Modeling Interdependencies between Information Processes and Communication Paths in Hospitals

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

Summary

Objectives: For planning hospital information systems it is important to recognize the interrelation between business processes and the communication needs between supporting application systems. We therefore present an approach to model, visualize and analyze those interdependencies.

Methods: The approach is based on the concepts defined in 3LGM², a meta-model to describe health information systems (HIS). An information process is defined as a sequence of functions using or updating information; a communication path as a sequence of communication links between interfaces belonging to application systems. The search for communication paths belonging to an information process is interpreted as an all-pairs shortest-paths problem. To solve this problem the Floyd-Warshall algorithm is applied.

Results: The resulting algorithm has been implemented as function of the 3LGM² tool, a tool to create 3LGM² compliant models. With it, it is possible to interactively define information processes at the domain layer and to analyze step by step whether the infrastructure at the logical tool layer is sufficient to communicate necessary data between application systems.

Conclusions: The presented approach enables the representation and analysis of dependencies between information processes and communication paths. With it, the HIS architecture is directly associated with the business needs. This is an important condition for the systematic planning of hospital information systems.

Keywords

Hospital information systems, architectural models, process modeling, information processes, communication paths

Methods Inf Med 2006; 45: 216–24

1. Introduction

The optimization and redesign of business processes in hospitals is an important challenge for hospitals in the next years. Although the support of business processes by computer-based information processing tools is not the cure-all [1], the implementation of business processes without them is not imaginable. This is where hospital information management must decide which application components suit best, which communication interfaces and communication links are necessary, which communication standards and which message types must be supported – in short: what is an optimal HIS architecture to support the business processes and fulfill the resulting information needs. In particular, the introduction of new application components and the replacement of legacy systems [2] need a detailed specification concerning the communication with other existing application components. This applies especially for hospital information systems following an architectural style, stamped by a great variety of application components of many different vendors which all have their own database systems. Such an architecture implicates distributed and redundant data storage, a serious problem concerning e.g. data integrity.

For business process modeling and simulation, there are a lot of useful tools available, which concentrate on the domain layer where conceptual models can be built, considering information processing tools as resources that do not have to be specified any more (see e.g. [3-5]). There are no tools available for modeling especially the information processing aspects of business processes and the consequences for the communication between application systems, and thus, can give answers to information management questions arising in this context, like:

- On what paths can data, representing needed information, be transported from the storing database system to the processing application systems?
- On what paths can data, representing produced information, be transported from the processing application component to the storing database system (or systems in case there is redundant storing)?
- Does the hospital information system of a certain hospital provide a suitable infrastructure (communication links, interfaces, message types, application components, etc.) for transporting the data? Will some additionally planned components be sufficient for constructing the needed infrastructure?

To overcome this problem we will present an approach that associates the information needs of business processes with their supporting application systems and the communication between them. In order to focus especially on the information processing aspects of business processes, we refer to those processes as information processes. The presented approach aims to visualize those processes, and also to evaluate if there are weaknesses concerning the information processing infrastructure which hinder a smooth implementation of the business processes.

After a short review on the state-of-the-art about process management in healthcare in chapter 2, we define information processes and communication paths in the context of 3LGM² – a metamodel to model hospital information systems. In chapter 3 we present a method to automatically derive communication paths from information processes. This chapter is accompanied by a fictive example. In chapter 5 we show the application within the 3LGM^2 tool. The paper finishes with a short discussion.

2. Process Management in Health Care

Traditionally, patient care is a process involving a lot of different actors (departments, persons, etc.). These actors must efficiently work together to improve quality of care [6] and patient satisfaction, and to reduce costs. To improve those processes, methods and tools for business process modeling have been successfully applied to health care projects (e.g. [7-9]), particularly to solve optimization problems in bounded areas.

As hospital functions are more and more supported by computer-based application systems, the integration of those information processing tools is also a great challenge. For health care a lot of standards for integrating application systems (see e.g. [10-13]) and also architectural approaches [14, 15] have been developed and are nowadays applied successfully in clinical routine. In information management sciences, these considerations are nowadays subsumed under the modern concepts of 'enterprise application integration' (EAI) (see e.g. [16, 17]) related to more technical aspects, and 'enterprise architecture planning' (EAP) (see e.g. [18-20]) related to more strategic aspects.

Even though today's hospital information systems are primarily department-oriented, the idea of process-oriented hospital information systems has become widely accepted. Software industries tend to develop integrated application systems which cover a wide range of hospital functions [21]. The usage of workflow management systems is discussed [22, 23], and the development of component-based hospital information systems is pushed [24, 25]. In the context of implementing clinical pathways, the IT architecture of hospital information systems must reflect those business processes [26, 27].

As hospital information systems get increasingly complex and probably will change to health information systems in the near future [28], hospital architecture planning will become a challenging task for information management and needs efficient methods and tools to design process-oriented architectures of hospital information systems which do not only support isolated hospital functions but also the information flow needed for efficient patient care.

Therefore, we need efficient planning tools to represent and analyze the static architecture as well as dynamical aspects of hospital information systems.

3. Information Processes and Communication Paths in the Context of 3LGM^2

3.1 Formalizing 3LGM^2 Concepts Using Algebraic Structures

In [29] we presented the three-layer graph-based meta model (3LGM^2) which defines an ontology to describe the static architecture of hospital information systems. Now we want to add means for representing and analyzing dynamic aspects to this approach by introducing information processes and communication paths. In the following we therefore will use and formalize those 3LGM^2 concepts out of [29] using algebraic structures that are relevant for their definition and the mapping of information processes to communication paths.

