Modeling Hospital Information Systems (Part 2): Using the 3LGM^2 Tool for Modeling Patient Record Management

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

Summary

Objectives: We introduce the 3LGM^2 tool, a tool for modeling information systems, and describe the process of modeling parts of the hospital information system of the Leipzig University Hospital (UKL[a]). We modeled the sub information systems of five patient record archiving sections to support the creation of a proposal for governmental financial support for a new document management and archiving system. We explain the steps of identifying the model elements and their relations as well as the analyzing capabilities of the 3LGM^2 tool to answer questions about the information system. ***Methods:*** The 3LGM^2 tool was developed on the basis of the meta model 3LGM^2 which is described in detail in [1]. 3LGM^2 defines an ontological basis, divided into three layers and their relationships. In addition to usual meta CASE tools, the 3LGM^2 tool meets certain requirements of information management in hospitals. The model described in this article was created on the base of on-site surveys in five archiving sections of the UKL. ***Results:*** A prototype of the 3LGM^2 tool is available and is currently tested in some projects at the UKL and partner institutions. The model presented in this article is a structured documentation about the current state of patient record archiving at the UKL. The analyzing capabilities of the 3LGM^2 tool help to use the model and to answer questions about the information system. ***Conclusions:*** The 3LGM^2 tool can be used to model and analyze information systems. The presentation capabilities and the reliability of the prototype have to be improved. The initial modeling effort of an institution is only valuable if the model is maintained regularly and reused in other projects. Reference catalogues and reference models are needed to decrease this effort and to support the creation of comparable models.

Keywords
Information management, hospital information systems, models (theoretical), information management, organizational model, 3LGM^2, computerized models, documentation, patient record

Methods Inf Med 2004; 43: 256–67

[a] UKL is the official abbrevations of the German name »Universitätsklinikum Leipzig«

Introduction

A hospital information system (HIS) should be considered as that socio-technical subsystem of a hospital which has to provide information, e. g. about patients, at the right time, in the right place to the right people [1-3]. Accordingly, a HIS is not only the hospital's ward management system or the patient management and accounting system but its whole system of information processing. It therefore comprises e. g. the paper-based patient records as well as the computer-based patient monitoring system in an intensive care unit.

Such a comprehensive and complex system needs a systematic information management approach. The information manager's task is comparable to the task of an architect, who has to construct a complex building from different and probably heterogeneous components. The information manager needs a blueprint or model for planning the information system but also for its direction and monitoring.

In [1] we proposed the 3LGM^2 as a meta model for modeling HIS. Preparing a model does not only need a meta model but also an appropriate tool. Using 3LGM^2 as the ontological basis this tool should enable information managers to graphically design even complex HIS. It should assist information managers similarly as computer aided design tools (CAD) support architects.

The aim of the paper is to present a prototype of the graphical 3LGM2 tool and to demonstrate its usability by presenting a model of a part of the Leipzig University Hospital Information System. The 3LGM^2 tool is freely accessible and readers are welcome to evaluate it[b].

We start with a requirements list on a 3LGM^2 *tool* and a short discussion of other modeling approaches in chapter two. The third chapter describes the *3LGM^2 tool* itself. The fourth and largest chapter reports on results from the work of the authors in information management at the Leipzig University Hospital (UKL): It describes a model of that sub HIS of the UKL that deals with archiving patient records. Analyzing and presenting capabilities that help to answer questions with a model are described in the fifth chapter. Benefits as well as shortcomings and needs for further development are discussed finally.

Requirements and Approaches

A 3LGM^2 compliant model describes an information system using three different layers [1]:

- a domain layer,
- a logical tool layer and
- a physical tool layer.

The domain layer consists of enterprise functions and entity types. An entity represents information about a physical or virtual thing (in a hospital) and an entity type is a class of entities. The logical tool layer focuses on application components and the physical tool layer describes physical data processing components. In contrast to other approaches the 3LGM^2 also defines inter-layer-relationships between the layers

