MEDINFO 2001
V. Patel et al. (Eds)
Amsterdam: IOS Press
© 2001 IMIA. All rights reserved

A UML-based Ontology for Describing Hospital Information System Architectures

Alfred Winter[a], Birgit Brigl[a], Thomas Wendt[a]

[a] *Institute for Medical Informatics, Statistics and Epidemiology, University of Leipzig, Leipzig, Germany*

Abstract

To control the heterogeneity inherent to hospital information systems the information management needs appropriate hospital information systems modeling methods or techniques. This paper shows that, for several reasons, available modeling approaches are not able to answer relevant questions of information management. To overcome this major deficiency we offer an UML-based ontology for describing hospital information systems architectures. This ontology views at three layers: the domain layer, the logical tool layer, and the physical tool layer, and defines the relevant components. The relations between these components, especially between components of different layers make the answering of our information management questions possible.

Keywords:

Hospital Information systems, information management, ontology

Introduction

Problems

Heterogeneity is inherent to hospitals. A hospital consists of various organizational units with different information processing profiles for various types of healthcare professionals using various computer-supported or conventional information processing tools. The care process of each patient is rather individual and demands for high interoperability among organizational units (internal communication). This is also true for the external communication (e. g. to insurances, practitioners, etc.). The fact that a patient is not only customer but that his or her health depends on successful care, which depends strongly on information available, leads to a high responsibility concerning information processing.

The heterogeneity of a hospital is reflected by its hospital information system as one of its socio-technical subsystems. Because of different requirements, their information processing tools base on various hardware platforms and software products offered by several vendors. In addition, legacy systems (see [1]) and modern information processing tools exist side by side.

To control this heterogeneity the information management needs tools for modeling, i.e. describing several aspects of hospital information systems. Suited models may have to answer e. g. the following questions of an information manager:

- What tasks of the health professionals does a certain health professional workstation (HPW) ([2]) in a certain ward of 'my' hospital support?
- How many different applications do 'my' health professionals need for completing their work at that ward?
- Which application is used for order entry and results reporting at that HPW? Who is the vendor of the respective software, what database management system is used, and what servers are engaged?
- The communication between the database on the server and the HPW happens to be erroneous due to timeout errors. Which network components could be contributing to those errors?
- What interfaces to the laboratory system are used for order and results communication? Is HL7 used as protocol on a server to server connection or is there a proprietary client based interface to the lab's intranet-server installed? How is visual integration ([3]) guaranteed in the latter case?
- To which network domain does the HPW belong and who is the responsible domain administrator?
- In order to realize another HPW an additional personal computer shall be installed at that ward. What plugs, wires and interface cards are necessary to connect the computer to the network at that site?

There are a lot of information system modeling approaches available. Unfortunately, models built using these approaches are not able to answer these questions for several reasons:

- Some approaches describe only what layers and views have to be modeled but not how (see e. g. [4],[5]).
- Some methods just concentrate on a domain layer,

considering information processing tools as resources that don't have to be specified (see e. g. [6], [7).

- Some approaches orientate just on syntactical integrity constraints to guarantee some kind of consistency uncovering modeling failures, but say nothing about the quality of a information system itself (see e. g. [8]).

Due to these deficiencies, we come to the following aims:

Aims of this paper

In this paper, we present an ontology based on [9} for describing hospital information systems architectures that allows the description of the essential information system components, their relationships and their dependencies. We show that hospital information systems models based on that ontology are sufficient to answer relevant information management questions as presented above. For the presentation of the ontology we use the Unified Modeling Language (UML) (see [10]), the examples are shown as intuitive graphical notations.

An ontology for describing hospital information systems architectures

Three layers of hospital information systems

In a hospital information system we can distinguish three layers of information management:

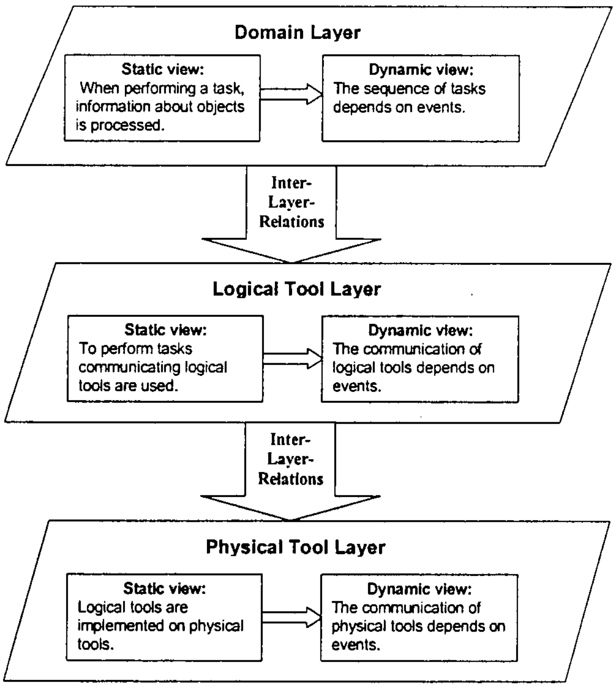

Figure 1 - Three Layers of hospital information systems.

