Unified Modeling Language and Design of a case-based retrieval system in medical imaging

Christel LeBozec, M.D., Marie-Christine Jaulent PhD., Eric Zapletal and Patrice Degoulet M.D., Ph.D.
Medical Informatics Department, Broussais Hospital, 75014 Paris, France
lebozec@hbroussais.fr

One goal of artificial intelligence research into case-based reasoning (CBR) systems is to develop approaches for designing useful and practical interactive case-based environments. Explaining each step of the design of the case-base and of the retrieval process is critical for the application of case-based systems to the real world. We describe herein our approach to the design of IDEM - Images and Diagnosis from Examples in Medicine - a medical image case-based retrieval system for pathologists. Our approach is based on the expressiveness of an object-oriented modeling language standard: the Unified Modeling Language (UML). We created a set of diagrams in UML notation illustrating the steps of the CBR methodology we used. The key aspect of this approach was selecting the relevant objects of the system according to user requirements and making visualization of cases and of the components of the case retrieval process.

Further evaluation of the expressiveness of the design document is required but UML seems to be a promising formalism, improving the communication between the developers and users.

INTRODUCTION

Developing sound knowledge based systems in radiology and pathology is difficult because these domains can hardly be modeled by logical formalization. Case-Based Reasoning (CBR) can be used in such weak-theory domains. CBR makes it possible to reuse prototype images and diagnosis of previously resolved cases to explain new situations [1]. CBR systems have been validated for intelligent reference image selection in MACRAD [2], IMAGECREEK [3], DIAGMED [4] and for retrieval of radiology reports in ISIS [5].

The status of methodology employed for designing real-world CBR systems is not achieved. There have been several attempts to define appropriate CBR systems development methodologies, mostly associated with specific projects [6], but sometimes also more formally [1,7].

A rapid prototyping strategy is currently considered to be the most effective way to design useful and practical interactive case-based environments [8,9]. A major drawback with this incremental development is that the user is only involved in the validation of prototypes. It is difficult to validate the deep components of the system based purely on the performance of a prototype. It is also vital that the system developers have a good understanding of the medical knowledge involved in the cases and in the retrieval process.

An alternative approach to CBR system development is to involve both users and developers in the design of the first prototype by producing a comprehensive document. The Unified Modeling Language (UML), a unified notation based on various Object-Oriented (OO) analysis and design methodologies, provides a representation of the components of a system that balances the expressiveness of the formalism and its computational tractability [10].

The aim of this work was to use UML, as a suitable formalism to improve the understanding by both users and developers of each step of the case-base and retrieval process design.

We have used UML to illustrate the design of IDEM "Images and Diagnosis from Examples in Medicine", a CBR application for "intelligent" content-based image retrieval in pathology. Based on the description of a new case, IDEM provides the images and diagnosis for the most similar cases in the database [11].

We first present the main CBR system development methodologies and the concepts of UML notation. Then, some examples of UML diagrams used to specify the relevant objects of the system are given as results. Our results show that the legibility of UML affords participants to visualize and evaluate the model before the implementation of the first prototype. Furthermore, the modular nature of the OO approach makes it possible to edit and modify selected implemented objects of the case-base or retrieval process during the development.

METHODS

A methodology provides guidelines for the successful development of a software system. It is usually presented as a series of steps, associated with specific techniques and notations. As the complexity of CBR systems increases in scope and scale, it becomes challenging to use such methodology to design the global architecture of the system as well as each specific component.

Bergman et al., as part of the APPLICUS project, designed a sample project plan for the development of CBR applications [8]. They use MILOS, a software process modeling system in which the model is defined in terms of processes, methods, products, goals and resources. Several types of processes can be identified (technical, managerial and organizational processes). The processes involved in this CBR application include for example*: defining the scope of the CBR project, background knowledge modeling, case acquisition.* For each process, input, output, resource consumption and methods are specified. A set of product types (e.g. descriptive models, case-bases, CBR prototypes, technical environment, documentation...) is provided. The methodology used in the APPLICUS project provides a global model for the development and deployment of a CBR system.

Other studies [1,12] have produced similar generic methodologies for developing CBR systems. The generic subprocesses and products flow involved in the design of a CBR system are depicted in Figure1.

