A Model-Driven Development Approach for Case Bases Engineering

Nikita O. Dorodnykh and Alexander Yu. Yurin
Matrosov Institute for System Dynamics and Control Theory,
Siberian Branch of the Russian Academy of Sciences (ISDCT SB RAS)
Irkutsk, Russia
tualatin32@mail.ru, iskander@icc.ru

***Abstract*—The paper discusses application of a model-driven development approach for engineering knowledge bases of case-based reasoning decision support systems. The conceptual models presented in XML-like formats are used as the initial data. The problem statement, main stages of the proposed approach and a tool are presented. An illustrative example describes an educational task.**

***Keywords*—model-driven development, case-based reasoning, intelligent system, knowledge base, conceptual model**

I. Introduction

The knowledge bases engineering for intelligent systems remains a time-consuming. This process requires the involvement highly qualified specialists in domain areas (e.g., domain experts, analysts, programmers) and also the use of specialized software. In this case, the development and use of software focused on non-programming users and implementing the principles of visual programming and automatic code generation are relevant. There are examples of such software: ExSys, Visual Expert System Designer, and others. In most cases, these systems are focused on a certain formalism of knowledge representation, in particular, logical rules. Software for conceptual modeling is another class of systems that uses visual programming for knowledge bases engineering. Such systems provide the ability to create models in the form of concept maps, mind maps, fuzzy maps, entity-relationship diagrams, tree-like semantic structures and schemes (fault trees, event trees), IDEF or UML models. Most of these systems are universal modeling software and don't take into account features of intelligent system engineering, i.e. formalisms, languages and program platforms, which are used in this area. In this connection, they provide a visual representation of domain-specific knowledge structures, but don't support an adequate knowledge codification (formalization) for knowledge representation languages.

The use of methods and approaches that implement the principles of a Model-Driven Engineering approach (MDE) (also known a Model-Driven Development, MDD) [1], [2] is a compromise in this case. Such compromise solutions [3] provide import (analysis and transformation) of conceptual models, which describing domain knowledge into knowledge base structures, and their specification in specialized editors for further execution. Model transformations [4] are implemented during the import, for example, by using Transformation Model Representation Language (TMRL) [5].

This paper discusses an example of a such compromise solution, in particular, we propose an approach and its implementation on the basis of a Personal Knowledge Base Designer (PKBD) platform [6]. Our proposals provide prototyping knowledge bases for case-based decision support systems. At that, concept maps and formalization from [7] are used as initial data.

II. State-Of-Art

Let's consider main concepts and definitions in the field of case-based reasoning and model-driven development (MDE/MDD) as a background for the research.

A. Case-Based Reasoning

Case-Based Reasoning (CBR) [8], [9] is a methodology for decision making that reuses and adapts (if necessary) of the previously obtained solutions of similar problems. This approach is based on the "by analogy" principle of decision making. The main concept is a case. The case is a structured representation of accumulated experience in the form of data and knowledge, providing its subsequent automated processing using specialized software.

The main case features are the following:

- A case represents special context-related knowledge that allows to use these knowledge at the application level.
- Cases can be represented by various forms: covering different time periods; linking solutions with problem descriptions; results with situations, etc.
- Case captures only the experience that can teach (to be useful), fixed cases can potentially help domain expert (decision-maker) to achieve the goal, facilitate its formulation in the future or warn him/hir about possible failures or unforeseen problems.

A case structures units of experience. At the same time, the used structure is a problem-specific. In general, a case structure includes two main parts:

- An identifying (characterizing) part describes the experience by a way that provides to estimate the possibility of its reuse in a particular situation.
- A learning part describes a lesson (learning knowledge, a solution) as a part of an experience unit, for example, a solution of a problem or its part, a decision proof (conclusion), an alternative or failed decisions.

CBR applied to solving problems in various subject domains, including planning [10], diagnostics [11] and others. There are some examples of CBR tools: CBR-Express, CasePoint, CasePower, Esteem, Expert Advisor, ReMind, CBR/text, ReCall, RATE-CBR, S3-Case, INRECA, and CASUEL.

