Reengineering an Industrial Legacy Software Towards an Object-Oriented Knowledge-Based System [*]

Hakim Lounis
Department of Computer Science
Université du Québec à Montréal, Canada
lounis.hakim@uqam.ca

Kaddour Boukerche
Centre de Recherche Informatique de Montréal (CRIM), Canada
kaddour.boukerche@crim.ca

Houari A. Sahraoui
Department of Computer Science and Operations Research
Université de Montréal, Canada
sahraouh@iro.umontreal.ca

Abstract. *A software product is expected to fulfill some need and meet some acceptance standards that dictate the qualities it must have. This paper presents a reengineering work tending to increase to a significant degree some software qualities relevant in the management of production of a hydroelectric network. An object-oriented knowledge-based architecture is proposed to ensure an intelligent and automatic management of the knowledge in use in the daily decisional process of a major Canadian company.*

1 Introduction

Reengineering is the examination and the modification of an existing system to reconstitute it in a new form and the subsequent implementation of the new form. The first phase of reengineering is some form of reverse engineering so as to abstract and understand the existing system. The second phase is traditional engineering or full restructuring using new specification and knowledge of the old system obtained from reverse engineering. This process is generally motivated by the will of moving old programs and systems to new platforms, as in source code translation, or restructuring programs that were corrupted by repeated maintaining activities. However, the most promising axis of reengineering is certainly moving legacy systems to emerging technologies and paradigms. Indeed, many organizations have been migrating their legacy systems to emerging technologies, e.g., Object-Oriented (OO) technology. Lehman and Belady present this migration as an economical choice through their three laws on the evolution of large systems [1].

OO approaches and languages have become quite popular, partially because of their potential benefits in terms of maintainability, reusability, separation of concerns, information hiding, etc. However, the vast majority of software available today is not OO. The effort to simply rewrite them from scratch using an OO approach would be prohibitive, and significant expertise recorded in the procedural software would be lost. The cost of manual conversion would also be prohibitive. Support coming from tools, documentation, and developers of the legacy software would ease the introduction of OO technology in many organizations. This kind of reengineering process could be especially helpful to integrate existing systems and new ones developed with OO approaches.

On the other hand, Knowledge-Based Systems (KBS) are used in numerous application domains, of which the field of hydroelectricity in which this work fits [2] [3] [4]. KBS are used to reproduce an expert's reasoning and are based on two distinct components: knowledge and reasoning. Separation between these two levels of intervention makes it possible to offer a flexibility of operation that many traditional software approaches are missing. KBS are presently an effective and useful solution to integrate the necessary analyses of hydropower experts and to meet the needs of the hydroelectric industry.

In the balance of this paper, we first present in section 2 the problem description. In section 3, we introduce what we consider as motivations for moving towards an OO knowledge-based architecture. In section 4, we describe the adopted solution and present its main features. Finally in

[*]This work is part of an industrial project between the ALCAN Ltd group and CRIM, a research center. It was supported by a joint grant from ALCAN Ltd and NSERC, operation grant #CRDPJ 228746-99.

section 5, we conclude and present some of the lessons we learned.

2 Problem description

Alcan is one of the two world biggest player in the aluminum industry. With a total surface area of 73 800 km², the Alcan hydropower network under study, constitutes a territory larger than the province of New Brunswick (Canada). The network has, on average, an annual energy capacity of approximately 2000 megawatts; it includes 6 hydroelectric power stations, 28 reserve installations, 43 turbine-alternators groups (TAG), 850 kilometers of energy transport lines, a network of about thirty hydro-meteorological stations, etc.

The objective of planning the operation of such a network can be summarized as the satisfaction of the following requirements:

- Effective use of water
- Account of future hydrological uncertainty
- Satisfaction of energy need
- Respect of safety constraints.

To reach these goals, a decision-making process of water stock management is used that consists of four steps:

1) Weather hydro measurements and gathering of the data;
2) Data analysis;
3) Weather and hydrological forecasting;
4) Planning.

In these planning tasks, information processing systems based on mathematical models tested for this kind of applications, are used for optimization and simulation purposes. This decision-making process is part of the knowledge management (KM) policy of Alcan Ltd. Currently, a lot of companies, often multinationals, face a significant problem of management of their knowledge, their know-how and their competences. They thus should constitute an alive and productive memory for their company, resting on three following main topics: the management of the experts and their expertise, the return of experiment, and the knowledge and information transfer in the company. From a theoretical point of view [5], KM makes it possible a company to manage (i) its specific expertise, which characterize the company capacities in the study, the realization, the sale and the support of its products and its services and (ii) individual and collective know-how, which characterizes the company capacities of action, adaptation and evolution.