[b] Please e-mail to thomas.wendt@imise.uni-leipzig.de

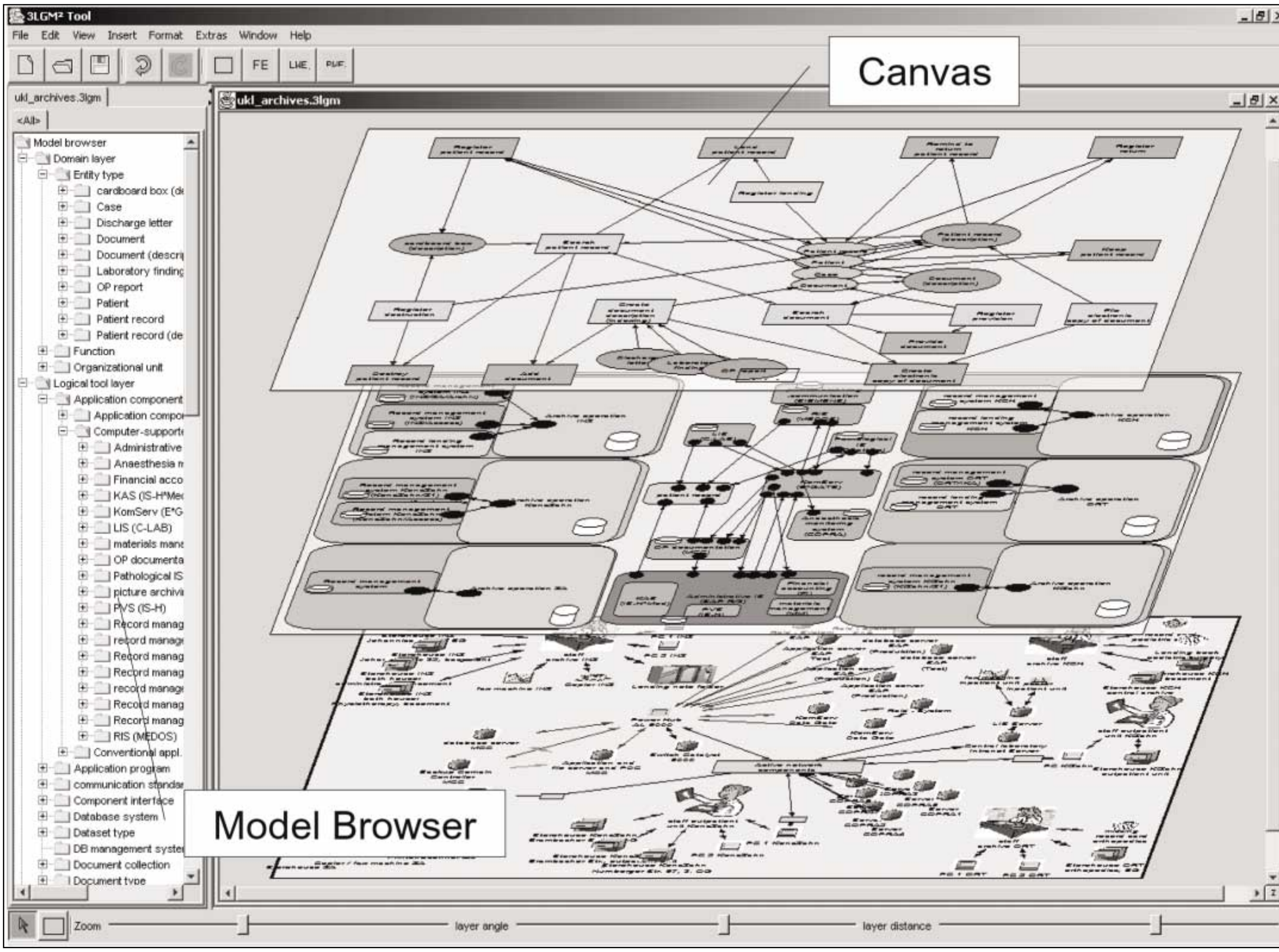

Fig. 1 The 3LGM² tool

to build integrated models of information systems[c].

A tool to efficiently support information managers in constructing 3LGM² models for HIS in real environments should

- provide a graphical representation of the most important concepts of 3LGM²;
- ensure that only the 3LGM² concepts can be modeled and that only those associations can be specified which are defined by the 3LGM²;
- be able to display the three layers of a HIS model separately but also combined in a multi-level view together with the inter-layer-relationships;
- help to manage even large models by supporting sub-models for various views;
- provide means for analyzing a completed model;
- provide means for documenting all needed properties of functions, entity types, application components, and physical data processing components and should support this by possibilities to create individual catalogs.

That means the modeling tool should 'know' the 3LGM². Consequently, a mere drawing tool would in fact be helpful for drawing diagrams but would not meet the requirements in total.

More appropriate candidates for tools may come from computer aided software engineering (CASE). It is widely accepted to use the Unified Modeling Language (UML). A lot of UML tools are offered by the industry (e. g. [4]). Using the concept of "stereotypes" [5] could make UML 'learn 3LGM²'. Stereotypes can extend UML's base modeling elements and thus extend the UML meta model layer by classes, i. e. concepts of the 3LGM². A more generic approach may come from so called meta CASE tools (e. g. KOGGE [6]). A meta CASE tool gets a meta model for a particular class of graphs as input and generates a CASE tool. The generated tool enables the construction of exactly those graphs, which are defined by the meta model.

Hence UML tools as well as meta CASE tools could be used as tools for constructing 3LGM² compliant models of hospital information systems. But we decided

[c] It is strongly recommended to read the original paper [1] to get deeper insight into details and the theoretical concepts of the 3LGM².

to develop a specific tool because Information management must not be confused with software engineering. Information managers usually do not develop software but plan, direct and monitor heterogeneous HIS. This means in detail:

- The full functionality of a CASE tool is not needed for modeling HIS and a reduction of complexity is necessary for users, as e.g. information managers.
- Means for analyzing a completed 3LGM² model and for extracting submodels cannot be generated by generic approaches.
- The 3LGM² implies that models are visualized in a specific way (e. g. displaying the three layers of a HIS model separately but also in a common view together with the inter-layer-relationships). This cannot be achieved by generic meta case tools.
- Models should also (at least in parts) be understandable by persons not specialized in information management, e.g. chief physicians or the CEO (chief executive officer).

The selection of a meta-language and tool for describing the meta model 3LGM², i. e. the meta-meta model, has been independent from the decision to develop a particular tool for making 3LGM² compliant models. There are different meta-languages respectively tools for describing languages or ontologies (e.g. UML or Protégé [7]). As explained in [1] we decided for UML in this case simply because it is widespread and proved to be sufficient in this case.

The 3LGM² Tool

General Features

The 3LGM² tool is a software product designed to create information system models on the basis of 3LGM² (Fig. 1). On the modeling canvas, which dominates the main window of the tool, an information system can be modeled and displayed on three layers. A model diagram is a graph, i. e. consists of nodes and edges. There are different node types that correspond to the element classes defined in the 3LGM²:

1) On the top layer – the *domain layer* – the hospital's enterprise functions and entity types used by these functions are modeled. An edge between a function and an entity type symbolizes that the function uses or creates information about entities of that entity type.
2) The middle layer – the *logical tool layer* – contains application components, database systems, document collections, component interfaces and communication links between them. An application component is an installation of application software or an implementation of an organizational plan. Application components support functions and store data about entity types in database systems respectively document collections. These relations are modeled explicitly by linking elements from the logical tool layer to elements of the domain layer.
3. The bottom layer – the *physical tool layer* – contains physical tools: record shelves, computers, network components and even personnel[d], i.e. 'touchable' components of the information systems. Physical data processing components are the basis for application components. Similar to the relations between the domain layer and the logical tool layer, relations between the latter and the physical tool layer are modeled explicitly by linking model elements.

Each of the element classes *function, entity type, application component, database system, document collection, component interface,* and *physical data processing component* is visualized in the model diagram and has a default geometric shape and a default background color. Since a fixed mapping of shapes and colors to the element types may be too inflexible, users can

- change the default mapping from element classes to shapes and colors,
- change the shape and the color of selected elements,
- assign bitmap symbols to selected elements, and
- change the line style and the color of selected edges.

A lot of detailed information is not graphically displayed but accessible via the model browser and by dialog windows (Fig. 2).

The three different layers can be viewed and edited separately but can also be combined in a multi-level view as shown in Figure 1. Thus, users may focus on a specific layer but also on relations between the

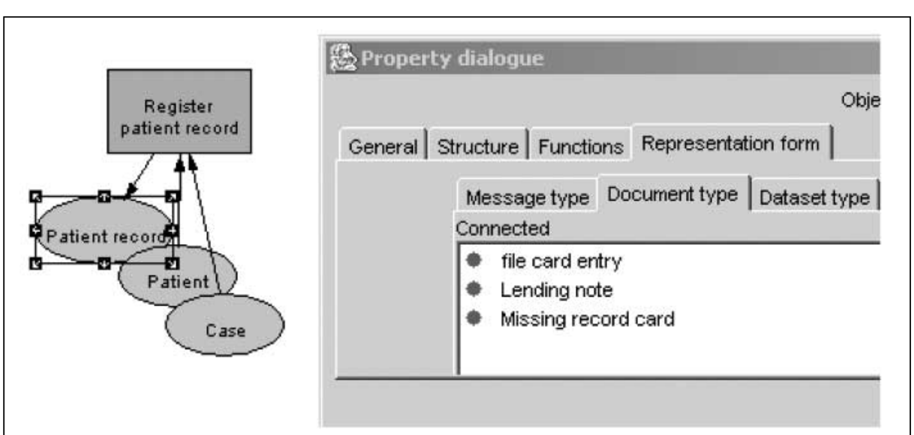

Fig. 2 A simple domain layer with a function, three entity types and a detail dialog for the entity type "Patient record"

[d] We apologize for denominating human beings as "tools" and "data processing components". This scandal seems to be the back side of the same medal showing the very importance of non-electronic information processing in hospitals. We are grateful for hints and comments leading to a nomenclature more adequate to human diginity.

different layers. This feature makes it easier to create an integrated model and present this model to partners. The angle of and the space between the layers in the multilevel view can be adjusted.

