MEDINFO 98
B. Cesnik et al. (Eds)
Amsterdam: IOS Press
© 1998 IMIA. All rights reserved

Hospital Information System: Resuability, Designing, Modelling, Recommendations for Implenting

B. Huet

Universite Paris XIII, Faculte de Medecine, Dept de Biostatistiques es Informatique Medicale, France

Abstract

The aims of this paper are to precise some essential conditions for building reuse models for hospital information systems (HIS) and to present an application for hospital clinical laboratories. Reusability is a general trend in software, however reuse can involve a more or less part of design, classes, programs; consequently, a project involving reusability must be precisely defined. In the introduction it is seen trends in software, the stakes of reuse models for HIS and the special use case constituted with a HIS. The main three parts of this paper are: 1) Designing a reuse model (which objects are common to several information systems?) 2) A reuse model for hospital clinical laboratories (a gen-spec object model is presented for all laboratories: biochemistry, bacteriology, parasitology, pharmacology, ...) 3) Recommendations for generating plug-compatible software components (a reuse model can be implemented as a framework, concrete factors that increase reusability are presented). In conclusion reusability is a subtle exercise of which project must be previously and carefully defined.

Keywords:

Hospital information systems; Design; Modeling;

1 - Introduction

Customer satisfaction is a top priority. In hospital information systems (HIS) customers are clinicians, nurses, engineers, health plan administrators for the benefit of patients. Under the influence of advances in information technology (object, CASE tools, modeling methods,...) several HIS paradigms are shifting (the computer-based patient record shifts to a larger structure, clinical information systems functions shift for larger functions integrating critical pathways and case management, EIS must be designed for hospital managers,...).
A new generation of software for HIS must be designed.
Design is hard and expensive [1]. One way to avoid the act of design is to reuse designs (many advantages: less expensive, better quality, easier maintenance, ...) this old idea becomes today more realistic thanks to recent advances in software: the promise of object technology is predicated on the creation of reusable, plug-compatible software components. Reuse-oriented design is a basic trend for software today.
Three points are seen in this paper: 1) conditions for designing a reuse model, 2) proposal of a reuse model for hospital clinical laboratories, 3) recommendations for implementing a reuse model

2 - Designing a Reuse model

2.1. Generalities about models

To model is defined [2] as « to build a particular and partial view of the Information System of an organization ». A model is characterized by its limits and conditions of validity , consequently, a modeling method provides a precise and partial view of an organization.
The right modeling is to choose the right method for the right objective of modeling. A good model is a right model, a right model is a useful model, a useful model is a customer satisfaction-oriented model.
There are four classes of modeling methods: functional methods of which key idea is a description of the functions to be performed by the system, relational methods of which key idea is a description of data as relations, object methods of which key idea is to capture those concepts from the real world that are important to an application, dynamic model of

which key idea is to describe those aspects of a system concerned with time and sequencing of operations.

2.2. Conditions for designing a reuse model

A reuse model is a generalization / specialization structure with a generalization class at the top and specialization classes below.
Each generalization / specialization structure forms either a hierarchy or a lattice.
To build a reuse model supposes that:

1) there is a precised project based on customer satisfaction of several information systems.

2) it is wanted to model several information systems with the same objective of modeling, i.e. the same point of view.

3) the objects of these information systems have common classes or attributes, i.e. there is an intersection between objects (e.g. hospital laboratory of biochemistry, hospital laboratory of bacteriology, hospital laboratory of haematology,...).

4) the designer can highlight this intersection between objects as superclass (the reusable component) and to identify specific components as subclasses. This designer can answer to difficult questions such as: which objects of this information system are common to that other information system [3], moreover he has to answer to more difficult questions such as: is there inheritance between this class and that class ? The designer must work in cooperation with an expert of the applied domain.

3 - Modeling a reuse model for hospital clinical laboratories

3.1 Why hospital clinical laboratories are adequate for a reuse model

Four arguments plead for designing a reuse model for hospital clinical laboratories:

1) Customer satisfaction is about easier access to all laboratories information systems, moreover there is an increasing interest for clinical laboratories decision support systems that implies an easy access to information of all laboratories information systems.

2) Hospital clinical laboratories constitute several information systems (biochemistry, bacteriology, parasitology, nuclear medicine, haematology, anatomo-pathology,...) with common features.

Biologists have the same basic function (to produce high quality information for completing patient's medical information), there are common steps for producing information in laboratories (checking specimen, producing results, quality control, interpreting results, reporting results), there is a same objective of modeling (to result to efficient clinical laboratory information systems). Consequently, for a designer there is the same point of view of modeling for all hospital clinical laboratories.

3) The objects to be modelled for hospital clinical laboratories have common classes and / or attributes [4], i.e. there is an intersection between objects of several laboratories (e.g. hospital laboratory of biochemistry, hospital laboratory of bacteriology, hospital laboratory of haematology,...).

Any sample has an identifier, a corresponding clinician and a ward, on an other hand any result must be controlled in such a way a status is attached to any result,...

