Structuring the electronic patient record

E. Kuikka, A. Eerola, J. Komulainen

Report A/2001/1

ISBN 951-781-618-9
ISSN 0787-6416

UNIVERSITY OF KUOPIO

Department of Computer Science
and Applied Mathematics

P.O.Box 1627, FIN-70211 Kuopio, FINLAND

Structuring the electronic patient record

E. Kuikka and A. Eerola
Department of Computer Science and Applied Mathematics
University of Kuopio

J. Komulainen
Department of Pediatrics
Kuopio University Hospital

Abstract

The paper presents a methodology for the structure definition of the electronic patient record. A patient record is typically a document which is updated by many users, required to be done in many different layouts, transferred from one place to another and archived for a long time. For these reasons, it is a very good candidate to be processed as a structured document and coded using SGML or XML document standards.

The design process starts at the most important issue as a goal to understand the work done in the hospital in such a level which makes it is possible to describe the information flow of the patient record and the content, structure and the relationships of the data needed in the treatment of the patients. This phase presents the information flow using data flow diagrams and the conceptual modeling of the content using object-oriented UML class diagrams. Next, the SGML/XML structure definition (Document Type Definition, DTD) is generated from the UML class diagrams using a set of correspondence rules.

The method gives a pragmatic way to design the structure of the electronic document. The results can be utilized for searching the possibilities to standardize the content of the electronic patient record.

1 Introduction

The patient record contains the identification information of a patient as well as information of the closest relatives, about medical facts, for example, allergies, blood group, etc. and about actions and diagnoses during clinical visits. Traditionally, a patient record is a pile of paper forms filled by the staff of various clinics and laboratories of a hospital. The main part consists of free text dictated or written by doctors and nurses but it can also contain drawings, pictures, photos and films. The textual content of the patient record consists of two types of data. The names of the patients, doctors, clinics, etc. are typical fixed-length data and suitable to be managed in the databases. The other type of information, such as entrance reason, preliminary information, problem, diagnoses, statements, etc., is represented in lines with variable lengths and written with the typewriter or a text processing system.

Many electronic patient record systems have been implemented using predefined techniques, for example, certain databases, certain programming languages, etc. [1, 2, 3, 4]. This approach has produced problems when additional features have been added into the system, when data has been transferred between various systems and when old data needs to be processed with new program versions.

An opposite approach to these specific implementations is taken in a set of systems or prototypes which represent the patient record as a hierarchically structured document containing the standardized markup [5, 6, 7, 8, 9, 10, 11, 12]. There exist currently two such standards. SGML (Standard Generalized Markup Language) [13] is an international standard to define the markup notation used in structured documents. XML (Extensible Markup Language) [14] as a subset of SGML is a World Wide Web Consortium Recommendation meant especially for structured Internet documents. For the SGML and

XML documents the structure can be defined in advance with the use of a Document Type Definition (DTD).

Very often the developing of the DTD is based on the paper form of a document. The designers use example documents or the knowledge of the user when they define a set of the potential semantic elements and hierarchical relationships between elements [15, 16]. There is a possibility that, for example for the patient record, the derived structure may conflict with what the doctors and nurses actually require in their everyday practises. The problem here is that the necessity of information is not considered and new information requirements may not be found. Further, the possibilities to evaluate the medical care and nursing routines may not be discovered.

The aim of this paper is to describe how an object-oriented method developed for the derivation of an SGML/XML DTD [17, 18] is applied in the structuring of the patient record. The method has the following steps:

1. The flow of the data in the clinic is considered from the point of view of the patient record. Requirements analysis [19], which resemples the traditional data flow analysis [20] is used in getting the general description of actions with the patient record.
2. The conceptual modeling is done using the Unified Modeling Language (UML) [21]. Here it is important that the model describes the utilization of the patient record in a natural way.
3. Elm (enables lucid models) tree diagrams [15] of the SGML/XML DTD are created from the UML class diagram using a set of correspondence rules introduced in [18].

After this introduction, Sections 3 and 5 review shortly the concepts which are important in understanding the method. Sections 2, 4 and 6 describe in detail the results of the three steps of the method. As the sample environment we have used the Kuopio University Hospital, especially its outpatient ward of the Department of Pediatrics. The conclusion in Section 7 finishes the paper.

2 Data flow in the clinic

The work started by making a careful analysis with the staff of the outpatient ward of the Department of Pediatrics in Kuopio University Hospital. Our target in the first step of the method was to analyze carefully where in the hospital the information of the patient record 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 patient record. To describe these issues, it is possible to use the object-oriented Unified Modeling Language (UML). In UML the analyst defines *use cases*, *class diagrams*, *state diagrams* and *interaction diagrams* [22, 23]. We felt, however, that the UML:

- does not show the boundaries of the system in consideration,
- does not give an overview of the functionality of the system, and
- is insufficient for the description of the flow of the information in the organization.

The use case diagram gives an overview of the interaction with the system not between the actors. Usually nonautomated parts are not included in the diagram. Each use case is described giving an exact information about the actors, pre- and post-conditions, special requirements, alternative scenarios (primary, secondary and exception handling), uses and extends relationships. From these data the analyst gets a lot of information for designing user interfaces and processing methods. However, the overview of the work and flow of information in the organization is missing. One possibility to describe this is business processes [24]. This was not suitable, because the Kuopio University Hospital has an organization with departments. Hence, we utilized the traditional *data flows* [20] in different levels in describing how the patient record is used and transferred in the hospital clinic. It was natural to ask from doctors, nurses and secretaries the following type of questions:

- What information do you need in the patient ward and in the developing of the action?
- What are the requirements to store the information?

- Why is the information important?
- From where does the information come and where does it go?
- What problems and development ideas do you have?

The *context diagram* is a good tool for describing the boundaries of the information system. What belongs to the target, for example, the clinic considered, and which are the external targets that send information to the target. Fig. 1 describes the persons or organizations which have connections to a hospital clinic when a patient comes to the clinic and when the patient record is processed. It also shows how the patient record and other involved documents are handled when doctors and nurses do their work.

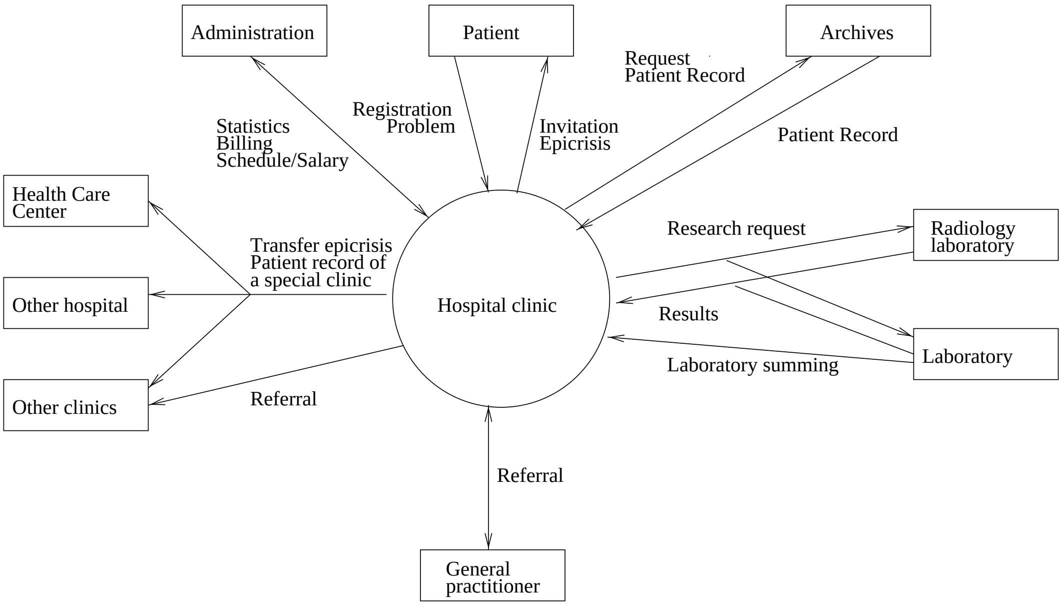


Figure 1: The context diagram of the hospital clinic

A more detailed description of the hospital clinic is represented in Fig. 2 as an *overview data flow diagram*. The clinic gets the patient record and some involved documents from outside organizations. Thereafter, the patient record is utilized and information is inserted into it in the work posts inside the clinic. The problem here is that the patient record is needed, perhaps even at the same time, in many places inside the clinic, inside the organization or inside the health care area. The secretaries do not always know where the patient records are. Thus, they cannot find them sufficiently fast. This causes that, for example, laboratory tests are performed many times increasing the costs.

Our analysis continued by describing in more detail the actions involved the patient record in various working posts of the clinic which are mainly responsible on the processing of the patient record. These results can be found in [25].

We noticed that the doctors and nurses felt the data flow analysis as a natural way to describe their work. They felt it easy and natural to describe what information they need in taking care of the patients. They also felt that the diagrams were sufficiently easy to read and understand and gave them a lot of information about their actions.