Similarly to a lot of other graphical modeling tools the 3LGM² tool also offers
- typical operations for graphical modeling, e.g. aligning elements, moving elements up and down, changing colors and fonts, adding icons, etc.,
- a model browser for hierarchical browsing through the model structure;
- dialog windows to enter and view detailed information about model elements;
- a menu bar and tool bars for accessing common operations.

Analysis parameter	Analysis result
A function	The function, all application components supporting the function, and all underlying physical data processing components
An entity type	The database systems, where this entity type is stored, the message types transporting them and the component interfaces sending or receiving these message types.
An application component	The functions supported by the component and the physical data processing components needed for this application component
A physical data processing component	The physical data processing component, all application components installed on that component and all supported functions
A physical data processing component	The entity types stored on this component
<no parameter>	All functions not accessing an entity type

Table 1 Some predefined analysis functions provided by the 3LGM² tool

Extraction of Submodels

Depending on the complexity of an information system and the required level of detail the model diagram may easily get unclear and confusing. The 3LGM² includes the functionality to extract subsets of models into submodels.

Submodels are presented in separate diagrams, have their own model browser trees and hold own layout data, e.g. element position and colors. The data elements from the underlying model are not separated, i. e. a change of an element name in a submodel affects the whole model.

Analyzing Models

The 3LGM² tool provides a set of predefined analysis functions (see Table 1). These functions are designed to answer specific questions arising in information management business. The result of an analysis can be highlighted in the model graphic but also be used to create a submodel.

The analysis feature applies search algorithms on the internal graph structure to find model subsets and paths between model elements. It takes into account
- names and descriptions,
- element classes, and
- connections to elements of specific classes.

This feature may be considered as the major criteria of differentiation from other graphic and modeling tools. In addition to the predefined function set the user may define customized analysis functions in the analysis definition dialog.

Exporting

Bitmap representations of model diagrams can be created via an export operation. This feature makes it easy to embed model graphics into slides or report documents. A report feature to create text lists and tables is provided: Since the model files are saved in XML format, we integrated an XSLT processor to apply XSL transformations to 3LGM² models. The list of extractable information is under continuous development.

A 3LGM² Based Model of the Patient Record Management at the Leipzig University Hospital

Several projects at the UKL focus on the topic of archiving patient records and administrative records: While the archives for paper-based records have to be restructured and to be equipped with a centralized electronic record management system, an electronic document management and archiving system shall be implemented to archive electronic documents and to step towards the electronic patient record [8, 9]. A model of the current sub information system for archiving paper-based and electronic documents may help to assess the situation and to elaborate requirements. A model of the target subsystem may help to state the requirements more precisely and to present them in discussions.

We describe the creation of the model of the current state. The model includes the central patient record archive, local patient record archives at the departments for internal medicine, children surgery, conservative dentistry, children dentistry, and orthopedics of the UKL, and parts of the information system that are used to fill and maintain the records. The modeling activities were preceded by on-site surveys in these departments. We interviewed persons responsible for record management. In some of the departments record management is performed by nurses, in others by specialized secretaries. Most of the interviews were not guided by a standardized questionnaire but the interviewer was a trained 3LGM² user.

The model is built in five major steps:

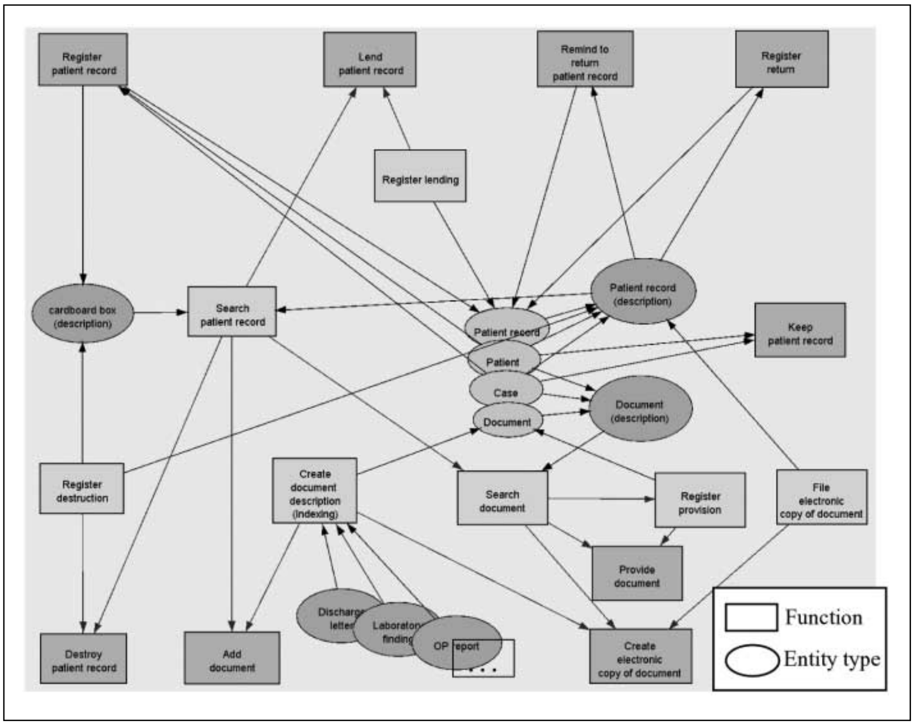

Fig. 3 The Domain Layer of the sub information system for archiving

Describing the Domain Layer: Enterprise Functions and Entity Types

From the survey data we gathered enterprise functions and associated entity types (Fig. 3). We describe five of the functions to illustrate how we identified them:

1) *Register patient record.* When a patient is admitted to the hospital the first time, the ward creates a record. After discharge of the patient his/her record is archived. Here we identified the first function: Patient records have to be registered so that they can easily be found when someone needs them. Of course, the records have to be filed when they have been registered. Since information about a record is created, the function itself leads to the first entity type: *Patient record.*
   Several information about the patient and his/her case are included in registering because these information are needed when records are searched. Thus we need two additional entity types: *Patient* and *Case.*
2) *Lend patient record.* When a patient is readmitted to the hospital, then his or her patient record, which has been created during an earlier case must be available on the ward. Usually the record has already been archived, i. e. it has to be lent out to the ward. We subdivided the function into two subfunctions:
   a) *Search patient record.* An archived record that ought to be used has to be searched for in the archive. This subfunction is performed in every archive and is part of several other functions. It uses information about the patient and the case the record belongs to and information about the patient record itself, e. g. its location.
   b) *Register lending.* Before handing out the patient record to a client (ward, laboratory, etc.), the information about this patient record is updated to reflect the new location respectively the department that is now responsible.
3) *Remind to return patient record.* Very often patient records are not returned to the archive when patients have been discharged. The wards have to be reminded to return the records. The reminder includes information about the patient, the case and the record itself. The record information in the archive is updated to reflect the reminder activity.
4) *Register return.* The return of a patient record is registered, i. e. the record information is updated. To find the record information that is to be updated, parts of the record information itself and, sometimes, the patient and her/his case are needed.
5) *Add document.* Very often documents are added to records that are already archived. These documents are delivered to the archive where they are indexed and filed in a record. We subdivided the function into two subfunctions:
   a) *Search for patient record.* This subfunction was already described above. Of course, the record that will receive a document has to be searched for.
   b) *Create document description (indexing).* Index information helps to find documents. They are created when documents are added to the record.

When modeling information systems it may be necessary to distinguish organizational units. They are also modeled on the domain layer, but not shown on the modeling canvas.

During the process of identifying functions and entity types we complied with the following principles:

1) A function usually needs information and creates or changes information. In 3LGM² terminology: A function uses or updates at least one entity type. The most of the functions use AND update entity types.
2) An entity type indicates that the information system contains *information about* real life objects or abstract things of a specific kind but not the objects or abstract things itself.
3) Functions may be subdivided into subfunctions (part-of-association). A set of subfunctions should describe its superior function completely.

Describing the Logical Tool Layer: Application Components

In the previous section we described WHAT is done in the context of patient record archiving. We now focus on WHEREBY functions are performed and WHERE data are stored. In this section we start with application components on the logical tool layer.

Our survey yielded a heterogeneous information system with a lot of different computer-based and paper-based application components. We describe those of the archive section of the Department for Internal Medicine to illustrate how we identified application components (Fig. 4):

Patient record archive INZ. The patient record archive of the Department for Internal Medicine (INZ[e]) is modeled as an application component with four subcomponents:

a) *Record management system INZ (INZ/Access).* This computer-based application component is used to document the locations of patient records and is needed for the function *Search patient record.* It will be replaced by the new *Record management system INZ (INZ/ZA/Archive)* after a period of parallel usage.
b) *Record management system INZ (INZ/ZA/Archive).* This computer-based application component is used more generally: to perform *Register patient record, Lend patient record, Register return,* and *Add document.*
c) *Record lending management system INZ.* Every lending of a record is documented on a lending form that is destroyed after the return of the record. This paper-based application component is used in parallel to the *Record management system INZ (INZ/ZA/Archive)* to perform *Lend patient record* and *Register return*.
d) *Archive operation INZ.* This paper-based application component coordinates the operation of the previously described components. It is needed for communicating with wards, outpatient care sections, external institutions like insurances, etc. There are no computer-based interfaces for communication with these partners.

The superordinate component *Patient record archive INZ* only aggregates its subcomponents and does not have an own database system or own interfaces.

The application component details are described using the tabs in the associated dialog windows. The list of application components described above includes two elements named *Record management system INZ:* One component based on the software product MS Access customized by staff members of the archive and one based

[e] INZ is the abbreviation of the German department name "Zentrum für Innere Medizin"

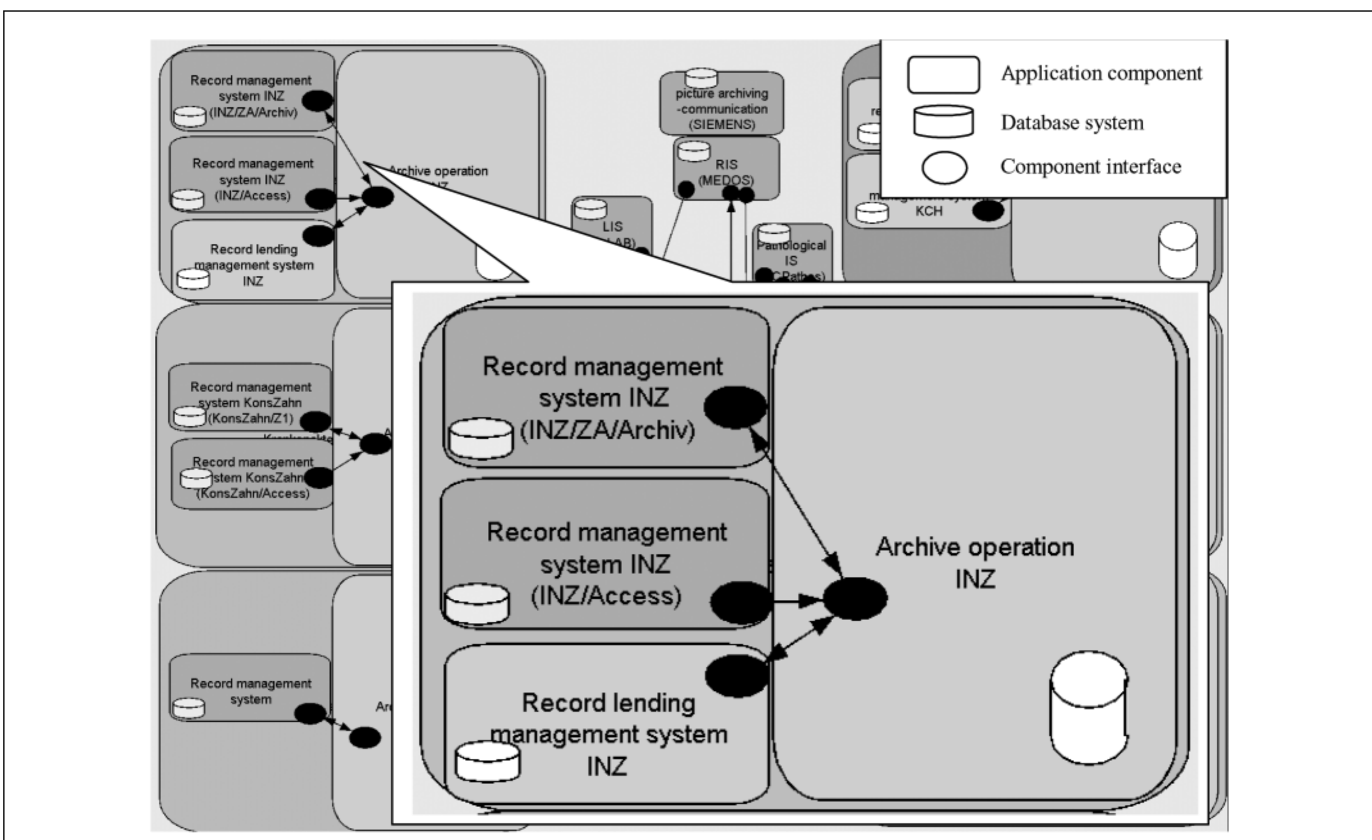

Fig. 4 A part of the logical tool layer of the sub information system for archiving

