Rule-based Systems, CASE tools,Visual Design

Grzegorz J. NALEPA, Antoni LIGEZA*

A VISUAL EDITION TOOL FOR DESIGN AND VERIFICATION OF KNOWLEDGE IN RULE-BASED SYSTEMS**

In order to address problems encountered in the design of rule-based and expert systems the paper discusses a proposal of a new approach to rule-based system design and verification using an integrated CASE tool supporting visual design of rule-based systems. The Mirella tool allows for dynamic on-line specification of components of the knowledge with simultaneous analysis of the rule-based system during the design phase. The main idea is to move the design procedure to a more abstract, logical level, where knowledge specification is based on use of abstract rule representation, called eXtended Tabular Trees, supported by Mirella CASE tool. It consists of visual design environment integrated with Prolog-based analysis and verification engine. Along with editor`s built in on-line checking capabilities it improves system performance and safety. Selected implementation aspects are also highlighted. The environment is implemented in ANSI C in the GNU/Linux environment using Gtk/GNOME/SWI-Prolog development platform. This makes it both efficient and portable. The tool has a multilayer, multimodule architecture which gives it flexibility and allows for future extensions.

1. INTRODUCTION

Rule-based systems (RBS) provide an omni-present methodological tool for thinking of, specifying, implementing and executing dynamic operations in the most general terms. Thinking in terms of rules while specifying behavior of systems constitutes practically useful, intuitive, and theoretically valid approach.

However, although rule-based technologies appear simple and intuitive at the first glimpse, designing a real-scale rule-base is both tedious and difficult task. The main problem is that in systems having more than several rules it becomes difficult to control the properties of them at the design stage. A well-defined system should be safe, reliable, and efficient and these features are further translated into a set of precisely defined characteristics which can be verified in a formal way. Here one typically considers that such a system must be complete, i.e. work in any input situation, deterministic, i.e. it s behavior must be predictable, correct, i.e. it must work according to desired specification, and it should be minimal, i.e. the size (e.g. number of rules) it should incorporate only necessary set of rules [1]. In practice there are practically no universal and efficient tools for support of edition of rule-based systems incorporating verification and design support capabilities.

---

* AGH University of Science and Technology, Institute of Automatics, <gjn@agh.edu.pl>,<ligeza@agh.edu.pl>
**Supported by KBN Research Project No 4 T11C 035 24

This paper is devoted to presenting , a tool for intelligent support of design, analysis and implementation of rule-based systems in general. It is oriented towards designing reliable and safe rule-based systems in general. The main goal of the system is to move the design procedure to a more abstract, logical and graphical level, where knowledge specification is based on use of abstract rule representation. The designed graphical specification is automatically translated into a predefined XML knowledge format, so the designer can focus on logical specification of safety and reliability; simultaneously, practical code can be generated for a wide class of systems. On the other hand, selected formal aspects may be automatically verified on-line during the design, so that it verifiable characteristics are preserved.

The organization of this paper is as follows: in Sect. 2 a critical perspective on the state of Rule-Based Systems (RBS) design methods is presented, eXtended Tabular-Trees which are a key aspect of new approach to RBS design are presented in Sect. 3, Sect. 4 discusses a intelligent CASE tool , in Sect. 5 concluding remarks are given.

2. A CRITICAL PERSPECTIVE ON RBS DESIGN METHODS

The following limitations concerning current approaches (and tools) for developing rule-based systems have been observed.

First, some most important limitations consist in using system-specific knowledge representation formalism. In fact, knowledge representation methods used are suitable only for systems of limited class, they have scalability issues-they are not suitable for large scale systems, and are not invented with CASE tools integration in mind.

Second, the problem with most tools is that no formal verification of system properties in early stages of the development cycle is carried out, which results in problems with late verification; in practice, the design-verification cycle must sometimes be repeated several times, since verification and correction of certain modules with respect to some characteristics influences their properties with respect to other features [3].

