Interdisciplinary Object-Oriented Domain Analysis for Electronic Medical Records

Jörn Bodemann
Fraunhofer Institut für Software- und Systemtechnik (ISST)
Joseph-von-Fraunhofer-Straße 20, D-44227 Dortmund, Germany
Tel.: ++49 +231 9700.734, Fax: ++49 +231 9700.798
bodemann@do.isst.fhg.de

Wilhelm Hasselbring
University of Dortmund, Department of Computer Science
Software Technology, D-44221 Dortmund, Germany
Tel.: ++49 +231 755.4712, Fax: ++49 +231 755.2061
hasselbring@ls10.informatik.uni-dortmund.de

Abstract

The experience gained during the domain analysis for data models of electronic medical records is discussed. Topics of interest are the way in which the domain is analyzed by means of expert interviews, the observed need for parallelism in the learning process, object-oriented modeling of the results, tool support for rapid evaluation of models with an object-oriented database, and an assessment of the requirements on electronic medical records.

The emphasis of this paper is to discuss the *process* of participatory analysis of the domain for electronic medical records in an interdisciplinary setting. Some extracts of the results of the domain analysis are presented.

Keywords: Object-oriented IS modeling, Domain analysis, Participatory development, Electronic medical records.

1 Introduction

Domain analysis in the process of software development is the activity of identifying the objects and operations of a class of similar systems in a particular problem domain. The goal of the presented work was to perform a domain analysis for electronic medical records which are central components of health care information systems [22]. A crucial problem is to meet the requirements of the users of health care information systems. Most health care information systems, which are currently in use, support the administrative work such as admission, discharge and transfer of patients, but not the medical work of physicians and nurses. To provide a basis for the development of electronic medical records which really meet the requirements of health care workers, a domain analysis for electronic medical records has been undertaken in close cooperation of computer scientists with several domain experts. This paper presents the experience with the applied development process and some of the results of this interdisciplinary cooperation.

As the first step, the domain of angiopathy (vascular diseases) has been analyzed. Angiopathy has been chosen, because a system for controlling therapy results in angiopathy was under development in a parallel project. This system is intended to directly support physicians with their medical work. This parallel project was the first system for which parts of the results of the presented domain analysis were (re)used.

During the domain analysis it became apparent that electronic medical records should always contain information about the complete basic case history of patients. Therefore, we extended the analysis of angiopathy to additionally capture the complete basic case history. However, this basic case history is independent of specific medical domains.

The results, which are object-oriented class diagrams for electronic medical records, are intended to enable software engineers without detailed medical knowledge to get known to specific medical domains for the development of health care information systems.

Section 2 discusses some general concerns of electronic medical records and Section 3 discusses the employed domain analysis techniques. The developed object-oriented data model and tool support for rapid evaluation of changes and extensions to this model are presented in Sections 4 and 5, respectively. Section 6 discusses related work and Section 7 draws some conclusions and indicates future work.

2 The electronic medical record

The purpose of electronic medical records is to store the information about patients that is generated by physicians, nurses, hospital administrators, etc. [3]. The information that originates from diagnosis and therapy is a central concern. Goals of digitizing medical records are, for instance, improving medical treatment of patients and the computerized evaluation of patient data to support research in medicine. Electronic medical records are not merely automated forms of today's paper-based medical records, but encompass the entire scope of health information in all media forms. Thus electronic medical records may include medical history, current medications, laboratory test results, etc.

The electronic medical record has several advantages over the conventional paper-based medical record, including:

- Patient information is available at several working places at the same time.
- The information is available within a short time. This is important in case of emergency.
- Acquisition of data may be improved by the use of advanced user interfaces.
- Reuse of results of medical operations is supported, even over the lifetime of a patient. This may relieve patients from being checked with the same medical operations several times.

- Medical research is supported. An application area is the control of the results of specific therapies.

However, the electronic medical record also has its disadvantages:

- It requires a larger initial investment than its paper counterpart because of hardware, software and training costs for the personnel.
- Capturing the physician-collected data for an electronic medical record can require a lot of time and effort: physicians often use a great deal of information to make one decision.

  New techniques to facilitate direct entry of such information (e.g. speech input) can reduce this problem.
- It is only possible to read and enter patient data where computer terminals are available.

  Mobile computing is a solution for this problem. For a discussion of problems and solutions of mobile computing refer to [15].
- Data security may be a problem when the system administration is not done carefully and responsibly.