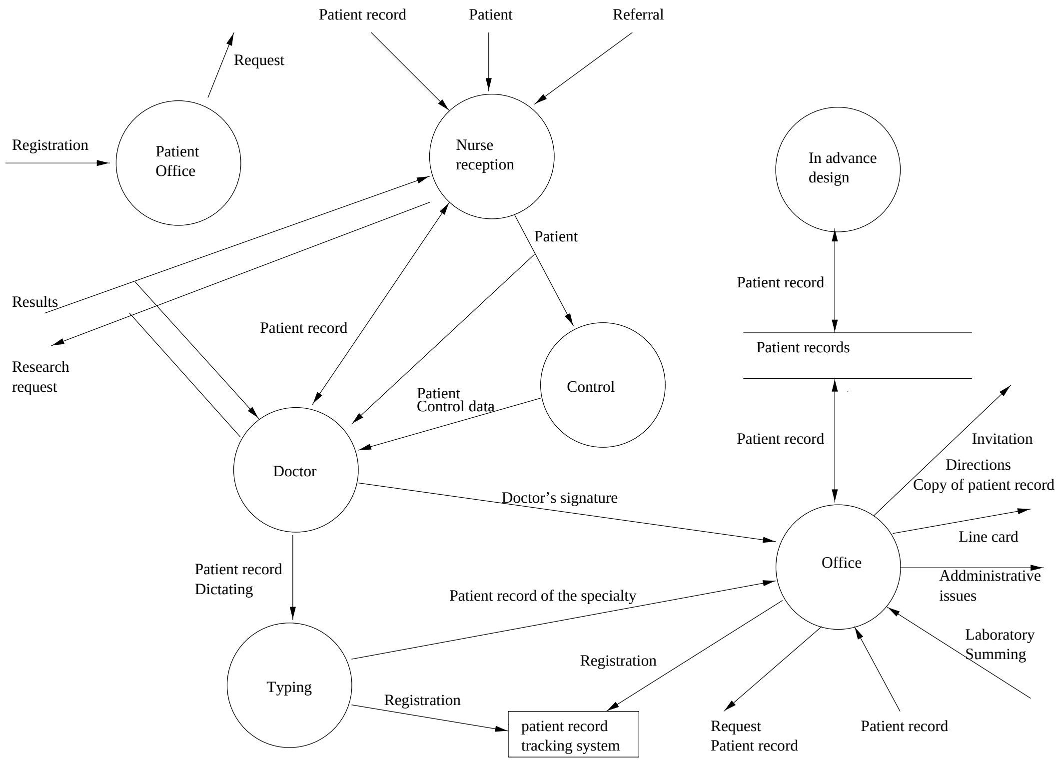


Figure 2: The overview data flow diagram in the hospital clinic

3 Object-oriented approach and UML disgrams

According to the object-oriented approach the concepts of the real world are *objects*. For example, the patient record, clinic, blood test, etc. are examples about objects. Each object possesses an *identity*, a *state* (defined by 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 association relationship may be one-to-one, one-to-many and many-to-many. The objects are defined in *classes* which form a class hierarchy. The objects sharing a common definition belong to the same class, while objects resembling each other belong to the same class hierarchy. The class hierarchy allows subclasses to inherit properties, i.e. attributes and methods, from super classes. The super class is a *generalization* of its subclasses. *Concrete classes* define objects. Other classes are *abstract classes* and they are used for the definition of the properties that subclasses inherit. The class hierarchy enables the analysis of the similarity of the objects. The class hierarchy is depicted by the UML class hierarchy [22].

In the graphical representation of the class hierarchy (see Fig. 3), classes are represented as rectangles; abstract classes with the bold borders. The class rectangles are divided into parts containing the name of the class, the attributes and the methods. The aggregation relationship is presented as a line with a diamond at one end and with numbers to indicate the type of association. The generalization relationship is depicted as a line with a triangle touched with the super class.

It is also possible to describe associations between objects in the class diagram; for example, the association "patient has a Medical record". Associations lead to references between objects. In Fig. 4 we have a class Patient associated with the Patient record class. This Patient class could have attributes, for example the name and address of the patient. However, if the objects of a class taking part in the relationship are not needed separately, it is possible to insert the information of the referred object in the aggregate object.

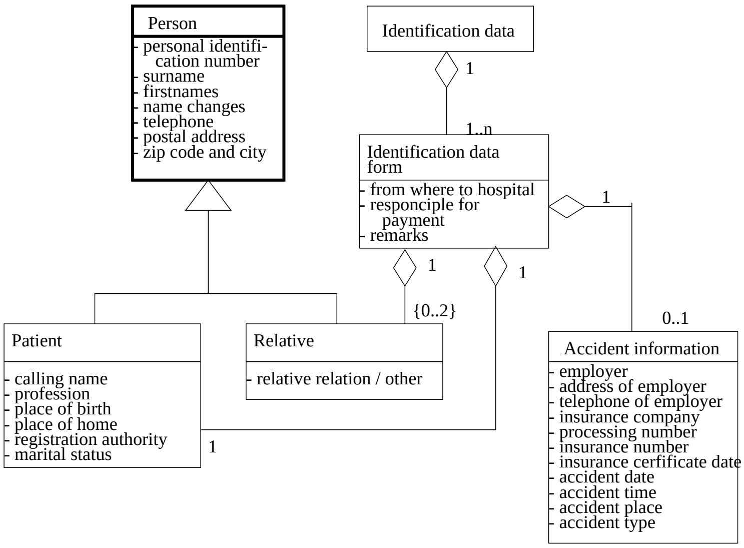


Figure 3: The class diagram of the identification data.

4 Conceptual modeling of the patient record using UML diagrams

In the second step of the method we modeled the conceptual content of the patient record of the departments in the Kuopio University Hospital using object-oriented UML class diagrams.

In the modeling of the patient record we depicted the facts collected while interviewing the nurses, doctors and secretaries. We also used all of the paper forms from the clinics. Our purpose was to increase profitability, usability, uniformity, comprehensibility and reusability of the system. For this reason we considered contents, structure and relationships of the concepts in the patient record as well as the queries that the doctors and nurses desired to ask from the electronic patient record data.

We first concluded that the patient record has the structure represented in Fig. 4. According to this class diagram, each patient has a patient record that consists of the identification data, the medical record and the nursing record.

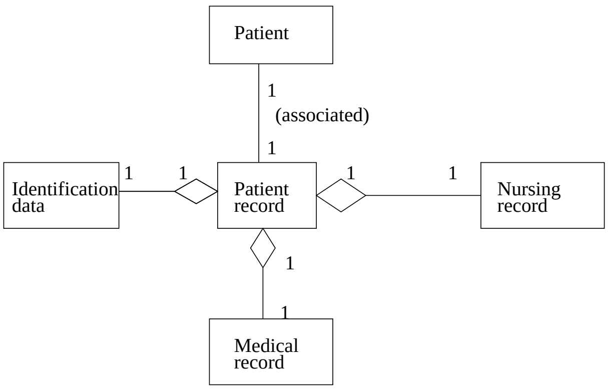


Figure 4: Class diagram of the patient record

Our analysis continued by modeling the three parts of the patient record. The class diagram of the identification data is represented in Fig. 3. Patient and relative subclasses are specifications for the abstract person class and inherit its attributes. The identification data forms collect information about the patient, her or his relatives as well as about the possible accident.

The class diagram of the medical record is represented in Fig. 5. For each special clinic the patient has visited, a continuing medical record is created. Further, the medical record consists of medical certificates, blood group and blood transfusion information (if needed) and a summary form with a set of summary markings. Summary markings (see Fig. 5) are derived from the visit notes of various clinics. The blood information consists of laboratory, subscriber and newborn's information.