The *domain layer* of 3LGM^2 describes a hospital independent of its implementation by its *enterprise functions*. An enterprise function is a kind of regulation for human or technical action to reach a certain goal. Enterprise functions therefore 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 an enterprise function to an entity type can be in a using or an updating manner.

In the following $\underline{ET}$ denotes a finite set of entity types, $\underline{EF}$ a finite set of enterprise functions, and $\underline{ACCESSES} \subset \underline{EF} \times \underline{ET} \times \{\text{using, updating}\}$ a relation that describes which entity types are used or/and updated by which enterprise function.

The *logical tool layer* concentrates on *application components* supporting enterprise functions. Application components are responsible for the processing, storage and transportation of data. *Component interfaces* ensure the communication among application components. A component interface can receive or send messages of a certain *message type.* For the communication among application components *communication links* can be defined as relations between two communication interfaces, one being the sender of a message, the other one being the receiver. Each communication link is specified by the entity types which in fact are communicated.

In the following, let $\underline{AC}$ denote a finite set of application components, and $\underline{CI}$ a finite set of component interfaces. Furthermore let **owns**: $\underline{CI} \rightarrow \underline{AC}$ be a function denoting the application component, which owns a certain component interface.

Between concepts of the different layers there exist so-called *interlayer relationships*, which describe the dependencies between model elements belonging to different layers. In this paper, we use the following interlayer relationships:

- An enterprise function is supported by a set of application components, expressed by the relation $\underline{SUPPORTS} \subset \underline{ACC} \times \underline{EF}$, $\underline{ACC} \subset P(\underline{AC})$[a]. Elements of $\underline{ACC}$ are called application component configurations.
- Entity types can be transported by a communication link between two application components, expressed by the relation $\underline{CL} \subset \underline{CI} \times \underline{CI} \times P(\underline{ET})$[b].

In the following, these concepts will be used to define information processes and communication paths.

[a] $P(\underline{AC})$ denoting the power set of $\underline{AC}$
[b] $P(\underline{ET})$ denoting the power set of $\underline{ET}$

3.2 3LGM² Information Process

We refer to an information process as a sequence of enterprise functions using and/or updating information about entities. Looking at 3LGM², an information process describes dynamic aspects of the domain layer. Formally, we define a 3LGM² information process as follows:

Again let $\underline{ET}$ be a set of entity types, $\underline{EF}$ be a set of enterprise functions, $\underline{ACCESSES}$ be a relation denoting which entity types are used or updated by which enterprise functions.

A tuple

$(ef_1, ..., ef_n)$, $ef_i \in \underline{EF}$, $i = 1..n$; $n \in \mathbb{N}$

is called *3LGM² information process* if, and only if

$\forall\, ef_i, i = 1..n-1: \exists(ef_i, et', \text{updating}), (ef_j, et'', \text{using}) \in \underline{ACCESSES}: et' = et'' \wedge j > i.$

This condition expresses a rather loose inner connectivity within the sequence of functions, saying that each function – except the last one of the sequence – must update at least one entity type, which has to be used by one of the succeeding functions.

Example

Given

$\underline{ET}$:= {order, result, patient, case}
$\underline{EF}$:= {patient admission, order entry, creation and dispatch of results, receipt and presentation of results}
$\underline{ACCESSES}$:= {(order entry, order, updating), (creation and dispatch of results, order, using), (creation and dispatch of results, result, updating), (receipt and presentation of results, result, using)},

the tuple

IP := (order entry, creation and dispatch of results, receipt and presentation of results)

is a 3LGM² information process. A graphical illustration is given in Figure 1. The entity types themselves do not belong to the information process, and are just shown to understand the condition which has to be fulfilled. The enterprise functions are taken from [30].

At this point we deliberately restrict us to rather elementary information processes without concurrencies and branches in order to reduce the complexity at the logical tool layer. If we want to examine more complex information processes we have to decompose them into those elementary ones. An information process is regarded on a rather abstract level. It just describes the interdependencies of enterprise function in terms of information. In this respect our definition of 'process' considerably differs from traditional views.

From an information process we can derive so-called information process steps. Let IP be an information process, IP = $(ef_1, \ldots, ef_n)$, , $ef_i \in \underline{EF}$, $i = 1..n$; $n \in \mathbb{N}$. Furthermore let **position**: $\underline{IP} \times \underline{EF} \rightarrow \mathbb{N}$ be a function which returns the position of *ef* within IP for a given enterprise function *ef*.

A tuple

(ef', ef'', et), $ef', ef'' \in \underline{EF}$, $et \in \underline{ET}$

is called *information process step of* IP if and only if

$(ef', et, \text{updating}) \in \underline{ACCESSES} \wedge (ef'', et, \text{using}) \in \underline{ACCESSES} \wedge \textbf{position}(IP, ef'') > \textbf{position}(IP, ef')$

Example

Let us again look at the information process

IP = (order entry, creation and dispatch of results, receipt and presentation of results).

We can determine two information process steps of IP, namely

ips_1 = (order entry, creation and dispatch of results, order) and

ips_2 = (creation and dispatch of results, receipt and presentation of results, result)

3.3 An Elementary 3LGM² Communication Path

We refer to an elementary communication path as a sequence of communication links between application components, necessary to satisfy the information needs given by an information process. An elementary communication path describes dynamic aspects of the logical tool layer of 3LGM².

Given are $\underline{AC}$ a set of application components, $\underline{CI}$ a set of components interfaces, $\underline{CL}$ a set of communication links and **owns** a function denoting the application component, which owns a certain component interface.