3 Domain analysis

Domain analysis in the process of software development is the activity of identifying the objects and operations of a class of similar systems in a particular problem domain [19]. To emphasize concerns of reuse, Priéto-Díaz defines domain analysis as follows:

> "We define domain analysis as a process by which information used in developing software systems is identified, captured, and organized with the purpose of making it reusable when creating new systems." [21]

Integrated into the process of domain-specific software architectures (DSSA) [30], domain analysis provides a general basis for the development of applications within the analyzed domain. Usually, design, not code, is reused for the development of specific applications [11].

However, it is not always clear, *how* to perform the domain analysis. Usually, domain analysis is divided into three steps:

1. The domain analysts get familiar with the domain through the development and analysis of several applications for the domain.
2. The experience made is used to identify objects and operations which occur in all or most of the analyzed applications for the domain.
3. The identified objects are generalized to obtain classes and standards are defined for their usage.

To undertake a domain analysis it is necessary to acquire and to represent the knowledge of the domain. In the presented project, acquisition and representation are accomplished as follows:

- A specific form of expert interview is the method for knowledge acquisition.
- An object-oriented modeling technique is employed for knowledge representation.

3.1 Problem areas of domain analysis

Our experience identifies some problems to be solved during domain analysis:

Vague domain borders: In the case of the presented project, the analysts are not domain experts. Therefore, it was necessary to define the area that is subject to analysis in cooperation with the domain experts. However, the exact definition of this area was not possible until the analysts had an rough overview of the part of the domain under consideration.

Parallelism in the learning process: It is not sufficient that the analysts learn about the problem domain. The domain experts need to get known to the modeling techniques and some requirements of computer science as far as possible.

Hardly any delegation feasible: The analysts need an overview of the domain under consideration. To cope with the complexity, it is necessary to divide the domain into independent parts, as far as possible and reasonable. For electronic medical records, such parts are angiopathy or cardiology, as well as the basic case history.

Natural domain evolution: It is important to realize that a domain analysis is never finalized. The results should be designed for change and extension; this is a general principle of software engineering [12].

3.2 Expert interviews for domain analysis

The goal of expert interviews is to elicit the domain knowledge from domain experts. The expert interview is a data capturing technique which has been investigated in the social sciences [26]. Social sciences distinguish three techniques for capturing data: interview, observation and content analysis. Two interview forms are commonly used: verbal and written. The verbal interview form is either loosely, partly or completely structured. For domain analysis, the loosely structured verbal interview form has proven to be most appropriate. This way, the domain experts are not tempted to leave their terminology.

Another concern is the form of the questions. With *closed* questions, the possible answers are predefined (multiple choice). With *open* questions, the domain analysts can chose how to structure the answer; this supports the parallelism in the learning process. Therefore, open questions should be preferred. It is important to note that it is necessary to plan a lot of time for the expert interviews. In addition, it is necessary to use a glossary to find a common terminology between computer scientists and domain experts.

3.3 Domain analysis as an iterative process

It is obvious that it is not possible to analyze a domain in one step. The following iterative process has been employed in the presented project:

1. Identification of the basic classes on the basis of paper dockets which are used in hospitals. For the basic case history, the textbook [9] was an excellent starting point for this task.
   (With the domain experts)
2. Detailed description of the identified classes.
   (With the domain experts)
3. Modeling the results with the object-oriented Unified Modeling Language (UML) notation [10, 24] using the ROSE tool [23].
   (Without the domain experts)
4. The results are checked according to additional paper dockets. When problems and/or additional questions are encountered, start again with step 1.
   (With the domain experts)

Steps 1 to 3 are executed until the basic information has been captured. Figure 1 illustrates this process.

4 Some results of the domain analysis

This section presents an extract of the data model for electronic medical records as a main part of the results of our domain analysis. The results for the basic case history and for angiopathy are discussed in sections 4.1 and 4.2, respectively.

4.1 Results for the basic case history

The case history covers the medical history of patients. However, the different medical areas use specific extensions for their area of interest. Consequently, we distinguish between basic and specific case history. Figure 2 displays an extract of the class diagram for the basic case history for medical information concerning the physical area of Mouth and Throat in the UML notation [10].

An explanation of the model in Figure 2 is given as follows. Rectangles are the UML symbols for classes. The UML allows specification of *non-shared* aggregation through filled diamonds, which is called *composition*, whereby the parts are expected to *live and die* with the whole [10]. Teeth, Lips, Tongue, etc. are included through composition into the Mouth/Throat Area, because their existence should be tied to the existence of the containing area. The Mouth/Throat Area is *composed of* these components.