The *domain layer* describes a hospital independent of its implementation. In the static view, a hospital is a accumulation of its functions or tasks (see [11]). To fulfil these tasks information about objects (e. g. patients, findings, etc.) is used, generated, or deleted. The dynamic view describes processes, i. e. the performance of tasks dependent on events.

To perform tasks, a hospital uses logical information processing tools, which are gathered in the static view of the *logical tool layer*. The dynamic view describes the invocation of and the communication among application components dependent on events.

The *physical tool layer* describes what physical tools for information processing are applied in the hospital, and how these have to be networked to ensure the communication described in the logical tool layer.

There exist various relations among the classes of these three layers, which we summarize as inter-layer-relationships.

Figure 1 shows the three layers of hospital information systems. In the following we take a look at the static view of the layers.

Domain layer

In the domain layer (see figure 2), two concepts are used: object types and tasks. An object type is the abstraction of objects, which share the same attributes. For example, PATIENT, CASE, or LABORATORY RESULT may be object types. A task (enterprise function, see [12]) is an instruction for human or machine action with no beginning and no end. We may understand a task as duty arising from the enterprise's goals and strategies. For example, PATIENT ADMISSION, NURSING, or TREATMENT may be tasks. Object types and tasks can be refined. A task accesses an object type to get the information needed for its execution. This access may just use information (type = reading) or change information (type = writing) of an object.

It depends on a certain hospital, which object types and which tasks are modeled. Reference models may offer recommendations about important object types and task for certain kind of hospitals.

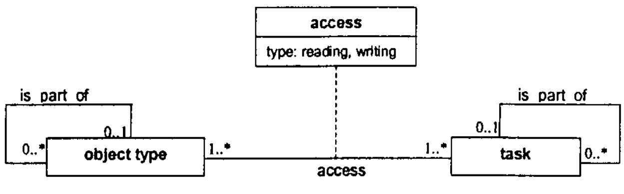

Figure 2 - Domain Layer- static view.

Fig. 3 shows an example of a domain layer. Rectangles represent object types and ovals represent tasks. An arrow from an object type to a task marks a writing access, from a task to an object type a reading access. Note that in the static view there are no direct relations between tasks. This would be part of the dynamic view. The domain layer is restricted to information about objects, and to tasks to be performed. There are no documents, databases, or organizational units. This would be part of the logical or physical tool layer.

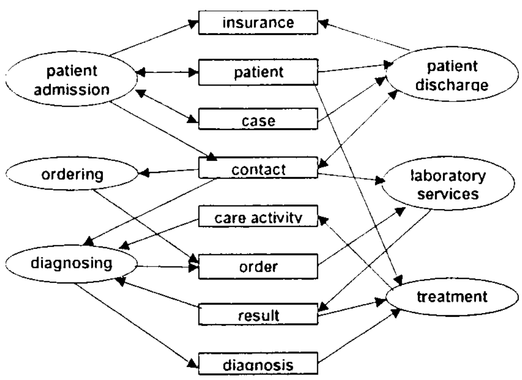

Figure 3 - Domain layer - example.

Logical tool layer

The logical tool layer (see figure 4) is built around application components which may be installed and adapted software products or conventional working plans. On this layer we describe, how information about objects is logically stored, and how tools have to communicate to ensure the access on information as described in the domain layer. Application components may have a local database system to store data. They are controlled by application programs, which are adapted software products (this is what we can buy). A software product may be installed multiple on one or more physical data processing components, as result we get several different application components. Communication interfaces ensure the communication among application components based on message types (like HL7 messages [13]), but also between a component and a user (user interfaces). Application components may be refined.

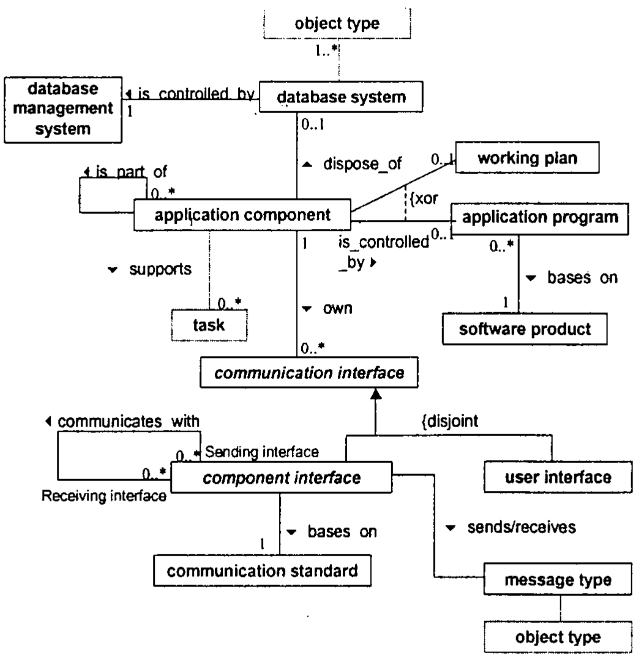

Figure 4 - Logical Tool Layer - static view.

