Formal design, Rule-based systems, Tabular-Trees

Grzegorz J. NALEPA, Antoni LIGEZA*

A GRAPHICAL TABULAR MODEL FOR RULE-BASED LOGIC PROGRAMMING AND VERIFICATION**

New trends in development of databases and expert systems seems to underline the role of graphical specification tools, visual information modeling and formal verification procedures. This paper incorporates these new ideas and, moreover, tries to present putting them in engineering practice. The main goal 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*. The main idea behind *XTT* is to build a hierarchy of Object-Attribute-Value Tables (OAV table). The basic component for knowledge specification is an OAV table. It is analogous to a relational database table; however, it contains conditional part and decision columns. Moreover, the attribute values can be non-atomic ones. Each row provides specification of a single rule. The OAV tables can be connected with one another through appropriate links specifying the control flow in the system. The design specification is automatically translated into Prolog code, so the designer can focus on logical specification of safety and reliability. On the other hand, formal aspects such as completeness, determinism, etc. are automatically verified on-line during the design, so that it verifiable characteristics are preserved. From practical point of view, the design process is performed with a intelligent tool named *Mirella*.

1. INTRODUCTION

Thinking in terms of rules while specifying formal specification of behavior seems to be both practically useful and intuitive, as well as theoretically justified. Rule-based approaches are omnipresent in science and technology, and computer programming technologies are perhaps one of brilliant examples of practical efficiency and effectiveness of this methodology. In fact, from the level of hard-wired micro-programs embedded in processors to abstract, logical knowledge specification in expert systems rule-based specifications serve as simple, yet powerful in the hands of experienced programmers.

However, although rule-based technologies appear simple and intuitive at the first sight, 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 their properties 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,

---

* 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

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 it should incorporate only necessary set of rules [1].

New trends in development of databases and expert systems seems to underline the role of graphical specification tools, visual information modeling and formal verification procedures. This paper incorporates these new ideas and, moreover, tries to present putting them in engineering practice. The main goal is to move the design procedure to a more abstract, logical level, where knowledge specification is based on use of abstract rule representation. The design specification is automatically translated into Prolog code, so the designer can focus on logical specification of safety and reliability. On the other hand, formal aspects such as mentioned above (completeness, determinism, etc.) are automatically verified on-line during the design, so that its verifiable characteristics are preserved. From practical point of view, the design process is performed with a full-screen intelligent graphical tool named .

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 given, Sect. 3 presents a new approach to RBS design, eXtended Tabular-Trees which are a key aspect of this approach are detailed in Sect. 4, Sect. 5 introduces an intelligent CASE tool, in Sect. 6 concluding remarks are given.

2. A CRITICAL PERSPECTIVE ON RBS DESIGN METHODS

On the base of research and evaluation of multiple RBS design methods [2,3], supported by development tools [4], a conclusion has been drawn, that existing methods and tools have some serious limitations. These limitations are located in three areas:

- knowledge representation method,
- verification of formal properties,
- computer tools supporting the design process.

Some most important limitations concerning the first 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.

With respect to the second, the problem 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.

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. CASE tools offer support for limited class of systems only, and are not designed for openness and availability, and are often user unfriendly.

3. NEW APPROACH TO RBS DESIGN

The approach to Rule-Based Systems design process proposed in this paper is based on the idea of integrated design and verification supported with an intelligent, interactive tool. It uses a domain-independent knowledge specification based on XML. To overcome these limitations 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 implementation processes is proposed, it includes analysis of selected formal properties,
- a new, visual, XTT-based CASE tool supporting this process is introduced.

The eXtended Tabular Trees-based design methods introduces on-line analysis of selected system properties during system design phase. The principal idea is to map system structure represented by XTT to Prolog-based representation, and to verify it on-line by an integrated Prolog-based inference engine. To fully support new design approach a modern CASE tool has been built. It supports both visual XTT knowledge representation method and integrated Prolog-based verification engine. More details are given in Sect. 5.

4. EXTENDED TABULAR TREES

In this section a new visual knowledge representation language is introduced. Some ideas used in its development were previously presented in [4,8,9]. They have been, however, vastly improved and refined. The main idea behind the 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 hierarchy of Object-Attribute-Tables [5,6,7]. This hierarchy is based on the concept of Psi-tree structure [5,6]. The new language has some unique features such as:

- simplicity and transparency - an intuitive way of knowledge representation,
- hierarchical, tree-like knowledge representation,
- highly efficient way of visualization with high data density,
- power of the decision table representation,
- flexibility with respect to knowledge manipulation,
- analogies to the RDB data representation scheme,
- direct knowledge representation mapping to PROLOG and RBSs.