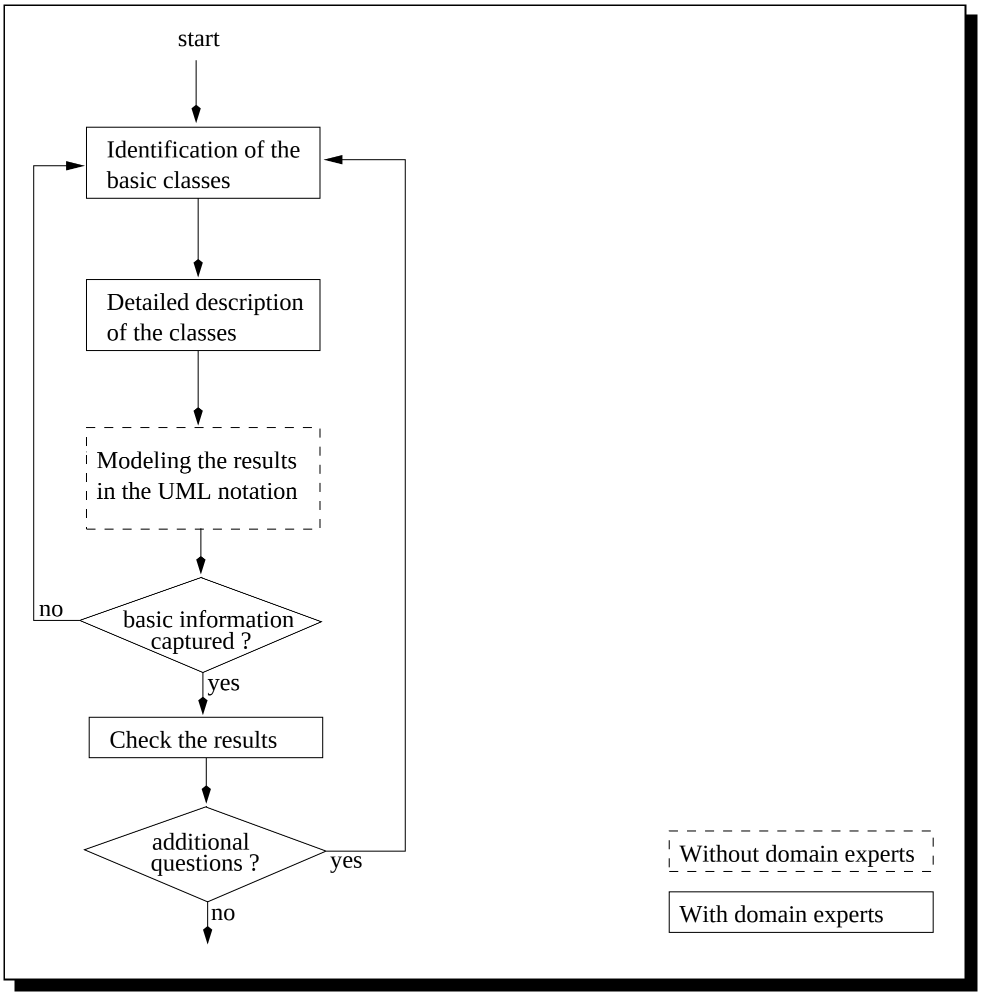

Figure 1: Our iterative process for domain analysis.