B. Model-Driven Engineering

Model Driven Engineering (MDE) or Model-Driven Development (MDD) is a software design approach that uses the information models as the major artifacts, which, in turn, can be used for obtaining other models and generating programming codes [2]. This approach enables programmers and non-programmers (depending on the implementation) to create software on the basis of conceptual models.

The main MDD concepts are the following:

- A model is an abstract description of a system (a process) by a formal language. As a rule, models are visualized with the aid of certain graphic notations and serialized (represented) in XML.
- A metamodel is a model of a formal language used to create models (a model of models).
- A four-layer metamodeling architecture is the concept that defines the different layers of abstraction (M0-M3), where the objects of reality are represented at a lowest level (M0), then a level of models (M1), a level of metamodels (M2) and a level of a meta-metamodel (M3).
- A model transformation is the automatic generation of a target model from a source model with the accordance of a set of transformation rules. In this case, each transformation rule describes the correspondence between the elements of a source and a target metamodels.

There are examples of a successful use of MDE for development of database applications (e.g., ECO, for Enterprise Core Objects), agent-oriented monitoring applications [12], decision support systems [13], [14], embedded systems (software components) for the Internet [15], rule-based expert systems [16].

C. Background

Proposals of the current research are based on the previous works. In particular, development of case-based expert systems and knowledge bases for petrochemistry are presented in [11]. Model transformations, as well as a specialized domain-specific declarative language for describing transformations of conceptual models are discussed in [5]. Application of MDD principles for engineering knowledge bases of rule-based expert systems are described in [3], the formalization of MDD for CBR is used from [7].

III. Prototyping Case Bases Using Transformation of Conceptual Models

A. Formalization

Most of decision-making tasks can be described by a set of characteristics, and can be formalized as follows:

$$\begin{array}{l} M^{Task} = \{p_1, ..., p_n\}, p_i \in Prop, \\ Prop = \bigcup_{i=1}^{M} p_i, i = \overline{1, N}, \end{array} \quad (1)$$

where M^{Task} is a task model; p_i is task properties (significant characteristics); $Prop$ is a set of properties.

The task model from a CBR point of view is defined as follows:

$$M^{TaskCBR} : Problem^{CBR} \to Decision^{CBR}, \quad (2)$$

where $M^{TaskCBR}$ is a task model in terms of CBR; $Problem^{CBR}$ is a task (problem) description; $Decision^{CBR}$ is a problem decision, while:

$$Problem^{CBR} = \langle c^*, C \rangle, C = \{c_1, ..., c_k\}, c^* \notin C, \quad (3)$$

where c^* is a new case; C is a case base.

$$\begin{array}{l} Decision^{CBR} = \{d_1, ..., d_r\}, \\ d_i = (c_i, s_i), c_i \in C, s_i \in [0, 1], \end{array} \quad (4)$$

where $Decision^{CBR}$ is a problem decision in the form of a set of retrieved cases with similarities s_i.

Let's formalize a case description:

$$c_i = \left\{ Prop_i^{Problem}, Prop_i^{Decision} \right\}, \quad (5)$$

where $Prop_i^{Problem}$ is a identifying (characterizing) part of a case; $Prop_i^{Decision}$ is a learning part of a case. In addition, each of these parts contains task properties:

$$\begin{array}{l} Prop_i^{Problem} = \{p_1, ..., p_m\}, \\ Prop_i^{Decision} = \{p_{m+1}, ..., p_n\}, \\ Prop_i^{Problem} \cup Prop_i^{Decision} = Prop, \\ Prop_i^{Problem} \cap Prop_i^{Decision} = \varnothing \end{array} \quad (6)$$

The composition of the parts has a problem-specific character.

There are many methods that can be used for case retrieval [17] nearest neighbour, decision trees, etc. The most popular method is the nearest neighbour, based on an assessment of similarity with the aid of different metrics, for example, Euclidean, City-Block-Metric, etc. The proposed approach uses the nearest neighbour method and Zhuravlev metric [18] with normalisation:

$$s_i(c^*, c_i) = \sum_{j=1}^{N} w_j h_i(p_j^*, p_{ij})/N,$$
$$h_i(p_j^*, p_{ij}) = \begin{cases} \text{for quantitative} \begin{cases} 1, if|p_j^* - p_{ij}| < \xi \\ 0, else \end{cases} \\ \text{for qualitative} \begin{cases} 1, p_j^* = p_{ij} \\ 0, p_j^* \neq p_{ij} \end{cases} \end{cases} \tag{7}$$

where w_i is an information weight, and ξ is a constraint on the difference between the property values. At the same time, the normalisation (standardisation) is follows:

$$p_{ik} = \left(p_{ik} - \min_k x_{ik}\right) / \left(\max_k p_{ik} - \min_k p_{ik}\right). \tag{8}$$

