

RE-USING EXPERIENCE IN INFORMATION SYSTEMS DEVELOPMENT

Paulo Tomé

Department of Computing, Polytechnic of Viseu, Viseu, Portugal

Ernesto Costa

Department of Computing, University of Coimbra, Coimbra, Portugal

Luís Amaral

Department of Information Systems, University of Minho, Guimarães, Portugal

Keywords: Information Systems Development, Case-Based Reasoning, Model development.

Abstract: Information Systems Development (ISD) is an important organization activity that generally involves the development of models. This paper describes a framework, supported by Case-Based-Reasoning (CBR) method, that enables the use of experience in model development in the context of ISD process.

1 INTRODUCTION

In the ISD process IT professionals develop models, like data and functional models, that express their conceptualization of some organizational aspects. The quality and development time of these models depends on the IT professionals experience. For example, several authors (Batra and Davis, 1992; Chaiyasut and Shanks, 1994; Venable, 1996) consider that expert data modellers perform better than novice data modellers. Besides that, in classical ISD domain bibliography (Kendall and Kendall, 1992; Downs et al., 1992) practical examples are used to illustrate how techniques/modelling tools can be applied.

The use of domain-specific knowledge within a KB-Case tool, and the ability to reason with and make use of this knowledge during a design session improves the efficiency of a tool (Lloyd-Williams, 1994b). This kind of knowledge is used by some tools, for example (Touzovich, 1990; Lloyd-Williams, 1994a), although these tools are oriented for specific types of models and based on rules. The CBR method has better performance in design tasks than rule-based systems (Watson, 1997).

This paper shows a framework, based on the CBR method, that enables the use of acquired experience to develop models in the ISD. The proposed frame-

work is not oriented for a specific software tool, nor for a specific modelling language and not even to a specific model type. In section 2 we briefly review the main ISD concepts needed to understand our contribution that is described in section 4. In section 3 we review some CBR concepts. Finally, in section 4 we show results of the ISMT (a software tool developed to support the framework) application to a set of data models.

2 INFORMATION SYSTEMS DEVELOPMENT

ISD is the fundamental process performed when engaging IT to achieve a specific purpose in a specific context (Fitzgerald et al., 2002). According to Fitzgerald et al. (Fitzgerald et al., 2002) ISD involves much more than simply the deployment of technology. The ISD generally involves several types of process. There is not a generally accepted process model. However the activities such as planning, analyzing, design and implementation are part of the ISD process. Each of these activities has a specific purpose and is generally implemented by making use of methods, techniques, modelling tools and software tools.

It must be noted that concepts like method, tech-

nique and modelling tool are not used with the same meaning in the ISD bibliography. In this paper, we consider the following:

- methods - define what must be done;
- techniques - define how it will be done;
- modelling tools - are the means used to implement techniques.

But despite this problem, it could be said that in the last few decades several authors have proposed methods, techniques and modelling tools that contribute to a better ISD process. Although some authors criticize the use of methods (Baskerville et al., 1992; Wastell, 1996), it is a fact that IT professionals make use of them. These methods and tools can be proprietary methods or created/adapted by the IT professional.

To support the application of techniques and modelling tools, some have authors proposed software tools, generally called CASE tools, which aim to improve the ISD processes. The first CASE tools were mainly design tools. In the nineties the first knowledge-based CASE (KB-CASE) tools were proposed. Generally, these tools are oriented for specific ISD processes, for example conceptual database modelling (Welzer et al., 1998) and are based on rules.

In CASE tools two types of modelling tools could be used: graphical and textual. Textual tools express models through text descriptions. While graphical tools define models through graphics constructors. There are a lot of graphical and textual tools described in ISD bibliography. For example, for defining the data aspect, there is IDEF1X (FIPS, 1993), Peter Chen notation (Chen, 1976) and UML class diagram (UML, 2006) graphical tools. The the BSP (IBM, 1984) method uses several textual tools.

3 THE CBR METHOD

CBR is a method (Watson, 1999) that tries to solve new problems based on solutions to similar previous ones (Kolodner, 1993; Aamodt and Plaza, 1994). CBR is based on two crucial aspects: the *cases* and the process model.

The *case* is formed by the *problem* and the *solution* (Kolodner, 1993). The *objective* and the *characteristics* of the situation are described by the *problem*. The *solution* consists of the solution itself, the solution evaluation and reasonings. The identification of *cases* types constitutes the major step forward in the development of the CBR system. The set of *cases* of the CBR system is called *case memory*. An important issue related to *cases* is the indexing. The index is

a label associated to the case that will allow it to be remembered.