Consequently, certain classes are common to all hospital clinical laboratories, they can be identified as the basis of a reuse model.

4) The designer can highlight a generalization / specification structure of classes and objects.

The analysis of attributes shows that there are common objects with inheritance to all clinical laboratories, these objects can be grouped in a global generalization pack, below there are several classes with several levels of generalization / specialization and lattices.

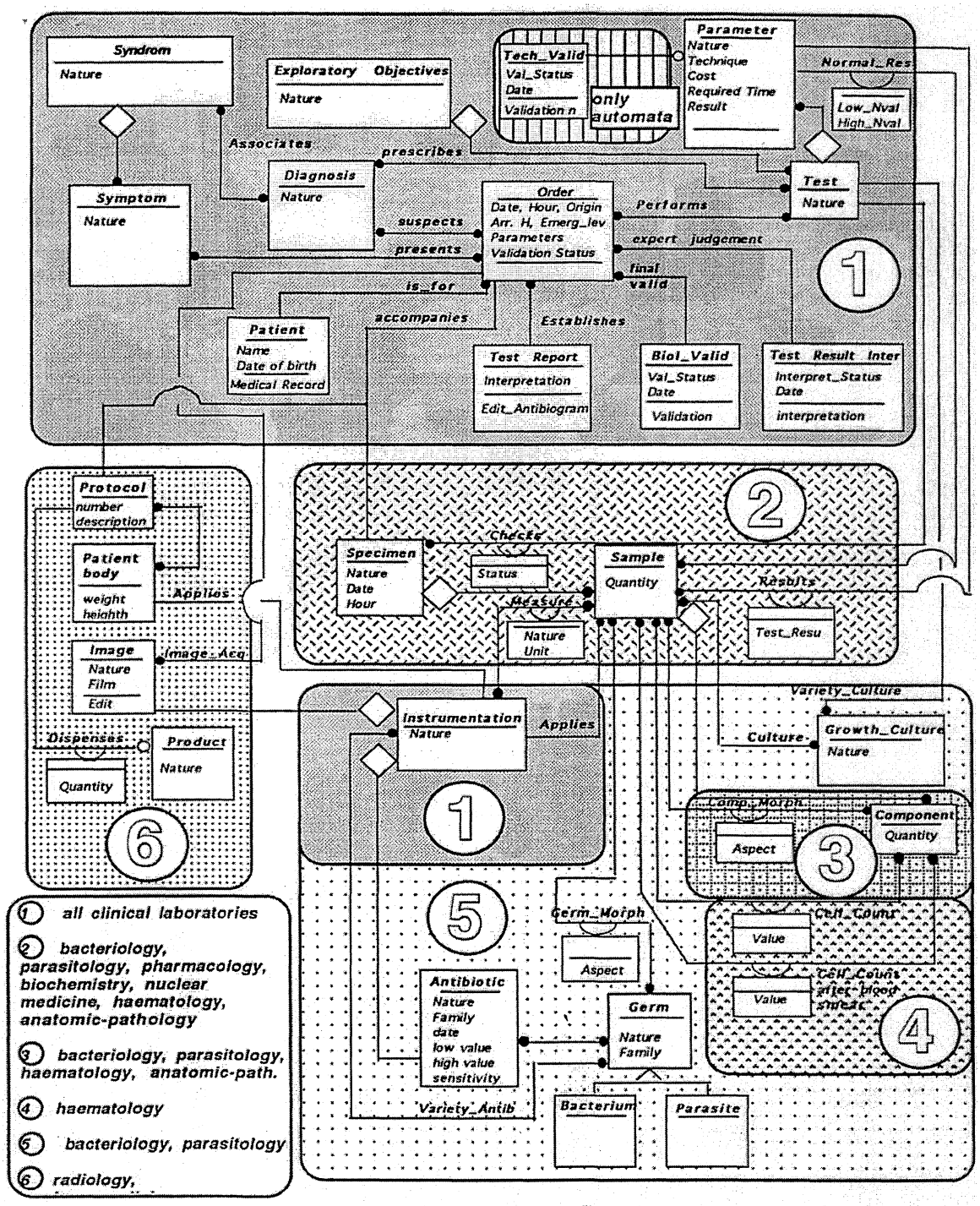
3.2 Designing a reuse model for hospital clinical laboratories

For designing a reuse model, it is necessary to be a good expert because of crossed questions, for example for hospital clinical laboratories one concrete question is: which objects of the clinical laboratory of biochemistry can be reused in the clinical laboratory of bacteriology ?

Classical other questions we tried to answer are: which aspects are common to all laboratories ?, which aspects differ from laboratory of biochemistry to other laboratories ? which aspects are common to automated laboratories ? which aspects are specific to each laboratory ? ...