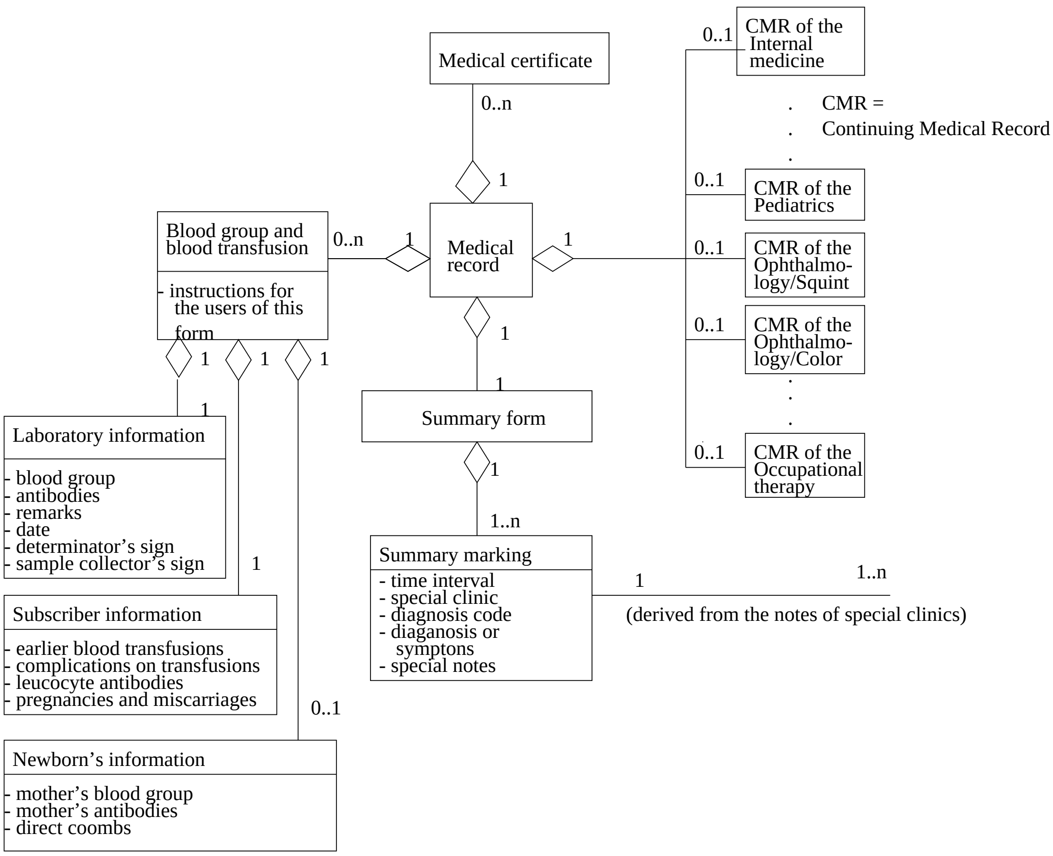


Figure 5: Class diagram of the medical record

The class diagram of the continuing medical record of the different special clinics is represented in Fig. 6. It specifies the structure of notes of various clinics. Certain clinics use the same structure which are defined by the general note type class. Here a negotiation question is, are the differences between clinics needed or could the clinics work in more standardised way using similar notes. Common information of subclasses is defined in the general information of the special clinic class and the confirmed note class. The last step is to optimise the class hierarchy. This problem is considered, for example, in [26, 27, 28].

The class diagram of the nursing record is represented in Fig. 7. As this diagram depicts, the nursing record consists on the nursing plan of the first aid, the diet information, entrance interview made by the nurse, the table of nursing and the continuing nursing plan. The continuing nursing plan contains a set of nursing plan notes. Each note consists of the nursing information, nursing plan as well as the progress, control and rating information of the nursing actions. The last mentioned class has the measurements class as a subclass.

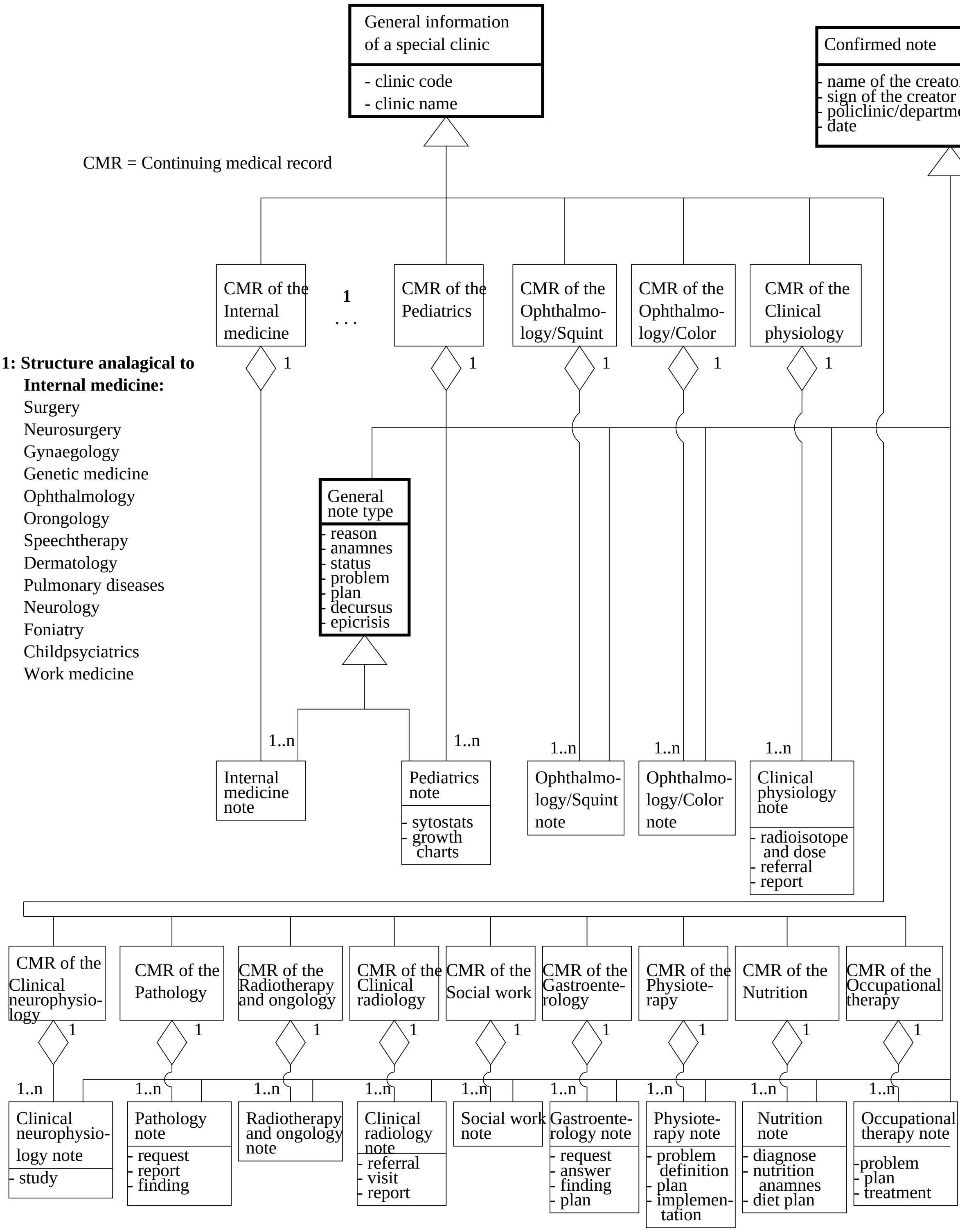


Figure 6: Class diagram of the continuing medical records of various special clinics

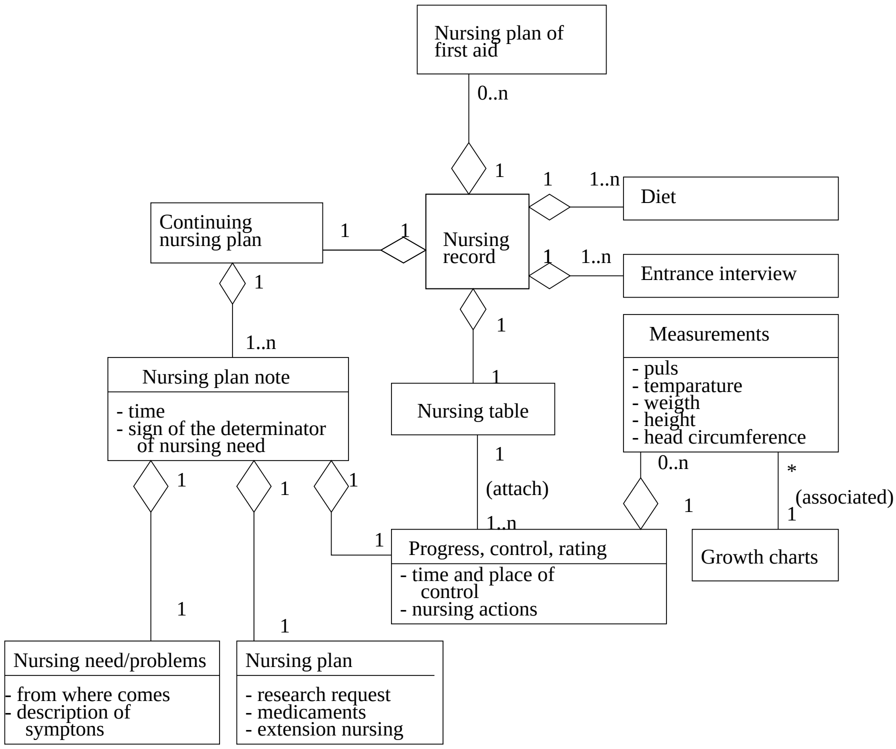


Figure 7: Class diagram of the nursing record

5 SGML and XML document DTDs

The SGML/XML DTD contains rules for elements of a document, for attributes of elements and for entities to replace character strings in a document instance as well as in a DTD. There are two types of elements: a *structure element* defines hierarchical structures and contains other elements and possibly also text, whereas a *content element* consists only of text. The SGML *element declaration* has three parts: the name of the element, indicators for the existence of tags in the SGML document instance (`-` for a compulsory tag and `O` for an optional tag), and the definition of the content model. According to the XML notation the second part is missing since all the tags must exist in the XML document instance. The content model (shown in parenthesis) of an element specifies the subelements, their order and existence conditions. The order of subelements is controlled using connectors between subelements: '`,`' for a sequence connector, '`|`' for a choice operator and '`&`' for a set (the free order) connector. Subelements can be repeated or be absent. The subelement name or a group (represented in parenthesis) is followed by `*` for zero or more repetitions, by `+` for one or more repetitions and by `?` for an optional existence. The content is defined using `#PCDATA`, `RCDATA` or `CDATA` data type in SGML and by `#PCDATA` in XML. To give metainformation usually not included in the actual content of a document, it is possible to define a list of *attributes* for each element.