A tuple

$(cl_1, ..., cl_n)$, $cl_i = (ci_1^{cl_i}, ci_2^{cl_i}, \{et_1^{cl_i}, ..., et_n^{cl_i}\}) \in \underline{CL}$, $i := 1..n$; $n, m \in \mathbb{N}$

is called *elementary communication path* if and only if

$\forall\, cl_j, cl_{j+1} \in \underline{CL}, j := 1..(n-1)$: $\textbf{owns}\,(ci_2^{cl_j}) = \textbf{owns}\,(ci_1^{cl_{j+1}})$

The condition expresses that for each pair of communication links cl_j, cl_{j+1} where cl_j is the direct predecessor of cl_{j+1}, the receiver of cl_j must be owned by the same application component as the sender of cl_{j+1}. This condition

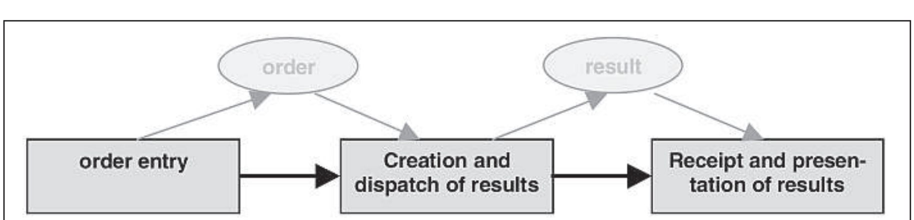

Fig. 1 Example of an information process (ovals: entity types; rectangles: enterprise functions)

ensures the inner connectivity of the communication path which is in fact guaranteed by the application components (see Fig. 2).

Example

Given

$\underline{AC}$:= (OES, CS, LIS, DMAS),

$\underline{CI}$:= (ci_1, ci_2, ci_3, ci_4, ci_5, ci_6, ci_7, ci_8, ci_9, ci_{10}),

$\underline{CL}$:= {(ci_1, ci_2, {order}), (ci_3, ci_4, {order}), (ci_5, ci_6, {result}), (ci_7, ci_8, {result}), (ci_9, ci_{10}, {result})},

$\underline{ACC}$:= ({OES}, {LIS}, {DMAS}).

owns with **owns**(ci_1) = **owns**(ci_{10}) := OES, **owns**(ci_2) = **owns**(ci_3) **owns**(ci_6) = **owns**(ci_7) := CS, **owns**(ci_4) = **owns**(ci_5) := LIS, **owns**(ci_8) = **owns**(ci_9) := DMAS

The tuples

ecp_1 := ((ci_1, ci_2, {order}), (ci_3, ci_4, {order})),

ecp_2 := ((ci_5, ci_6, {result}), (ci_7, ci_8, {result}), (ci_9, ci_{10}, {result}))

are elementary communication paths. A graphical illustration is given in Figure 3, which shows two elementary communication paths differentiated through the colors of the component interfaces and communication links. The light-gray communication interfaces and communication links represent ecp_1, the dark-gray communication interfaces and communication links represent ecp_2.

4. Mapping 3LGM² Information Processes on Elementary 3LGM² Communication Paths

If we understand a 3LGM² information process to be a kind of requirements specification for the logical tool layer, it is not sufficient to describe information processes and communication paths independently. The more important aspect is what kind of communication between application components is necessary to enable the execution of an information process. The structure of the resulting communication path depends on the individual architecture of the logical tool layer, i.e. where data are stored, processed and communicated.

In the following, we present an algorithm which – for a given 3LGM² information process – will result in a set of corresponding elementary communication paths at the logical tool layer. Therefore, we will define three steps:

- Step 1: Derive a communication matrix for each entity type belonging to an information process step of the given information process.
- Step 2: Calculate all shortest paths between communication interfaces.
- Step 3: Find an elementary communication path for each information process step.

In the following, again let $\underline{ET}$, $\underline{EF}$, $\underline{ACCESSES}$, $\underline{AC}$, $\underline{CI}$, $\underline{CL}$ and **owns** be as before, IP := (ef_1, …, ef_n), $n \in \mathbb{N}$ an information process and $\underline{IPS}$ the set of information process steps of IP.

4.1 Step 1: Derive a Communication Matrix for each Entity Type Belonging to an Information Process Step of the Given Information Process

For each $et \in \underline{ET}$ we can define a communication matrix R^{et} representing all communication links which can transport *et*. The rows and columns of such a matrix represent the communication interfaces in $\underline{CI}$. A communication matrix is defined as follows:

A matrix

$$R^{et} := (r_{ij}^{et}),\ i, j := 1..|\underline{CI}|$$

with

$$r_{ij}^{et} = \begin{cases} 1 \Leftrightarrow (\exists (ci_i, ci_j, \underline{ET}^*) \in \underline{CL}: \\ et \in \underline{ET}^*) \vee \mathbf{owns}(ci_i) = \mathbf{owns}(ci_j) \\ 0 \text{ otherwise} \end{cases}$$

is called *communication matrix* for *et*.