3.3. Proposal for a reuse model

Figure 1 shows general and specific objects using O.M.T. [5] for all clinical laboratories (biochemistry, pharmacology, haematology, bacteriology, parasitology, anatomic pathology, radiology and nuclear medicine). This model highlights a structure of 6 groups (fig. 2) with inheritance [6], one group is composed of objects found in one or several clinical laboratories (table 1).



the reuse-oriented object model

fig 1

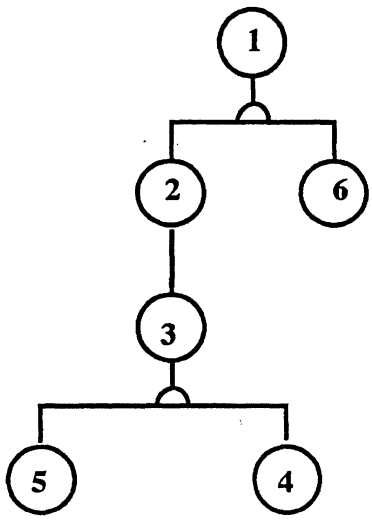


figure 2: inheritance between objects

group	laboratories
1	all clinical laboratories
2	bacteriology, parasitology, pharmacology, biochemistry, nuclear medicine, haematology, anatomic-pathology
3	bacteriology, parasitology, haematology, anatomic-pathology
4	haematology
5	bacteriology, parasitology
6	radiology, nuclear medicine

table 1: groups and laboratories

4 - Recommendations for implementing a reuse model

4.1. How to implement a reuse model

Reusability of components across applications in open environments mandates that the internals of the component be hidden from the user. It is the responsibility of software to supply an architecture to hide the internals. An architecture is the way the parts work together to make the whole: frameworks are a particular way of representing architectures.
Beck and Johnson [7] define a framework as « the reusable design of a system or a part of a system expressed as a set of abstract classes and the way instances of those classes collaborate ».
There are architectures that cannot be expressed as frameworks; design patterns are an other approach to design, with an overlap between frameworks and patterns.
Patterns are both a description of a recurring pattern of architectural elements and a rule for how and when to create this pattern.

4.2. Concrete aspects for plug-compatible software components

Using survey data (16 questions) collected recently from 29 organizations in the U.S. and in Europe, Frakes and Fox [8] consider that several factors increase reuse:
1) programming languages: it is argued that features of Ada, C++, Smalltalk, Eiffel, (e.g. support for abstraction, inheritance, strong typing, gen-spec structure,...) provide better reuse support.
2) common software process: there is more reuse in organizations with a common software process that promotes reuse; consequently a defined software process that promotes reuse does affect software reuse level.
3) CASE tools: these tools ease very much designer's tasks for modeling, simulation, code generation, moreover these tools help to design functional and dynamic models that highlight common software process.

Schmidt [9] discusses the lessons learned for problems encountered using design pattern-based reuse in production software environments.
1) the context where the patterns apply and do not apply must be carefully documented
2) implementing patterns efficiently requires careful selection of language features
3) patterns help to alleviate software complexity at several phases in the software life cycle

5 - Conclusion

Reusability is a strong trend in software today because of the stakes of plug-compatible software components, however to build a reuse model is a difficult and subtle exercise.
Many conditions must be satisfied, an important one is to work with an expert of the domain.
For implementing, CASE tools can help developer's tasks.

---

6 - References

[1] Garlan D. Research directions in software architecture, *A.C.M. Computing Surveys*, June 1995, vol 27, n°2, 257 - 261

[2] Huet B. Hospital Information Systems modeling and simulation: the right method for the right objective, in eds: Greenes, Peterson, Protti, *MEDINFO 95*, Amsterdam, North - Holland, 1995, 506 - 510

[3] Pree W. *Design patterns for object-oriented software development*, New York, Addison Wesley, 1994

[4] Huet B. A reuse model for hospital clinical laboratories, *Meth. Inf. Med.*, 1997, (submitted)

[5] Rumbaugh J.E., Blaha M., Premerlani W., Eddy F., Lorensen W. *Object-Oriented Modeling and Design*, Englewood-Cliffs NJ, Prentice-Hall, 1991

[6] Coad P., Yourdon E. *Object-Oriented Analysis*, Englewood-Cliffs NJ, Prentice-Hall, 1991

[7] Beck. K. and Johnson R. Patterns generate architectures, in eds: M. Takoro, R. Pareschi, *ECOOP'94*, Berlin, Springer-Verlag, 139 - 149 .

[8] Frakes W.B. and Fox J. Sixteen questions about software reuse, *A.C.M. Com.*, June 1995, vol. 38, n°6, 75 - 88

[9] Schmidt D.C. Using design patterns to develop reusable object-oriented communication software, *A.C.M. Com.*, October 1995, vol. 38, n°10, 65 - 74

Adress for correspondence

Pr. HUET Bernard
Université PARIS XIII
Faculté de Médecine
74 Av. Cachin
93012 BOBIGNY Cedex
FRANCE

tél: 01.48.95.51.97
fax: 01.48.38.77.77
e-mail: huet@laforia.ibp.fr