The DTD also contains *entity declarations* for defining the identifiers of a character (for example, characters not in the keyboard), of a character string and of a content model. The last one is called a *parameter entity* and it exists in the DTD only for the purpose of shortening the DTD. In the parameter entity declaration, the name is preceded by a `%` character and the content model is delimited by quotation marks. The reference to a parameter entity in a content model of another element is made using the name, which is delimited by `%` and `;` characters.

The following SGML DTD defines that the `MedRec` element consists of the identification data of the patient (`IdentData`) followed by a sequence of continuing medical records from different clinics (for the pediatric clinic `ContMedRecPediatrics`). The name and address of the patient are character sequences.

The parameter entity `GenInfSpecClinic` defines the identifier for two sequential elements, the clinic code (`ClinicCode`) and the clinic name (`ClinicName`); these are used for every clinic. The continuing medical record from each clinic contains general information (`GenInfSpecClinic`) and at least one note for each visit at the clinic (`PediatricsNote`). The content of the note consists of general information (entities `ConfirmNote1` and `ConfirmNote2`) and some specific information (`Problem`, `History`, and `Status`).

```
<!ELEMENT MedRec                  - - (IdentData,
                                       ContMedRecPediatrics,...)>
<!ELEMENT IdentData               - - (Name,Address)>
<!ELEMENT (Name,Address)          - O (#PCDATA)>
<!ENTITY % GenInfSpecClinic "ClinicCode,ClinicName">
<!ELEMENT (ClinicCode,ClinicName) - O (#PCDATA)>
<!ELEMENT ContMedRecPediatrics    - - (%GenInfSpecClinic;,
                                        PediatricsNote+)>
<!ENTITY % ConfirmNote1 "Clinic,Date">
<!ENTITY % ConfirmNote2 "CreatorName,CreatorSign">
<!ELEMENT (Clinic,Date,CreatorName,
          CreatorSign)             - O (#PCDATA)>
<!ELEMENT PediatricsNote           - - (%ConfirmNote1;,Problem,
                                        History,Status,
                                        %ConfirmNote2;)>
<!ELEMENT (Problem,History,Status) - O (#PCDATA)>
```

An elm (enables lucid models) tree diagram for a DTD, developed by [15], defines graphical representations for the items of the DTD. As an example, Fig. 8 shows a part of the previous DTD as an elm tree diagram. Structure elements are illustrated as rectangles and content elements and parameter entities as ovals. Existence indicators are written next to these. The sequential order in the content model is denoted by a horizontal line. Although not shown in Fig. 8, each alternative subelement is connected to the element with a line and a set of subelements is enclosed in a round "bag".

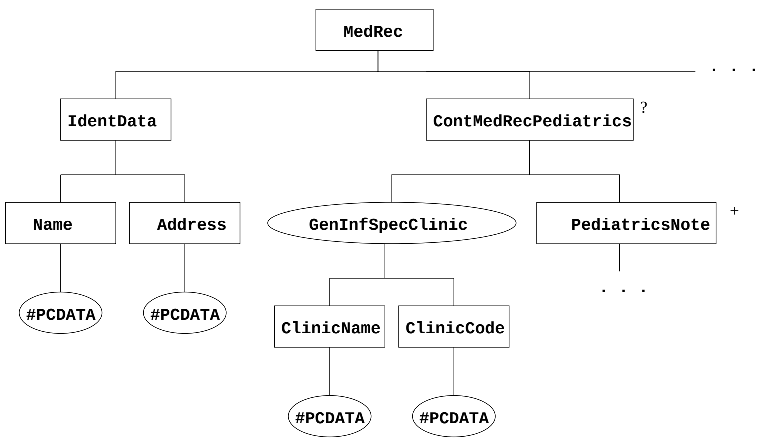


Figure 8: The elm tree diagram of a DTD

6 Transformation of UML diagrams to DTD diagrams

In the last step of the method, we used a following set of correspondence rules represented in detail in [17, 18] to transform the UML diagrams to an elm tree representation of the DTD.

Transformation rules define how an element (for example, class, attribute, etc.) of the UML diagram is transformed to a node of a elm tree diagram.
Rule 1: *A concrete class in a UML class diagram generates an SGML/XML structure element.*
Rule 2: *An abstract class in a UML class diagram generates an SGML/XML parameter entity.*
Rule 3: *An attribute in a UML class diagram generates an SGML/XML content element.*
Rule 4: *An SGML/XML parameter entity may be divided into many parameter entities.*

Placement rules define how the attributes as well as aggregate and generalization relationships specify the location of the generated nodes in the tree.
Rule 5: *The SGML/XML element corresponding to the attribute in a UML class diagram in placed in the content model of the element or the parameter entity corresponding to the class of the attribute. The place of the subelement among other subelements is fixed and the connector is selected. The repetition and optionality indicator for the SGML/XML subelement is as indicated for the attribute.*
Rule 6: *The SGML/XML parameter entity corresponding to a super class of a generalization relationship in a UML class hierarchy is placed in the content models of the SGML/XML elements or SGML/XML parameter entities corresponding to each of the subclasses. The place of the parameter entity among other elements or parameter entities is fixed and the connector is selected.*
Rule 7: *The part in an aggregate relationship in a UML class diagram is placed as an SGML/XML element or an SGML/XML parameter entity in the content model of the element or parameter entity corresponding to the aggregate. The place for the element or parameter entity among other elements or parameter entities of the content model is fixed and the connector is selected.*

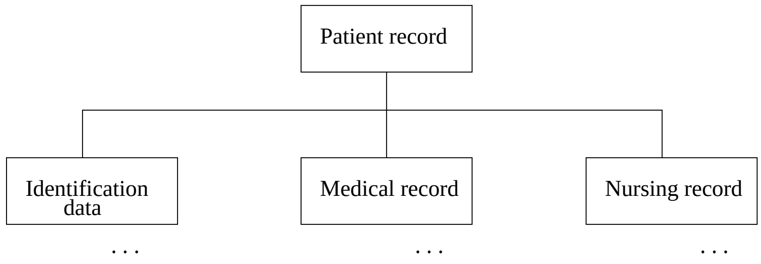
Processing rules define how the hierarchies of classes are processed.
Rule 8: *The generalization rule 6 is applied before aggregation rule 7.*
Rule 9: *The aggregation hierarchies are processed top-down and the classification hierarchies are processed bottom-up.*

Two issues concerning the application of the rules should be mentioned. Firstly, using the rules above, it is possible to generate the DTD which preserves the names of the classes and attributes defined in UML diagrams. However, although it is natural to use explanatory, and usually long names in UML diagrams, the same names are not suitable in the final DTD. When selecting names for the elements in the SGML DTD in Appendix we have used the naming rules represented in [30]. Secondly, whenever several elm tree nodes corresponding to the attributes or subclasses in a UML class diagram are placed as subnodes of an SGML/XML element, we have always used a sequential order. Other possibility was to connect the attributes using the operators as a purpose to allow their free order. The UML diagrams as their selves do not define any orders for its attributes or subclasses of a class.

Fig. 9 represents the elm tree diagram and the corresponding SGML DTD rule generated from the patient record class diagram in Fig. 4. According to this diagram the identification data, medical record and nursing record exist always in this order in the patient record. Their structures are depicted in subsequent diagrams.

Fig. 10 depicts the structure diagrams generated from the identification data class diagram in Fig. 3. The identification data consists of at least one group of data. Each group contains information about the patient, her or his relatives, how the patient came to the hospital, who is responsible about payments, remarks and, if an accident has been happened, information about it. In addition to their own data, the patient and her or his relative have common information which is defined as `Person 1` and `Person 2` parameter entities.