B. Methodology

Generally, the methodology for expert systems engineering consists of the following main stages [19]: analysis and identification, conceptualization (structuring), formalization (representation), implementation and testing. In turn, in accordance with the formalization [7] the process of intelligent systems engineering based on model transformations can be considered as the sequential formation and transformation of models with varying degrees of abstraction:

- A computation-independent model (CIM) describes key abstractions. Some problem-oriented notations are used for its representation: UML, concept and mind maps, event trees. In turn, these models will be serialized in XML-like formats: XMI (StarUML, IBM Rational Rose), CXL (IHMC CmapTools), ETXL (ET-Editor), etc.
- A platform-independent model (PIM) based on a CIM. This model depends on the formalism of knowledge representation, but does not take into account features of languages and the tools for implementing these formalisms.
- A platform-specific model (PSM) takes into account features of languages and means for implementing formalisms.

At the same time, source codes or specifications of knowledge bases and intelligent systems generated on the basis of these models.

Figure 1 shows a comparison of the main stages of case bases and expert systems engineering and also their relationships with MDE artifacts.

C. Implementation

Implementation of the proposed methodology is made on the basis of a Personal Knowledge Base Designer (PKBD) platform [6]. PKBD is designed for the end-users and provides the creation of knowledge bases and domain-specific editors by means of an implementation of the main principles of the Model Driven-Architecture (MDA) [20] (a separate direction within the MDE/MDD). The MDA assumes a clear definition of three levels (viewpoints) of the software representation:

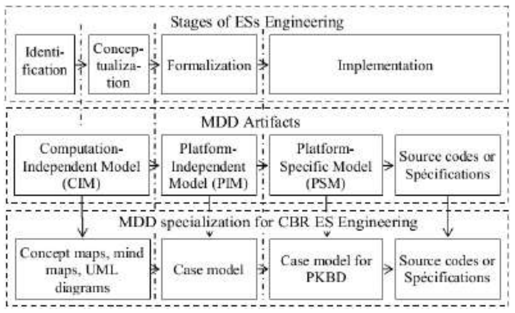

Stages of ESs Engineering
Identi-fication
Concep-tualiza-tion
Formalization
Implementation
MDD Artifacts
Computation-Independent Model (CIM)
Platform-Independent Model (PIM)
Platform-Specific Model (PSM)
Source codes or Specifications
MDD specialization for CBR ES Engineering
Concept maps, mind maps, UML diagrams
Case model
Case model for PKBD
Source codes or Specifications

Figure 1. The main stages of case bases and expert systems engineering and their relationships with MDE artifacts.

- A computation-independent level that describes the basic concepts and relationships of the subject domain expressed in the form of conceptual models. Models of this level can be form automatically on the basis of the analysis of XML-like formats, in particular: XMI (XML Metadata Interchange) for UML class diagrams, CXL (Concept Mapping Extensible Language) for concept maps, ETXL (Event Tree Mapping Extensible Language) for event trees.
- A platform-independent level that represents a domain model in the context of knowledge representation formalism used, in particular, logical rules or cases.
- A platform-dependent level that represents a formalized description of knowledge bases taking into account features of a certain software platform (for example, PKBD).

The PKBD architecture includes the main modules for: knowledge base management, knowledge representation languages support, dictionaries management, integration with CASE-tools, interpretation of models and the graphical user interface (GUI) generation. The interface is generated both in the form of pop-up windows and wizards. Wizards represent sequences of GUI forms, which segment and order the processes of entering and editing elements of a knowledge base. In particular, the user is consistently asked to specify: the name of a case template (used for display in the editor), the short name (used in the process of inference), the description and the properties (slots) of the template when entering it; the property name, its short name, description, a type (string, symbol or number), a possible default value and a constrain for this value (more, less, equal, unequal) when entering a slot. The similar wizards are used when entering and editing cases.