However, the knowledge used in the studied decision-making process, is many and varied. The problem is that this knowledge is often hidden in the code of the programs which use it, or consigned in internal documents, or even used implicitly by the experts, as in our case. This situation becomes more problematic, when the Alcan hydrological resources analysts wants to explore new scenarios, while modifying a little one of this knowledge. It has no other choice than to traverse the source code of the implemented programs, in order to make the discounted modifications there. It is, for example, the case of the operation rules of each power station and tank.

3 The Reengineering Motivations

The main disadvantage of the solution used since several years and described above, is the absence of separation between the knowledge level and the inference or reasoning one, in a product used in a knowledge intensive process! An immediate consequence is the restriction in the possibilities of investigation and exploration wanted by the Alcan analysts. Moreover, the adopted "black box" architecture produces a lack of flexibility of the whole decisional process. A consequence of that is the difficulty of maintaining and making evolve such a system.

An essential requirement of the KBS design process is the use of efficient representations of large amounts of knowledge; this ensures the consistency and effective exploitation of the KBS algorithms. The available knowledge can be explicit or implicit. An explicit representation consists of a symbolic expression of human expert knowledge. Rules are an example of that; they allow you to separate the expertise from the application code. This makes the application adaptable and maintainable. Since expert rules are externalized from the application code, they can be changed independently without recompiling the application. An implicit representation is knowledge that is usually hidden in numerical data. It requires further processing of the data before useful information can be extracted from it. For that, Machine-Learning (ML) techniques have been widely used to capture hidden knowledge from stored historical data. In each case, the goal was to determine trends or behaviour patterns that would allow the improvement of KBS procedures. For instance, ML techniques have been used in hydroelectricity to produce rules from a power generation database [6] [7].

On the other hand, adding new functionalities is not the only goal of such a reengineering process. We also want to reach some quality attributes in the reengineered product. Some of these qualities are:

- Evolvability: software is evolvable if it allows changes that enable it to satisfy new requirements. It is a quality attribute close to flexibility. The initial design of the product as well as any succeeding changes must be done with evolvability in mind. The object-oriented and knowledge-based architecture we have chosen intends to foster evolvability, especially by offering the means of updating knowledge.
- Usability: a software system is usable if its human users find it easy to use. In our case, users are Alcan experts and analysts, and they have special needs. For instance, the user-friendliness (i) of the configuration process of the network before each simulation (see use-case <*define hydropower network*> in figure 3), and (ii) of the updating process of the knowledge base (see use-case <*edit rule-set base*> in figure 3), are important factors.
- Reusability: this quality attribute is close to evolvability. It may be applied at different levels of granularity, from whole applications (including pieces of knowledge) to individual routines. For instance, in our context, mathematical models implemented in Fortran reusable components are part of our global architecture.
- Understandability: the activity of software maintenance is dominated by the subactivity of program understanding. Understandability is an internal product quality, and it helps in achieving many of other qualities, such as evolvability. Reaching understandability is a difficult task in complex systems with multiple functionalities. However, separating knowledge from the procedures that use it, and producing a new OO design are undeniable steps towards this objective.

All these qualities are in fact achieved thanks to internal software attributes, which deal largely with structure of the new software architecture.

4 The OO Knowledge-Based Solution

As stated above, this work deals partly with the knowledge management of the Alcan experts, including data, thus allowing the hydropower resources planning or simulation. To help perform the planning tasks, we have developed a KBS called HYPERPIK (Hydro Power Resources Planning based on Inference and Knowledge). Figure 1 summarizes our system architecture. It consists of an inference engine that is coupled with a knowledge base resulting from the problem modeling. The knowledge base contains an explicit knowledge that is the symbolic expression of Alcan experts' know-how. A machine-learning framework exploits a historical database and produces explicit or implicit knowledge, depending on the selected learning mechanism [7]. The produced knowledge is then used in the decision process. In particular, it uses natural contributions flow values predicted from the historical database. These contributions flow values help evaluate the ability of the power system to face various contingencies and to propose appropriate remedial actions.