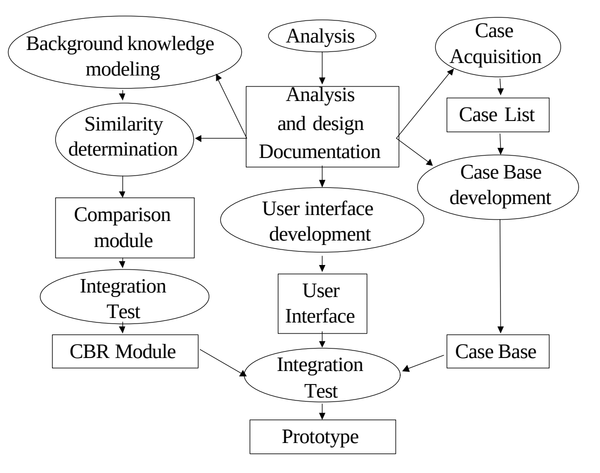

Figure 1- CBR system development processes

We used the concepts of OO software development within this framework. This approach involves modeling objects from the real world and then using the model to build a system application based on those objects. OO modeling and design lead to a better understanding of requirements, simpler designs, and more maintainable systems.

UML is a language for specifying, constructing, visualizing, and documenting the objects of a software system readable by humans and machines. It combines the concepts of the main available OO methodologies [13-15]. The authors of UML didn't produce a standard process but they did promote a development process that is use-case driven, architecture centric, iterative and incremental. OO methodology formalized by UML can be used to define the objects involved in the various processes using a series of diagrams specified by user requirements. User requirements are initially expressed in terms of **use cases**. Specification of a use case defines the possible sequences of interactions between participants and the system.

A specific pattern of interactions is shown on **interaction diagrams**. There is at least one interaction diagram for each significantly different kind of use case instance. A **summary table** is then proposed from all the interaction diagrams that produce the effects of the use cases.

There are two types of interaction diagram: **sequence diagrams** and **collaboration diagrams**.

A **sequence diagram** shows the explicit chronological sequence of messages between participants and the system. Sequence diagrams are suitable for real time specification and for complex scenarios.

A **collaboration diagram** highlights the set of objects that implement the use case. It also specifies the way they are linked together by the sequence of message flows. Collaboration diagrams are more useful than sequence diagrams for understanding all the effects on a given object and for procedural design. Each collaboration diagram provides a preliminary class model. The objects emerging from the various collaboration diagrams of the use cases are compiled into class diagrams for the **object model**.

RESULTS

This method was applied to our specific case-based retrieval system project to produce an analysis and design document. In IDEM, the medical image retrieval process is based on a comparison between meaningful image features. The main use case is "Case comparison" and is involved in several other use cases occurring in daily practice as "Standardization", "Quality assurance", "Diagnosis support", "Education"... In this section, we illustrate the expressiveness of the UML formalism by a series of diagrams used to specify the relevant objects. The set of objects has been implemented in an object database (O2 system)[16].

Object selection by UML diagrams.

The participants involved in the use case diagrams of IDEM are users (students, pathologists and experts) and people involved in the realization and maintenance of the system (experts and developers) (Figure 2).

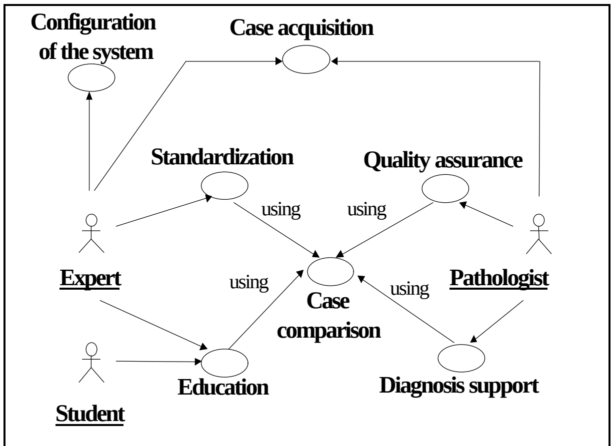

Figure 2 - Use cases diagram of the image retrieval system

A set of scenarios is attached to each use case (Table 1). Using the summary table the scenarios are proposed by the experts and validated by the other participants.

