Improving Data Modelling Through the use of Case-Based-Reasoning

Paulo Tomé[1], Ernesto Costa[2], and Luís Amaral[3]

[1] Instituto Politécnico de Viseu, Escola Superior de Tecnologia de Viseu,
Departamento de Informática, Campus Politécnico, 3504-510 Viseu, Portugal
ptome@di.estv.ipv.pt,

[2] Universidade de Coimbra, Departamento de Engenharia Informática, Polo II -
Pinhal de Marrocos, 3030-290 Coimbra, Portugal
ernesto@dei.uc.pt,

[3] Universidade do Minho, Departamento de Sistemas de Informação, Campus de
Azurém, 4800 Guimarães, Portugal
amaral@dsi.uminho.pt

Abstract. Experience plays an important role in Information Systems data modelling activity. This role is justified by the fact that determining the correct and consistent information requirements is a difficult and a challenging task. Currently three types of data modelling techniques are widely used: entity-attribute-relationship, object-relationship and object-oriented. There is not a consensus about which one is the best. This article proposes a framework, supported by a software tool, that uses Case-Based-Reasoning (CBR) methodology to represent and use experience in the data modelling task. The proposed framework does not depend on the data modelling technique nor on the modelling tool.

Key words: Data Modelling; Information Systems Development; Modelling process; Case-Base Reasoning

1 Introduction

It is commonly accepted that data modelling plays an important role in Information Systems Development (ISD). The adequacy of an Information System (IS) depends on the information that it can provide. However, the information given by the IS depends on the stored data. The ability to store data is determined by the data modelling activity. To support this there are several techniques that make use of modelling tools.

Besides the techniques and tools used in ISD, an important issue is the experience of the IS professionals [1]. It is clear that when we refer to the IS Professionals we mean the people and not the techniques nor the tools. It should be emphasized that the skills of an IS professional were acquired through experiencing with case studies [2–5].

The development of a data model can be classified as a designing task. This kind of task can not be formalized by means of a set of description rules that embody how the designer responds to the client needs. These kinds of domains are classified in Information Technology (IT) terminology as ill defined domains. CBR is commonly used in uncompleted described domains. CBR is a methodology developed by the Artificial Intelligence community that uses the experience of past resolved situations in new situations.

This paper proposes a framework based on CBR for the data modelling task that uses experience. This framework could be applied in conjunction with techniques that use graphical data modelling tools. Additionally a software tool called ISMT (Information Systems Modelling Tool) was developed to support the data modelling activity.

2 Data Modelling

In the IS domain there are a lot of ISD methods, techniques and modelling tools [6]. In this paper we consider methods as something that defines what must be done, technique to refer to the way in which tasks are performed and a modelling tool as the instrument to carry out the tasks. The ISD methods generally involve data modelling [3]. Lucas considers that *one of the major design tasks in building an information system is determining the contents and structure of a database.* The data modelling task can be achieved using several modelling techniques.

There are three main types of techniques: entity-relationship, object-relationship and object-oriented. The entity-relationship (ER) technique, proposed by Chen [7] is an extension of Codd's relational model [8], which considers a data model as a set of entities and relationships. The object-relationship technique implements some of the entity-relationship and object-oriented principles, so it can be considered an extension of entity-relationship technique. The Object-Oriented (OO) technique in data modelling is the application of the OO paradigm to this particular field.

For each of the previous afore types of techniques, it is possible to say that several modelling tools exist, most of which are graphical. For example, the ER technique is supported by IDEF1X [9], by Case*Method Entity Modelling [10] and by Chen Entity-Relationship notation [7].

It is also important to mention that, each technique, besides enabling the data model structure description, permits the expression of its semantic aspects. It is important to notice that the semantic aspects were one of the shortcomings of the first proposed data modelling techniques [11].

Another common feature of data modelling techniques/tools is the possibility of expressing data models according to several levels of detail. Although there is not a common taxonomy for these levels, generally a technique/modelling tool can be applied from a low level of detail to a level where the data model is close of to the Data Base Management System representation. For example, with IDEF1X the low level detail is called ER Level while the opposite is called the fully-attributed level.

3 The CBR Methodology

CBR is a methodology [12] that tries to solve new problems based on solutions for similar previous ones [13, 14]. CBR is based on two crucial aspects: the *cases* and the resolve process model.