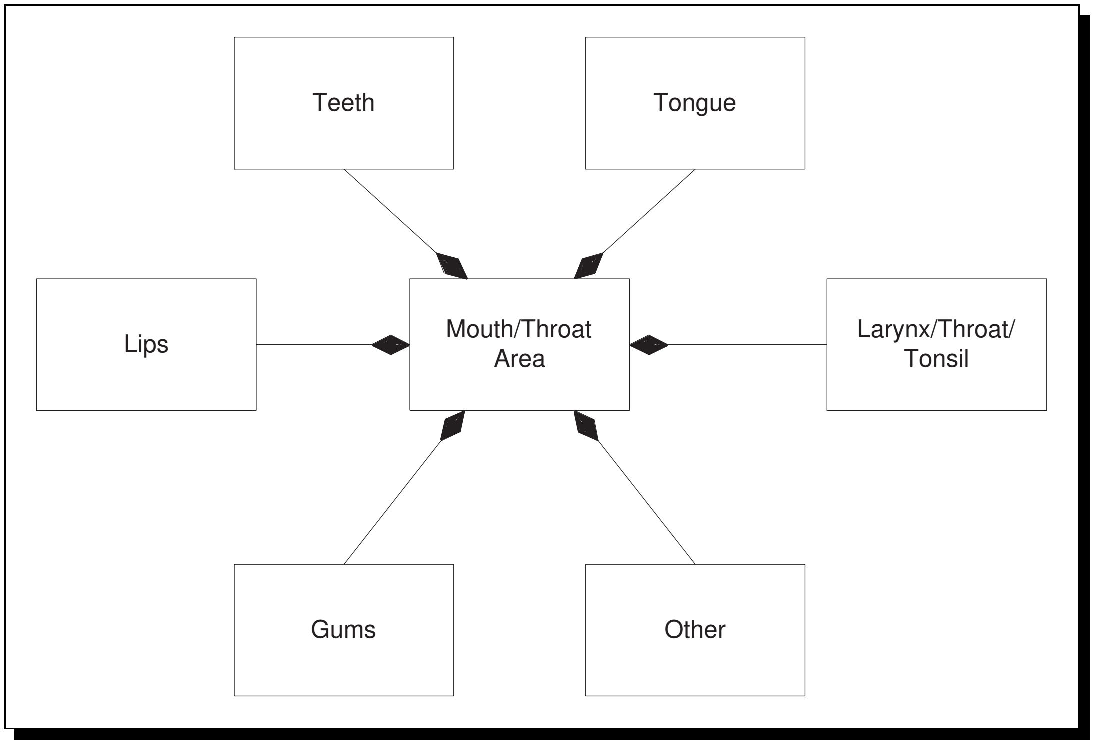

Figure 2: Extract of the basic case history: the area of mouth and throat.

Our class diagram for the basic case history contains more than 100 classes. It is important to note that for each class a detailed description has been produced which includes detailed lists for attributes.

According to Dahmer [9], about 70% of all diagnostic decisions could be made on the basis of the basic case history and a short clinical examination. However, every clinical diagnosis can be a contra-indication for specific medical operations. A concrete basic case history is used for decision making and can be used as a communication medium among different medical specialists. Medical specialists need extensions to the basic case history to cover their specific area of interest.

4.2 Results for the specific area of angiopathy

Figures 3 and 4 display small parts of the class diagram for the specific area of angiopathy. The class 'Operation' in both figures describes any activity which can be carried out on a patient with respect to angiopathy.

An explanation of the model in Figure 3 is given as follows. In the UML, multiplicities for associations are specified through numerical ranges at the association links. The default multiplicity is 1. If the multiplicity specification comprises a single star, then it denotes the unlimited non-negative integer range (zero or more). The arrows attached to the association names indicate the direction for reading the names which are annotations to associations (called *name direction*) [24]. For example, an Operation can use multiple Materials and the same Material can be used for multiple Operations.

