*Medical Infobahn for Europe*
*A. Hasman et al. (Eds.)*
*IOS Press, 2000*

# An Object-Oriented Approach for Structuring the Electronic Medical Record

Friedemann Banhart, Rüdiger Lohmann

*Dräger Krankenhaus-Informationssysteme GmbH*
*Schreberweg 1, 24119 Kiel-Kronshagen, Germany*

**Abstract.** We implemented a framework for modelling the electronic medical record on top of an object-oriented model. Clinical patient data are structured in a uniform way through the use of a comprehensive data model. The meaning of the information elements is explicitly determined by a medical data dictionary. The data structures of both, medical record and data dictionary are implemented, using a semantically rich, object-oriented data model. We examined several possibilities for the graphical preparation of the inherently recursive data structures. Again, we use object-oriented frameworks for the implementation of flexible user interfaces to the electronic medical record with a consistent look-and-feel.

## 1. Introduction

Our overall goal is to implement an electronic medical record that supports individual patient care. We want to represent all clinically significant information in a structured form. Meaning and context of any medical information element is explicitly captured and will be preserved, even if a statement is reused in many different ways. See [1] and [2] for a discussion of the requirements for medical records.

Medical data shall be captured only once in order to build up a consistent and redundancy-free collection of patient data. Once recorded, one data element can be inserted in different documents, used by different applications and transmitted to other systems, that take part in the process of patient treatment. A reusability of medical data is the precondition to define different views to the patient data, dependant on the actual application scenario.

## 2. Medical Data Dictionary

A central part of our clinical information system is a medical data dictionary. The data dictionary is used for the explicit representation of meaning and terminology of medical data. See [3] and [4] for a discussion of medical dictionaries and controlled vocabularies.

The relationships between the categories of the data dictionary are expressed in the formalism of a semantic network. Our data dictionary supports versioning, and its vocabulary can be mapped onto the international nomenclatures SNOMED, ICD and ICPM using ID-SNOMED™ and ID-DIACOS™.

The data dictionary can be decomposed into sections, using different containers. A container comprises a subset of concepts, and puts them together as sets, lists or trees. These containers are part of the data dictionary by themselves. Such a container can e.g. be used to define the value range for choice lists.

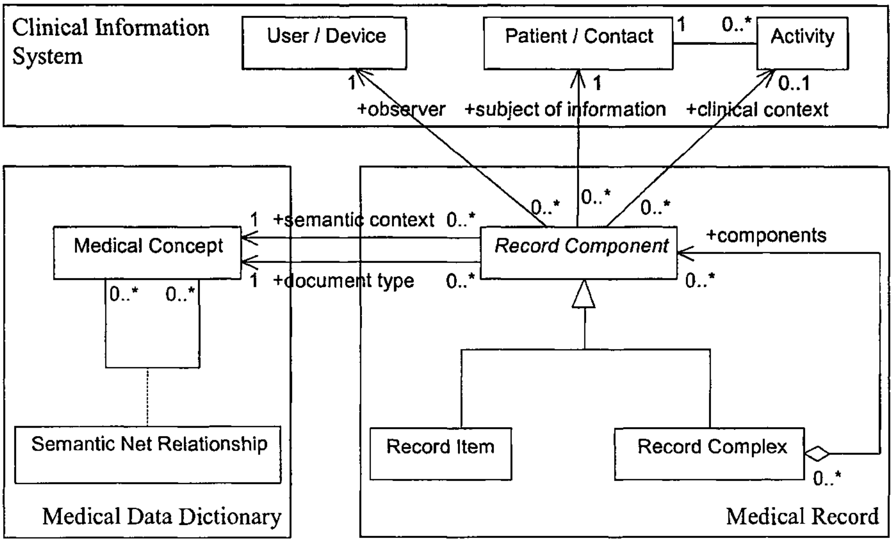


**Fig 1:** Simplified UML diagram of the conceptual data model: record components, items and complexes constitute the medical record. There are relationships to the medical data dictionary and to the rest of the clinical information system (containing a description of clinical resources, patients, the process of patient care and the clinical dialogue).

## 3. Conceptual Data Model

The design of our conceptual data model is influenced by the work of the CEN/TC251 project teams on the European prestandard for an architecture for electronic healthcare information and communication systems [5]. This model is capable of structuring medical data in a uniform way, presenting the multitude of different facts while preserving meaning and context of the data.