The *case* is formed by the *problem* and the *solution* [13]. 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 us remember it.

The resolve process, called CBR Cycle, begins with the problem description and ends with the solution. The CBR cycle has two principal models: 4Rs proposed by Aamod & Plaza [14] and the one proposed by Kolodner [13]. The CBR cycles generally involves the following activities:

- case search to find similar cases;
- similarity evaluation to measure the level of similarity between the problem that needs solving 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 [15]. 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* on 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 [13, 16, 17]. The CBR application areas consist of, for example, software development, architectural design, meal planning and legal reasoning systems.

The tool presented in this paper could be classified as belonging to the design class of the classification schema proposed by Althoff [15]. The data modelling activity 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 [18] is an example of this. This project aims to use the CBR methodology in the development of UML diagrams [19–21]. The Experience Factory [22] proposes a structure and a software application that aims to reuse experience in the context of software development processes. Krampe and Lusti [23] applied CBR in the IS design. The emphasis of this work was on the use of design specifications. Their focus is also putted 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 a common aspect 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 [24]. This tool allows us a to maintain a "experience base" platform that facilitate ISD projects.

4 The use of Experience in Data Modelling

In this section is described the framework that enables the use of experience in data modelling tasks. The framework developed was supported by software application - a CBR system - that will be described.

The first decisions to make in the CBR systems design is the identification of *case* types. Considering the information systems analyst activities, the following *case* types are proposed: *model*, *entity*, *relationship* and *attribute*. The last three types are proposed to cope with tasks where is not possible to re-use a similar model. It must to be noticed that *case* characteristics were derived from modelling tools grammar specification through synthesized and inherited attributes of the Attribute Grammar formalism [25]. It should also be noticed that the developed grammar can be applied to any data modelling tool as it is not oriented to a specific tool.

As previously mentioned, the data modelling techniques and the tools used to apply it, emphasise two aspects. One of the aspects is their graphical notation. The other is the fact that they permit us to capture the semantical aspect of the problem domain.

In a CBR system a case is divided into the *problem*, with *objective* and *characteristics*, and the *solution*. The structure of the four case types are described in table 1. The *characteristics Modelling tool* and *Model type* are control parameters used to identify the tool and the model, respectively. In general, the *characteristics* related to keywords represent the semantic aspect of the data model, while most the of the others represent the structural aspect of the data model. The characteristics *Attribute keywords* and *Relationship keywords* of the model case structure are not included because they are considered low level features. To facilitate the interface with several software modelling tools, the *solution* is the XML description code.

We developed a software tool whose structure is illustrated in figure 1, which supports the use of CBR in data modelling activity. It was our intention to build a flexible tool that permits the use of several case types, data modelling tools and software modelling tools. As illustrated in figure 1 the software tool has two main parts: client and server.

The user can do two main types of tasks: configure new data modelling tools or data modelling. The configuration of new data modelling tools is responsible for the creation of knowledge domain. This task is detailed below. The development of the data models begins with the definition of the modelling tool and the

[4] Characteristic in brackets are optional.

Case Type	Case Problem	Case Solution
Model	Objective: Model definition Characteristics: Entities keywords: Model type: Number of entities: Modelling tool: Number of relationships: Number of relationships by entity:	Model XML description
Entity	Objective: Entity definition Characteristics: Keywords: Entity type: Modelling tool: Number of attributes: Attributes keywords: Relationships with:	Entity XML description
Attribute	Objective: Attribute definition Characteristics: Keywords: Attribute type: Modelling tool: Belong to: Data type: [Length]:	Attribute XML description
Relationship	Objective: Relationship definition Characteristics: Keywords: Relationship type: Modelling tool: Parent cardinality: Child cardinality: Relates: Relationship attributes keywords:	Relationship XML description

Table 1. Case Description[4]

data model level. After that, the user dialogues with the server to get help in its model development. The help could be the entire model or a specific element, i.e, a model constructor.

The server component has seven elements: *Modelling 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: 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 the 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 SQL Anywhere Technology [26]. Two resources were used from the SQL Anywhere: data tables and stored procedures/functions. Data tables were used to store the domain knowledge, while stored procedures/functions were used to implement the system functionalities.