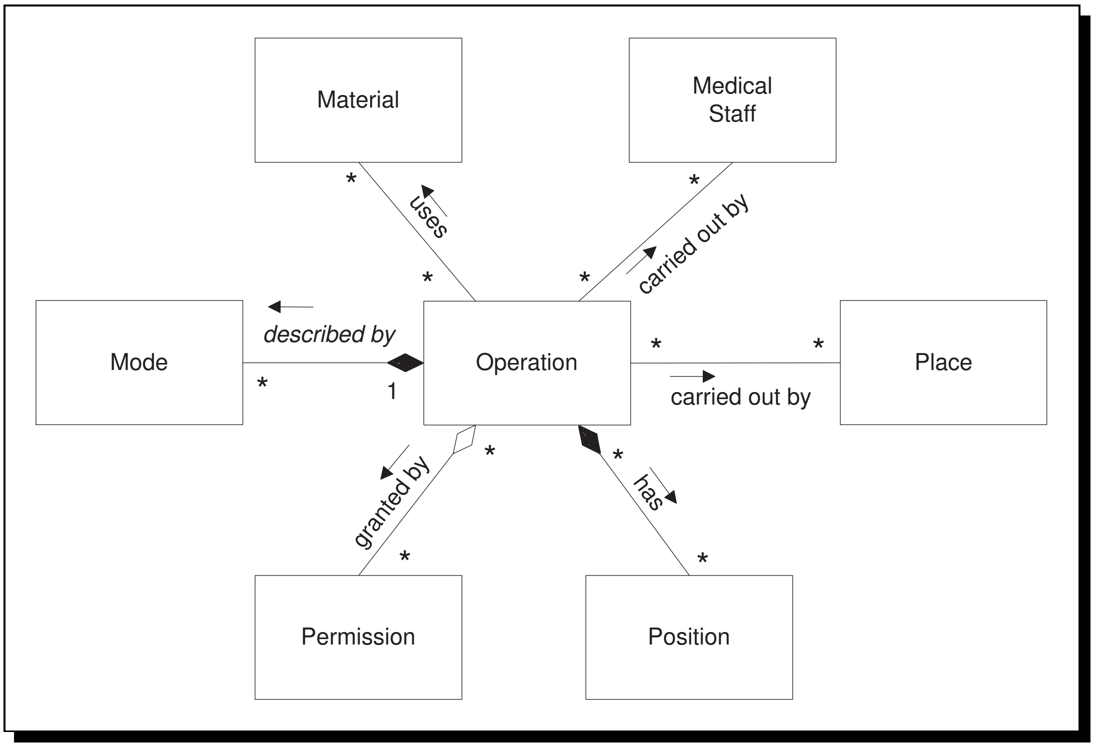

Figure 3: Associations of medical operations in angiopathy.

In addition to non-shared aggregation, the UML allows specification of *shared* aggregation through hollow diamonds which indicate part-of relations, whereby the existence of the parts is not tied to the existence of the whole. An Operation may require multiple Permissions and a Permission may allow multiple Operations. Permissions are considered to be part of Operations, but they may exist on their own, before the Operations are performed. Permissions may also belong to multiple Operations. Different to this relation, the existence of Positions is tied to the existence of Operations. The associations without diamonds are general associations with equal rights for the involved classes.

An explanation of the model in Figure 4 is given as follows. Inheritance for specialization and generalization is shown in UML as a solid-line path from the subclass to the superclass, with a hollow triangle at the end of the path where it meets the superclass [10]. All operations are defined by inheriting from the class Operation, as shown in Figure 4. Usually operations are distinguished in either diagnostic or therapeutic. Our domain analysis showed that a distinction between these types often cannot be made. Many medical operations in angiopathy are diagnostic in one context and therapeutic in another. Therefore, a distinction between invasive and non-invasive operations is made on the first level.

The class diagram for the basic case history is integrated through aggregation into the class diagram of the electronic *patient* record which is used in health care applications (this is not shown in the presented extract). To save space we cannot present the complete class diagrams in this paper. Since the emphasis of this paper is to discuss the *process* of participatory analysis of the domain for electronic medical records in an interdisciplinary setting, only extracts of the results of the domain analysis are presented.