The basic elements of a medical record are *record-items* and *record-complexes*. Record item and record complex are subclasses of the abstract class *record-component*, thus building up a *composite*-design pattern [6] (see Fig. 1).

a.) **Record-Component** is the common superclass for all elements of an electronic patient record. Each record component contains attributes describing its context and state:

- The **clinical context** describes the subject of information (either a patient or a specific hospital contact), together with the observer (user or device) and the localisation or responsible organisation. Furthermore, the originating activity, that triggered the recording of the data will be captured. Precondition is the explicit representation of each activity on a patient in our data model.
- The **semantic context** is a link to a category of the data dictionary. This qualifies the medical data according to a concept of the controlled vocabulary. Medical data are no longer described by the name of their attributes in relational tables, but can be identified and interpreted in the context of the data dictionary.
  We implemented two links: One describes the specific type of the record component (e.g. blood pressure, systolic). The other one specifies the document type, as described in the domain termlists in [7] (e.g. a document is classified as a subjective finding, a problem, symptom, patient history, drug prescription, etc.).
- The **status** describes what has been done by whom to that item, together with the date and time of the recording in order to implement a simple versioning system. The status also implies, what shall or can be done with the record component in the future.

- **Data security** information describe, which user, agent, and / or organisational unit has read or write access to the component.
- **Content type** information describes the data type of the content and is e.g. used to create an appropriate graphical presentation for displaying the data.
- **Comments** can be attached to record component (*annotations*), and explicit **links** (*link items*) can be defined to connect different constructs.

b.) A **Record-Item** (or clinical statement) is the indivisible basic information element of the medical record. A record item represents an observation by an agent at a particular time and place, e.g. a symptom, diagnosis or a laboratory result. Each record item can be individually recorded, presented and interpreted. Record items can be distinguished according to their content type:
- **Simple** values: text strings, booleans, integers, float, date or time.
- **Composed** data types enumerate two or more simple values. E.g. a measurement consists of a numeric value plus a unit statement.
- **Unstructured** multimedia data like texts (narrative notes, discharge letters, etc.) or pictures. The record item extends unstructured clinical data with annotations in order to classify the content and context.
- **Coded** information is represented by a reference to a category of the data dictionary. E.g. a SNOMED code can represent a compound clinical expression. The value range of a coded data type is structured as a set, list, tree or semantic network, using the above mentioned containers. So called *multiple-choice-lists* are supported, i.e. a data item can take more than one value at the same time.
- A record item can contain a **reference** to any object in our system (described by the class name plus object identifier). Thus, you can define arbitrary complex objects in the usual way, and include them in the framework of the medical record. This is especially valuable to solve the "legacy"-problem of merging existing data that is stored in specific relational tables with the medical record.
- **Command** data type: a command opens e.g. an internal frame on the desktop. The idea behind this special data type is to apply the flexible architecture of medical records to the implementation of adaptable user interfaces. Generally, all of the presented concepts can also be applied to non-medical data. The boundaries between the patient record and the configurable user environment of our software system become blurred.

c.) A **Record-Complex** is a container for different record components. Complexes can be arbitrarily nested in order to build up the medical record. A complex can be treated either like a document, a section of a document, a working list, a table of content, a directory, or as the medical record itself. A component can simultaneously be part of different complexes. There are different possibilities to define the membership of a component in a complex:
- **Static**: there is a many-to-many relationship in the database connecting record complex and record components.
- **Dynamic**: an inter-object relationship is specified, that starts at the actual patient and looks for a set of record components, fulfilling the search criteria of the relationship. Such a relationship is nothing else but a parameterised SQL-query, taking the actual patient (plus the current user, agent and organisation if required) as input parameters and giving back instances of record-components (i.e. items or complexes).
- A **repeating group** contains components of one type. Repeating groups are extensible, in the sense of that it's always possible to add new instances of the specified object type to them.

## 4. Object-Oriented Implementation

The recursive data structures of the medical record are implemented, using an object-oriented data model All our patient data are stored in a central medical database, using a relational database system. We implemented a module called *object server*, that supports the projection of the relational to an object-oriented model. Persistent data are stored in a relational database, but the application uses a semantically rich object-oriented model. The object server is a new and extended implementation of the ideas, presented in [8] and [9].

## 5. User-defined data entry

A user can define different document templates. She can do this, using a graphical editor, without the help of any technical support. A document template contains content and layout information, necessary to create instances of record items and record components at run time, together with instructions to layout the final document. Precondition is the possibility, to assemble graphical user interfaces at run-time, according to the rules in the document template:

- **Record-Items**: We use a model-view-controller pattern to present our data at the user interface. Dependant on the item type, a record item is assigned to a controller, that creates a specific graphical component at the user interface that can be used to view and edit the value of the information element.
- **Record-Complexes**: A record-complex contains different record-components. Each component is capable of presenting its own view to manage an independent sub-dialog with the user for data entry. The only task of the controller of a record-complex is to gather the controllers of its components and assemble the graphical components into a *compound document*. Record-complexes can be nested. Consequently, their compound document views can be nested, too.