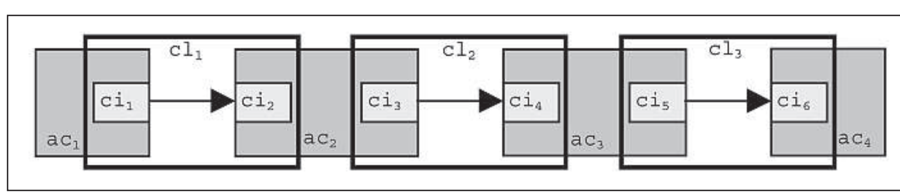

Fig. 2 The inner connectivity of a communication process (ci: communication interface; cl: communication link; ac: application component)

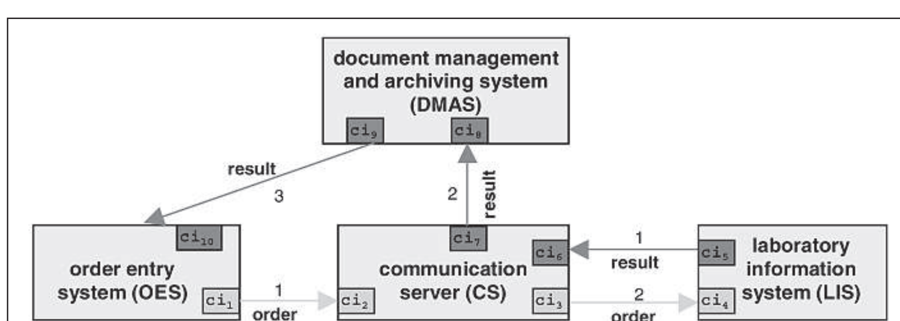

Fig. 3 Elementary communication paths. Large boxes: application components, small boxes: communication interfaces; arrows: communication links, arrow labels: entity types transmitted. Arrow numbers: sequence of communication links within a communication process

	ci_1	ci_2	ci_3	ci_4	ci_5	ci_6	ci_7	ci_8	ci_9	ci_{10}
ci_1	0	0	0	0	0	0	0	0	0	1
ci_2	0	0	1	0	0	1	1	0	0	0
ci_3	0	1	0	0	0	1	1	0	0	0
ci_4	0	0	0	0	1	0	0	0	0	0
ci_5	0	0	0	1	0	1	0	0	0	0
ci_6	0	1	1	0	0	0	1	0	0	0
ci_7	0	1	1	0	0	1	0	1	0	0
ci_8	0	0	0	0	0	0	0	0	1	0
ci_9	0	0	0	0	0	0	0	1	0	1
ci_{10}	1	0	0	0	0	0	0	0	0	0

Fig. 4 Communication matrix of the entity type result.

Example

In our example we will get the communication matrix R^{result} for the entity type result as shown in Figure 4.

4.2 Step 2: Calculate All Shortest Paths between Communication Interfaces

Let us now return to our information process $IP = (ef_1, \ldots, ef_n)$, the set of information process steps $\underline{IPS}$, and the communication matrix R^{et} for each $ips = (ef_1, ef_2, et) \in \underline{IPS}$. For each *ips* we can determine a corresponding elementary communication path ecp^{ips} as follows:

The elementary communication path will have to start at a communication interface owned by an application component supporting ef_1 and will have to end at a communication interface owned by an application component supporting ef_2. All communication interfaces belonging to the process will have to send or receive the entity type *et*.

To derive an elementary communication path between a start and an end component interface to transport *et*, we have to know possible paths between any two component interfaces. To solve this all-pair shortest-path problem we apply the Floyd-Warshall algorithm [31].

This algorithm needs a cost matrix C^{et} and a predecessor matrix P^{et}.

Therefore, let us return to the communication matrices R^{et}. Each matrix R^{et} is an adjacency matrix of a directed, labeled graph. In our case, the rows and columns represent the component interfaces $ci_1, \ldots, ci_n \in \underline{CI}$, $n = |\underline{CI}|$. Communication between two component interfaces ci_i and ci_j is possible if $r_{ij} = 1$, i.e., if there exists a communication link between these component interfaces or if the component interfaces belong to the same application component. For the sake of simplicity we fix the cost for each existing communication link to 1. So, it is sufficient to transform the given adjacency matrix to a modified adjacency matrix, where all 1 and all 0 of the diagonal persist, and all other 0 are replaced with ∞. This modified adjacency matrix is our *cost matrix*.

$$C^{et} := (c_{ij}^{et}) \text{ with}$$

$$(c_{ij}^{et}) := \begin{cases} 1, & \text{if } r_{ij}^{et} = 1 \\ 0, & \text{if } i = j \\ \infty, & \text{if } r_{ij}^{et} = 0 \quad \wedge i \neq j \quad i,j = 1,\ldots,n \end{cases}$$

The *predecessor matrix* can be derived from the cost matrix.

$$P^{et} := (p_{ij}^{et}) \text{ with}$$

$$(p_{ij}^{et}) := \begin{cases} ci_i, & \text{if } c_{ij}^{et} = 1 \\ ci_i, & \text{if } i = j \\ 0, & \text{otherwise} \quad i,j = 1,\ldots,n \end{cases}$$

Example

The cost matrix and the predecessor matrix of the entity type *result* look as shown in Figure 5.