Fig. 11 and 12 represent the diagrams for the medical record generated from the UML diagrams in Fig. 5 and 6. However, saving the drawing space, Fig. 11 does not depict all the clinics. First, the medical record contains the summary of all the visits in the clinics. Second, blood information may be defined several times. Thereafter continuing medical records for each special clinics visited by the patient follow. The order of clinics is fixed. A group of clinics share the common structure of the visit note; it is defined by `General note type` parameter entity. Other clinics have their own note structure. According to these diagrams, the notes contain certain data in certain order. This kind of the approach is very suitable when the user writes the content; the application may direct the author and make her or him to




```
<!ELEMENT PatientRec - - (IdentificationData,MedicalRec,NursingRec)>
```

Figure 9: The elm tree diagram and the SGML DTD rule corresponding to the patient record class

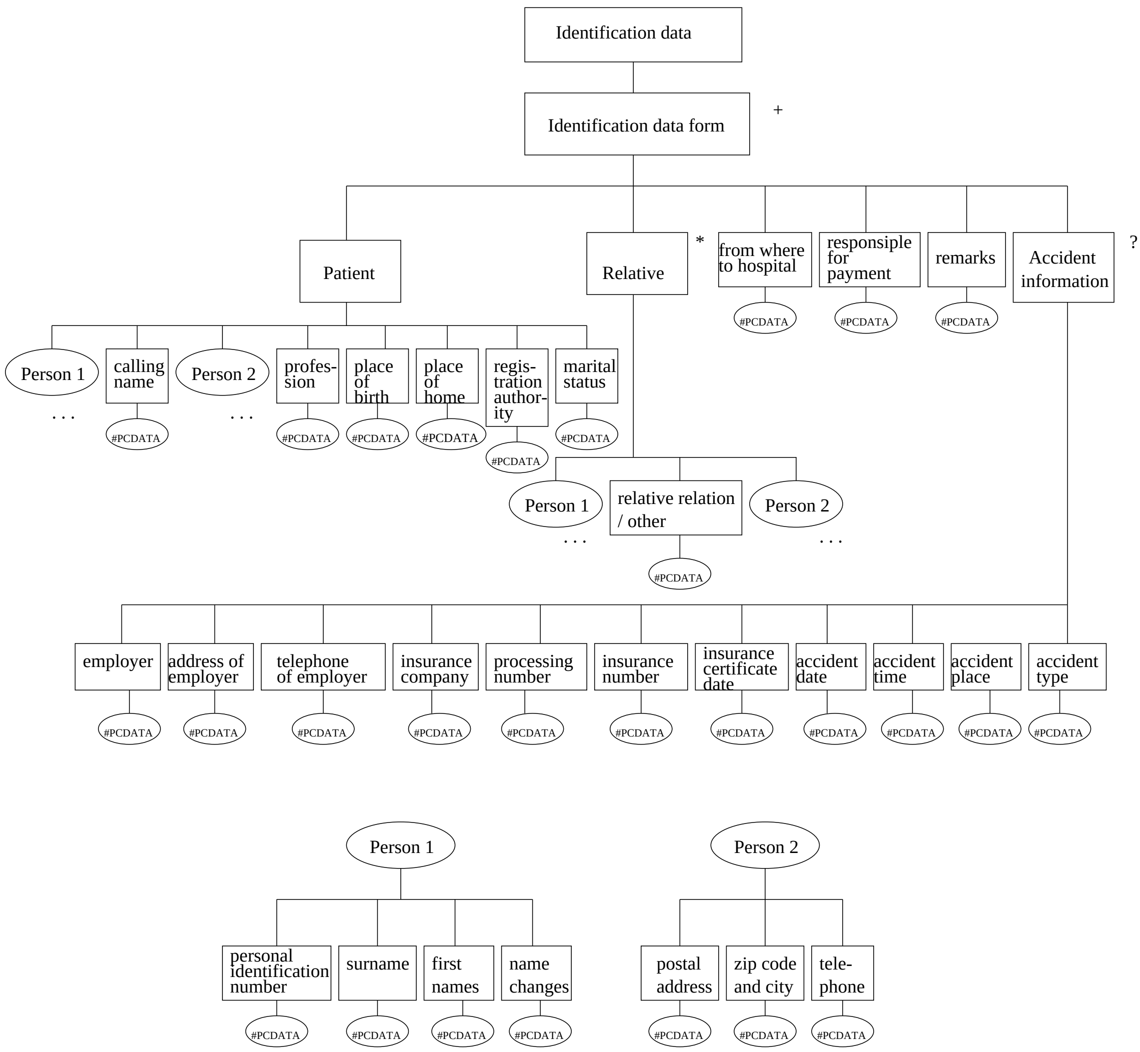


Figure 10: The elm tree diagram of the identification data element

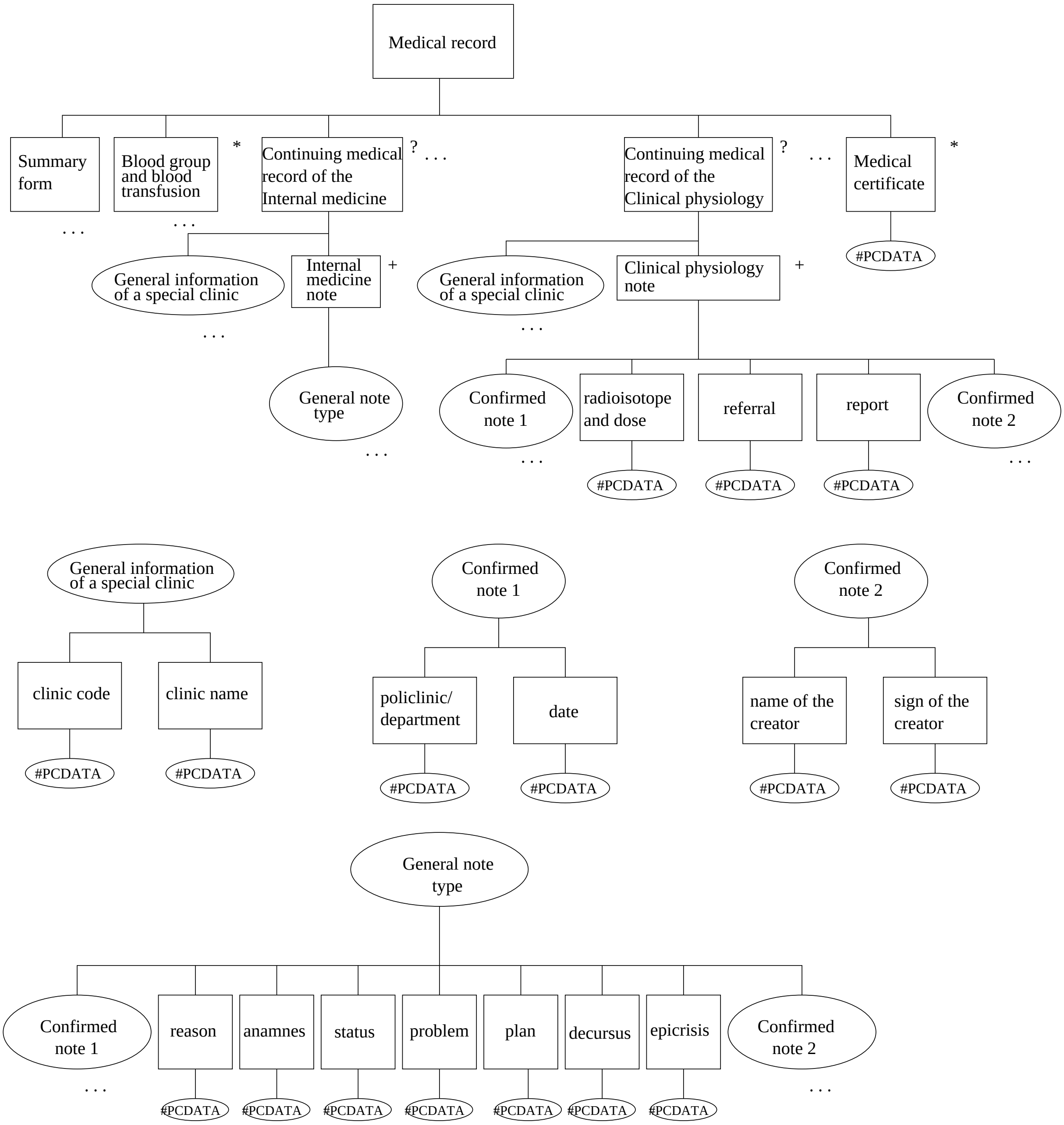


Figure 11: The elm tree diagram of the medical record

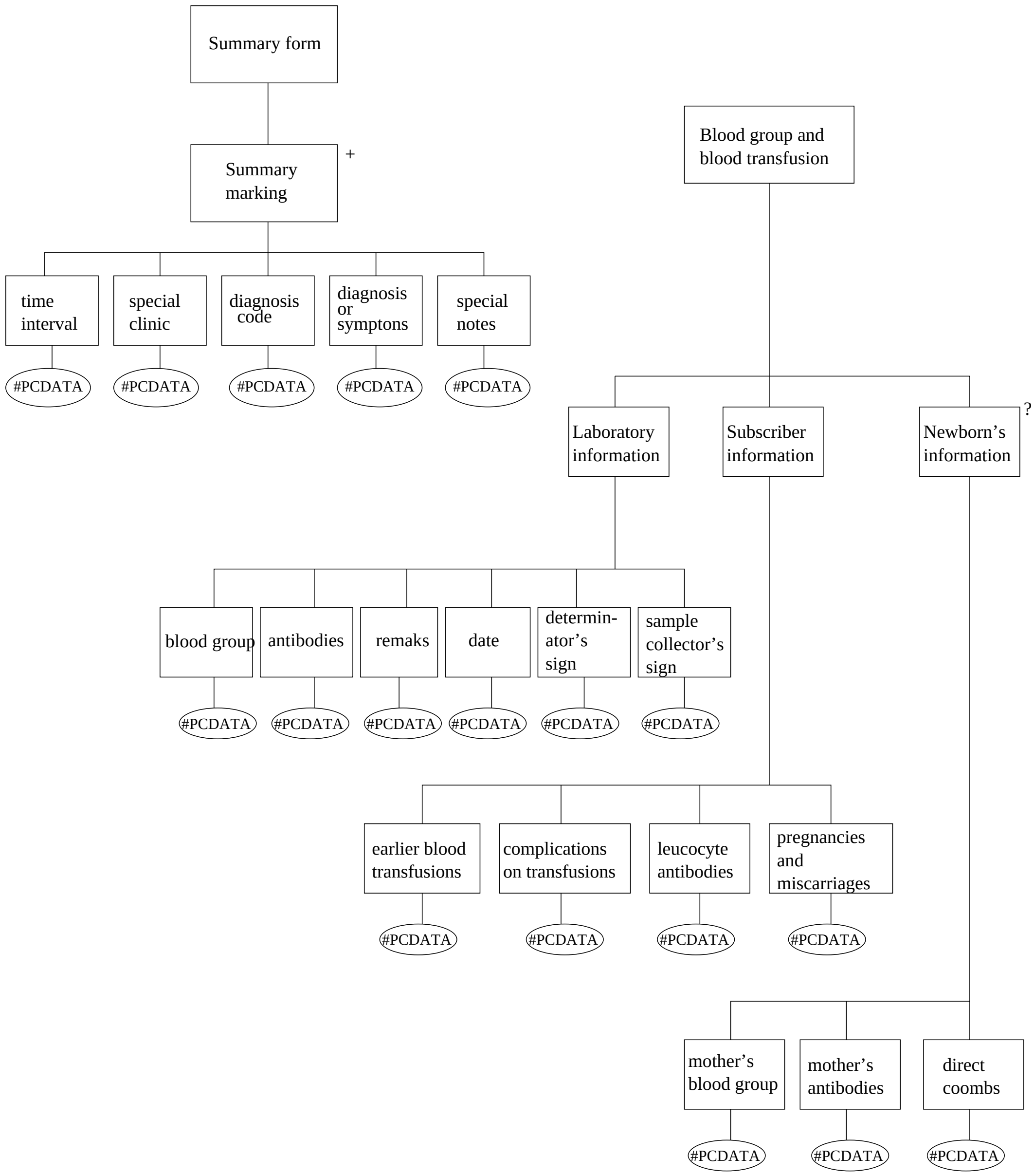


Figure 12: The elm tree diagrams of the summary form and the blood group and blood transfusion

input all the necessary information. This advantage is lost partly if the data of the notes are allowed to be in any order. The last information contains the given medical certificates.

Fig. 13 represents the generated nursing record diagram from Fig. 7. Up to now in our research, we have concentrated mainly on the medical record. For this reason, this diagram depicts the detailed structure only for the continuing nursing plan which have been modeled in Fig. 7.

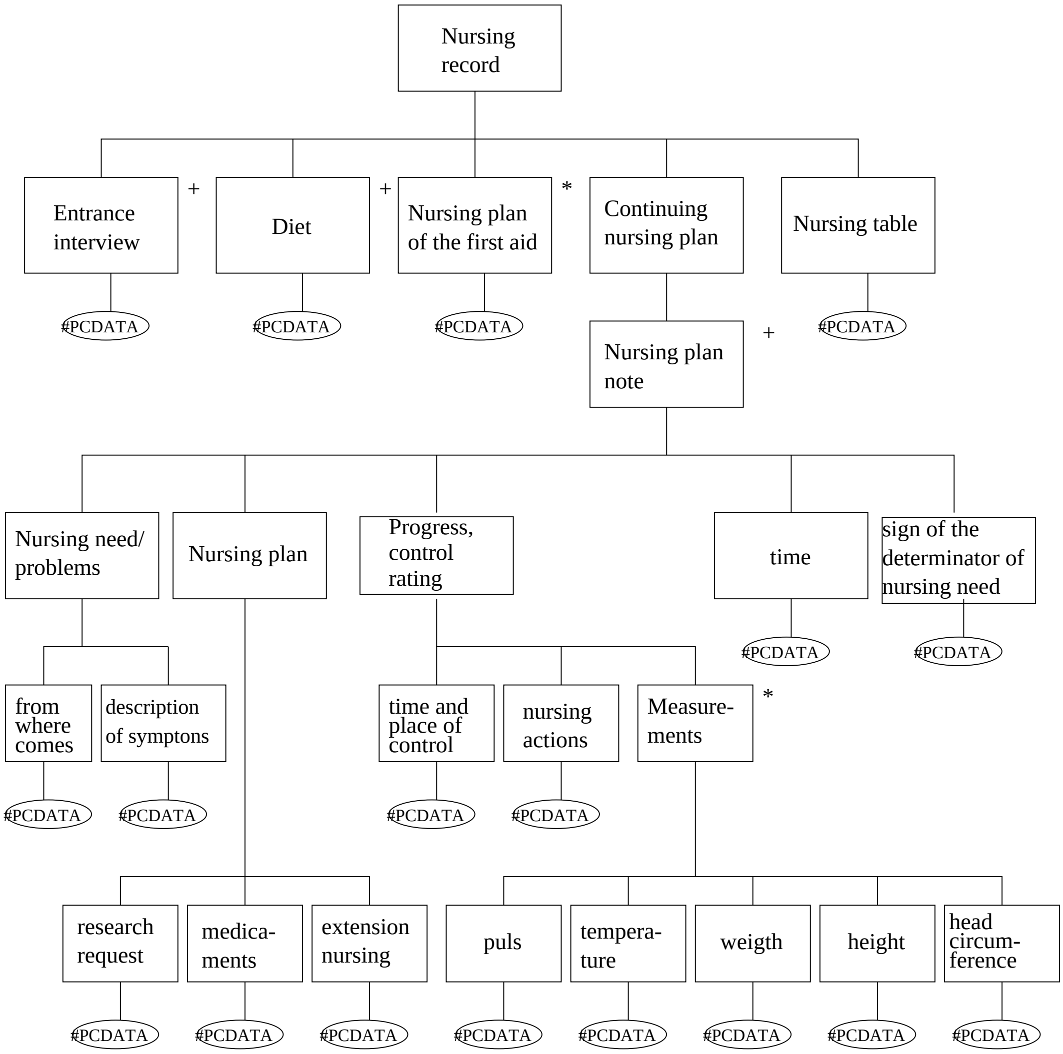


Figure 13: The elm tree diagram of the nursing record

The UML case tools, for example Rational Rose, allow the user to export the models. This is done with the XMI interchange format [29]. This specification defines how to create XML DTDs for the stream-based data format when models are transferred from one system to another. XMI is standardized by the Object Management Group (OMG).