The third issue is that design approaches do not offer integrated development process covering all the stages from design to implementation phase. Such methodologies support mainly subsequent stages of the conceptual design in case of large systems, while direct technical support of the logical design and during the implementation phase is mostly limited to providing a context-sensitive, syntax checking editors [4].

To overcome limitations discussed above new solutions in all the three areas mentioned above are offered:

- a new knowledge representation method for graphical knowledge specification, called eXtended Tabular Trees (XTT for short), is offered,
- an integrated RBS design and verification processes is proposed, it verifies properties such as completeness, determinism, and subsumption,
- Mirella a new, visual, XTT-based CASE tool supporting this process is introduced.

The proposed tool, discussed in detail in this article, solves all of the above problems.

3. INTRODUCING EXTENDED TABULAR TREES

The main idea behind new visual knowledge representation language called Extended Tabular-Trees aims at combining some of the existing approaches such as decision-tables and decision-trees by building a special a hierarchy of Object-Attribute-Tables [4,5,6]. This hierarchy is based on the Psi-tree structure.

The new language has some unique features such as: simplicity and transparency, due to an intuitive way of graphical knowledge specification, hierarchical, tree-like knowledge representation, highly efficient way of visualization with high data density using tables similar to Relational Database tables, power of the decision table representation, flexibility with respect to knowledge manipulation, analogies to the RDB data representation scheme, direct knowledge representation mapping into PROLOG schemes and rule-based systems.

The language plays a key role in the new approach to RBS design method, on which the Mirella tool is based. Some ideas used in its development were previously presented in [4,8,9]. They have been, however, vastly improved and refined.

4. THE MIRELLA TOOL

4.1. SYSTEM ARCHITECTURE

The architecture of the new Environment consisting of several modules and layers.
There are four main system layers:

- **environment** it provides ways of importing, exporting and presenting knowledge base in both human and machine readable formats,
- **visual -** it allows for visual design of knowledge base,
- **logic** it provides means of formal verification of knowledge base,
- **semantics** it helps in adding domain-specific semantics to the knowledge base.

The **knowledge base** itself is a repository for XTT based system description.
There are four principal modules of the system:

- **Mirella Creator -** which supports defining of system attributes with specific constraints, and semantic information,
- **Mirella Designer -** which allows for visual design of system knowledge base using XTT schema,
- **Mirella Validator -** which performs on-line analysis of system formal properties via an integrated Prolog compiler,
- **Mirella Exchanger -** which consists of number of import and exports plugins, exchanging data with other knowledge representation languages, and domain specific system prototypes.

4.2. DEVELOPMENT PLATFORM

The development platform for the environment has been chosen with flexibility, portability and efficiency in mind. The platform consists of several middleware layers supporting GUI, canvas management, Prolog integration and XML handling; main components are:

- *GLib* is a low-level general-purpose core utility library, providing a portable implementation of fundamental data types and algorithms and an advanced and extensible type/object system for ANSI C language.
- *GDK* is a middleware that implements a portable general drawing toolkit providing an abstraction over native operating system drawing and windowing system.
- *GTK2* is an advanced library for building complex multiplatform graphical user interfaces, it provides multiple GUI components, widgets. GTK2 has a C-based object-oriented architecture that allows for maximum flexibility.
- *libGlade* allows for dynamic runtime GUI building using an XML-based interface description; it also provides functions that can be used to connect signal handlers to parts of the interface.
- *GNOMECanvas* provides a high-level engine for creating structured graphics that offers high performance rendering and rich imaging model;
- *libXML2* is a portable and efficient parser toolkit for XML that implements all core XML technologies such as XML-NameSpaces, XPath, and XSLT.

The architecture described above is implemented in pure, ANSI C language, and is able to run on multiple platforms. While the environment is being implemented in the GNU/Linux environment, it can be easily ported to Sun Solaris, and possibly Windows platform.