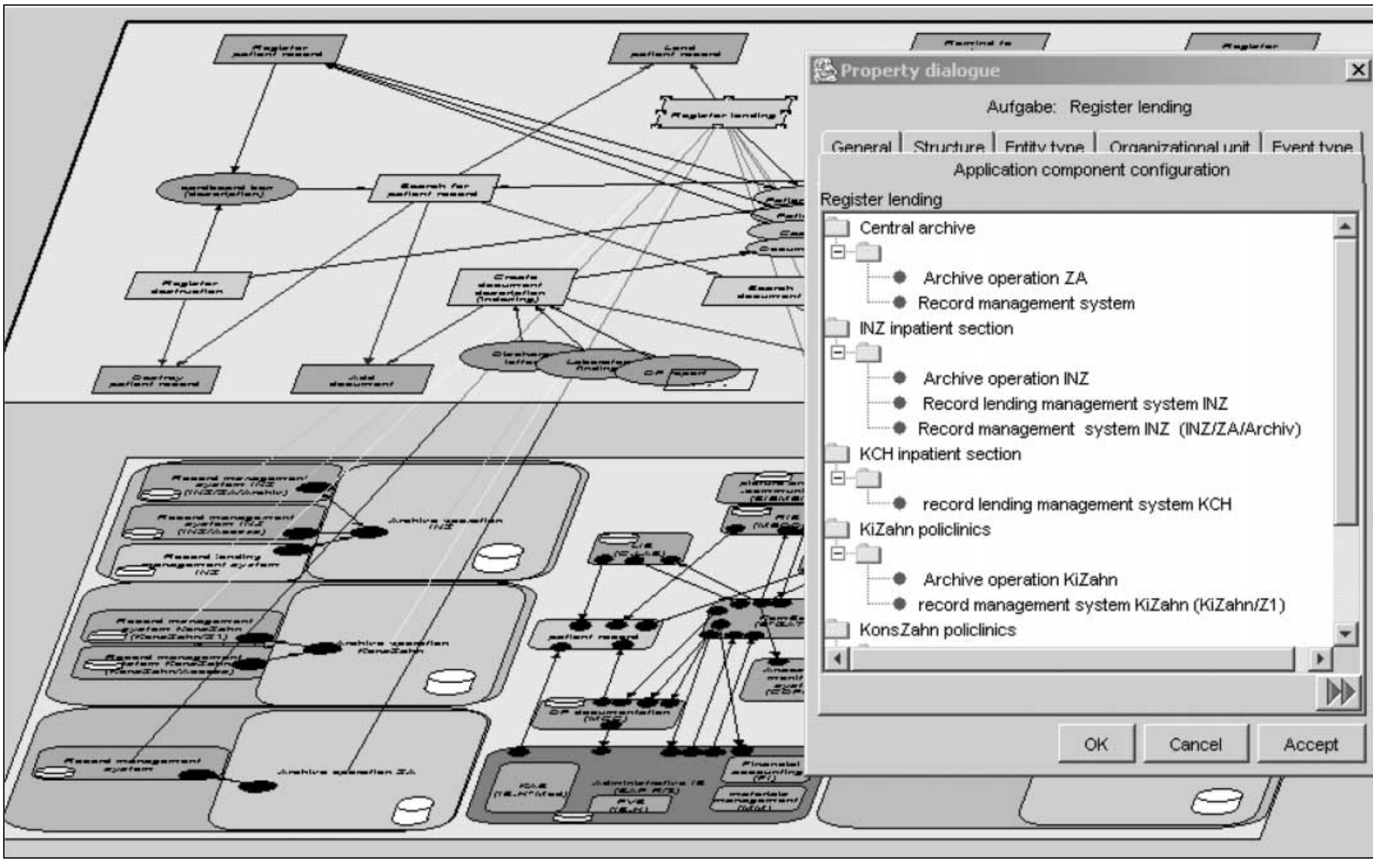

Fig. 5 Links between the enterprise function *Register lending* and application components

on a software product created by the head of the central archive department. The corresponding application program descriptions hold information about the adaptation of the software product. All of the four subcomponents of *Patient record archive INZ* have a database system respectively document collection. The computer-based record management system that was implemented using MS Access, for example, stores datasets of the type *Patient record description INZ/Access* in its database. The paper-based record lending management system stores documents of the type *Lending form.* The dataset type and document type model elements are mainly needed to link database systems to entity types on the domain layer (see section about linking application components to enterprise functions and entity types).

Another important part of the application component description is that about interfaces. The application component *Archive operating INZ,* for example, communicates documents of the type *Lending form* with the component *Record lending management system INZ* when records are returned to the archive. It communicates messages of the type *Visual data output (Patient Record description)* when patient records are searched. Similar to dataset type and document type model elements, message type elements can be linked to entity types.

The model shows the dual role of document type elements: They cover the storage aspect AND the communication aspect of paper-based data representations. Computer-based data representations are divided in dataset types for storing and message types for communicating.

During the process of identifying application components we complied with the following principles:

1) Application components can be imagined as aggregation of
   - algorithms that describe how a function (or a number of functions) is to be performed

   and, not necessarily but typically,
   - a storage that holds data representing the information needed to perform the functions(s) and/or
   - component interfaces to communicate data to other application components.
2) Identifying application components depends on the required level of detail of a model. An application component does not necessarily have to have a database component/document collection AND interfaces, but should have at least one of them. But: Application components that aggregate other application components via the Part-of-association should not have own storage components or interface components.