	ci_1	ci_2	ci_3	ci_4	ci_5	ci_6	ci_7	ci_8	ci_9	ci_{10}
ci_1	0	∞	∞	∞	∞	∞	∞	∞	∞	1
ci_2	∞	0	1	∞	∞	1	1	∞	∞	∞
ci_3	∞	1	0	∞	∞	1	1	∞	∞	∞
ci_4	∞	∞	∞	0	1	∞	∞	∞	∞	∞
ci_5	∞	∞	∞	1	0	1	∞	∞	∞	∞
ci_6	∞	1	1	∞	∞	0	1	∞	∞	∞
ci_7	∞	1	1	∞	∞	1	0	1	∞	∞
ci_8	∞	∞	∞	∞	∞	∞	∞	0	1	∞
ci_9	∞	∞	∞	∞	∞	∞	∞	1	0	1
ci_{10}	1	∞	∞	∞	∞	∞	∞	∞	∞	0

	ci_1	ci_2	ci_3	ci_4	ci_5	ci_6	ci_7	ci_8	ci_9	ci_{10}
ci_1	ci_1	0	0	0	0	0	0	0	0	ci_1
ci_2	0	ci_2	ci_2	0	0	ci_2	ci_2	0	0	0
ci_3	0	ci_3	ci_3	0	0	ci_3	ci_3	0	0	0
ci_4	0	0	0	ci_4	ci_4	0	0	0	0	0
ci_5	0	0	0	ci_5	ci_5	ci_5	0	0	0	0
ci_6	0	ci_6	ci_6	0	0	ci_6	ci_6	0	0	0
ci_7	0	ci_7	ci_7	0	0	ci_7	ci_7	ci_7	0	0
ci_8	0	0	0	0	0	0	0	ci_8	ci_8	0
ci_9	0	0	0	0	0	0	0	ci_9	ci_9	ci_9
ci_{10}	ci_{10}	0	0	0	0	0	0	0	0	ci_{10}

Fig. 5 Left: cost matrix of the entity type result; right: predecessor matrix of the entity type result

	ci_1	ci_2	ci_3	ci_4	ci_5	ci_6	ci_7	ci_8	ci_9	ci_{10}
ci_1	0	0	0	0	0	0	0	0	0	ci_{10}
ci_2	ci_7	0	ci_3	0	0	ci_6	ci_7	ci_7	ci_7	ci_7
ci_3	ci_7	ci_1	0	0	0	ci_6	ci_7	ci_7	ci_7	ci_7
ci_4	ci_5	ci_5	ci_5	0	ci_5	ci_5	ci_5	ci_5	ci_5	ci_5
ci_5	ci_6	ci_6	ci_6	ci_4	0	ci_6	ci_6	ci_6	ci_6	ci_6
ci_6	ci_7	ci_2	ci_3	0	0	0	ci_7	ci_7	ci_7	ci_7
ci_7	ci_8	ci_2	ci_3	0	0	ci_6	0	ci_8	ci_8	ci_8
ci_8	ci_9	0	0	0	0	0	0	0	ci_9	ci_9
ci_9	ci_{10}	0	0	0	0	0	0	ci_8	0	ci_{10}
ci_{10}	ci_1	0	0	0	0	0	0	0	0	0

	ci_1	ci_2	ci_3	ci_4	ci_5	ci_6	ci_7	ci_8	ci_9	ci_{10}
ci_1	0	∞	∞	∞	∞	∞	∞	∞	∞	1
ci_2	5	0	1	∞	∞	1	1	2	3	4
ci_3	5	1	0	∞	∞	1	1	2	3	4
ci_4	7	3	3	0	1	2	3	4	5	6
ci_5	6	2	2	1	0	1	2	3	4	5
ci_6	5	1	1	∞	∞	0	1	2	3	4
ci_7	4	1	1	∞	∞	1	0	1	2	3
ci_8	3	∞	∞	∞	∞	∞	∞	0	1	2
ci_9	2	∞	∞	∞	∞	∞	∞	1	0	1
ci_{10}	1	∞	∞	∞	∞	∞	∞	∞	∞	0

Fig. 6 Left: path matrix of the entity type result; right: distance matrix of the entity type result

As result of the Floyd-Warshall algorithm we get the path matrix $W^{et} = (w_{ij}^{et})$ and the distance matrix $D^{et} = (d_{ij}^{et})$ for each $et \in \underline{ET}$ as shown in Figure 6.

The distance matrix quotes one shortest path between two communication interfaces ci' and ci''. If there are more shortest paths between two communication interfaces it belongs to the processing sequence of the communication interfaces which path is determined.

The path matrix may be used to track back the path between ci' and ci'' as follows: the element p_{ij}^{et} of P^{et} represents that communication interface which is predecessor of ci_j on the shortest path from ci_i to ci_j. So, we can determine the predecessor of ci'' on the shortest path from ci' to ci'', the predecessor of the predecessor of ci'' on the shortest path from ci' to ci'' etc. until we reach ci'. If we read the resulting sequence of communication interfaces starting from the back, we get a tuple of communication interfaces $(ci', \ldots, ci'')$ representing the shortest path between ci' and ci'' communicating et, and thus, the way from the application component which owns ci' to the application component which owns ci''. The resulting tuple of communication interfaces can easily be transformed into a tuple of communication links.

4.3 Step 3: Find an Elementary Communication Path for each Information Process Step

Let $(ci_1, \ldots, ci_m)$, $m \in \mathbb{N}$ be a tuple of communication interfaces resulting from step 2.

A tuple

$ecp := (cl_1, \ldots, cl_{n-1})$ with $cl_i := (ci_i, ci_{i+1}, \underline{ET}^*) \in \underline{CL}$

is the corresponding tuple of communication links. *ecp* is an elementary communication process.

Now let $\underline{acc}'$, $\underline{acc}''$ be application component configurations with $(\underline{acc}', ef')$, $(\underline{acc}'', ef'') \in$ SUPPORTS[c]. We assume that application components belonging to one application component configuration are able to communicate unrestrictedly with each other and thus, the application component configuration can be considered as one (large) application component.

Furthermore, let

$\underline{CI}_{acc'} := \{ci \in \underline{CI} \mid \mathbf{owns}(ci) \in \underline{acc}'\}$ be the set of all component interfaces belonging to any of the application components of $\underline{acc}'$,

$\underline{CI}_{acc''} := \{ci \in \underline{CI} \mid \mathbf{owns}(ci) \in \underline{acc}''\}$ be the set of all component interfaces belonging to any of the application components of $\underline{acc}''$.

