Medical Informatics Europe '99
P. Kokol et al. (Eds.)
IOS Press, 1999

Design of the SGML-based electronic patient record system with the use of object-oriented analysis methods

Eila KUIKKA, Anne EEROLA, Jari PORRASMAA, Ari MIETTINEN
Department of Computer Science and Applied Mathematics,
University of Kuopio, P.O.Box 1627, 70211 Kuopio, Finland
Jorma KOMULAINEN
Department of Pediatrics, Kuopio University Hospital
P.O.Box 1777, 70211 Kuopio, Finland

Abstract. Since a patient record is typically a document updated by many users, required to be represented in many different layouts, and transfered from place to place, it is a good candidate to be represented structured and coded using the SGML document standard. The use of the SGML requires that the structure of the document is defined in advance by a Document Type Definition (DTD) and the document follows it. This paper represents a method which derives an SGML DTD by starting from the description of the usage of the patient record in medical care and nursing.

1. Introduction

The electronic patient record (EPR) is an answer for the well-known difficulties in the use of the paper patient record. Numerous electronic patient record systems [2, 3, 4] have been implemented in particular environments with the use of some predefined techniques. An opposite approach to these specific implementations is taken in a set of systems or prototypes which represent the patient record as a structured document containing the standardized markup [7, 11]. SGML (Standard Generalized Markup Language)[5] is an international standard to define the markup notation used in structured documents. An SGML document follows the structure which is defined with the use of a Document Type Definition (DTD). The DTD can be designed by considering the paper patient record and by defining the potential semantic elements and their relationships [6, 10]. The derived structure may, however, conflict with what doctors and nurses actually require in their everyday practices. The problem here is that the necessity of information is not considered, new information requirements may not be found, and the possibilities to evaluate the medical care and nursing routines may not be discovered.

The aim of the paper is to develop a method for the derivation of an SGML DTD by starting from the description of the utilization of the patient record in medical care and nursing. The method has the following steps:

1. The flow of the data in the clinic is considered from the patient records point of view.

2. The conceptual modeling is done using the Unified Modeling Language (UML) [1] to describe the medical treatment in a natural way.
3. The tree diagram [6] of the SGML DTD is created from the UML class diagram using a set of rules presented in Section 4.

2. Analysis and design

Our target was to analyze carefully where in the organization the information of the EPR is utilized and in which order, what is the importance of the information and when, where and why the information about the patient is gathered to the EPR. Therefore we made a careful analysis with doctors, nurses and secretaries of the Department of Pediatrics of the Kuopio University Hospital. We felt the UML [1] insufficient for the description of the flow of the information in the organization. Hence we utilized the traditional data flows [12].

According to the object-oriented approach the concepts of the real world are *objects*, for example patient record, clinic, blood test, etc. Each object possesses an *identity*, a *state* (consisting of the values of its *attributes*), and a *behavior* (methods). An attribute of an object can be an object. This leads to the *aggregation relationship* between objects. The objects are defined in *classes* which form a class hierarchy. The class hierarchy allows subclasses to inherit properties from superclasses. The superclass is a *generalization* of it's subclasses. *Concrete classes* define objects. *Abstract classes* are used for the definition of the properties that subclasses inherit. These concepts are usually described using UML class diagram [1].

Our work has similarities with the work of Salminen *et al.* [8] who appied the metodology of Shlaer and Mellor [9] for document analysis.

3. SGML DTD

According to the SGML standard, a structured document consists named elements whose identifiers and hierarchical relationships are defined in the DTD. All the documents of a type have to follow the same DTD. The DTD defines for each element *a content model.* It specifies the subelements, their order and existence conditions. The following sample DTD defines that the `patient` element consists of the name of the patient (`name`) and zero or more visits (`visit*`) in the clinic. The patient name is a sequence of characters (`#PCDATA`) defining a *content element.* The `day`, `clinic`, `doctor?` and `problem` elements are contained in the `visit` element, in this order and by the condition, that the `doctor` element is optional.

```
<!ELEMENT patient         - - (name, visit*)>
<!ELEMENT (name)          - - (#PCDATA)>
<!ELEMENT visit           - - (day,clinic,doctor?,problem)>
```

A document instance according to this DTD would start, for example, as follows

```
<patient><name>John Hui</name><visit><day>21.12.1998</day><clinic>...
</clinic><doctor>...</doctor><problem>...</problem></visit><visit>...
</visit>...</patient>
```

If the same content model exists in many places of the DTD, it is possible to define a *parameter entity* to shorten the DTD. The parameter entity is used like an element in the DTD, but the names of parameter entities are not used in the document instance.

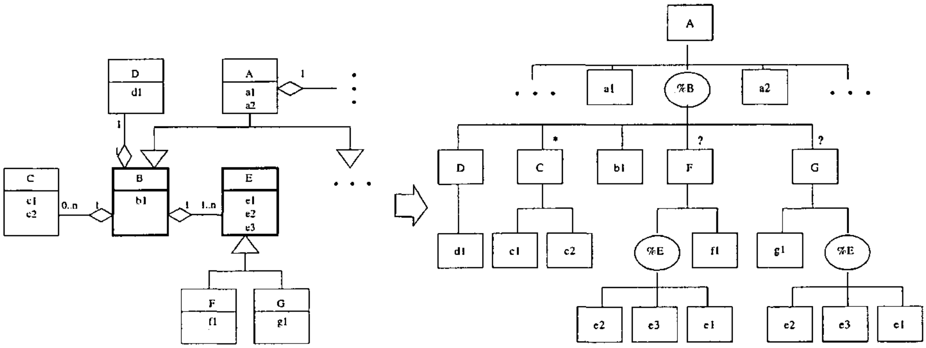

Figure 1: An UML class diagram and the corresponding DTD tree representation

4. From the UML class diagram to the SGML DTD

The potential semantic components are designed in the UML class diagram. The following two sets of rules can be used by the analyst to generate from it the DTD tree representation. Examples in parentheses refer to Figure 1. The rules 1–4 are *transformation rules* and 5–7 *placement rules*.