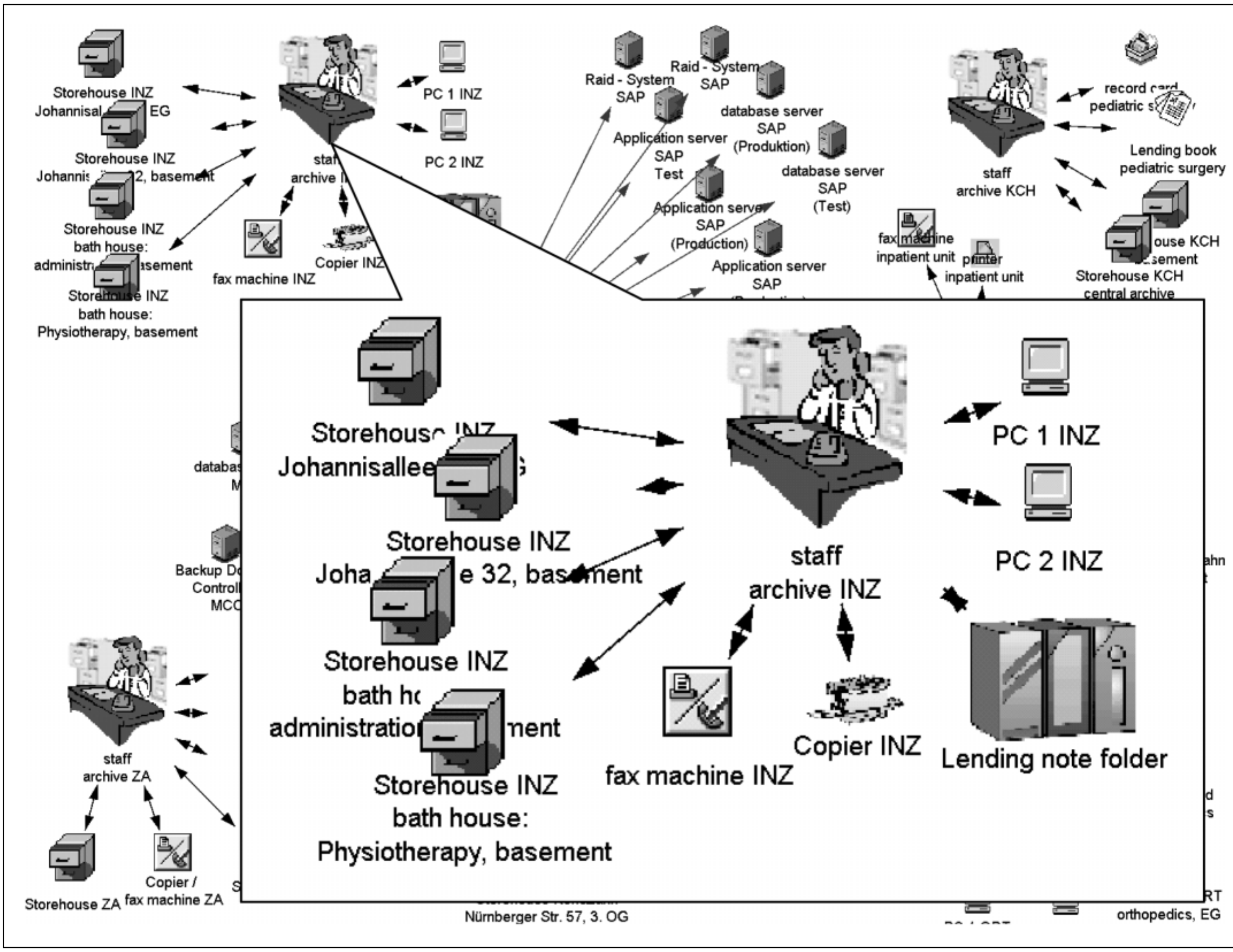

Fig. 6 A part of the physical tool layer of the sub information system for archiving

Linking Application Components to Enterprise Functions and Entity Types

In the chapter about requirements and approaches we emphasized that the 3LGM² defines relations to link the different layers. The links between the domain layer and the logical tool layer describe

- which application components are needed to perform what functions and
- which database systems respectively document collections store what entity types, i. e. information.

In the Department for Internal Medicine the function *Register lending* is performed by the application components *Archive operating INZ, Record lending management system,* and *Record management system INZ (INZ/ZA/Archive)* in combination (Fig. 5). The function *Remind to return patient record* is performed by the components *Archive operating INZ* and *Record lending management system* in combination.

Via the dataset types and document types the entity types are linked to database systems respectively document collections. The entity type *Patient record,* for example, is linked to the dataset types *Lending form* and *Patient record description INZ/Access.* Via message types and document types the entity types are linked to component interfaces. The entity type *Patient record* is additionally linked to the message types *Manual data input (Patient record description)* and *Visual data output (Patient record description).*

During the process of identifying links between application components and enterprise functions/entity types we complied with the following principles:

1) As defined in the 3LGM², functions are linked to application component configurations, which may contain one or several application component(s). This concept is used to express that performing a function may need more than one application component. There may be alternative configurations for the same function.