  Record-complexes can also be displayed using a *tree view*. This is especially useful, if the user wants to change the structure of a document, i.e. cutting, copying and pasting sub-components into other documents, creating subfolders and rearranging the structure of a document.

  Finally, complexes can be displayed by *table views*. Inside a table, the inherent component-controllers will be used as table cell renderers or editors respectively.
- **Inspector**: The context information of a record-component is not included into a document, but can be edited using an inspector. A popup window will show the context of the actually marked record-component. It is possible to step through the items of a compound document and "touch" a specific component in order to view or edit the context information of that information element.

Document templates are related with the categories of the data dictionary. This enables a *knowledge-driven data entry* (see [10]): the user browses semantic network of the data dictionary for a concept and instantiates the patient data for the chosen concept, according to the attached document template.

## 6. Conclusions: what does this mean for the electronic record?

The above described structures are used to build up the complete electronic medical record. A medical record is a record-complex, containing a set of nested complexes (or folders).

The medical record is typically presented by a tree view. Folders can be used as working lists, problem lists, mail folders or archives. They can contain medical documents, arranged according to their scope, content, time stamp or source.

There is a high flexibility in specifying the structure of the patient record. A set of predefined templates for record complexes will be used to build up the skeleton of the patient record. This skeleton will be instantiated and filled with patient data, documents and folders in order to reflect the specific situation of one patient.

Different (orthogonal) views on the patient record can be combined in order to support the needs of different users, specialities or different application scenarios. We provide two standard views to the record: a source-oriented view (arranging data according to their sources) and a time-oriented view (arranging data according to their type and time stamps).

The data dictionary can be used, to connect the document templates to different disease categories, procedures or concepts. Each clinical site or speciality can define its own templates. Ultimately, the clinicians will be responsible for what shall be documented in which clinical context and what level of detail will be necessary

## 7. Discussion

The core of our data model is similar to the proposals of CEN/TC251. The effectiveness and validity of this approach could be verified. However, multiple extensions have been necessary in order to implement a comprehensive electronic patient record.

The flexible data model can be used to merge free text data entries with structured data entries. Unstructured (multimedia) data will be presented as specific record item types, labelled with context and content information and embedded in the context of the medical record. Thus, structured data entry can be used as far as possible, unstructured documents of arbitrary content can be included as far as necessary, using a consistent data model.

The key to success is twofold: on the one hand the flexible recursive data structures of the composite pattern of the medical record, and on the other hand the use of a data dictionary to explicitly present the semantics of the data. The implementation of these structures requires the use of an extensible, object-oriented data model.

## References


[1] AL Rector, WA Nowlan, S Kay. Foundations for an Electronic Medical Record. Meth Inform Med 1991; 20, pp. 179 – 186.

[2] AL Rector, WA Nowlan, S Kay, CA Goble, TJ Howkins. A Framework for Modelling the Electronic Medical Record. Meth Inform Med 1993, 32, pp 109-119.

[3] HU Prokosch, T Bürkle, J Storch, A Strunz, M Müller, J Dudeck, B Dirks, F Keller: MDD-GIPHARM: Design and Realization of a Medical Data Dictionary for Decision Support in Drug Therapy. Informatik, Biometrie und Epidemiologie in Medizin und Biologie 1995: 16 (3), pp 250 – 261.

[4] JJ Cimino: Data Storage and Knowledge Representation for Clinical Workstations. International Journal of Bio-Medical Computing, 1994;34, pp 185-194.

[5] CEN/TC251/WG I and project team 26: Health Informatics. Electronic Healthcare Communication – Part 1: Extended Architecture 1998. http://www.centc251.org

[6] E Gamma, R Helm, R Johnson, J Vlissides. Design Patterns. Addison Wesley 1995.

[7] CEN/TC251/WG I and project team 27: Health Informatics: Electronic Healthcare Communication – Part 2: Domain Termlist 1998. http://www.centc251.org

[8] F Banhart, H Klaeren, E Zrenner: An Object-Oriented Database Front End for Clinical Research Information Systems. In: P Barahona et al (eds). MIE 94 Proceedings, pp 506 – 511.

[9] F Banhart, H Klaeren: A graphical query generator for clinical research databases. Methods Inf Med 1995; 34, pp. 328 – 339.

[10] PW Moorman, AM van Ginneken, J van der Lei, JH van Bemmel: A Model for Structured Data Entry Based on Explicit Descriptional Knowledge. Meth Inform Med 1994; 33, pp 454 – 463.