To find a shortest path between any application component of $\underline{acc}'$ and any application component of $\underline{acc}''$ to communicate *et* we can transform the distance matrix D^{et} to $D^{et'}$:

$D^{et'} := (d_{ij}^{et'})$ with

$$(d_{ij}^{et'}) := \begin{cases} d_{ij}^{et}, \text{ if } ci_i \in \mathrm{CI}_{acc'}, \\ \qquad \wedge\, ci_j \in \underline{\mathrm{CI}}_{acc''} \\ \infty, \quad otherwise \end{cases}$$

The shortest path between any component interface belonging to an application component of $\underline{acc}'$ and any component interface belonging to an application component of $\underline{acc}''$ is the path between those two component interfaces ci' and ci'' where $d(ci', ci'') = \min(d_{ij})$.

We can now determine a tuple of communication interfaces $(ci_1, \ldots, ci_k)$, $k \in \mathbb{N}$, from the path matrix representing a shortest path between ci' and ci'', $ci' = ci_1$, $ci'' = ci_k$ and transform this tuple into a tuple of communication links as shown.

Example

In our example we will look now at the following information process step:

[c] If there exist more then one application component configuration for an enterprise function, it is the task of the user of the algorithm to choose one of them.

	ci_1	ci_2	ci_3	ci_4	ci_5	ci_6	ci_7	ci_8	ci_9	ci_{10}
ci_1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
ci_2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
ci_3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
ci_4	7	∞	∞	∞	∞	∞	∞	∞	∞	6
ci_5	6	∞	∞	∞	∞	∞	∞	∞	∞	5
ci_6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
ci_7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
ci_8	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
ci_9	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

Fig. 7
Transformed distance matrix for the entity type result

(creation and dispatch of results, receipt and presentation of results, result).

The enterprise function creation and dispatch of results is supported by the application component configuration {LIS}, the enterprise function receipt and presentation of results is supported by the application component configuration {OES}. In this example, all application component configurations consist of just one application component. The transformed distance matrix looks as shown in Figure 7.

$min(d_{ij}) = 5, i = 5, j = 10.$

The shortest path between LIS and OES is the path ($ci_5, ci_6, ci_7, ci_8, ci_9, ci_{10}$) from ci_5 to ci_{10}. The corresponding elementary communication path is ((ci_5, ci_6, {result}, (ci_7, ci_8, {result}, (ci_9, ci_{10}, {result})).

5. Application

The presented algorithm was implemented within the 3LGM² tool [32], a tool for modeling 3LGM² compliant models of hospital information systems. At the domain layer, information processes can be modeled in-