We have implemented an semi-automatic mapping system [18] which uses the XMI-based DTD for the text export and applies our rules to the exported XML document to produce a textual representation for the DTD. Until now we have tested the system using simple diagrams. Our approach, using the XMI standard for the textual form, would make it possible to use our prototype also with other case tools which utilize XML as an export format. However, one problem is that XMI does not define a fixed DTD but only the rules for defining DTDs. Various systems can use their own DTDs or modify the existing ones. For example, the XMI outputs of Rational Rose and IBM XMI Toolkit differ from each other: the output of the latter does not contain all the information that our rules need.

7 Conclusion

We have created the structure of the patient record using an object-oriented DTD design method. We started by increasing comprehension of the work done in the hospital and described how, why, when the doctors and nurses read and update the patient record. The motivation to this was: 'If you do not understand the work you cannot model it.' The collected information was presented using data flow diagrams that also doctors and nurses understood. After this phase, we had a general overview of the system (both manual and automated parts). Further, we derived the conceptual model of the content of the patient record using object-oriented UML class diagrams. The SGML/XML DTD diagrams corresponding to the UML class diagrams were generated using a set of correspondence rules. For this phase, we have made a mapping system which generates the SGML DTD rules, not elm tree diagrams, from the XML representation of UML class diagrams.

Our method gives a pragmatic new way to design structured patient records. The results can be used also for searching the possibilities to standardize the content. Parallel to our work, the ISIS European XML/EDI Pilot Project [30, 31] has also developed the mapping rules from the UML diagram of a message to XML DTD. They, however, concentrate on the description of a message using UML diagrams, whereas we consider more carefully the analysis and design phases and semantics of electronic patient record.

Acknowledgements

The research was carried in the research project on Electronic Documents in Hospital financed by the University of Kuopio. The authors would like to thank Jari Porrasmaa, Ari Miettinen and Merja Ek for their contribution in this work and the personality of the Department of Pediatrics of the Kuopio University Hospital for their kind help.