4.3. PROLOG INTEGRATION

The SWI-Prolog is a modern ISO-compliant portable Prolog compiler. The SWI-Prolog was selected due to its: flexibility, portability, availability, and possibility of integration with other software. It offers a bi-directional programming interface to and from ANSI C language allowing for easy integration with other code. It provides a built-in XML/RDF parser, suitable for direct XML processing and applications in the field of *Semantic Web.*

4.4. MIRELLA USE CASES

Mirella can vastly improve knowledge engineering process in the design of RBS. It already has several scientific and engineering applications in the fields of visual design support, web security [9], and control systems. So far Mirella has been used to design several types of rule-based systems. An example is shown in Fig. 1.

Another effort aims at combining some of Mirella features with the Adder project (*http://home.agh.edu.pl/~adder*) developed as a part of KBN Research Project No 4 T11C 035 24. The project deals with application of colored Petri nets as a formal method for requirements specification of real-time systems, and as an algebraic and graphical language for design of executable models of such systems [10,11].

Mirella home page, hosting information about current status of the project, is located at *http://mirella.ia.agh.edu.pl.* The Mirella project is developed as a part of KBN Research Project No 4 T11C 027 24.

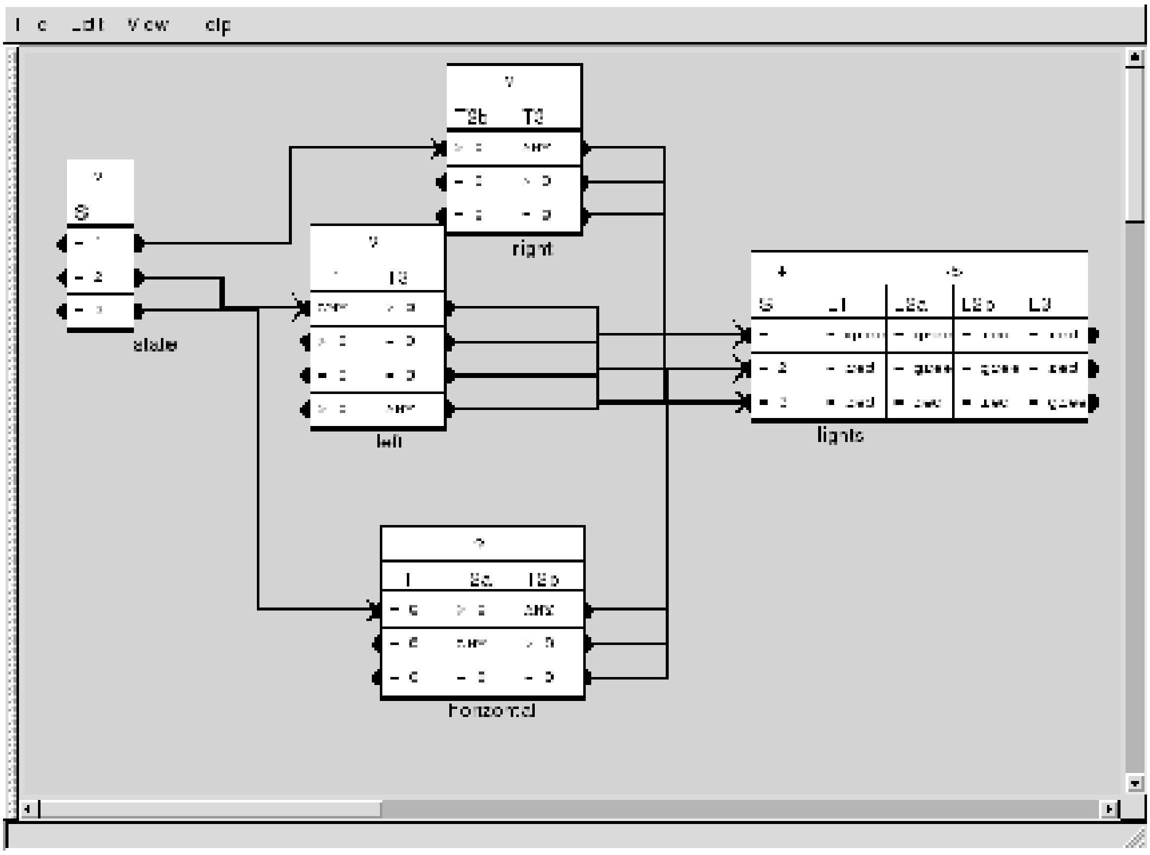

Fig. 1: Example of Rule-based system designed in Mirella

5. CONCLUDING REMARKS

The paper presents the design and implementation of the Mirella tool: an integrated visual editor for RBS design and verification. It is based on the idea of a new, integrated approach to RBS design and uses the XTT visual knowledge representation language, which can be used as a modern knowledge acquisition tool. Mirella tool has both scientific and engineering applications, e.g. in the field of web security [9], and control systems. Through integrating design and analysis within a single CASE tool, the work presented in the paper opens new horizons in the domain of architecture of expert systems design.

REFERENCES

[1] LIGEZA, A.: *Intelligent data and knowledge analysis and verification; towards a taxonomy of specific problems.* Validation and Verification of Knowledge Based Systems: Theory, Tools and Practice (1999) 313-325