Rule 1: The concrete class generates an element. (Classes A, C, D, and F, G)
Rule 2: The abstract class generates a parameter entity. (Classes B, and E)
Rule 3: The attribute generates a content element (Attributes a1, a2 of A)
Rule 4: The parameter entity may be divided into many parameter entities.
Rule 5: The parameter entity corresponding to the super class in the generalization relationship is placed into the content model of the element or parameter entity corresponding to the subclass. (The super class E of the subclasses F and G)
Rule 6: The part in the aggregate relationship is placed as an element or a parameter entity of the content model of the element corresponding to the aggregate. (The aggregation between the class B and classes C, D and E)
Rule 7: The element corresponding to the attribute is placed into the content model of the element or the parameter entity corresponding to its class. (The attributes e1, e2 and e3 of the class E)

When we applied the rules to the class hierarchy on the left side of Figure 1 the tree diagram of the DTD on the right side was produced. The UML class diagram does not, however, define all the information needed to generate the DTD. First, the order of the attributes or parts is not fixed. Orders were defined by discussing with the users and examing the paper patient record. Second, elements corresponding attributes of the abstract class may be grouped in various ways when they are placed into the content models of the subclasses.

The relationship between the aggregate and the part defines how many elements are generated into the content model. If a part may exist many times there are two possibilities. If the relationship is 0..n, then the repeating element is marked by a star character * (for example, the class C). If the relationship is 1..n, then the repeating elements is marked by a plus character +. If the repeating part is abstract it generates a parameter entity into the content model of the element of its subclasses. This situation exists in our example for the class E.

5. Results and conclusions

We applied the method to define the SGML DTD for the patient record of the Department of Pediatrics in the Kuopio University Hospital.

First, we made a careful analysis with the staff of the outpatient ward of the Department of Pediatrics. We utilized the traditional data flows in describing how the EPR would be used and transfered in the department. We noticed that the doctors and nurses felt the data flow analysis as a natural way to describe their work.

Second, we modeled the content of the EPR with UML class diagrams. Here we tried to increase uniformity, reusability and comprehensibility of the system. At the same time we took into the consideration the queries that the doctors and nurses desired to ask from the EPR data.

Third, we applied the rules represented in Section 4 to the UML class diagrams to generate the SGML DTD of the EPR. The rules produced the tree diagram almost automatically without requiring "too much intuitive design".

The method produces a DTD which is very suitable to be used by the users. Since the classes and attributes in the UML diagram have very content-based names the DTD also uses the specific identifier names. The SGML editor programs use these names to advice the user. However, we are convinced that using a slightly modified set of rules to the same UML diagrams, also a DTD more flexible for queries can be generated. This will be the next step in our work. To get more material to test the method we are currently analyzing the work flows in the inpatient ward of the Department of Pediatrics. The interesting point will be whether the generated DTD differs from the DTD generated for the outpatient word.

Acknowledgments

This work was financed by the University of Kuopio. The authors would like to thank the persons of the ELSA (electronic patient record) project of the Kuopio University Hospital and the Department of Pediatrics for their help.

References

[1] G. Booch. *UML Users Guide.* Addison Wesley, Chicago, Il, 1998.

[2] V. Bradley. Innovative informatics CPR: Computerized patient record. *Journal of Emergency Nursing* **20** (1994) 230–232.

[3] T. Bürkle, H.-U. Prokosch, and J. Dodeck. Steps towards integration of knowledge based functions into a hospital system. In J. Brender *et al.* (ed), *MIE'96.* IOS Press, 1996, pp. 286–290.

[4] P.C. Carpenter. The electronic medical record: perspective from Mayo Clinic. *International Journal of Bio-Medical Computing* **34** (1994) 159–171.

[5] ISO 8879. *Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML).* ISO, Geneva, 1986.

[6] E. Maler and J. El Andaloussi. *Developing SGML DTDs from Text to Model to Markup.* Prentice Hall, Upper Saddle River, New Jersey, 1996.

[7] D. Pitty, C. Gordon, P. Reeves, A. Capey, P. Vieyra, and T. Rickards. The place of SGML and HTML in building electronic patient records. In C. Pappas *et al.* (ed), *MIE'97.* IOS Press, 1997, pp. 329–333.

[8] A. Salminen, K. Kauppinen, and M. Lehtovaara. Towards a methodology for document analysis. *Journal of the American Society for Information Science* **48(7)** (1997) 644–655.

[9] S. Shlaer and S.J. Mellor. *Object Lifestyles – Modeling the World in States.* Yourdon Press, Englewood Cliffs, NJ, 1992.

[10] B. Travis and D. Waldt. *The SGML Implementation Guide.* Springer-Verlag, Berlin, 1996.

[11] H. Yoshihara, K. Obe, K. Ohashi, R. Yamamoto, S. Yamazaki, Y. Hirose, K. Matsui, T. Hishiki, Y. Yamashita, M. Kimura, K. Minagawa, and H. Oyama. Standardization of exchange procedures of clinical information using Medical Markup Language (MML). In B. Cesnik *et al.* (ed), *MEDINFO 98 Proceedings.* IOS, Press, 1998, pp. 129.

[12] E. Yourdon. *Modern Structured Analysis.* Yourdon Press, Englewood Cliffs, NJ, 1989.