References

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

[2] Bürkle T., Prokosch H.-U., and Dodeck J., 1996, Steps towards integration of knowledge based functions into a hospital system. *Medical Informatics Europe '96*, Copenhagen. August 1996, pp. 286–290.

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

[4] Paty D., Studney D., Redekop K., and Lublin F., 1994, MS COSTAR. A computerized patient record adapted for clinical research purposes. *Annals of Neurology*, **36**, S134–S135.

[5] Benson T., 1996, SGML, HTML and EPR, application of the standard generalized markup language (SGML) in electronic patient records. Available at http://www.mcis.duke.edu/standards/HL7/committees/sgml/references/sgmlepr.htm.

[6] Pitty D., Gordon C., Reeves P., Capey A., Vieyra P., and Rickards T., 1997, The place of SGML and HTML in building electronic patient records. *Medical Informatics Europe '97*, Thessaloniki. June 97, pp. 329–333.

[7] TadekaYH., Matsumura T., Okada T., and Kuwata S., 1997, Dynamic template driven data entry system for an EPR system. *Proceedings of Toward an Electronic Health Record Europe'97, Vol. 2*, London. October 1997, pp. 139–143.

[8] Tadeka H., Matsumura T., Okada T., Kuwata S., and Ioue M., 1997, Approach to establish an electronic patient record system in an intelligent hospital. *Proceedings of IMIA-WG4: Common Security Solutions for Communication Patient Data.* Available at http://dchps.hops.med.osaka-u.ac.jp/home/hp-info/paper/epr2/body.htm.

[9] Yoshihara H., Obe K., Ohashi K., Yamamoto R., Yamazaki S., Hirose Y., Matsui K., Hishiki T., Yamashita Y., Kimura M., Minagawa K., and Oyama H., 1998, Standardization of exchange procedures of clinical information using Medical Markup Language (MML). *MEDINFO 98 Proceedings*, Seoul. August 1998, pp. 129.

[10] Scott & White Hospital and Clinic, 1998, *Scott and White Electronic Medical Record Project.* Available at http://swinfo.tamu.edu/news/ options/july98/empx_his.htm.

[11] Labreze L., Lagouarde P., Dakin C., and Renaud-Salis J.-L., 1998, A web interface for multimedia electronic patient record: conceptual validation of the Aquitaine Health Iinformation Network prototypes. *Medical Informatics*, **23(1)**, 75–84.

[12] Jung B. and Grimson J., 1999, Synapses/SynEx goes XML. *Medical Informatics Europe '99*, Ljublijana. August 1999, pp. 906–911.

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

[14] W3C Recommendation, 1998, *Extensible Markup Language (XML) 1.0.* Available at http://www.w3.org/TR/REC-xml.

[15] Maler E. and el Andaloussi J., 1996, *Developing SGML DTDs from Text to Model to Markup* (Upper Saddle River;Prentice Hall).

[16] Travis B. and Waldt D., 1996, *The SGML Implementation Guide* (Berlin:Springer-Verlag).

[17] Kuikka E., Eerola A., Porrasmaa J., Miettinen A., and Komulainen J., 1999, Design of the SGML-based electronic patient record system with the use of object-oriented analysis methods. *Medical Informatics Europe '99*, Ljublijana. August 1999, pp. 838–841.

[18] Kuikka E. and Eerola A., 2000, A correspondence between UML diagrams and SGML/XML DTD. *Preliminary proceedings of the Fifth International Workshop on Principles of Digital Document Processing (PODDP'00)*, Munich, Germany. September 2000.

[19] Robertson S. and Robertson J., 1999, *Mastering the requirements process* (Harlow:Addison-Wesley).

[20] Yourdon E., 1989, *Modern Structured Analysis* (Englewood Cliffs:Yourdon Press).

[21] Booch G., 1998, *UML Users Guide* (Chicago:Addison Wesley).

[22] Booch G., Jacobson I., and Rumbaugh J., 1999, *The Unified Modeling Language Users Guide* (Reading:Addison Wesley).

[23] Jacobson I., Booch G., and Rumbaugh J., 1999, *The Unified Software Development Process* (Reading:Addison Wesley).

[24] Berztiss A., 1996, *Software methods for Business Reengineering* (Berlin:Springer-Verlag).

[25] Porrasmaa J., 1998, The description of the data flow and the current and future state of the patient record in the outpatient ward of the Department of Pediatrics of the Alava Hospital. Unpublished special exercise (in Finnish) (Kuopio, Finland:University of Kuopio, Department of Computer Science and Applied Mathematics).

[26] Putkonen A. and Kiekara M., 1997, A case-tool for supporting navigation in the class hierarchy. *Software Engineering Notes*, **22(1)**, 77–84.

[27] Chen J.-L. and Wang F.-J., 1998, An inheritance flow model for class hierarchy analysis. *Information Processing Letters*, **66**, 309–315.

[28] Eerola A., 1999, A disciplined approach to the maintenance of the class hierarchy. Technical Report Raport A/1999/2 (Kuopio, Finland:University of Kuopio, Department of Computer Science and Applied Mathematics).

[29] Object Management Group, 1998, *XML Metadata Interchange (XMI). Proposal to the OMG Object Analysis & Design Task Force RFP 3: Stream-based Model Intechange Format (SMIF).* Available at http://www.omg.org.

[30] ISIS European XML/EDI Pilot Project, 1999, *Mapping from UML Generalised Message Descriptions to XML DTDs.* Available at http://www.tieke.fi/isis-xmledi/D2/D22.htm.

[31] ISIS European XML/EDI Pilot Project, 1999, ISIS XML/EDI Project Website. Available at http://www.tieke.fi/ovt/.