Let's consider one of the educational examples of case-base engineering with the aid of PKBD from a course of "Information Systems Toolkits" of the Irkutsk National Research Technical University (IrNITU).

IV. Case Study

The educational task of case base engineering of a static expert system for crystal minerals identifications is considered as an example.

A crystal mineral model in the form of a concept map is constructed at the conceptualization stage with the aid of IHMC CmapTools. A concept map fragment corresponding to a CIM is presented in Figure 2.

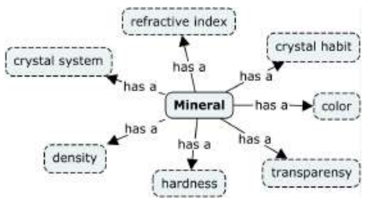
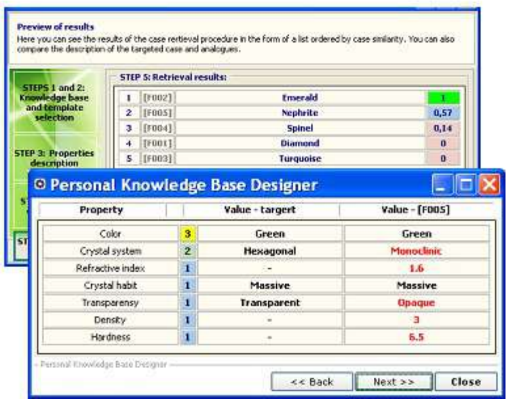


Figure 2. A fragment of a model of a "Mineral" concept.

Then, a PIM is formed in correspondence with the "Case" concept. In particular, all properties of the considered concept added to the identifying (characterizing) part, while the learning part consists only of a mineral name.

Cases can be added to case base either manually using a special wizard or by importing from text files. Testing the developed case base carried out in a special wizard by building queries (see Fig. 3).



STEPS 1 and 2: Knowledge base and template selection
STEP 3: Properties description
Personal Knowledge Base Designer
<< Back
Next >>
Close

Preview of results
Here you can see the results of the case retrieval procedure in the form of a list ordered by case similarity. You can also compare the description of the targeted case and analogues.

STEP 5: Retrieval results:

1	[F002]	Emerald	1
2	[F005]	Nephrite	0,57
3	[F004]	Spinel	0,14
4	[F001]	Diamond	0
5	[F003]	Turquoise	0

Property		Value - targert	Value - [F005]
Color	3	Green	Green
Crystal system	2	Hexagonal	Monoclinic
Refractive index	1	-	1.6
Crystal habit	1	Massive	Massive
Transparensy	1	Transparent	Opaque
Density	1	-	3
Hardness	1	-	6.5

Figure 3. Preview query results and comparison of cases.

V. Conclusion

Knowledge base engineering, in particular, case bases engineering for CBR decision support systems remains to be a time-consuming process, which requires the use of specialized tools. The effectiveness of the use of such tools can be improved by applying approaches based on generative programming and visualization.

The paper proposes the MDE application for development of case bases. The conceptual models in the form of concept maps presented in XML-like formats are used as the source data. PKBD (new module for case base engineering) is used as a mean for implementing the proposed approach. The tool tested on educational tasks and on the problem of structural material selection for a petrochemical systems design [21].

References

[1] A. R. Da Silva, *Model-driven engineering: A survey supported by the unified conceptual model*. Computer Languages, Systems & Structures, 2015, vol. 43, pp. 139-155.

[2] M. Brambilla, J. Cabot and M. Wimmer, *Model Driven Software Engineering in Practice*. Morgan & Claypool Publishers, 2012.

[3] A. Yu. Yurin, N. O. Dorodnykh, O. A. Nikolaychuk and M. A. Grishenko, *Designing rule-based expert systems with the aid of the model-driven development approach*. Expert Systems, 2018, vol. 35, no. 5, pp. 1-23.

[4] K. Czarnecki and S. Helsen, *Feature-based survey of model transformation approaches*. IBM Systems Journal, 2006, vol. 45, no. 3, pp. 621-645.

