A method and tools for prototyping components of intelligent systems based on transformations

I. V. Bychkov¹, A. Yu. Yurin¹

¹ Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of the Russian Academy of Sciences (ISDCT SB RAS), 134, Lermontov str., Irkutsk, 664033, Russia

E-mail: iskander@icc.ru

Abstract. The paper describes a method and tools for prototyping components of intelligent systems. These means are the parts of the PESoT technology based on generative and visual programming as well as the model transformations. The proposed method redefines the main stages and models of the standardized model-driven approach, in particular, Model Driven Architecture in the context of developing knowledge bases and expert systems that use the logical rules formalism. The description of the method is presented. Proposed tools include languages and software. Specifically, the Rule Visual Modeling Language and Personal Knowledge Base Designer are used for knowledge bases engineering including modeling and codification; the Transformation Model Representation Language and Knowledge Base Development System support model transformations and the rapid creation of computation-independent models. Described tools were applied to solve problems in the field of reliability and safety of technical systems.

1. Introduction
Research aimed at improving the efficiency of knowledge bases and intelligent systems engineering has been conducted since their origin. However, despite some success in this area, the existence of specialized techniques (AT-TECHNOLOGY, MOKA, MIKE, OSTIS, IACPaaS, etc.) and tools (Expert System Designer, Expert System Creator, ARITY Expert Development Package, CxPERT, Exsys Developer, etc.), there has been no transition to industrial technology for creating this type of software. The process of developing intelligent systems and their components (software modules) continues to be a complex and time-consuming task, and its solution requires the development of programming theory. This is mostly due to the limitations of existing solutions related to the selectivity of supported development platforms, high qualification requirements for developers, weak support for visual and generative programming, low integration capabilities with software for conceptual and cognitive modeling, CASE tools, and other general-purpose systems that provide the representation of domain information.

The main way to overcome these limitations is to use the principles of generative and visual programming, as well as the concept of model transformations in the context of conceptualization, formalization, and automatic codification. The integrated use of these principles is implemented within a group of approaches based on model transformations. Currently, this direction in software
engineering is known as the model-oriented or model-driven approach (Model-Driven Engineering, MDE) [1].

This report describes a method and tools for developing intelligent system components based on MDE and software engineering standards such as UML, XMI, and MOF. The created components are decision-making modules that contain rule-based knowledge bases. Described tools were applied to solve problems in the field of reliability and safety of technical systems.

2. A method and tools

To overcome the shortcomings described in the introduction we developed a method and tools for creating components of intelligent systems in the form of software modules for decision-making. The developed means are based on the principles of model transformations, and form the PESoT (Prototyping Expert Systems Based on Transformations) technology.

The following tasks were solved: analysis of modern approaches, methods, and tools for creating knowledge bases with the use of model transformations; development of a method for creating decision-making modules containing knowledge bases with logical rules; development of languages and software for supporting the proposed method; evaluation of the developed method and tools when solving real-world and educational tasks [2-10].

The main results of the solved tasks are the following:

- A method for creating expert systems and knowledge bases using the sequential transformation of conceptual models.
- Domain-specific languages that support this method.
- Software that supports the method and languages.
- Evaluation results of the method, languages, and software.

Let's describe the results in detail.

2.1. A method

The method is based on a chain of sequential model transformations of a standardized model-driven approach (namely Model-Driven Architecture, MDA), providing a transition from models with more abstraction to models with less abstraction and obtaining interpreted source codes and specifications at the end of the chain. According to the method, the chain consists of the following main stages (Fig. 1) [6]:

- Building computation-independent models (CIM), in particular: domain models and expert system models. These are the most general models that contain information about the main concepts and relationships, as well as the main modules of the software that provide business logic. To build models of this stage, OWL ontologies, UML class diagrams, concept maps, decision tables, which can be sterilized in XML-like formats (in particular, XMI and XTM) or typed text files (for example, CSV) can be used.
- Building domain-specific platform-independent models (PIM) by specialization of computation-independent models, taking into account the tasks to be solved and the type of software to be created. In our work, the formalism of logical rules and the architecture of rule-based expert systems are the target artifacts. The Rule Visual Modeling Language (RVML) is used to represent logical rules [5], UML (Unified Modeling Language) is used to describe the architecture. Models of the stage are obtained as a result of the transformation of computation-independent models. The Transformation Model Representation Language (TMRL) [7] can be used to describe these transformations.
- Building platform-specific models (PSM) that take into account the features of the software implementation for a specific language (or a platform). Our method considers the features of the CLIPS (C Language Integrated Production System) and DROOLS, as well as the PHP (Hypertext Preprocessor) general-purpose language. Information about the Personal Knowledge Base Designer (PKBD) [4] specifications is used when creating expert systems.
3

Figure 1. Main stages and models of the method of the PESoT technology [6].