The process model, called the CBR cycle, begins with the problem description and ends with the solution. The CBR cycle has two principal models: 4Rs proposed by Aamodt & Plaza (Aamodt and Plaza, 1994) and the one proposed by Kolodner (Kolodner, 1993). The CBR cycles generally involve the following activities:

- case search to find similar cases;
- similarity evaluation to measure the level of similarity between the problem that must be resolved and the stored ones;
- adaptation to adjust one or several solutions to the current problem;
- case retain to store the new resolved problem.

The case search is based on the problem description. The similarity evaluation is based on similarity functions (Althoff et al., 1995). Consequently, the new solution is built by adapting old solutions to the needs of the current problem. The last task of the CBR cycle is the inclusion of the *case* in the cases memory. Given the fact that a new *case* is added to the system, it could be said the CBR systems have the ability to learn.

It is important to mention, that there are a lot of domains where the CBR methodology has been used (Kolodner, 1993; Watson, 1996; Mntaras and Plaza, 1997). The CBR application areas consist of, for example, software development, architectural design, meal planning and legal reasoning systems.

The problem presented in this paper could be classified as belonging to the design class of the classification schema proposed by Althoff (Althoff et al., 1995). The development of models is a design task because the model conception is carried out without any guidelines. There are several CBR systems that share this property. These are mainly found in the software development environments where it is possible to reuse software code. The Rebuilder project (Rebuilder, 2006) is an example of this. This project aims to use the CBR methodology in the development of UML diagrams (Gomes et al., 2003). The Experience Factory (Althoff et al., 1999) proposes a structure and a software application that aims to reuse experience in the context of software development processes. Krampe and Lusti (Krampe and Lusti, 1997) applied CBR in the IS design. The emphasis of this work was on the use of design specifications. Their focus is also on software development process.

Regarding these works, it is important to say that ISMT is not concerned with the software development

process (i.e code writing). It is meant to help the development of data models. However, the use of UML diagrams could be an aspect that it has in common with the Rebuilder project. We may consider ISMT as a tool that contributes to a good Knowledge Management (KM). The KM leads to rational allocation of organisational knowledge assets (Althoff and Weber, 2005), and this tool allows us to maintain a "experience base" platform that facilitate ISD projects.

4 THE USE OF EXPERIENCE IN ISD

The use of experience in ISD increase the quality of IS and reduces the costs of development. This section shows a framework that will enable the use of experience in model developments.

The proposed framework is formed by four main modules: a *Model repository*, a *Conversation module*, a *Knowledge management module* and an *Interface module*. The *Model repository* stores models. The *Conversion module* deals with the conversion between notations. The *Knowledge management module* is responsible for the management of the information about developed models. The *Interface module* implements the interface mechanisms with the software modelling tools.

The *Model Repository* is implemented by the *Case memory* of the CBR system. As previously mentioned, the *Case memory* stores *cases* and general knowledge domain used by the CBR system. Before describing the knowledge domain, it is important to state that it was our intention to implement a system independent of application domain and also independent of the type of modelling tool. Because of this, the following description is not oriented to a specific modelling tool.

One part of the knowledge domain is the system vocabulary (Richter, 1995). The observation of the IT professionals modelling activity, leads us to conclude that we should consider two kinds of cases: *models* and *constructors*. The case *model* is naturally implemented because the IT professionals major goal is the development of models. But as it is not always possible to get an entire model, it is useful to extract individual constructors.

It was also stated in section 2 that IT professionals develop textual or graphic models. As we considered it important to build an independent system tool, three types of constructors were implemented: *nodes*, *arcs* and *attributes*. It could be considered that textual models are only composed of *nodes*. For each constructor type a case is defined.

To define a case we followed Kolodner's (Kolodner, 1993) suggestion, where a case consists of: *problem*, *solution*. For the solution part we considered the XML code that describes the constructor. The problem characteristics were derived from the formulation of graphic and textual tools through attribute grammar (Wilhelm and Maurer, 1996). It is important to note that attribute grammars were used because some of the modelling tools enable the capture of semantic aspects. Besides that, it also enables the description of the structure of the aspect under analysis. After this step, the system vocabulary is defined as shown in tables 1 and 2.