Fig. 5 shows an example of a logical tool layer. In this example we just look at the application components depicted as large rounded rectangles and the relationships between them via communication interfaces (small rectangles), depicted as arrows. The communication is based on HL7 or proprietary interfaces. The direction of the arrow represents the direction of the communication. For reasons of clearness, this example does not include database system aspects.

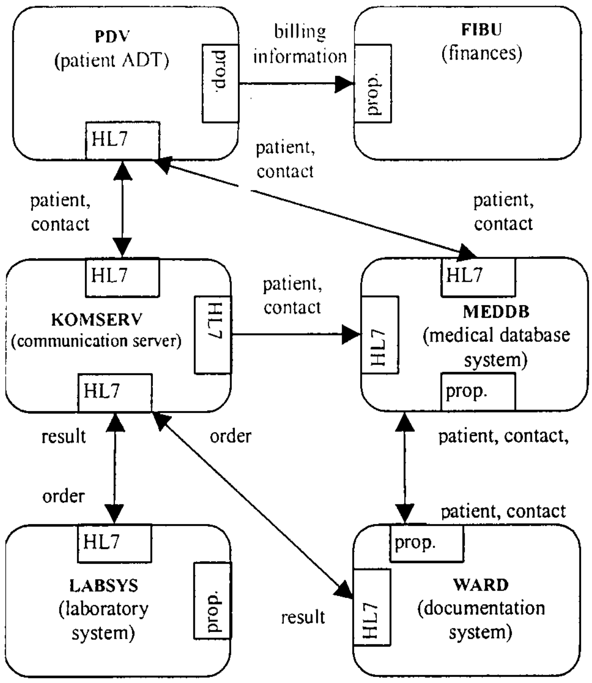

Figure 5 - Logical Tool Layer - example.

This example is simplified and fictive, but reflects a typical situation: indeed there is a HL7 based communication server, but obviously not all application components are able to use this communication service. As a consequence, a lot of proprietary interfaces are needed. Additionally, some application components communicate directly via HL7 interfaces.

Physical tool layer

The physical tool layer (see figure 6) is a set of physical data processing components (like personal computers, servers, switches, routers, etc), which are physically connected via data wires. The constellation of these connections leads to physical networks, which base on net 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.

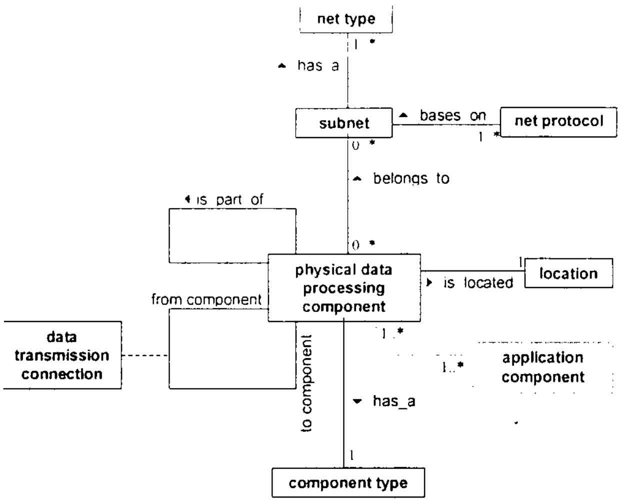

Figure 6 - Physical Tool Layer - static view.

Fig. 7 shows an example of a physical tool layer. In this example we distinguish two component types: the rectangles represent servers and PCs, the black dots represent connection points. Data transmission connections are shown as lines. In this example, all physical data processing components belong to one network, i.e. there are no subnets. Information about net type, or net protocol is not represented.

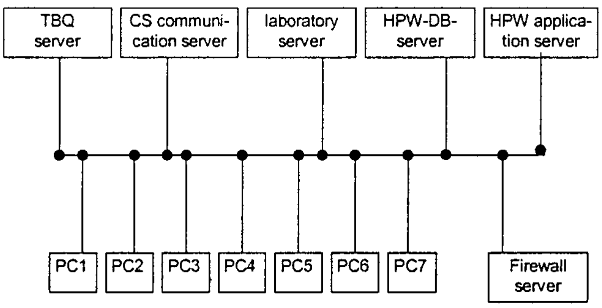

Figure 7 - Physical Tool Layer - example.