4.1. SYNTACTIC STRUCTURE

The basic syntactic notions of XTT are: *Attribute*, *Cell*, *Header*, *Row*, *Table*, *Connection*, *Tree.*

4.2. VISUAL REPRESENTATION

A XTT Visual Representation is crucial from the RBS design point of view.
An example of a Table is shown in Fig. 1.

4.3. SEMANTIC INTERPRETATION

Semantic interpretation of XTT uses some well established concepts.

Rule Mapping A Row on a Table is interpreted as a production rule, of a form:
The condition part of the rule is mapped to the Conditional context of the Row. However the conclusion part is mapped to Assert, Retract, and Decision contexts of the Row. Use of Assert/Retract contexts allows for dynamic modification of RBS knowledge base.
So in practice it is a extended rule, allowing for non-monotonic reasoning, with explicit control statements.

Table Mapping A Table is simply interpreted as a set of rules, where rule *j+1* is processed after rule *j*. However rules grouped in a Table share the same attributes. This concept is similar to Decision Tables and to RDBMS knowledge representation.

Tree Mapping A concept of Tree allows for building a hierarchy of Tables. Each Row *j* of a Table *x* can have a right Connection to the other Row *k* in other Table *y*. Such a connection implies logical AND relation in between. Rule processing is then transferred from Row *j* in Table *x* to Row *k* in Table *y*. This concept is similar to Decision Trees.

Prolog Mapping Another semantic interpretation comes with XTT to Prolog mapping. Any subset of an XTT Tree hierarchy can be mapped to corresponding Prolog code. This representation is crucial to the formal verification of XTT.

XML Mapping It is worth noting, that while this does not introduce a new semantical interpretation, XTT can be represented in an XML-based XTTML (XTT Markup Language) suitable for import, export operations as well as translated to XML-based formats such as RuleML.

Logical Aspect An important feature of XTT is the fact, that besides its visual representation they have a well defined, logical form which may be formally analyzed.

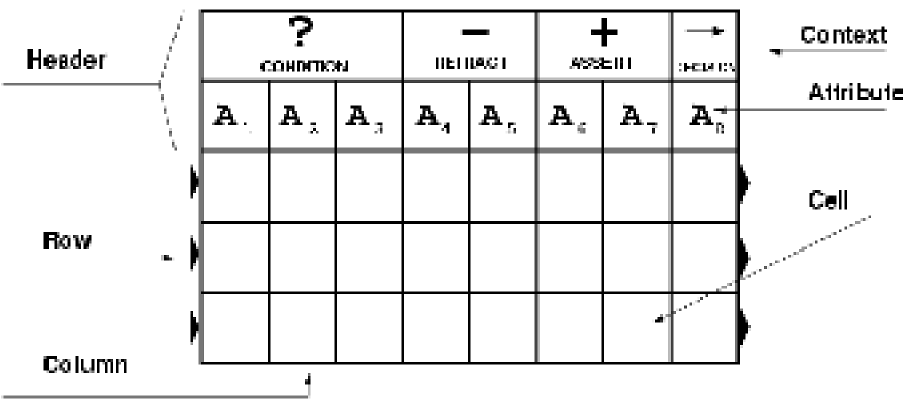

Fig. 1. An Example of XTT Table

5. INTRODUCING MIRELLA TOOL

Mirella is an intelligent visual design tool supporting on-line verification of rule-based systems, based of the XTT knowledge representation. 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 (XTTML) knowledge format, so the designer can focus on logical specification of safety and reliability; simultaneously, practical code can be generated form a wide class of systems. On the other hand, formal aspects such as completeness, determinism, etc. may be automatically verified on-line during the design, so that it verifiable characteristics are preserved.

Mirella features can be combined with the Adder project (*http://home.agh.edu.pl/~adder*) which 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's multilayer and multimodule architecture has been detailed in the other article entitled: ``A Visual Edition Tool for Design and Verification of Knowledge in Rule-based Systems'' submitted to ICSS 2004 Conference.

6. CONCLUDING REMARKS

The paper presents a new, integrated approach to rule-based systems design. It introduces the XTT knowledge representation language, which can be used as a modern knowledge acquisition tool. The Mirella tool, introduced in the paper has both scientific and engineering applications, e.g. in the field of web security [9], and control systems. Through integrating design and verification 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.