Table 1. The main use cases and the corresponding scenarios

Use cases	Scenarios
Configuration of the system	Definition/Modification of area descriptors Definition/Modification of descriptor values
Case acquisition	Definition/Modification of case description Definition/Modification of case indexing
Case comparison	Description of a new case Retrieval of similar cases Similarity justification Comparison of descriptions for a single case
Standardization	Comparison/Standardization of the descriptions for a single case Comparison/Standardization of the descriptions of cases from the same diagnostic class Comparison/Standardization of the indexing of the cases of a single diagnostic class
Quality assurance	Comparison/Standardization of the descriptions for a single case
Diagnosis support	Retrieval of similar cases
Education	Comparison/Standardization of the descriptions for a single case

Scenarios corresponding to the two main processes have been identified (Figure 1): *constitution of the case base* and *retrieval process determination*.
The key object that implements the case base constitution is “case”. A case is a set of empirical data, gathered by an experienced pathologist while solving a situation. A case is a collection of macroscopic areas, each associated with a collection of histologic areas. A histologic area may contain several other histologic areas as well as cytological descriptions (class “ cell ”). Each type of area is defined by a set of about ten specific features. For instance, (*nuclear_size: large)* is one of the features of histologic areas of the proliferation type.
On the legible object model of the domain (Figure 3) the user can visually identify a missing class or attribute. The main classes can be edited thanks to the O2 interface so that the user adds or modifies components of the cases.

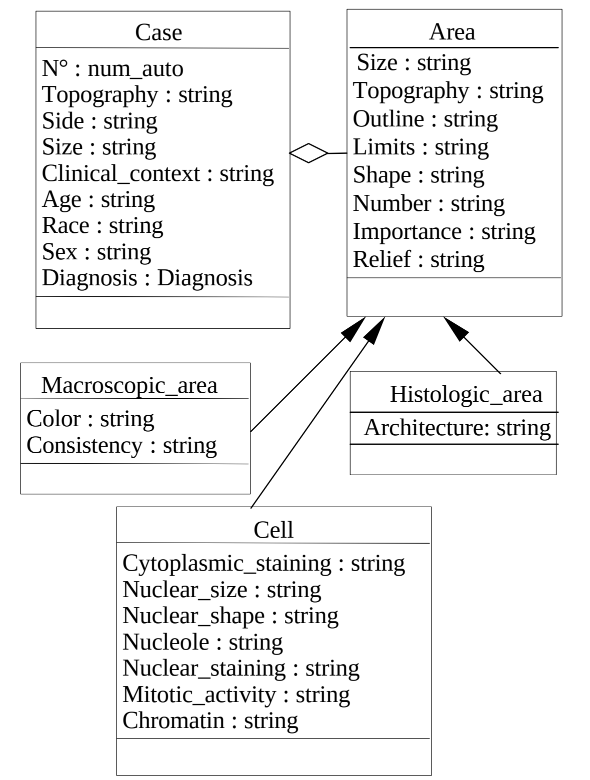

Figure 3 - The object model of a pathological case

Figure 4 illustrates an aspect of the definition of the second component of our CBR system: *the retrieval component*. Retrieval capacity depends on the definition of similarity between cases. Overall similarity is computed as the aggregation of local similarities between morphological features. A similarity table is defined for each feature such that all the possible values of the feature can be compared. For instance, the possible values for the feature “nuclear size” are “small”, “medium”, “large”, very_large”. “Defining/Modifying local similarity of an attribute” is one of the sequence diagrams that explicitly emphasizes a particular aspect of the use case “Configuring the system”.

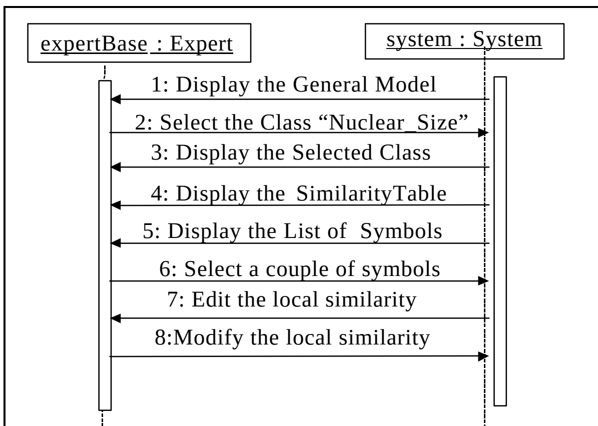

Figure 4 - The sequence diagram of “Defining/Modifying the local similarity of an attribute”

Figure 5 shows the corresponding collaboration diagram. The objects describing the use case, such as Similarity Table, Symbol... are emerging. Some objects do not belong to the domain, but to the human-computer-interaction. In Figure 5, “L_Objects” (for List_Objects) corresponds to a set of instances of the class “Object”.