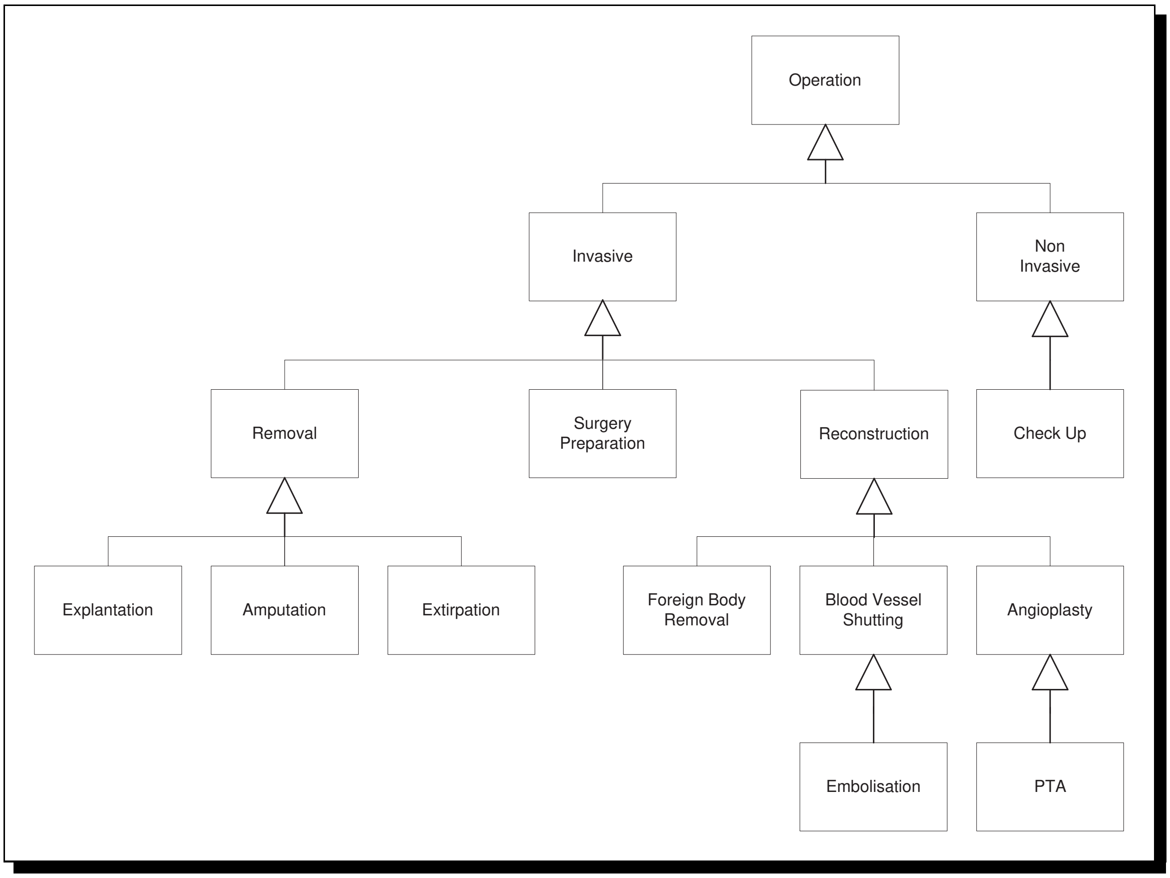

Figure 4: Extract from the hierarchy of medical operations in angiopathy.

5 Tool support for rapid evaluation of changes and extensions to the model

To allow the rapid evaluation of changes and extensions to the model when extending and modifying the domain model towards the needs of specific health care information systems which need electronic medical records, a new tool has been developed. This tool translates the model from the ROSE internal repository [23] into a database schema and some application code for the object-oriented database O_2 [4]. The O_2Look [20] module of O_2 then automatically generates a simple application which can be used to experimentally evaluate the changes made to the domain model. This simple application has a graphical user interface which allows data entry, display and navigation within the data structures.

The generator has been implemented in a platform independent way with the Java language [1]. The front-end of the generator parses the ROSE internal repository and transforms it into an intermediate format. The back-end generates specific code from this intermediate representation to allow the generation of different target codes. Note, however, that concerns of schema evolution in existing databases [5] are not dealt with in this approach.

6 Related work

There exist several initiatives for standardizing components of health care systems [2, 7, 17]. Standards are under development for transferring information across subsystems, codifying text, describing the content of the medical record, etc. The following subsections take a short look at three standardization initiatives which are related to our work: coding systems for medical information, the HL7 protocol, and CORBAmed of the Object Management Group.

6.1 Coding systems for medical vocabulary

Several initiatives for standardizing coding systems for medical vocabulary exist. The Systematized Nomenclature of Human and Veterinary Medicine (SNOMED) [6] is a code structure which is widely accepted for describing pathological test results. It has a multi-axial coding structure for symptoms, diagnoses and procedures. The Unified Medical Language System (UMLS) [16] contains a metathesaurus that links biomedical terminology, semantics, and formats of the major clinical coding and reference systems. It links medical terms (e.g. SNOMED) to so-called medical index subject headings (MeSH codes) and to each other. The UMLS also contains a specialist lexicon, a semantic network, and an information sources map. Together, these elements are intended to represent all of the codes, vocabularies, terms, and concepts to become a foundation for the health care informatics infrastructure [16].

These coding systems are used to encode medical information independent of any specific natural language. They can, for instance, be used to allow the exchange of medical records across different language areas.

6.2 The Health Level 7 protocol

The Health Level 7 (HL7) protocol has been designed to standardize the data transfer within hospitals [13]. It is based on level seven of the ISO/OSI-protocol hierarchy [29]. HL7 covers various aspects of data exchange in hospital information systems, e.g. admission, discharge and transfer of patients, as well as the exchange of analysis and treatment data. The HL7 standard represents hospital related transactions as standardized messages. HL7 is a de-facto standard for data exchange between commercial systems for hospitals [17].

HL7 defines messages as strings to be exchanged by subsystems. The messages themselves contain standardized information, but do not invoke specific methods at the destination. HL7 can be used to transfer electronic medical records across subsystems in hospitals and, therefore, implicitly defines a data model for electronic medical records. However, the detailed structure of medical information is not explicitly defined. The medical information is encoded within the strings to be exchanged by the subsystems.