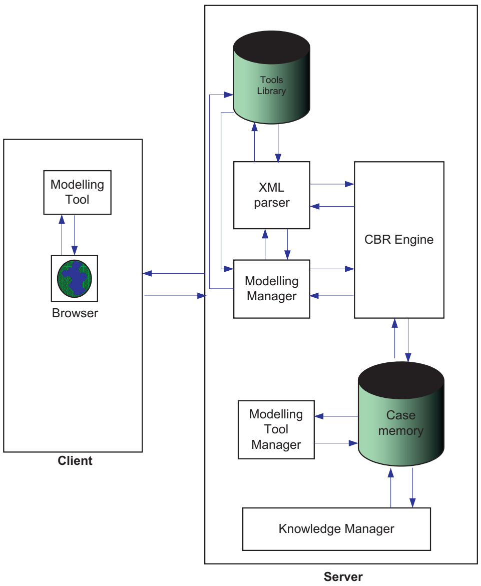

Fig. 1. ISMT Structure

The *Case memory* is structured through frames [27, 28]. Richter's container concept [29] is implemented in textitCase 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 new frames.

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 procedure names that implement the adaptation rules are also stored in the mentioned frames. These procedures are implemented using stored procedures of the SQL Anywhere engine.

The CBR engine implements the 4Rs cycle [14]. The *Recall* phase begins with a request to the *Modelling manager* which specifies a set of problem characteristics. The recall phase 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, the correspondent knowledge domain 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. Every one was designed for a different organisation/domain. Each data model has a different number of entities, attributes and relationship as described in rows 1, 2 and 3 of table 2, respectively.

We used two strategies to test the ISMT. In the first strategy we introduced each data model separately and without any data model in the memory. In the second strategy the data models are introduced sequentially from M_1 to M_{15} and each introduced data model stays in memory.

As can be see in table 2, even considering each data model separately, it is possible to take advantage of previous experience. We can see that the percentage of *Adapted attributes* and *Adapted relationships* is significant. In the latter case type the percentage is higher than 50% because within each data model the range of values is short. In the former case type the percentage of adaptation is higher than 22%. In *Adapted entities* experience was not so relevant because within each data model it is not so common to find similarities.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
Entities	4	24	17	13	10	6	22	16	7	8	4	4	30	4	4
Attributes	6	86	113	44	48	60	102	67	22	74	25	22	130	13	53
Relationships	4	32	12	10	9	4	12	13	3	7	3	3	44	4	3
Adapted entities[2]	0	0	5.9	0	10	0	9.1	6.3	28.6	0	0	0	10	0	0
Adapted attributes[2]	66.7	24.4	39.8	52.3	22.9	25.0	34.3	32.8	45.5	28.4	32.0	22.7	36.2	23.1	58.5
Adapted relationships[2]	75.0	96.9	91.7	90	88.9	50	91.7	92.3	66.7	85.7	66.7	66.7	97.7	75.0	66.7

Table 2. Results of strategy one: each data model separately.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
Entities	4	24	17	13	10	6	22	16	7	8	4	4	30	4	4
Attributes	6	86	113	44	48	60	102	67	22	74	25	22	130	13	53
Relationships	4	32	12	10	9	4	12	13	3	7	3	3	44	4	3
Adapted Entities[2]	0	8.3	5.9	7.7	20	0	9.1	12.5	28.6	25	0	0	20	25	25
Adapted attributes[2]	66.7	26.7	48.7	70.5	43.8	45	41.2	50.8	77.3	43.2	60	68.2	48.5	92.3	83.0
Adapted relationships[2]	75	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 3. Results of strategy two: data models introduced sequentially M_1 to M_{15}

As we can see in table 3, when the models are launched sequentially the percentage of adapted cases increases significantly. For instance the relationships are, in almost all situations, derived by adapting cases contained in the case memory. The same behaviour happens in the *adapted attributes* where percentage increases for the sequentially launched data model strategy. By contrast for the entity cases the use of past cases is lower than the cases relationship. However this can be justified by the heterogeneity of IS domains data models. Notice also that the order of data models created was not considered an issue. In spite of that the percentage of *Adapted entities* increases if the data models are launched sequentially. The zero entries in the table are justified because the applications fields are distinct.

We have 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 since he/she do not need to provide that information manually to the system. Therefore the ISD can be focus on the new elements.

In order to increase the reliability of the system more components have to be added, namely a conversion module. This module will allow the translation of knowledge between two different modelling tools.