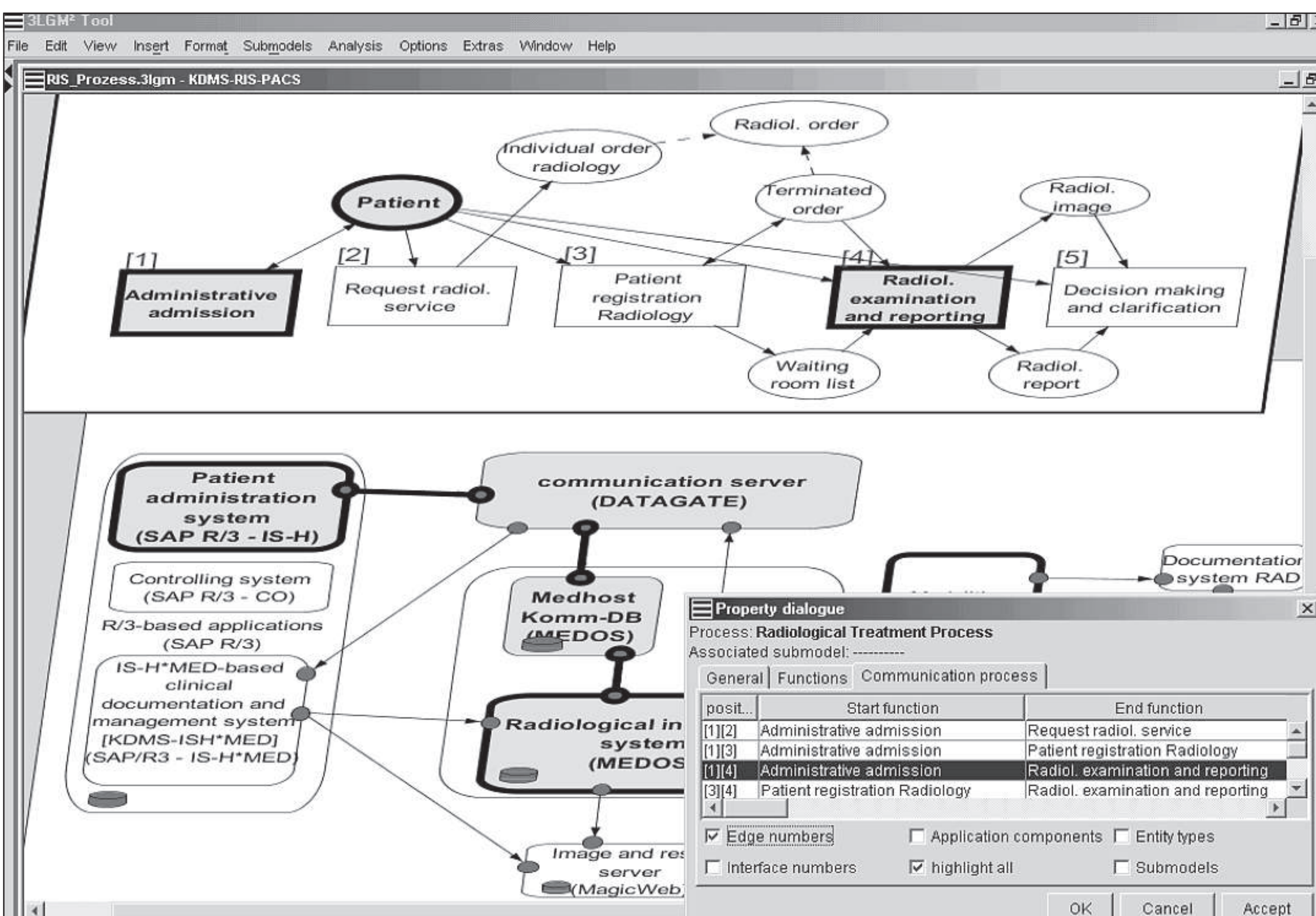

Fig. 8 Modeling and analyzing information processes with the 3LGM² tool

teractively. The 3LGM² tool evaluates the correctness of an information process during the modeling activity and proposes just such functions to be included to the information process which fulfill the precondition defined in section 3.2. The information process in Figure 8 consists of the functions numbered by [1-5]. The information process step (Administrative Admission, Radiol. Examination, {patient}) that is looked at is highlighted. Our algorithm determines automatically, that for performing a radiological examination, patient data which has been generated during patient admission has to be transported from the Patient Management System to the Radiology Information System via the communication server and the Medhost component. In Figure 8 the associated elementary communication path at the logical tool layer is highlighted. The communication links used to transport necessary data from the PMS to the RIS are also numbered consecutively.

In the dialogue window of the information process you can navigate through the information process steps and trace the communication paths at the logical tool layer.

6. Discussion

In this paper we presented an approach which can support information managers modeling the architecture of their information system, focusing on the interdependencies between information processes and communication paths. For a given 3LGM² model of a hospital information system and a given information process, this approach enables us to evaluate if there are weaknesses at the logical tool layer, which hinder the implementation of an information process. In this respect, it covers new features for information system modeling. Whereas traditional business modeling tools like ARIS [33] or IDEF [34] mainly focus on the optimization and redesign of business processes on a small-grained level including aspects of time and available resources, our approach concentrates on an architectural level looking at the IT infrastructure. For it, the analysis does not include process instances and their variability. Even if this approach originates in the scope of the 3LGM² research activities it may be applicable to other approaches if the necessary concepts can be modeled.

Particularly to answer questions like those mentioned in the introduction, our approach can be useful because the necessary analysis of the hospital information system can be done (semi-)automatically, provided that there is a tool available which implements the algorithm. The user should be able to analyze the model interactively from several points of view without being forced to know the underlying algorithm. For this reason, the 3LGM² tool which supports modeling of hospital information systems was extended. Each 3LGM² model created with the 3LGM² tool provides the information necessary for that algorithm.

The definition of information processes restricts the common interpretation of the concept 'business process' as we just look at the information processing aspects of enterprise functions as well as information-based dependencies between enterprise functions. Other events like the availability of physical resources or the termination of activities are not considered as these kinds of events are outside the scope of our approach.

Within a hospital information system this approach will mainly be used for documentation and presentation purposes. Particularly if a communication server is installed, the communication paths at least for the computer-based part of the information system are rather clearly defined. The visualization may nevertheless help to get a better insight. It will even be more interesting if we look at regional health information systems (rHIS) which in most cases do not have any integration engines to control the communication between the components involved. In this context, an analysis if e.g. a digital image of an modality located in hospital A can be transported to the diagnosing system of hospital B may facilitate the overall planning of the rHIS (also see [35]).

Furthermore, the algorithm presented is suitable to answer further questions of communication and data quality, e.g. the suitability of a hospital information system architecture to ensure data integrity. Data integrity is especially at risk if the same data are redundantly stored in different database systems and/or if multiple application systems are allowed to modify the same data. In this case, every activity which changes data must be followed by a set of communication activities to update all database systems storing these data. On an architectural level, our algorithm can detect if these communication activities can be carried out, provided that the underlying 3LGM² model keeps information about which data are stored in which database systems and which application components are permitted to update which data. 3LGM² offers the required concepts and relations.

Admittedly, as an analysis tool our approach offers no solutions to ensure data integrity on a semantic or technical level. For this, we need, on the one hand data dictionaries and terminology servers as integral parts of an information system and, on the other hand techniques for the management of distributed database systems. Unfortunately, both are not yet in wide-spread use in the health information systems area.

References

1. Hammer M, Champy J. Reengineering the corporation. New York: Harper Business; 1994.
2. Brodie ML, Stonebraker M. Migrating legacy systems. San Francisco: Morgan Kaufmann; 1995.
3. 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.
4. Ferstl OK, Sinz EJ. Modeling of Business Systems Using SOM. In: Bernus P, Mertins K, Schmidt G, editors. Handbook on Architectures of In formation Systems. International Handbook on Information Systems. Berlin: Springer; 1998: pp 339 -58.
5. Scheer AW. ARIS Toolset: A Software Product Is Born. Information Systems 1994; 19 (8): 607-24.
6. Ball MJ, Douglas JV. Redefining and improving patient safety. Methods Inf Med 2002; 41 (4): 271-6.
7. Staccini P, Joubert M, et al. Modelling health care processes for eliciting user requirements: a way to link a quality paradigm and clinical information system design. Int J Med Inf 2001, 64 (2-3): 129-42.
8. Maij E, van Reijswoud VE, et al. A process view of medical practice by modeling communicative acts. Methods Inf Med 2000, 39 (1): 56-62.