[2] ANDERT, E.P.: *Integrated knowledge-based system design and validation for solving problems in uncertain environments*. Int. J. of Man-Machine Studies 36 (1992) 357-373

[3] VAN HARMELEN, F.: *Applying rule-based anomalies to kads inference structures.* ECAI'96 Workshop on Validation, Verification and Refinement of Knowledge-Based Systems (1996) 41-46

[4] LIGEZA A., WOJNICKI I., NALEPA, G.: *Tab-trees: a case tool for design of extended tabular systems.* In et al., H.M., ed.: Database and Expert Systems Applications. Volume LNCS 2113 of Lecture Notes in Computer Sciences. Springer-Verlag, Berlin (2001) 422-431

[5] LIGEZA, A.: *Towards design of complete rule-based control systems.* In J. Kocijan, R.K., ed.: IFAC/IMACS International Workshop on Artificial Intelligence in Real-Time Control. IFAC, Bled, Slovenia (1995) 189-194

[6] LIGEZA, A.: *Logical support for design of rule-based systems. reliability and quality issues.* In Rousset, M., ed.: ECAI-96 Workshop on Validation, Verification and Refinment of Knowledge-based Systems. Volume W2. ECAI'96, Budapest1996 28-34

[7] LIGEZA, A.: *Toward logical analysis of tabular rule-based systems.* International Journal of Intelligent Systems 16 (2001) 333-360

[8] NALEPA, G.J., LIGEZA, A.: *Graphical case tools for integrated design and verification of rule-based systems.* In T. Burczynski, W. Cholewa, W.M., ed.: AI-METH 2002 : Methods of Artificial Intelligence, 13-15 November, Gliwice, Poland, Gliwice, Poland, Silesian University of Technology. SUT DSMCM DFMD, Polish Association for Computational Mechanics (2002) 307-313

[9] NALEPA, G.J., LIGEZA, A.: *Designing reliable web security systems using rule-based systems approach.* In Menasalvas, E., Segovia, J., Szczepaniak, P.S., eds.: Advances in Web Intelligence. First International Atlantic Web Intelligence Conference AWIC 2003, Madrid, Spain, May 5-6, 2003. Volume LNAI 2663 of Lecture Notes in Artificial Intelligence., Berlin, Heidelberg, New York, Springer-Verlag (2003) 124-133

[10] SZPYRKA M., NALEPA G.J., *On Two Formal Approaches to Design and Verification of Embedded Rule-based Systems,* accepted to 28th IFAC/IFIP Workshop on Real-Time Programming, WRTP2004

[11] SZPYRKA M., SZMUC T. *RTCP-nets as a Tool for Real-Time Systems Modelling and Analysis.* In Colnaric M., Adamski M., W?grzyn M. (Eds.) Real-Time Programming 2003, Elsevier Science Ltd. pp. 21-26.