The set of characteristics listed above, are synthesized and inherited attributes of the formalized modelling tools. For example, *Number of nodes* and *Node keywords* are synthesized attributes. While the *Modelling tool*, *Type of IS* and *Modelled aspect* node characteristics are inherited attributes. It is important to say that *Node properties*, *Attribute properties* and *Arc properties* intend to store the specific information of each constructor.

ISMT was developed under some of the baselines generally accepted to CBR tools, namely the 4Rs cycle of Aamodt and Plaza (Aamodt and Plaza, 1994) and some of aspects mentioned by Mantaras et al (Mantaras et al., 2006) were implemented. Besides that, it was our aim to develop a tool that could be case type independent. As shown in tables 1 and 2, the ISMT tool supports several objectives.

The ISMT structure is illustrated in figure 1. The ISMT has two main components: the client and the server. The user uses a browser and a software modelling tool (like Visio, PowerDesigner or ER-win). The communication between client and server is done through the http protocol. The ISMT was implemented with ASPX (Microsoft, 2002) and Oracle technologies.

The ISMT user can do two main types of tasks: modelling tasks or configuring new modelling tools. As already mentioned, the modelling task is the main aim of IT professionals. The configuration of new modelling tools is the creation of knowledge domain. This task will be detailed below.

When the user does the modelling task he intends to build a model of an organizational aspect. The development of the model begins with the definition of the IS type, modelling tool and the software modelling tool. After that, the user dialogues with the server to get help in its model development, which could be the entire model or a specific element, i.e, a model constructor.

The server component has seven elements: *Mod-*

¹If is a graphic model.

Table 1: Case Description.

Case type	Description
Model	<ul style="list-style-type: none"> Problem: Objective: Model definition Characteristics: Type of IS: Modelling tool: Modelled aspect: Number of nodes: Nodes keywords: Number of arcs¹: Number of arcs by nodes¹: Solution: Solution: Model XML description
Nodes	<ul style="list-style-type: none"> Problem: Objective: Node definition Characteristics: Type of IS: Modelling tool: Modelled aspect: Node properties: Keywords: Number of attributes: Attributes keywords: Arcs with¹: Solution: Solution: Node XML description

Table 2: Case Description (Cont.).

Case type	Description
Attribute	<ul style="list-style-type: none"> Problem: Objective: Attribute definition Characteristics: Modelling tool: Keywords: Belong to: Attribute properties: Solution: Solution: Attribute XML description
Arc	<ul style="list-style-type: none"> Problem: Objective: Arc definition Characteristics: Type of IS: Modelling tool: Modelled aspect: Keywords: Arc properties: Arc attribute keywords: Links: Solution: Solution: Relationship XML description

elling Manager, XML parser, CBR engine, Modelling tool manager, Case Memory, Tools Library and Knowledge manager.

The *Modelling Manager* is responsible for all communication with the user during the modelling task. This element has two major functionalities: to accept complete models and process user requests (model or constructor). The complete models are transmitted to the *XML parser*. The user requests are communicated to the *CBR engine* and after its response the related http code is generated and passed to the browser.

The *XML parser* treats model files using the parsing rules of the *Tools library*. In the starting up phase complete models are transmitted to the *XML parser* to develop the initial *Case memory*.

The *Case Memory* component stores cases and all knowledge domain. This component is implemented through Oracle technology. Two Oracle resources were used: data tables and stored proce-

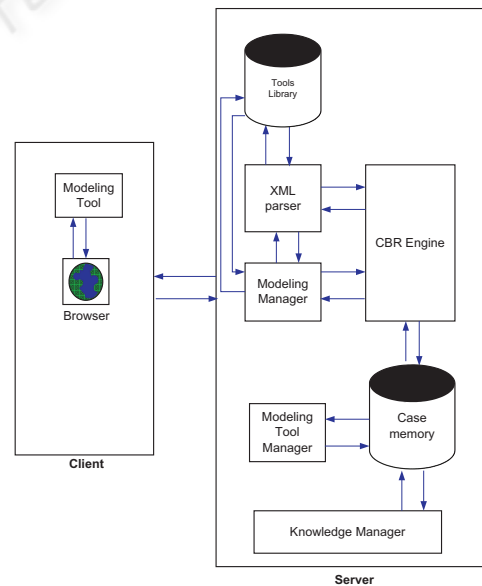


Figure 1: ISMT Structure.

dures/functions. Data tables were used to store the domain knowledge, while stored procedures/functions

were used to implement the system functionalities.