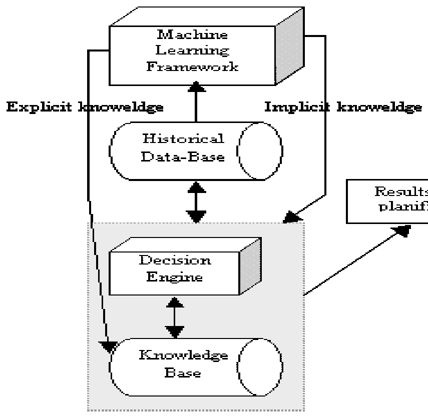

Figure 1. The HYPERPIK architecture

On the other hand, the planning step exploits explicit knowledge. It takes the form of rules. Rule technology is based on the philosophy of providing fast and flexible software components to empower computer applications with "business" or "expert" rules capabilities. The general idea of a rule is that actions on the right-hand side are carried out whenever all the patterns on the left-hand side are successfully matched. A *pattern* is an expression that is capable of designating one or more objects. The objects result from our modelization of the hydropower domain and the following classes diagram is an illustration of them.

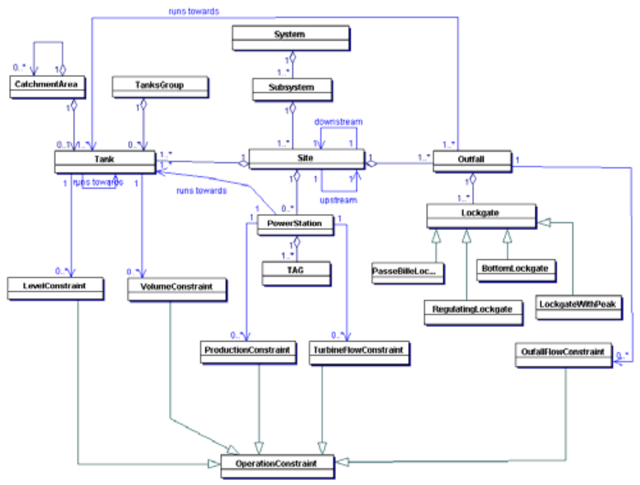

Figure 2. Objects involved in the planning process

The decision or inference engine processes the rules using the objects in a working memory. It implements a RETE algorithm (it is widely recognized as by far the most efficient algorithm for the implementation of production systems) [8] where rules are compiled into a network. Input data to the network consists of changes to working memory. Objects are inserted, removed and modified. The network processes these changes and produces a new set of rules to be fired. This process continues cyclically until there are no further rules to be fired.

The rules have a simple structure, composed of a header, a condition part and an action part. The header part defines the name of the rule, the eventual packet to which the rule is attached, and its priority (if needed). The condition part utilizes the object-oriented structure of Java to carry out pattern matching on class instances, i.e. objects. This pattern matching binds (instantiates) variables to objects and field values. Rule conditions are also used to test field values. This provides a filtering mechanism for objects. When the condition part of a rule is verified, i.e. valid objects have been found, the action part of the rule may be executed. Actions may vary from simple to complex, e.g. printing a message to creating new objects or calling a pre-existing Fortran routine (through a Java method). The rules are written in the Ilog rule language[1] and the following one illustrate their structure:

```
rule short_TermDischargeRiskStJeanLake {
  packet = shortTermRiskSJL;
  when {
    site(name =="LSJ");
    predicted_averageDischarge_InNextdays(currentDate
    + 7) > -100 ;
    precited_averageDischarge_InNextdays(currentDate
    + 7) <= 0); }
  then {
    modify { shortTermRiskDischarge(currentDate) = 50;
    } } };
```

Our reengineering work yields to about 150 rules organized in 16 packets. A packet allows us to group rules with regard to their goal in the whole process. Examples of packets are: short-term risk at St-Jean Lake (see the rule above), overflow risk, Saguenay sub-system production, etc. This reengineering work yields also to a new interaction model between Alcan analysts (final users of the system) and the system. The following use-cases illustrate these new functionalities and particularly the flexible way that the experts from now on have to configure their network or run a simulation session.

[1] Ilog Jrules is a general-purpose expert-system generator that combines rule-based techniques and object-oriented programming (www.ilog.com).