6.3 CORBAmed of the Object Management Group

CORBA is the 'Common Object Request Broker Architecture' of the Object Management Group to standardize interoperability among heterogeneous hardware and software systems [18]. Simply stated, CORBA allows applications to communicate with one another no matter where they are located or who has designed them. CORBA defines the Interface Definition Language (IDL) and the Application Programming Interfaces (API) that enable client/server object interaction within a specific implementation of an Object Request Broker (ORB). CORBA also defines interoperability by specifying how ORBs from different vendors can interoperate.

The ORB is the middleware that establishes the client-server relationships between objects. Clients can transparently invoke a method on a server object, which can be on the same machine or across a network. The ORB intercepts the call and is responsible for finding an object that can accept the request, pass the parameters, invoke its method, and return the results. The client does not have to be aware of where the object is located, its programming language, its operating system, or any other system aspects that are not part of an object's interface.

CORBAmed is the Health Care Special Interest Group for CORBA [27]. In 1996, CORBAmed started the process of adopting standard interfaces for health care related objects by issuing a 'request for information' which requested the health care and information technology industry to give the OMG guidance in its upcoming standardization efforts for CORBAmed.

It can be expected that the CORBAmed services within the CORBA framework will be an important standard for the interoperation among subsystems in health care. However, in the current stage of standardization it is not clear what exactly CORBAmed will contribute to the challenge of integrating information in health care information systems.

The HL7 Special Interest Group for Object Brokering Technologies is mapping the forthcoming HL7 version 2.3 onto the IDL of CORBA and version 3 of HL7 will be based on an object-oriented model of the underlying data [25]. So, we can expect a combination of CORBAmed and HL7 in the future.

7 Conclusions and future work

The experience gained during the domain analysis for class diagrams of electronic medical records is presented. Problem areas of domain analysis, a specific form of expert interviews for domain analysis, our analysis process for domain analysis, and extracts of the resulting class diagrams for electronic medical records are discussed.

The yielded object-oriented analysis model provides a basis for the development of electronic medical records. The use of an object-oriented modeling technique for the description of the results allows 'natural' modeling: the identified classes correspond to medical terms as you can see in the presented extracts of the class diagrams in Section 4. It has been suggested to use object-oriented techniques for modeling health care information systems [14]. We used the UML for modeling, which offers a rich set of notations that are in the process for standardization by the Object Management Group (OMG) [10, 24].

To support extensions and modifications to the class diagrams, a new tool has been developed which allows to translate changed models from the ROSE internal repository into a schema for the object-oriented database O_2 which automatically generates a simple application to be used for experimentally evaluating the changes made to the model. This tool supports early *prototyping* of extensions and modifications. Prototyping is an important concern in participatory development to involve the users of health care information systems [8, 28].

However, when extending or modifying the model it is reasonable to follow the process presented in the paper. In particular, the observed need for parallelism in the learning process is important: analysts must get known to the application domain *and* the domain experts must get known to concerns of data modeling to some extent.

The results of the domain analysis were used as one basis for re-engineering a system for controlling therapy results in cardiology. The experience with this project will enable us to extend and refine the class hierarchy, since information from additional domain experts is gathered to extend our angiopathy model with cardiology data. The core model for the complete basic case history is directly reusable, because it is independent of specific medical domains.

An additional concern for future research is the construction of *adaptive* user interfaces for the complete basic case history. It is important to introduce appropriate views to the model for different user groups.

Acknowledgements

The authors would like to thank Thorsten Jahnke, Detlef Mehlstäubler and Arnd Röser for spending a lot of time to serve as medical domain experts, and Jan Neuhaus and Bernhard Holtkamp for the comments on drafts of this paper.

References

[1] K. Arnold and J. Gosling. *The Java Programming Language.* Addison-Wesley, Reading, 1996.

[2] A.R. Bakker, W.E. Hammond, and M. Ball. Summary report of observations, conclusions and recommendation of the IMIA WG 10 Hospital Information Systems Working Group Conference, Durham, NC, August 1994. *International Journal of Bio-Medical Computing*, 39(1):11–15, 1995.

[3] M.J. Ball and M.F. Collen, editors. *Aspects of the computer-based patient record.* Springer-Verlag, New York, 1992.

[4] F. Bancilhon, C. Delobel, and P. Kanellakis. *Object-Oriented Database system - The story of O_2.* Morgan Kaufmann Publishers, 1992.

[5] J. Banerjee, W. Kim, H.J. Kim, and H.F. Korth. Semantics and implementation of schema evolution in object-oriented databases. In U. Dayal and I. Traiger, editors, *Proc. ACM SIGMOD International Conference on Management of Data*, pages 311–322, San Francisco, CA, 1987.