The *Case memory* is structured through frames (Minsky, 1974; Minsky, 1975). Richter's (Richter, 1995) concept of container is implemented in *Case memory* to store all types of knowledge. Every frame has a flag that specifies which type of knowledge it stores.

As mentioned the *Cases* are structured according to the frame mechanism. The data is stored in each frame case according to the approach attribute/value. The composed attributes are split into a new frame.

Besides the cases, the *Case memory* has knowledge related to the metric and adaptation rules. The metric is stored on the frame that describes the domain knowledge according to attribute value approach. The procedures names that implement the adaptation rules are also stored in the mentioned frames. These procedures are implemented using stored procedure of the Oracle engine.

The CBR engine implements the 4Rs cycle (Aamodt and Plaza, 1994). The *Recall* phase begins with a request to the *Modelling manager* which specifies a set of problem characteristics. The recall is implemented based on non-structured *case memory*. The adaptation is done according to a set of pre-defined rules. The case evaluation and final adaptation is done by the system user.

The *Modelling tool manager* is responsible for the management of the domain knowledge. Every time that a new modelling tool is created, it is through this module that the domain knowledge is specified. Using this module it is possible to define: the vocabulary, the weights and the adaptation rules.

Finally, the *Knowledge manager* intends to manage all the stored knowledge related to cases.

5 RESULTS AND CONCLUDING REMARKS

The system was tested with fifteen IS data models. The data models belong to different organizations/domains. Each data model has a different number of entities, attributes and relationship as described in rows 1, 2 and 3 of table 3, respectively.

The ISMT was used with two strategies. In the first strategy the system was launched separately for each data model (no models in memory). In the second strategy the data models are introduced sequentially from M_1 to M_{15} .

As we can see in table 4 when the models are launched sequentially the percentage of adapted cases increases significantly. For instance the *cases* relationship in almost all situations is derived by adapting

Table 3: Situation 1 Results.

	Total of entities	Total of attrib.	Total of relat.	Adapted entities	Adapted attrib.	Adapted relat.
M_1	4	6	4	0	66.7	75
M_2	24	86	32	0	24.4	96.9
M_3	17	113	12	5.9	39.8	91.7
M_4	13	44	10	0	52.3	90
M_5	10	48	9	10	22.9	88.9
M_6	6	60	4	0	25	50
M_7	22	102	12	9.1	34.3	91.7
M_8	16	67	13	6.3	32.8	92.3
M_9	7	22	3	28.6	45.5	66.7
M_{10}	8	74	7	0	28.4	85.7
M_{11}	4	25	3	0	32	66.7
M_{12}	4	22	3	0	22.7	66.7
M_{13}	30	130	44	10	36.2	97.7
M_{14}	4	13	4	0	23.1	75
M_{15}	4	53	3	0	58.5	66.7

cases contained in the case memory. By contrast for entity cases the use of past cases is very low. This can be justified by the heterogeneity of IS domains. Notice also that the order of data models created was not considered an issue.

Table 4: Situation 2 Results.

	Adapted entities	Adapted attrib.	Adapted relat.
M_1	0%	66.7%	75%
M_2	8.3%	26.7%	100%
M_3	5.9%	48.7%	100%
M_4	7.7%	70.5%	100%
M_5	20%	43.8%	100%
M_6	0%	45%	100%
M_7	9.1%	41.2%	100%
M_8	12.5%	50.8%	100%
M_9	28.6%	77.3%	100%
M_{10}	25%	43.2%	100%
M_{11}	0%	60%	100%
M_{12}	0%	68.2%	100%
M_{13}	20%	48.5%	100%
M_{14}	25%	92.3%	100%
M_{15}	25%	83%	100%

We have empirically shown that the inclusion of a CBR methodology providing a memory of past experience greatly improves the task of data modelling within ISD. The use of adapted cases benefits the user once he/she does not need to provide that information manually to the system. Therefore the ISD can be focus on the new elements.