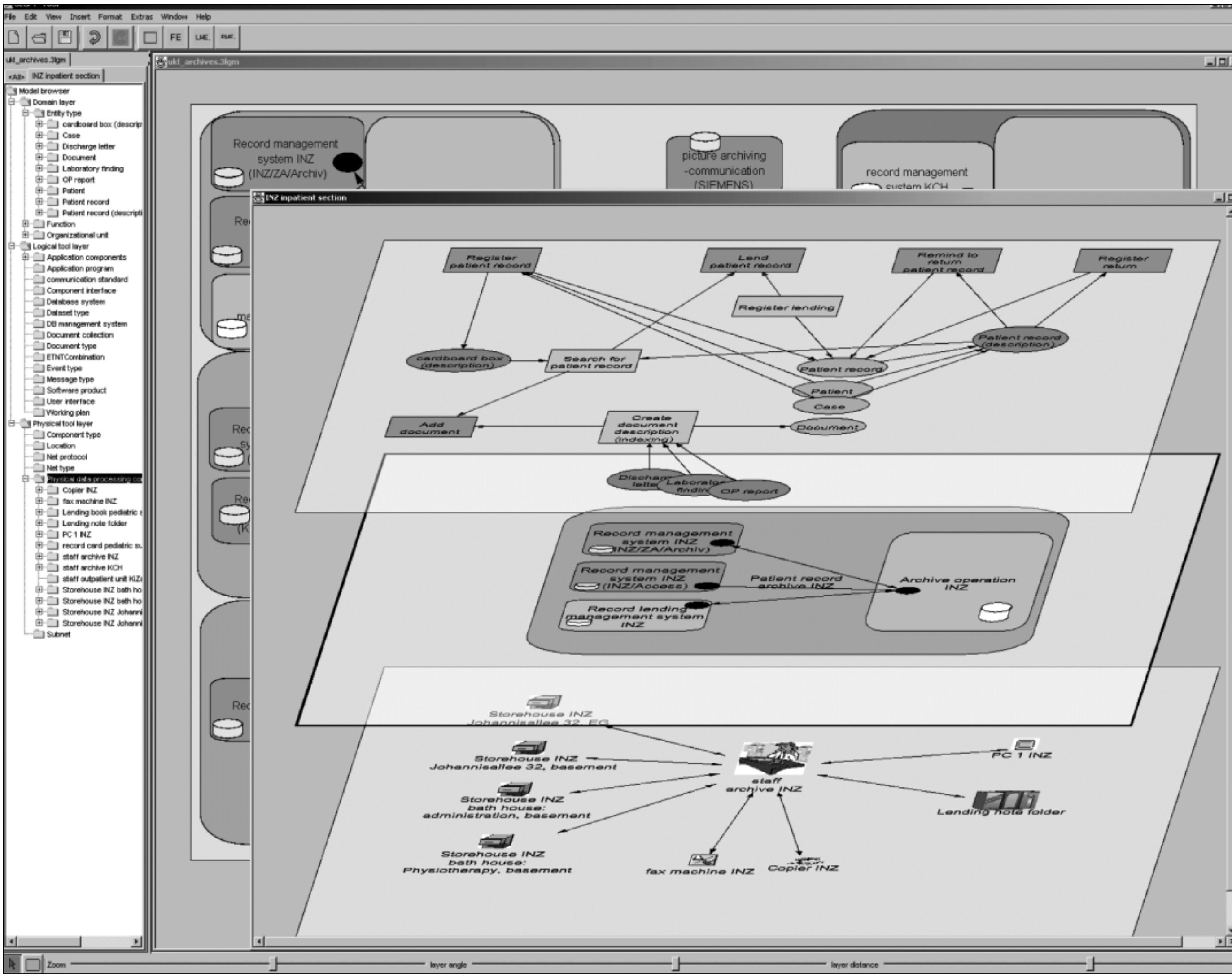

Fig. 7 Result of the analysis searching the sub information system for patient record archiving in INZ, inpatient section, presented in a submodel window

2) As defined in the 3LGM_, entity types are linked via dataset type and document type to database systems. Message type and document type are used to link entity types to component interfaces.

Describing the Physical Tool Layer: Physical Data Processing Components

On the physical tool layer we describe the physical data processing components that are needed to operate the application components described on the logical tool layer. For the Archive Section of the Department for Internal Medicine our survey yielded the following physical data processing components (Fig. 6):

1) *Staff INZ.* In the archive section there work two persons.
2) *Storehouse INZ (…).* The patient records of the Department for Internal Medicine are stored in four storehouses. The primary storehouse contains inpatient records of the past three years; the secondary storehouses contain older inpatient records and outpatient records.
3) *PC 1 INZ, PC 2 INZ.* There are two personal computers that are not connected to a network.
4) *Lending note folder.* Lending notes are stored in a standard office folder.
5) *Fax machine INZ, Copier INZ.* A fax machine and a copier are used as physical base for copying and sending ordered documents.

The model expresses ONE of two roles of staff members in relationship to the information system: As part of the information system they are modeled on the physical tool layer. We describe persons as part of the information system when they execute organizational plans to have paper-based application components run. The 3LGM^2 does not define elements to model staff members as users of the information system.

As every data transfer is carried out by the staff members we modeled physical data transfer connections between *Staff INZ* and the other physical data processing components.

Also for physical data processing components there are associated dialog windows to describe details. The component *Primary storehouse INZ,* for example, is located on the ground floor of building no. 4271. It has the component type *Storehouse.* For computers there is a tab with technical details about the operating system, memory, etc. *PC 1 INZ,* for example, has the operating system Windows 98, a memory capacity of 128 MB, and a Pentium III processor.

During the process of identifying physical data processing components we complied with the following principles:

1) All devices like PCs, server computers, copiers, fax machines, etc. should be modeled as physical data processing components.
2) Depending on the required level of detail devices may be combined to abstract components without modeling every physical part. For example, instead of record shelves, cartons, etc. a model could contain elements describing their combination as a storehouse.
3) When persons act as part of the information system, i. e. execute organizational plans to have paper-based application components run, they should be modeled as physical data processing components.

Linking Physical Data Processing Components to Application Components

The links between the logical tool layer and the physical tool layer describe what physical data processing components the application components are installed on, i. e. what persons, computers, record shelves, etc. are needed to have the application components run.

In the Department for Internal Medicine the paper-based application component *Record lending management system INZ* is installed on the physical data processing components *Lending note folder, PC 1 INZ* and *Staff INZ* in combination. The computer-based application component *Record management system INZ (INZ/ZA/Archive)* is installed on the component *PC 1 INZ.*

At a first glance it may be confusing that we added the item *Staff INZ* to the combination of physical data processing components needed to run *Record lending management system INZ,* but not to a combination of components needed to run *Record management system INZ (INZ/ZA/Archive).* We had in mind that

- an application program controlling a computer-based application component is usually executed by computers that in most cases also include the physical media to store data, while
- an organizational plan controlling a paper-based application component is usually executed by persons who need additional physical media to store data.

During the process of identifying links between physical data processing components and application components we complied with the following principles:

1) As defined in the $3LGM^2$, application components are linked to physical data processing component configurations, which may contain one or several physical data processing component(s). This concept is used to express that an application component may be installed on more than one physical data processing components. There may be alternative configurations for the application component.
2) Computer-based application components usually need computer(s) but no persons to run; paper-based application components usually need persons and additional physical tools.

Using the Model: Analyzing and Presenting

In the previous chapter we described the steps for creating a $3LGM^2$ model. Now we want to show how the modeling effort may remunerate by extracting useful information from our model. We try to answer two questions.

(Q1) Our model describes the sub information system for patient record archiving in several organizational units. The director of the Department of Internal Medicine wants to know the subsystem only for his department.

In terms of the model structure the questions may be formed like “Which functions are linked to the organizational unit *INZ, inpatient section,* which entity types are linked to these functions, which application components (including detail information about database systems, interfaces, etc.) are linked to these functions, and which physical data processing components are linked to the selected application components?”

As already described in the chapter introducing the $3LGM^2$ tool, there is a library of predefined queries. In case there is no predefined query fitting our question, one might define it in the analysis dialog and add it to the analysis repository. Figure 7 shows the result of the query in a submodel window.

(Q2) An important aspect of information management is the reliability of application components. In this context one might ask: Is searching of patient records at the department INZ, inpatient section, still supported by computer-based application components when PC 2 INZ fails?

To answer this question we have to create a query similar to the first one. We would have to add an additional restriction: From the functions only *Search for patient record* shall be included. The result submodel shows that for searching patient records three application components are needed. One of them *(Record management system INZ (INZ/Access))* is installed on *PC 2 INZ.* Without any detail information the answer to our question would be “NO”. But the description text for *Record management system INZ (INZ/Access)* explains that the information found only in this application component is transferred to *Record management system INZ (INZ/ZA/Archive).*

The presentation capabilities are very important for using query results. In the chapter introducing the $3LGM^2$ tool we already mentioned the multilevel presentation feature. The three levels of a model and connections between them can be presented in one diagram. It is not difficult to imagine that the graphical presentation features are less informative for complex models. The submodel feature makes the graphical presentation more valuable: As a presentation of the entire archive model proved to be confusing, a presentation of a submodel for each organizational unit was useful.