Appendix: The SGML DTD of a patient record

```
<!DOCTYPE PatientRec [
<!ELEMENT PatientRec - - (IdentificationData,MedicalRec,NursingRec)>
<!-- Structure Definition of Identification Data -->
<!ELEMENT IdentificationData - - (IdentificationDataForm+)>
<!ELEMENT IdentificationDataForm - - (Patient,Relative*,FromWhereHospital,
             PaymentResponsible, Remarks,AccidentInfo?)>
<!ELEMENT (FromWhereHospital,PaymentResponsible,Remarks) - O (#PCDATA)>
<!ENTITY % Person1 "PersonalIdentificationNumber,Surname, FirstNames,
             NameChanges">
<!ENTITY % Person2 "PostalAddress, ZipCodeCity, Telephone">
<!ELEMENT (PersonalIdentificationNumber,Surname,FirstNames,NameChanges,
           Telephone,PostalAddress,ZipCodeCity) - O (#PCDATA)>
<!ELEMENT Patient - - (%Person1,CallingName,%Person2,Profession,BirthPlace,
             HomePlace,RegistrationAuthority,MaritalStatus)>
<!ELEMENT (CallingName,Profession,BirthPlace,HomePlace,RegistrationAuthority,
           MaritalStatus) - O (#PCDATA)>
<!ELEMENT Relative - - (%Person1,RelativeRelationOther,%Person2)>
<!ELEMENT RelativeRelationOther - O (#PCDATA)>
<!ELEMENT AccidentInfo - - (Employer,EmployerAddress,EmployerTelephone,
             InsuranceCompany,ProcessingNumber,InsuranceNumber,
             InsuranceCertificateDate,AccidentDate,AccidentTime,AccidentPlace,
             AccidentType)>
<!ELEMENT (Employer,EmployerAddress,EmployerTelephone,InsuranceCompany,
           ProcessingNumber,InsuranceNumber,InsuranceCertificateDate,
           AccidentDate,AccidentTime,AccidentPlace,AccidentType) - O (#PCDATA)>
<!-- Structure Definition of the Medical Record -->
<!ELEMENT MedicalRec - - (SummaryForm,BloodGroupTransfusion*,
           InternalMedicineContinuingMedicalRec?,SurgeryContinuingMedicalRec?,
           NeuroSurgeryContinuingMedicalRec?,GynaegologyContinuingMedicalRec?,
           GeneticMedicineContinuingMedicalRec?,PediatricsContinuingMedicalRec?,
           OphthalmologyContinuingMedicalRec?,
           OphthalmologySquintContinuingMedicalRec?,
           OphthalmologyColorContinuingMedicalRec?,
           OrongologyContinuingMedicalRec?,FoniatryContinuingMedicalRec?,
           SpeechTherapyContinuingMedicalRec?,DermatologyContinuingMedicalRec?,
           RadioTherapyOngologyContinuingMedicalRec?,
           PsychiatryContinuingMedicalRec?,ChildrenPsychiatryContinuingMedicalRec?,
           NeurologyContinuingMedicalRec?,PulmonaryDiseasesContinuingMedicalRec?,
           ClinicalPhysiologyContinuingMedicalRec?,
           ClinicalNeuroPhysiologyContinuingMedicalRec?,
           ClinicalRadiologyContinuingMedicalRec?,
           GastroenterologyContinuingMedicalRec?,
           PathologyContinuingMedicalRec?,SocialWorkContinuingMedicalRec?,
           PhysiatricsContinuingMedicalRec?,WorkMedicineContinuingMedicalRec?,
           MedicalRehabilitationContinuingMedicalRec?,
           NutritionContinuingMedicalRec?,
           OccupationalTherapyContinuingMedicalRec?,MedicalCertificate*)>
<!ENTITY % SpecialClinicGeneralInfo "ClinicCode,ClinicName">
<!ELEMENT (ClinicCode,ClinicName) - O (#PCDATA)>
<!ELEMENT InternalMedicineContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
              InternalMedicineNote+)>
```

```
<!ELEMENT SurgeryContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        SurgeryNote+)>
<!ELEMENT NeuroSurgeryContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        NeuroSurgeryNote+)>
<!ELEMENT MedicalRehabilitationContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        MedicalRehabilitationNote+)>
<!ELEMENT GynaegologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        GynaegologyNote+)>
<!ELEMENT GeneticMedicineContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        GeneticMedicineNote+)>
<!ELEMENT PediatricsContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        PediatricsNote+)>
<!ELEMENT PsychiatryContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        PsychiatryNote+)>
<!ELEMENT ChildrenPsychiatryContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        ChildrenPsychiatryNote+)>
<!ELEMENT OphthalmologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        OphthalmologyNote+)>
<!ELEMENT OrongologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        OrongologyNote+)>
<!ELEMENT SpeechTherapyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        SpeechTherapyNote+)>
<!ELEMENT DermatologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        DermatologyNote+)>
<!ELEMENT PulmonaryDiseasesContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        PulmonaryDiseasesNote+)>
<!ELEMENT NeurologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        NeurologyNote+)>
<!ELEMENT FoniatryContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        FoniatryNote+)>
<!ELEMENT WorkMedicineContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        WorkMedicineNote+)>
<!ELEMENT OphthalmologySquintContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        OphthalmologySquintNote+)>
<!ELEMENT OphthalmologyColorContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        OphthalmologyColorNote+)>
<!ELEMENT ClinicalPhysiologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        ClinicalPhysiologyNote+)>
<!ELEMENT ClinicalNeuroPhysiologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        ClinicalNeuroPhysiologyNote+)>
<!ELEMENT PathologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        PathologyNote+)>
<!ELEMENT RadioTherapyOngologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        RadioTherapyOngologyNote+)>
<!ELEMENT ClinicalRadiologyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        ClinicalRadiologyNote+)>
<!ELEMENT SocialWorkContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        SocialWorkNote+)>
<!ELEMENT GastroenterologyContinuingMedicalRec  - - (%SpecialClinicGeneralInfo;,
        GastroenterologyNote+)>
<!ELEMENT PhysiatricsContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
        PhysiatricsNote+)>
<!ELEMENT NutritionContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
```

```
              NutritionNote+)>
<!ELEMENT OccupationalTherapyContinuingMedicalRec - - (%SpecialClinicGeneralInfo;,
              OccupationalTherapyNote+)>
<!ENTITY % ConfirmedNote1 "PoliclinicDepartment,Date">
<!ENTITY % ConfirmedNote2 "CreatorName,CreatorSign">
<!ELEMENT (PoliclinicDepartment,Date,CreatorName,CreatorSign) - O (#PCDATA)>
<!ENTITY % GeneralNoteType "%ConfirmedNote1;,Reason,Anamnes,Status,Problem,
              Plan,Decursus,Epicrisis,%ConfirmedNote2;">
<!ELEMENT (Reason,Anamnes,Status,Problem,Plan,Decursus,Epicrisis) - O (#PCDATA)>
<!ELEMENT (InternalMedicineNote,SurgeryNote,NeuroSurgeryNote,PsychiatryNote,
           ChildrenPsychiatryNote,GynaegologyNote,GeneticMedicineNote,
           OphthalmologyNote,OrongologyNote,SpeechTherapyNote,DermatologyNote,
           PulmonaryDiseasesNote,NeurologyNote,FoniatryNote,
           PediatricsPsychiatryNote,WorkMedicineNote) - - (%GeneralNoteType;)>
<!ELEMENT PediatricsNote - - (%GeneralNoteType;,Sytostats,GrowthCharts)>
<!ELEMENT OphthalmologySquintNote - - (%ConfirmedNote1;,%ConfirmedNote2;)>
<!ELEMENT OphthalmologyColorNote - - (%ConfirmedNote1;,%ConfirmedNote2;)>
<!ELEMENT ClinicalPhysiologyNote - - (%ConfirmedNote1;,RadioIsotopeDose,
               Referral,Report,%ConfirmedNote2;)>
<!ELEMENT ClinicalNeuroPhysiologyNote - - (%ConfirmedNote1;,Study,
               %ConfirmedNote2;)>
<!ELEMENT PathologyNote - - (%ConfirmedNote1;,Request,Report,Finding,
               %ConfirmedNote2;)>
<!ELEMENT RadioTherapyOngologyNote - - (%ConfirmedNote1;,%ConfirmedNote2;)>
<!ELEMENT ClinicalRadiologyNote - - (%ConfirmedNote1;,Referral,Visit,Report,
               %ConfirmedNote2;)>
<!ELEMENT SocialWorkNote - - (%ConfirmedNote1;,%ConfirmedNote2;)>
<!ELEMENT MedicalRehabilitationNote - - (%ConfirmedNote1;,%ConfirmedNote2;)>
<!ELEMENT GastroenterologyNote - - (%ConfirmedNote1;,Request,Answer,Finding,
               Plan,%ConfirmedNote2;)>
<!ELEMENT PhysiatricsNote  - - (%ConfirmedNote1;,ProblemDefinition,Plan,
               Implementation,%ConfirmedNote2;)>
<!ELEMENT NutritionNote - - (%ConfirmedNote1;,Diagnose,NutritionAnamnes,
               DietPlan,%ConfirmedNote2;)>
<!ELEMENT OccupationalTherapyNote - - (%ConfirmedNote1;,Problem,Plan,
               Treatment,%ConfirmedNote2;)>
<!ELEMENT (Sytostats,GrowthCharts,RadioIsotopeDose,Referral,Report,Study,
           Request,Answer,Finding,Visit,ProblemDefinition,Implementation,
           Diagnose,NutritionAnamnes,DietPlan,Treatment) - O (#PCDATA)>
<!ELEMENT SummaryForm - - (SummaryMarking+)>
<!ELEMENT SummaryMarking - - (TimeInterval,ClinicSpeciality,DiagnosisCode,
                DiagnosisSymptoms,SpecialNotes)>
<!ELEMENT (TimeInterval,ClinicSpeciality,DiagnosisCode,DiagnosisSymptoms,
           SpecialNotes) - O (#PCDATA)>
<!ELEMENT BloodGroupTransfusion - - (LaboratoryInfo,SubscriberInfo,NewbornsInfo?)>
<!ELEMENT LaboratoryInfo - - (BloodGroup,Antibodies,Remarks,Date,
                DeterminatorsSign,SampleCollectorsSign)>
<!ELEMENT (BloodGroup,Antibodies,DeterminatorsSign,SampleCollectorsSign)
                - O (#PCDATA)>
<!ELEMENT SubscriberInfo - - (EarlierBloodTransfusions,TransfusionsComplications,
                LeucosyteAntibodies,PregnanciesMiscarriages)>
<!ELEMENT (EarlierBloodTransfusions,TransfusionsComplications,
           LeucosyteAntibodies,PregnanciesMiscarriages) - O (#PCDATA)>
```

```
<!ELEMENT NewbornsInfo - - (MothersBloodGroup,MothersAntibodies,DirectCoombs)>
<!ELEMENT (MothersBloodGroup,MothersAntibodies,DirectCoombs) - O (#PCDATA)>
<!ELEMENT MedicalCertificate - - (#PCDATA)>
<!-- Structure Definition of the Nursing Record -->
<!ELEMENT NursingRec - - (EntranceInterview+,Diet+,FirstAidNursingPlan*,
                ContinuingNursingPlan,NursingTable)>
<!ELEMENT (EntranceInterview,Diet,FirstAidNursingPlan,NursingTable) - O (#PCDATA)>
<!ELEMENT ContinuingNursingPlan - - (NursingPlanNote+)>
<!ELEMENT NursingPlanNote - - (NursingNeedProblems,NursingPlan,
                ProgressControlRating,Time,NursingNeedDeterminatorsSign)>
<!ELEMENT NursingNeedProblems - - (ComesFrom,SymptomsDescription)>
<!ELEMENT NursingPlan - - (ResearchRequest,Medicaments,ExtensionNursing)>
<!ELEMENT ProgressControlRating - - (ControlTimePlace,NursingActions,
                Measurements*)>
<!ELEMENT Measurements - - (Puls,Temparature,Weigth,Height,HeadCircumference)>
<!ELEMENT (Time,NursingNeedDeterminatorsSign,ComesFrom,SymptomsDescription,
           ResearchRequest,Medicaments,ExtensionNursing,ControlTimePlace,
           NursingActions,Puls,Temperature,Weigth,Height,HeadCircumference) - O (#PCDATA)>
]>
```