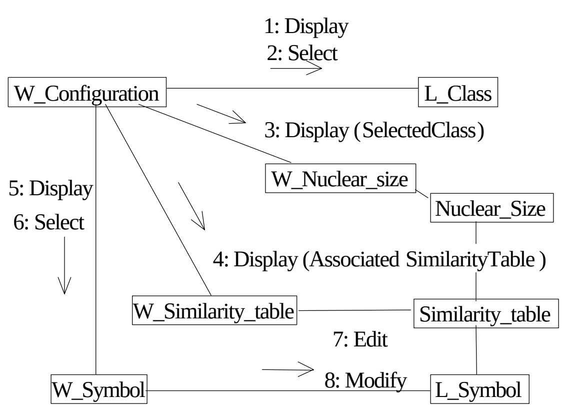

Figure 5 - The collaboration diagram “Defining/Modifying the local similarity of an attribute ”

“W_objects” (for (Window_Objects)) correspond to the interface objects that are successively displayed on the screen (W_Configuration) for a particular scenario (Figures 6 and 7). The visualization of the “W_objects” gives an idea of the fonctionalities of the future prototype.

On Figure 7, the parameters of “W_Similarity_table”, once edited, can be changed by the expert.

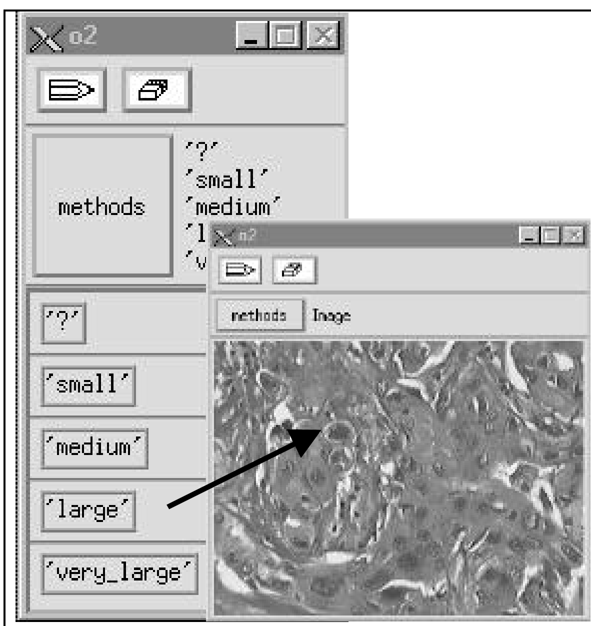

Figure 6 - Copy screen of the interface object “W_Symbol”

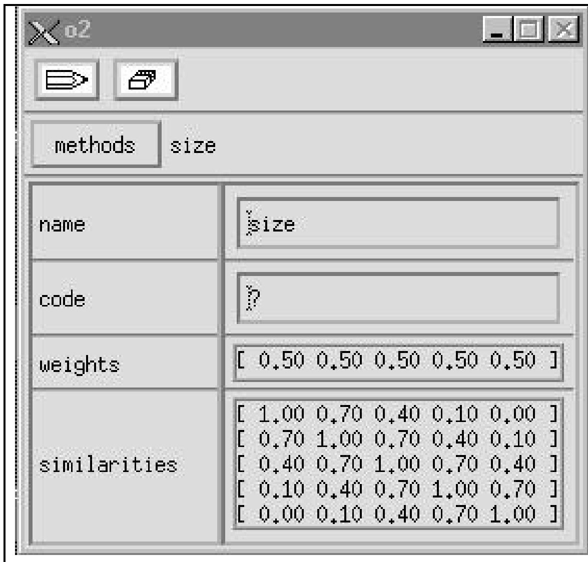

Figure 7 - Screen copy of the interface object “W_Similarity_table”

DISCUSSION AND CONCLUSION

Developing CBR systems is more than just delivering prototypes for the retrieval of cases, based on case comparison. It is important to make sure that all the components of the system and the steps in the process are convenient and comprehensible to experts, users and developers alike.