REFERENCES

- Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations and systems approaches. *AI-Communications*, 7(1):39–52.
- Althoff, K. D., Auriol, E., Barletta, R., and Manago, M. (1995). A review of industrial case-based reasoning tools. Technical report, AI Intelligence.
- Althoff, K. D., Nick, M., and Tautz, C. (1999). Cbr-peb: An application implementing reuse concepts of experience factory for the transfer of cbr system know-how. In *7th German Workshop on Case-Based Reasoning*, Wurzburg.
- Althoff, K. D. and Weber, R. O. (2005). Knowledge management in case-based reasoning. *The Knowledge Engineering Review*, 20:305–310.
- Baskerville, R., Travis, J., and Truex, D. (1992). Systems without method: The impact of new technologies on information systems development projects. *The Impact of Computer Supported Technologies on Information Systems Development*, pages 241–270.
- Batra, D. and Davis, J. G. (1992). Conceptual data modelling in database design: Similarities and differences between expert and novice designers. *International Journal of Man-Machine Studies*, 37:83–101.
- Chaiyasut, P. and Shanks, G. (1994). Conceptual data modelling process: A study of novice and expert data modellers. In Halpin, T. and Meersman, R., editors, *First International Conference on Object-Role Modelling*, pages 310–323, Brisbane - Queensland.
- Chen, P. P.-S. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36.
- Downs, E., Clare, P., and Coe, I. (1992). *Structured Systems Analysis and Design Method Application and Context*. Prentice Hall, 2nd edition.
- FIPS (1993). *Integration Definition for Information Modeling (IDEFIX)*. Federal Information Processing Standards Publications.
- Fitzgerald, B., Russo, N., and Stolterman, E. (2002). *Information Systems Development: Methods in Action*. McGraw-Hill.
- Gomes, P., Pereira, F. C., Paiva, P., Seco, N., Carreiro, P., Ferreira, J. L., and Bento, C. (2003). Case-based reuse of uml diagrams. In *Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'03)*.
- IBM (1984). *Business Systems Planning: Information Systems Planning Guide*. IBM Cooperation.
- Kendall, K. E. and Kendall, J. E. (1992). *Systems Analysis And Design*. Prentice-Hall.
- Kolodner, J. (1993). *Case-Based Reasoning*. Morgan Kaufmann Publishers.
- Krampe, D. and Lusti, M. (1997). Case-based reasoning for information systems design. In *ICCBR-97*.
- Lloyd-Williams, M. (1994a). Expert system support for object-oriented database design. *International Journal of Applied Expert Systems*, 1(3).
- Lloyd-Williams, M. (1994b). Knowledge-based case tools: Improving performance using domain-specific knowledge. *Software Engineering Journal*, July 1994:167–173.
- Mantaras, R. L., Mcsherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M. L., Cox, M. T., Forbus, K., Keane, M., Aamodt, A., and Watson, I. (2006). Retrieval, reuse, revision and retention in case-based reasoning. *The Knowledge Engineering Review*, 20(3):215–240.
- Microsoft (2002). www.asp.net.
- Minsky, M. (1974). A framework for representing knowledge. Technical report, Massachusetts Institute of Technology.
- Minsky, M. (1975). A framework for representing knowledge. In Winston, P., editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill. ISBN 0070710481.
- Mntaras, R. L. and Plaza, E. (1997). Case-based reasoning: An overview. *AI Communication*, 10(1):21–29.
- Rebuilder (2006). rebuilder.uc.pt.
- Richter, M. M. (1995). The knowledge contained in similarity measures. *ICCBR 95*.
- Tauzovich, B. (1990). An expert system for conceptual data modelling. In *8th International Conference on the Entity-Relationship Approach*, Toronto, Canada.
- UML (2006). www.uml.org.
- Venable, J. R. (1996). Teaching novice conceptual data modellers to become experts. In *Software Engineering Education and Practice*, Dunedin. IEEE Computer Society Press.
- Wastell, D. G. (1996). The fetish of technique: methodology as a social defense. *Information Systems Journal*, 6:25–40.
- Watson, I. (1996). Case-based reasoning tools: an review. In *2nd UK Workshop on Case-Based Reasoning*, pages 71–78, University of Salford. AI-CBR/SGES Publications.
- Watson, I. (1997). *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. Morgan Kaufmann Publishers.
- Watson, I. (1999). Cbr is a methodology not a technology. *Knowledge Based Systems Journal*, 12(5-6).
- Welzer, T., Stiglic, B., Druzovec, M., and Takac, I. (1998). Database reusability in intelligent medical systems. In *Intelligent Systems for Humans in a Cyberworld: SM'98*, volume 4, pages 4075–4079. IEEE.
- Wilhelm, R. and Maurer, D. (1996). *Compiler Design*. Addison-Wesley.