9. Jerva M. BPR (business process redesign) and systems analysis and design: making the case for integration. Top Health Inf Manage 2001, 21(4): 30-7.
10. Health Level 7. HL7 Reference Information Model V01–20; http://www.hl7.org.
11. Siegel EL, Channin DS. Integrating the Healthcare Enterprise: a primer. Part 1. Introduction. Radiographics 2001; 21 (5): 1339-41.
12. Flanders AE, Carrino JA. Understanding DICOM and IHE. Semin Roentgenol. 2003; 38 (3): 270-81.
13. Dudeck J. Aspects of implementing and harmonizing healthcare communication standards. Int J Med Inf 1998; 48 (1-3): 163-71.
14. CEN TC251. N-97–024, Healthcare Information System Architecture Part 1 (HISA) Healthcare Middleware Layer – draft. Report no. PrENV12967–1 1997E. Brussels: European Commitee for Standardisation. http://www.centc251.org.
15. Klingler A. An Open, Component-based Architecture für Healthcare Information Systems. In: Hasman A, Blobel B, Dudeck J, Engelbrecht R, Gell G, Prokosch H-U, editors. Medical Infobahnfor Europe. Amsterdam: IOS Press; 2000. pp 997-1001.
16. Grimson J, Grimson W, et al. A CORBA-based integration of distributed electronic healthcare records using the synapses approach. IEEE Trans Inf Technol Biomed 1998, 2(3): 124-38.
17. Heitmann KU. The Role of Communication Servers in the Architeure of Healthcare Information Systems. In: Dudeck J, Blobel B, Lordieck W, Bürkle T, editors. New Technologies in Hospital Information Systems. Amsterdam: IOS Press; 1997. pp 156-62.
18. Zachman JA. A framework for information systems architecture (Reprint). IBM systems journal 1999; 38 (2 & 3): 454-70.
19. Spewak SH, Hill SC. Enterprise Architecture Planning: Developing a blueprint for Data, Applications and Technology. New York: John Wiley; 1992.
20. Chief Information Officer Council. A Practical Guide to Federal Enterprise Architecture. Boston: Chief Information Officer Council c/o Rob C. Thomas, U.S. Customs Service 7681 Boston Boulevard Springfield, VA 22153; 2001 February 2001. http://www.cio.gov.
21. Haux R, Seggewies C, Baldauf-Sobez W, et al. Soarian™ – Workflow Applied for Health Care. Methods Inf Med 2003; 42: 25-36.
22. Toussaint PJ, Lodder H. Component-based development for supporting workflows in hospitals. Int J Med Inf 1998; 52 (1-3): 53-60.
23. Wong ST, Tjandra D, Wang H, Shen W. Workflow-enabled distributed component-based information architecture for digital medical imaging enterprises. IEEE Trans Inf Technol Biomed 2003; 7 (3): 171-83.
24. Degoulet P, Marin L, Lavril M, et al. The HEGP component-based clinical information system. Int J Med Inf 2003; 69 (2-3): 115-26.
25. Xu Y, Sauquet D, Degoulet P, Jaulent MC. Component-based mediation services for the integration of medical application. Artif Intell Med 2003; 27 (3): 283-304.
26. Chu S, Cesnik B. A three-tier clinical information systems design model. Int J Med Inf 2000; 57 (2-3): 91-107.
27. Maij E, Toussaint PJ, Kalshoven M, Poerschke M, Zwetsloot-Schonk JHM. Use Cases and DEMO: aligning functional features of ICT infrastructure to business processes. Int J Med Inf 2002, 65: 179-191.
28. Kuhn KA, Giuse DA. From Hospital Information Systems to Health Information Systems. Methods Inf Med 2001; 40: 275-87.
29. 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 (5): 544-51.
30. Ammenwerth E, Buchauer A, Haux R. A Requirements Index for Information Processing in Hospitals. Methods Inf Med 2002; 41 (4): 282-8.
31. Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms. Cambridge: MIT Press; 2001.
32. Wendt T, Häber A, Brigl B, Winter A. Modeling Hospital Information Systems (Part 2): Using the 3LGM^2 Tool for Modeling Patient Record Management. Methods Inf Med 2004; 43 (3):256-67.
33. Scheer AW. ARIS Toolset: A Software Product Is Born. Information Systems 1994; 19 (8), 607-24.
34. Pitt M. A Generalized Simulation System to Support Strategic Resource Planning In Healthcare. In: Andradóttir S, Healy KJ, Withers DH, Nelson BL. Proceedings of the 1997 Winter Simulation Conference; 1997, 1155-1162. http://www.informs-cs.org/wsc97papers/1155.PDF (Last visit: February 8, 2005)
35. Winter A, Brigl B, Heller O, Mueller U, Struebing A, Wendt T (2004): Supporting Information Management for Regional Health Information Systems by Models with Communication Path Analysis. In: 2004 IDEAS Workshop on Medical Information Systems: The Digital Hospital (IDEAS-DH 2004). IEEE Computer Society, pp 139-46.

Correspondence to:
Dr. Birgit Brigl
Institute for Medical Informatics, Statistics and Epidemiology
University of Leipzig
Härtelstr. 16-18
04107 Leipzig
Germany
E-mail: birgit.brigl@imise.uni-leipzig.de