[5] N. O. Dorodnykh, S. A. Korshunov, N. Yu. Pavlov and A. Yu. Yurin, *Model Transformations for Intelligent Systems Engineering*. Open Semantic Technologies for Intelligent Systems, 2018, vol. 2, no. 8, pp. 77-81.

[6] M. A. Grishenko, N. O. Dorodnykh and A. Yu. Yurin, *Software for rule knowledge bases design: Personal Knowledge Base Designer*. Processing of the 6th International Scientific and Technical Conference: Open Semantic Technologies for Intelligent Systems (OSTIS-2016), 2016, pp. 209-212.

[7] N. O. Dorodnykh and A. Yu. Yurin, *About the specialization of model-driven approach for creation of case-based intelligence decision support systems*. Open Semantic Technologies for Intelligent Systems, 2017, vol. 7, pp. 151-154.

[8] A. Aamodt and E. Plaza, *Case-based reasoning: foundational issues, methodological variations, and system approaches*. Artificial Intelligence Communications, 1994, vol. 7, no. 1, pp. 39-59.

[9] R. Bergmann, *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*. Springer. 2002.

[10] A. Popa and W. Wood, *Application of case-based reasoning for well fracturing planning and execution*. Journal of Natural Gas Science and Engineering, 2011, vol. 3, no. 6, pp. 687-696.

[11] O. A. Nikolaychuk and A. Yu. Yurin, *Automating the identification of mechanical systems' technical state using case-based reasoning*. Processing of the 3rd International IEEE Conference Intelligent Systems, 2006, pp. 687-696.

[12] J. M. Gascuena, E. Navarro and A. Fernandez-Caballero, *Model-driven engineering techniques for the development of multi-agent systems*. Engineering Applications of Artificial Intelligence, 2012, vol. 25, no. 1, pp. 159-173.

[13] J. Baumeister and A. Striffler, *Knowledge-driven systems for episodic decision support*. Knowledge-Based System, 2015, vol. 88, pp. 45-56.

[14] R. Neto, P. J. Adeodato and A. C. Salgado, *A framework for data transformation in credit behavioral scoring applications based on model driven development*. Expert Systems with Applications, 2017, vol. 72, pp. 293-305.

[15] J. Canadas, J. Palma and S. Tunez, *InSCo-Gen: A MDD Tool for Web Rule-Based Applications*. Web Engineering, 2009, vol. 5648, pp. 523-526.

[16] M. A. Nofal and K. M. Fouad, *Developing web-based Semantic and fuzzy expert systems using proposed tool*. International Journal of Computer Applications, 2015, vol. 112, no. 7, pp. 38-45.

[17] L. R. Da Mantaras, D. Mcsherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. T. Cox, K. Forbus, M. Keane, A. Aamodt and I. Watson, *Retrieval, reuse, revision and retention in case-based reasoning*. Knowledge Engineering Review, 2005, vol. 20, no. 3, pp. 215-240.

[18] I. Yu. Zhuravlev and I. B. Gurevitch, *Pattern recognition and image recognition*. In Pattern recognition, classification, forecasting: Mathematical techniques and their application. Nauka: Moscow, 1989, vol. 1, pp. 5-72. (In Russ.)

[19] P. Jackson, *Introduction to expert systems*, 3rd ed. Harlow: Addison-Wesley, 1999.

[20] *MDA Specifications*. Available at: http://www.omg.org/mda/specs.htm (accessed 2018, Dec).

[21] A. F. Berman, G. S. Maltugueva and A. Yu. Yurin, *Application of case-based reasoning and multi-criteria decision-making methods for material selection in petrochemistry*. Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications, 2018, vol. 232, no. 3, pp. 204-212.

РАЗРАБОТКА ПРЕЦЕДЕНТНЫХ БАЗ ЗНАНИЙ С ИСПОЛЬЗОВАНИЕМ MDE-ПОДХОДА

Дородных Н.О., Юрин А.Ю.

В работе рассмотрено применение модельно-управляемого подхода для создания прецедентных баз знаний. Концептуальные модели, представленные в XML-подобных форматах, использованы в качестве исходных данных. Представлена формализованная постановка задачи, основные этапы подхода и инструментальное средство. Приведен пример решения учебной задачи.

Received 17.12.18