[6] K.E. Campbell and M.A. Musen. Representation of clinical data using SNOMED III and conceptual graphs. In M.E. Frisse, editor, *Proc. Sixteenth Annual Symposium on Computer Applications in Medical Care*, pages 354–358, Baltimore, MD, 1992. McGraw-Hill.

[7] CEN TC/251 Working Group 1. Healthcare Information Framework (HIF). Draft Prestandard, CEN, European Committee for Standardisation, 1995.

[8] A. Clement and P. Van den Besselaar. A retrospective look at PD projects. *Communications of the ACM*, 36(4):29–37, June 1993.

[9] J. Dahmer. *Anamnese und Befund.* Thieme, 4th edition, 1978. (in German).

[10] M. Fowler and K. Scott. *UML Distilled: Applying the Standard Object Modeling Language.* Object Technology Series. Addison-Wesley, Reading, MA, 1997.

[11] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns – Elements of Reusable Object-Oriented Software.* Addison Wesley, Reading, MA, 1995.

[12] C. Ghezzi, M. Jazayeri, and D. Mandrioli. *Fundamentals of Software Engineering.* Prentice-Hall, 1991.

[13] W.E. Hammond. Health Level 7: A protocol for the interchange of healthcare data. In G.J.E. De Moor, C.J. McDonald, and J.N. van Goor, editors, *Progress in Standardization in Health Care Informatics*, pages 144–148. IOS Press, Amsterdam, 1993.

[14] J. Harrington. Recommendations for an object oriented healthcare information model. In G.J.E. De Moor, C.J. McDonald, and J.N. van Goor, editors, *Progress in Standardization in Health Care Informatics*, pages 52–59. IOS Press, Amsterdam, 1993.

[15] T. Imielinski and B.R. Badrinath. Mobile wireless computing: Challenges in data management. *Communications of the ACM*, 37(10):18–28, October 1994.

[16] D.A.B. Lindberg, B.L. Humphreys, and A.T. McGray. The unified medical language system. *Methods of Information in Medicine*, 32:281–291, 1993.

[17] C.J. McDonald. News on U.S. health informatics standards. *M.D. Computing*, 12(3):180–186, 1995.

[18] T.J. Mowbray and R. Zahavi. *The Essential CORBA: Systems Integration Using Distributed Objects.* Wiley, 1995.

[19] J. Neighbors. *Software Construction Using Components.* PhD thesis, Department of Information and Computer Science, University of California, Irvine, 1981.

[20] O_2 Technology. *O_2Look User Manual.* 1996.

[21] R. Priéto-Díaz. Domain analysis: An introduction. *ACM SIGSOFT/Software Engineering Notes*, 15(2):47–54, April 1990.

[22] H.U. Prokosch and J. Dudeck, editors. *Hospital Information Systems: Design and Development Characteristics; Impact and Future Architecture.* Elsevier, Amsterdam, 1995.

[23] Rational Software Corporation, Santa Clara, CA. *Rational Rose User's Guide*, 1997.

[24] Rational Software Corporation. The Unified Modeling Language. UML Document Set Version 1.1, Santa Clara, CA, September 1997. (available from `www.rational.com`).

[25] W. Rishel and J. Quinn. Software components, the clinical workstation and healthcare networks: How HL7 is helping you get there. In *Proc. Healthcare Information and Management Systems Society's Annual Conference*, Atlanta, Georgia, March 1996.

[26] R. Schnell, P.B. Hill, and E. Esser. *Methoden der empirischen Sozialforschung.* R. Oldenbourg, 1989. (in German).

[27] R. Sokolowski. Integration services for clinical data exchange. OMG/CORBAmed DTF White Paper 97-01-01, Kurzweil Applied Intelligence, January 1997. (available as `ftp.omg.org/pub/archives/docs/corbamed/97-01-01.pdf`).

[28] G. Symon, M. Fitter, C. Radstone, I. Kunkler, and B. Hancock. The process of deriving requirements for a hospital information system. *Behaviour and Information Technology*, 11(3):131–140, 1992.

[29] A.S. Tanenbaum. *Computer Networks.* Prentice Hall, third edition, 1996.

[30] R.N. Taylor, W.J Tracz, and L. Coglianese. Software development using domain-specific software architectures. *ACM SIGSOFT Software Engineering Notes*, 20(5):27–38, December 1995.