Before its recent adoption as a standard by the Object Management Group (OMG), UML was already a *de facto* standard embraced by many organizations and vendors due to its expressiveness, conciseness and modular nature. However, these advantages have not been exploited in developing CBR systems. We found that UML could be useful to display objects implementing the use cases of a daily practice and to extend the object model to the level of a programming language.

In this paper we have presented a selection of diagrams illustrating the expressiveness of the design document.

Representing the design with UML has considerable advantages. First, it is possible for experts to validate and modify the case model and to act on the various steps of the retrieval process during the development. Second, the knowledge included in the system is evaluated at the level of the interface objects. The performances of the first resulting prototype are not adversely affected by the use of a great deal of preliminary and simplified information to evaluate the validity of the system.

At last, the legibility of UML affords the developers to appreciate the complexity of the future implementation of the system.

CBR systems in similar application domains such as radiology or pathology necessarily have use cases and scenarios in common. Unifying the design language may be useful for experts and system designers of various application domains to combine their respective experience and to solve their problems in building medical image retrieval systems.

We are currently studying human computer interactions. Before people can decide whether to rely on a CBR system's advice, they must understand the criteria by which the system identifies a particular case as being the most relevant to a given problem [17]. We think it would be valuable to allow users to tune the retrieval process. The users could select their own criteria for relevant cases by changing a predefined set of degrees of similarity.

References

1. Kolodner JL. *Case-basedreasoning*. Morgan Kaufmann, 1993.
2. Macura RT, Macura KJ. MacRad : Radiology image resource with a case-based retrieval system. In: First International Conference on Case-Based Reasoning: Research and Development, Veloso M, Aamodt A (eds), Berlin: Springer Verlag, 1995; 43-54.
3. Grimnes M, Aamodt A. A two layer case-based reasoning architecture for medical image understanding, In: Case-Based Reasoning- Third European Workshop, Smith I, Faltings B (eds), Berlin: Springer Verlag, 1996; 164-178.
4. Maulmy O. Un système d'aide à l'interprétation d'images médicales, Mémoire de diplôme d'ingénieur CNAM, 1997.
5. Kahn CE, Anderson GM, Case-based reasoning and imaging procedure selection. Investigative Radiology, 1994; 29: 643-7.
6. Althoff KD, Wess S., Weis KH, Auriol E, Bergman R, Holtz H et al. An evaluation of the final integrated system. Technical report. Esprit Project INRECA Deliverable D6, 1995.
7. Bergmann R, Wilke W, Schumacher J. Using software process modeling for building a case-based reasoning methodology: Basic approach and case study. In: International Conference on Case-Based Reasoning, Springer Verlag, 1997; 509-519.
8. Bartsch-Spröl B. How to make CBR systems work in practice. In: 4th German Workshop on Case-Based Reasoning: System development and evaluation; Burkhard H.D, Lenz M. (eds), Berlin, 1996; 36-42.
9. Boehm BW. A spiral Model of Software Development and Enhancement, Computer, 1988; 21(5): 61-72.
10. The Rational Rose documentation home page: http://www.rational.com/uml/documentation.html
11. Jaulent MC, Le Bozec C, Zapletal E, Degoulet P. Case-based diagnosis in histopathology of breast tumours. Accepted to MEDINFO 1998.
12. Goel AK, Murdock JW. Meta-Cases: Explaining Case-Based Reasoning. In: Case-Based Reasoning- Third European Workshop, Smith I, Faltings B (eds), Berlin: Springer Verlag, 1996; 150-163.
13. Booch G. *Object-oriented analysis and design with applications*, 2nd edition, Addison-Wesley, 1994.
14. Jacobson I, Ericson M, Jacobson A. *The Object advantage*. Business Process reengineering with Object technology, Addison-Wesley, 1994.
15. Rumbaugh J, Blaha M, Premerlani W, Lorensen W. Object-oriented modeling and design. Prentice Hall, 1991.
16. The O2 technology home page. http://www.O2tech.fr/index.html.
17. Ashley KD, Aleven V. A logical representation for relevance criteria. In: Case-Based Reasoning- First European Workshop, Wess S, Althoff KD, Richter MM (eds), Berlin: Springer Verlag, 1993; 338-52.