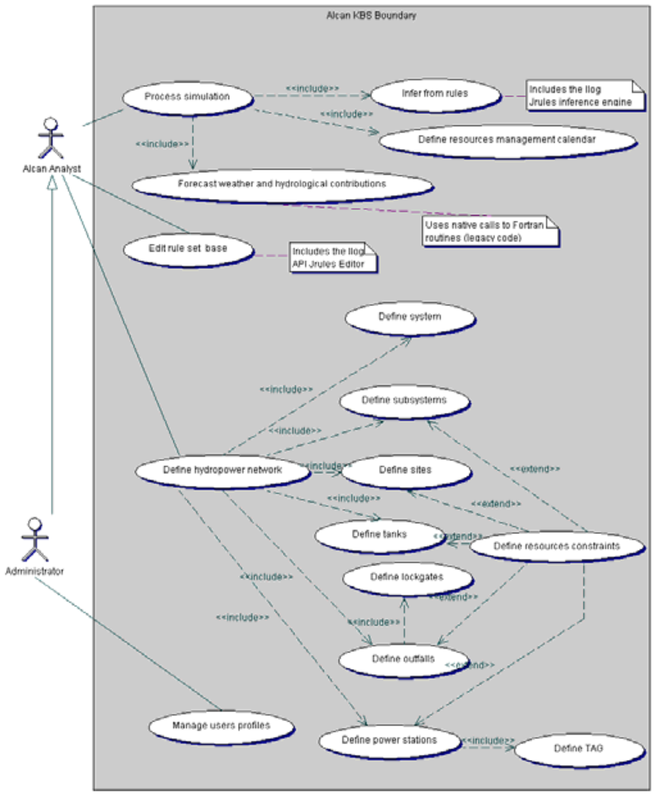

Figure 3. The Reengineered system use-cases

5 Conclusion and Lessons Learned

The reengineered system is currently used within the hydropower resources management group at Alcan. It results from a long collaborative process between authors of this paper and Alcan analysts. The latter were active and decisive actors; they have to maintain their Fortran routines (e.g., short term evaluation functions, water rise rate calculation functions, volume calculation functions, etc.) and we have to build a bridge between these functions and the objects methods we have implemented. The exercise was not so easy; we have to keep a good separation between what we consider as an expert knowledge and the procedures that exploit this knowledge. This critical step of the project was iterative and it requires, even now, many adjustments.

The following points could summarize the strengths of the proposed solution:

- a greater flexibility of the tool during its use within the decisional process, by facilitating the exploration of new power network management scenarios. It is the main need of Alcan analysts;
- better user interfaces allowing better usability during system configuration and simulations;
- a better evolvability of the system thanks to the OO rules-based architecture. It results from our objects/rules modelization and it allows updating easily expert knowledge to explore new planning schemes. The general understandability of the system is also much higher than it was in the previous system;
- a current and future better reusability of different components of the system.

This work is still in progress. We are working now on some extensions of the system. By adding priority factors to the rules, mainly to those dealing with management instructions, we expect to improve the whole performance of the tool. In fact, the tuning of such a system is a long and meticulous work; it is actually one of the main tasks of Alcan analysts. Finally, a next step will be to produce a rules verification module, in order to maintain the knowledge base free of anomalies (redundancy, inconsistency, etc.).

Acknowledgement: The authors want to thank Ms. Louise Rémillard and Janine Dufour from Alcan Ltd for their precious collaboration.

References

[1] M. M. Lehman and L. A. Belady, "Program evolution". *Academic Press, New York, 1985.*

[2] Samarasinghe, S.; McKinnon, A.; Bright, J. Expert System for Flood Management in Lake Manapouri ».. Lincoln Univ. Canterbury, New Zealand. IEEE Comput. Soc. Press, Los Alamitos, CA, USA. First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, 1993, Dunedin, New Zealand.

[3] Chang, Tiao J.; Moore, David. Reservoir Operation by the Use of an Expert System. Ohio Univ, Athens, OH, USA. Proc. of the 1994 ASCE National Conference on Hydraulic Engineering. Buffalo, NY, USA.

[4] Leslie, A.S.; Moyes, A.; McDonald, J.R.; Burt, G.M.; McGowan, J.; Charlesworth, W. CEPE, Intelligent System for the Management of a Hydro-electric Scheme. Strathclyde Univ., Glasgow, UK. 31st Universities Power Engineering Conference, Iraklio, Greece, 1996.

[5] Grundstein M., "La capitalisation des connaissances de l'entreprise, système de production de connaissances", 1995.

[6] Mejia-Lavalla M., Rodriguez-Ortiz G., ,Obtaining expert system rules using data mining tools from a power generation database, Expert systems with applications, 14, 1998, 37-42. 1517.

[7] Lounis H., Boukadoum, M. & Siveton, V., Assessing Hydro Power System Relevant Variables: a Comparison Between a Neural Network and Different Machine Learning approaches, Proc. International Conference on Neuro-Fuzzy Technology (Neuro-Fuzzy2002), Havana (Cuba), January 2002. 45-51.

[8] Forgy, C.L.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem Artificial Intelligence, 19(1982) 17-37.