structure of Strategic Plans for Information Management in Hospitals. In: Hasman A, Blobel B, Dudeck J, Engelbrecht R, Gell G, Prokosch H-U, editors. Medical Infobahn for Europe. Amsterdam: IOS Press; 2000. pp. 880-4.
9. Oberweis A, Scherrer G, Stucky W. INCOME/STAR: Methodology and Tools for the Development of Distributed Information Systems. Information Systems 1994; 19 (8): 643-60.
10. Ferstl OK, Sinz EJ. Modeling of Business Systems Using the Semantic Object Model (SOM) – A Methodological Framework. In: Bernus P, Mertins K, Schmidt (eds). Handbook on Architectures of Information Systems. International Handbook on Information Systems: Springer; 1997.
11. Gartner Group. Three Documents for Healthcare IT Planning. Gartner Group's Healthcare Executive and Management Strategies Research Note; 1998. Report No.: KA-03-5074.
12. Zachman JA. A framework for information systems architecture (Reprint). IBM systems journal 1999; 38 (2&3): 454-70.
13. Ferrara FM. Healthcare Information Systems Architecture. In: Dudeck J, Blobel B, Lordieck W, Bürkle T, editors. New Technologies in Hospital Information Systems. Amsterdam: IOS Press; 1997. pp. 1-10.
14. Grimson W, Jung B, van Mulligen EM, van Ginneken A, Pardon S, Sottile PA. Extensions to the HISA standard – the SynEx computing environment. Methods of Information in Medicine 2002; 41 (5): 401-10.
15. Wang C, Ohe K. A CORBA-Based Object Framework with Patient Identification Translation and Dynamic Linking. Methods of Information in Medicine 1999; 38 (1/1999): 56-65.
16. Scheer A-W. Business Process Engineering, Reference Models for Industrial Enterprises. Berlin: Springer; 1998.
17. Weber H, editor. The Software Factory Challenge. Amsterdam: IOS Press; 1997.
18. Winter A, Brigl B, Wendt T. A UML-based Ontology for Describing Hospital Information System Architectures. In: Patel VL, Rogers R, Haux R (eds). MEDINFO 2001. Amsterdam: IOS; 2001. p. 778-82.
19. Object Management Group (OMG). Unified Modeling Language Specification. Version 1.3. www.omg.org; 1999.
20. Ammenwerth E, Buchauer A, Haux R. A Requirements Index for Information Processing in Hospitals. Methods of Information in Medicine 2002; 41 (4): 282-8.
21. Haux R, Winter A, Ammenwerth E, Brigl B. Strategic Information Management in Hospitals. Submitted for publication.
22. Winter A, Ammenwerth E, Brigl B, Haux R. Krankenhausinformationssysteme. In: Lehmann T, Bexten EMz (eds). Handbuch der Medizinischen Informatik. München: Hanser; 2002. pp. 473-552.

Correspondence to:
Prof. Dr. Alfred Winter
Institute for Medical Informatics, Statistics and Epidemiology
University of Leipzig
Liebigstr. 27
D-04103 Leipzig, Germany
E-mail: alfred.winter@imise.uni-leipzig.de