- Generating platform-specific source codes and specifications for knowledge bases and expert systems. CLIPS, DROOLS, PHP, PKBD are used as target platforms.
- Testing generated codes and specifications in the development tool (PKBD) or after integration to the application.

MDA specialization involves the use of metamodels and a four-level metamodeling scheme [1]. An example of this scheme for our method is shown in Fig.2. The method involves the use of some OMG standards: UML, XMI, MOF, but also the additional and new elements, such as OWL, CSV, RVML, and TMRL (that replaces QVT). Features of our method (its difference from the standard four-level metamodeling scheme) are highlighted in Fig.2.

2.2. Languages for the PESoT method
The method includes the development and the use of new languages:

- A UML-based graphical notation for modeling domain knowledge in the form of logical rules, it is called Rule Visual Modeling Language [5]. This notation differs from the known ones by representing logical rules components in the form of specialized graphical elements and clearly defines the rule core. RVML is described in detail in [5].
- A domain-specific language for describing model transformations, namely Transformation Model Representation Language (TMRL). This language provides a declarative description of transformation rules and focuses on XML-like standards (XMI, XTM, etc.). A detailed description of TMRL is made in [7].
2.3. Software for the PESoT method

The special software was created to support our method and languages:

- Personal Knowledge Base Designer (PKBD) [4] is a software tool for prototyping knowledge bases and expert systems that use logical rules. The main features of PKBD are RVML support, integrated models and metamodels of rule-based and case-based expert systems.
- Knowledge Base Development System (KBDS) [3] is a tool for designing model transformations components. KBDS is used for forming computation-independent models by transforming various conceptual models serialized in XML-like formats. The features of KBDS are the TMRL and RVML support, as well as built-in metamodels of OWL ontologies and RVML logical rules.

2.4. Evaluation

The method and tools were evaluated when solving real-world tasks in collaboration with the Irkutsk research and design institute of chemical and petrochemical engineering (IrkutskNIIHimmash) in the field of investigating reliability and safety of technical systems. In particular, the knowledge bases and expert systems were created for:

- a module of the IS (Industrial Safety) Expertise software [8] designed for defining the causes of damages and destructions of technical systems (our first case study);
- predicting degradation processes for petrochemical technical systems (our second case study) [9];
- a failure analysis process intelligent scheduler [10] (our third case study).
As a result of the evaluation, it was concluded that the method and tools reduce the time spent on prototyping intelligent components, as well as when creating applications for multiple platforms.

3. Conclusion
In this paper, we propose a method and tools for improving the efficiency of creating components of intelligent systems in the form of knowledge bases and expert systems that use the logical rules formalism. The proposed means implement model transformations in terms of a standardized model-driven approach. The novelties of our proposals are the following: the redefined chain of model transformations, original languages and software, as well as their application technique.

The proposed methods and tools can reduce the time of prototyping decision support modules containing knowledge bases. At the same time, the maximum effect is observed when reusing previously developed conceptual models, as well as when developing applications for multiple platforms. However, the resulting source codes require further modification and integration.

4. Acknowledgments
The research was supported by the Program of the Fundamental Research of the Siberian Branch of the Russian Academy of Sciences, project no. IV.38.1.2 (reg. no.AAAA-A17-117032210079-1), project no. IV.38.1.3 (reg. no. AAAA-A17-117032210077-7). Results are achieved using the Centre of collective usage «Integrated information network of Irkutsk scientific educational complex».

References
[1] Da Silva A R 2015 Model-driven engineering: A survey supported by the unified conceptual model Computer Languages, Systems & Structures 43 139–155
[2] Dorodnykh N O and Yurin A Y 2019 Technology for creating production expert systems based on model transformations. (Novosibirsk: SB RAS)
[3] Bychkov I V et al 2016 Computational Technologies 4 16–36
[4] Yurin A Y and Dorodnykh N O 2020 SoftwareX 11 100411
[5] Yurin A Y et al 2018 Expert Systems 35(5) 12291
[6] Yurin A Y 2020 CEUR Workshop Proceedings (ITAMS-2020) 2677 36–50
[7] Dorodykh N O and Yurin A Y 2018 CEUR Workshop Proceedings (ITAMS-2018) 2221 70–75
[8] Berman A F et al 2015 Chemical and Petroleum Engineering 50(1-2) 730–738
[9] Nikolaychuk O.A and Yurin A Y 2008 Expert Systems with Applications 34 635–642
[10] Berman A F et al 2010 Expert Systems with Applications 37 7101–7107