The XSL-based export feature mentioned above may also increase the reusability of models: via text lists and tables the model data can easily be transferred to text processors.

Discussion

We presented a prototype of the graphical 3LGM² tool which is to build comprehensive 3LGM² compliant models and intends to support information management especially in hospitals. We reported on modeling the sub-information system of the UKL for archiving paper-based and electronic documents of the patient record. Using the model as an example, means for analyzing the model and reducing its complexity could be demonstrated.

But in our modeling activities we have been confronted with two major challenges:

1) *Hospital functions and entity types:* As already stated in [10], determining hospital functions and entity types "is a task the complexity of which should not be underestimated". It is important to keep in mind, that an information system model should contain only those enterprise functions that deal with information processing. Hence, the entity types represent information about e. g. patient records, but do not represent the records themselves. Publications like [11] may help to create better 'formulas' to find functions and entity types. A more valuable approach may be the development of reference models for the domain layer of hospital information systems. They could base on common standards [12] like the Reference Information Model (RIM) of the HL7 standard [13] and the Good European Health Record (GEHR) [14] and on the requirements index for information processing in hospitals [15].
2. *Paper-based application components:* 3LGM² claims to consider not only computer-based but also paper-based information and data processing in hospitals. Consequently we introduced paper-based application components as an analogy to computer-based application components. Whereas it is no problem to identify a computer-based application component, which is an installation of a software product, it is difficult to identify paper-based analogies. It turned out, that (sub-)organizations dealing with information processing in the hospital can usually be considered to be paper-based application components. Again the demand for better 'formulas' respectively for reference models arises. A reference model for the logical tool layer should also provide users with catalogues of message- and event-type combinations – basing on HL7, DICOM, etc.; this would reduce efforts for modeling interfaces dramatically.

The model proposed here has been developed in the context of applying for getting financial support for a new digital Document Management and Archiving System for the UKL. Diagrams like in Figure 5 have been used to illustrate the current and future way of integration. But our experience showed clearly, that it cannot be recommended to use 3LGM² and the 3LGM² tool for illustration purposes only. Real benefit gains only from continuously documenting the knowledge of the information system in a clearly structured way and therefore from using the 3LGM² tool as a valuable repository of knowledge about the hospital and its information system. The usage of the 3LGM² tool itself seemed to be rather trouble-free. Certainly, this can be traced back to the fact that the developers of the 3LGM² and the 3LGM² tool were part of the project team and could directly help if any technical or handling problems appeared. The general usability of the 3LGM² tool has to be analyzed in the context of an evaluation study incorporating information managers of other healthcare organizations.

Currently we are working on some enhancements to increase usability; this is strongly influenced by the collaboration with Austrian consultants applying the tool in two projects. A template library for common model elements, based on the reference models mentioned above, may shorten the modeling process. Exporting features as mentioned in the chapter about the 3LGM² tool will ease to reuse the model data in other documents. The current research activities on the 3LGM² include, for example, the coverage of business and communication processes and the coverage of different architectural styles for information systems, especially for better modeling the interaction between application components.

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

We thank the people doing their important jobs in the patient record archives for their cooperation. We also thank our partners at the University for Health Sciences, Medical Informatics and Technology and the Information Technology for Healthcare GmbH (Innsbruck, Austria) for fruitful discussions about the functionality of the 3LGM² tool.

References

1. 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. 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.
3. Berg M. Patient care information systems and health care work: A sociotechnical approach. International Journal of Medical Informatics 1999; 55: 87-101.
4. Quatrani T. Visual Modeling with Rational Rose 2002 and UML. Boston: Addison-Wesley; 2002.
5. Heverhagen T, Tracht R. Using Stereotypes of the Unified Modeling Language in Mechatronic Systems. In: Proc. of the 1. International Conference on Information Technology in Mecha-

tronics, ITM'01, October 1-3, 2001, Istanbul, UNESCO Chair on Mechatronics. Istanbul: Bogazici University, Istanbul, Turkey; 2001; pp. 333-8.
6. Ebert J, Süttenbach R, Uhe I. Meta-CASE in Practice: a Case for KOGGE. In: Olive A, Pastor JA (eds.). Advanced Information Systems Engineering, Proceedings of the 9th International Conference, CAiSE'97, Barcelona, Catalonia, Spain, June 16-20, 1997. 1250 ed. Berlin; 1997; pp. 203-16.
7. Noy NF, Grosso W, Musen MA. Knowledge-Acquisition Interfaces for Domain Experts: An Empirical Evaluation of Protégé-2000. Twelfth International Conference on Software Engineering and Knowledge Engineering (SEKE2000), Chicago, IL, 2000.
8. Dujat C, Schmücker P, Haux R, Winter A. Digital Optical Archiving of Medical Records in Hospital Information Systems – A Practical Approach towards the Electronic Patient Record? Methods Inf in Med 1995; 34 (5): 489-97.
9. Atkinson CJ, Peel VJ. Transforming a hospital through growing, not building, an electronic patient record system. Methods Inf Med 1998; 37 (3): 285-93.
10. Winter A, Haux R. A Three-Level Graph-Based Model for the Management of Hospital Information Systems. Methods Inf Med 1995; 34 (4): 378-96.
11. Spewak SH, Hill SC. Enterprise Architecture Planning: Developing a blueprint for Data, Applications and Technology. New York: John Wiley & Sons; 1992.
12. Klein GO. Standardization of health informatics – results and challenges. In: Haux R, Kulikowski C (eds.). Yearbook of Medical Informatics 02. Stuttgart: Schattauer 2002, pp. 103-114.
13. Schadow G, Russler DC, Mead CN, McDonald CJ. Integrating medical information and knowledge in the HL7 RIM. Proc AMIA Symp 2000: 764-8.
14. Griffith SM, Kalra D, Lloyd DS, Ingram D. A portable communicative architecture for electronic healthcare records: the Good European Healthcare Record project (Aim project A2014). Medinfo 1995; 8 Pt 1: 223-6.
15. Ammenwerth E, Buchauer A, Haux R. A Requirements Index for Information Processing in Hospitals. Methods Inf Med 2002; 41 (4): 282-8.

Correspondence to:
Thomas Wendt
Institute for Medical Informatics, Statistics and Epidemiology
University of Leipzig
Liebigstr. 27
04103 Leipzig
Germany
E-mail: thomas.wendt@imise.uni-leipzig.de