References

1. B. Fitzgerald, N. Russo, and E. Stolterman, *Information Systems Development: Methods in Action.* McGraw-Hill, 2002.
2. T. A. Halpin and G. M. Nijssen, *Conceptual Schema and Relational Database Design.* Prentice-Hall, 1989.
3. H. C. Lucas, *The Analysis, Design, and Implementation of Information Systems.* McGraw Hill, 4 ed., 1992.
4. R. Veryard, *Information Modelling: Pratical Guidance.* Prentice Hall, 1992.
5. E. Downs, P. Clare, and I. Coe, *Structured Systems Analysis and Design Method Application and Context.* Prentice Hall, 2 ed., 1992.

[2] Percentage values.

6. N. Jayaratna, *Understanding and Evaluating Methodologies: A Systemic Framework*. McGraw-Hill, 1994.
7. P. P.-S. Chen, "The entity-relationship model - toward a unified view of data," *ACM Transactions on Database Systems*, vol. 1, no. 1, pp. 9–36, 1976.
8. E. F. Codd, "A relational model of data for large shared data banks," *Communications of ACM*, vol. 13, no. June, 1970.
9. FIPS, *Integration Definition for Information Modeling (IDEF1X)*. Federal Information Processing Standards Publications, 1993. ISBN.
10. R. Barker, *Case*Method - Entity Relationship Modelling*. Addison-Wesley, 1995. ISBN 0-201-41696-4.
11. D. Batra, J. A. Hoffer, and R. P. Bostrom, "Comparing representations with relational and eer models," *Communications of ACM*, vol. 33, no. Number 2, 1990.
12. I. Watson, "*CBR* is a methodology not a technology," *Knowledge Based Systems Journal*, vol. 12, no. 5-6, 1999.
13. J. Kolodner, *Case-Based Reasoning*. Morgan Kaufmann Publishers, 1993.
14. A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations and systems approaches," *AI-Communications*, vol. 7, no. 1, pp. 39–52, 1994.
15. K. D. Althoff, E. Auriol, R. Barletta, and M. Manago, "A review of industrial case-based reasoning tools," tech. rep., AI Intelligence, 1995.
16. I. Watson, "Case-based reasoning tools: an review," in *2nd UK Workshop on Case-Based Reasoning*, (University of Salford), pp. 71–78, AI-CBR/SGES Publications, 1996.
17. R. L. Mntaras and E. Plaza, "Case-based reasoning: An overview," *AI Comunication*, vol. 10, no. 1, pp. 21–29, 1997.
18. Rebuilder, "rebuilder.uc.pt," 05-01-2006 2006.
19. P. Gomes, F. C. Pereira, P. Paiva, N. Seco, P. Carreiro, J. L. Ferreira, and C. Bento, "Using wordnet for case-based retrieval of uml models," in *STarting Artificial Intelligence Researchers Symposium (STAIRS'02)*, 2002.
20. P. Gomes, F. C. Pereira, P. Paiva, N. Seco, P. Carreiro, J. L. Ferreira, and C. Bento, "Case-based reuse of uml diagrams," in *Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'03)*, 2003.
21. P. Gomes, F. C. Pereira, P. Paiva, N. Seco, P. Carreiro, J. L. Ferreira, and C. Bento, "Human-machine interaction in a case environment," in *International Joint Conference on Artificial Intelligence IJCAI'03 Workshop: "Mixed-Initiative Intelligent Systems"*, 2003.
22. K. D. Althoff, M. Nick, and C. Tautz, "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), 1999.
23. D. Krampe and M. Lusti, "Case-based reasoning for information systems design," in *ICCBR-97*, 1997.
24. K. D. Althoff and R. O. Weber, "Knowledge management in case-based reasoning," *The Knowledge Engineering Review*, vol. 20, pp. 305–310, 2005.
25. R. Wilhelm and D. Maurer, *Compiler Design*. Addison-Wesley, 1996.
26. Sybase, "www.sybase.com/products/databasemanagement," 1-11-2006 2006.
27. M. Minsky, "A framework for representing knowledge," tech. rep., Massachusetts Institute of Technology, 1974.
28. M. Minsky, "A framework for representing knowledge," in *The Psychology of Computer Vision* (P. Winston, ed.), pp. 211–277, McGraw-Hill, 1975. ISBN 0070710481.
29. M. M. Richter, "The knowledge contained in similarity measures," 1995. Comunicao por convite na ICCBR 95.