Inter-layer-relationships

Inter-layer-relationships are relations among classes of different layers. In the diagrams, the following relations are shown:

- Information about objects has to be stored logically in some kind of database (relation between object type and database system).
- Information about objects has to be transmitted via communication interfaces (relation between object type and message type).
- The performance of a task (or part of a task) is supported by one or more application components (relation between task and application component).
- Application components are installed on one or more physical data processing tools (relation between application component and physical data processing tools).

These relationships between two classes may be very complex. E.g. a certain task can be supported by exactly one application component, by several different application components (based on software products of the same or of different vendors), by a set of application components or by a combination of these.

All these relationships are controlled by integrity constraints, which guarantee some kind of correctness.

Discussion

The selection of the concepts defined in the presented ontology for describing hospital information systems architectures, and the relationships among these concepts is orientated on information management questions which have to be answered and information management tasks which have to be supported in daily work.

In this respect our approach differs from other information systems planning and modeling approaches as mentioned in the introduction. Many of these approaches place modeling aspects in the foreground, whereas interpretation and evaluation aspects are neglected.

Especially the inter-layer-relationships offer a great range of applications. The answers of nearly all of these information management questions depend on them. Solely these controlled relations make it possible to define semantic integrity constraints, i.e. criteria for the quality of a hospital information system, which can be verified.

Nevertheless, we have to qualify that we just looked at the static components of hospital information systems. Thereby, we are able to support many important information management tasks. The inclusion of dynamic aspects in this approach, i. e. processes that occur at the domain layer, and the consequences for the logical and the physical tool layer will expand these possibilities. So, in the next steps, we will concentrate on the following aspects:

- What aspects of a business process on the domain layer are relevant for information management, and how can we describe them?
- Looking at business processes, what do they effect on the logical and physical tool layers?
- How can we grasp the quality of a hospital information system, what semantic criteria are there?

Additionally, first evaluation results showing the suitability of our approach for the development of a strategic information management plan in a clinic for pediatric surgery will be available in June 2001.

We hope that our approach contributes to a more systematic information management of hospital information systems.

Acknowledgments

This work is granted by the DFG (Deutsche Forschungsgemeinschaft) under the grant no. WI 1605/2-1 (Title of the project: Integrative Modellierung von Strukturen und Prozessen in Krankenhausinformationssystemen - integrative modeling of structures and processes in hospital information systems).

References

[1] Brodi ML, Stonebraker M. *Migrating Legacy Systems: Gateways, Interfaces & the incremental approach.* Morgan Kaufmann, 1995.

[2] Winter A, Lagemann A, Budig B, Grothe W, Haux R, Herr S, Pilz J, Sawinski R, Schmücker P. Health professional workstations and their integration in a hospital information system: the pragmatic approach MEDIAS. *Computer Methods and Programs in Biomedicine* 1996. 51(3), 193-209.

[3] Reynolds, M., I. Wejerfeld. Health Informatics – Short Strategic Study – Health Information Infrastructure. CEN/TC 251/SSS-HII INR, 2000.

[4] Gartner Group. *Three Documents for Healthcare IT Planning*, Gartner Group's Healthcare Executive and Management Strategies, Research Note KA-03-5074, 1998.

[5] Zachman JA. A framework for information systems architecture. IBM systems journal 1999: 38 (2&3), pp. 454-470 (Reprint).

[6] Oberweis A, Scherrer G, Stucky W. INCOME/STAR: Methodology And Tools for the Development of Distributed Information Systems. *Information Systems* 1994. 19(8), 643-660.

[7] 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

[8] Scheer AW. ARIS - Business Process Frameworks. Berlin: Springer, 1998.

[9] Winter A, Haux R. A Three-Level Graph-based model for the management of Hospital Information Systems. *Meth. Inf. Med. 1995.* 34(4), 378-396.

[10] OMG. Unified Modelling Language Specification. Version 1.3, 1999. (www.omg.org)

[11] Porter M. *Competitive Advantage.* The Free Press, 1985.

[12] Martin J. *Information Engineering, Book II: Planning and Analysis.* Englewood cliffs: Prentice Hall, 1990.

[13] Bealer GW, Huff S, Rishel W, Shakir AM, Walker M. Message Development Framework. HL7 Modeling & Methodology Committee, 1999.

Address for correspondence

Prof. Dr. Alfred Winter, University of Leipzig, Institute of Medical Informatics, Statistics and Epidemiology, Liebigstr. 27, 04103 Leipzig, Germany

winter@imise.uni-leipzig.de