Visual Design and On-line Verification of Tabular Rule-Based Systems with XTT*

Antoni Ligęza and Grzegorz J. Nalepa
Institute of Automatics,
AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland
ligeza@agh.edu.pl, gjn@agh.edu.pl

Abstract: The paper is dedicated to presentation of a new approach to joint design and verification of rule-based systems. The principal idea is that verification should be performed on-line, incrementally, during system design. This allows for early detection and handling of knowledge base anomalies and inconsistencies. The proposed approach offers also an innovative visual tool for computer-aided design. Knowledge representation is based on eXtended Tabular Trees (XTT) a very flexible and powerful knowledge representation paradigm combining the expressive power of attributive decision tables and decision trees. Design and verification process with an XTT design tool and the visual editor is discussed.

1 Introduction

Over thirty years rule-based systems (RBS, [Lie98]) prove to constitute one of the most substantial technologies in the area of applied Artifi cial Intelligence (AI). Rules of various particular forms implement the core of numerous applications, including expert systems, decision support systems, control and monitoring systems and knowledge-based systems in general. Modern rule-based systems fi nd applications ranging from medicine to fi nance, and economy, going well beyond traditional rule-based programming. Some interesting recent applications of rule technology are the ones of business and semantic web [Bea04]. Both research and applications go far beyond the traditional paradigm [HS02, BM04]. Knowledge specifi cation with rules is used both for defi nition of domain knowledge as well as meta-knowledge concerning inference control.

Although rules constitute one of the simplest and most transparent programming paradigms, practical implementation of rule-based systems (RBS) encounters important problems. A non-trivial system may contain less than fi fty rules and simultaneously it may be diffi cult to handle. The main problems encountered concern *complete* specifi cation of *non-redundant* and *consistent* set of rules [Lie98, Lig05]. This turns out to be a tedious task requiring far-going efforts.

*Research supported from KBN Research Project *Adder* No.: 4 T11C 035 24

The approaches to verifi cation and validation of RBS [Lie98, GS00, PBS92, SG02, VC99] do not solve the problem entirely. Verifi cation performed after the system is designed is both costly and late. Moreover, after introducing the corrections of detected errors, the verifi cation cycle must be repeated. The problem consist in the possibility of introducing new errors through correction of the old ones – there in no warranty that the verifi cation-correction cycle is fi nite.

This paper is dedicated to presentation of a new approach to the issue of verifi cation and assuring system correctness. The principal idea is that *verification should be performed on-line, incrementally, during design of the system*; moreover, error detection should lead the designer back through the design, towards error elimination.

Research towards amalgamating the design and verifi cation stages has been undertaken several years ago [Lig96, LWN01]. It resulted with elaboration of a new approach combining the expressive power of decision trees and attributive decision tables of non-atomic attribute values [Nal04]. A more complete presentation is also given in recent book [Lig05].

The approach presented in this paper allows for early detection and handling of knowledge base anomalies and inconsistencies by incorporating formal validation of RBS in the system design phase. A formal concept of a design tool for specifying attributive rule-based systems (called XTT) with a visual editor is outlined. These concepts have been implemented in a prototype CASE tool called MIRELLA. Design and verifi cation process is also discussed.

2 Foundations of the XTT Approach

Although some expert system shells provide specialized editors for defi ning the rules, majority of the specifi cations are in pure, textual form. Contrary to some grammar-checking proposals, the presented approach is based on introducing well-defi ned structure into both knowledge representation and the design process. This leads to reduction of the number of errors on one hand, and allows to verify even partially specifi ed knowledge on the other.

For knowledge specifi cation an attributive logical language allowing for non-atomic attributes values is selected [Lig05]. Rules of similar attribute scheme can be easily combined into a special form of decision table. The *Extended Attributive Table* (XAT) (or *eXtended Table* (XT), for short) (see [Lig05, Nal04]) takes the following form (Tab. 1).

Each rule is represented by one row. In the fi rst column there is the rule identifi er, the *Ctx* is the context common for all the rules, $A_1 \ldots A_n$ are the preconditions attributes, and $B_1 \ldots B_b$ are the retract, $C_1 \ldots C_c$ are the assert, and $H_1 \ldots H_h$ are the conclusion-defi ning ones. The *Ctrl* part defi nes the (N) next rule to be executed (if present) or the (E) else rule to be executed in case of failure. Hence, the table can specify both declarative knowledge (rules) and control knowledge (the *Ctrl* column).

In order to illustrate the idea of the XT knowledge representation let us show its use for specifi cation of the Thermostat system ([Neg02]). The system contains 18 rules for defi ning the set-point (required temperature) for a thermostat system. The original rules can be

Table 1: The basic form of an XAT

Info		Prec	Retract	Assert	Decision	Ctrl	
I	Ctx	$A_1 \dots A_n$	$B_1 \dots B_b$	$C_1 \dots C_c$	$H_1 \dots H_h$	N	E
1	ψ	$t_{11} \dots t_{1n}$	$b_{i1} \dots b_{1b}$	$c_{11} \dots c_{1c}$	$h_{11} \dots h_{1h}$	g_1	e_1
$\vdots$	$\vdots$	$\vdots \ddots \vdots$	$\vdots \ddots \vdots$	$\vdots \ddots \vdots$	$\vdots \ddots \vdots$	$\vdots$	$\vdots$
i	ψ	$t_{i1} \dots t_{in}$	$b_{i1} \dots b_{ib}$	$c_{i1} \dots c_{ic}$	$h_{i1} \dots h_{ih}$	g_i	e_i
$\vdots$	$\vdots$	$\vdots \ddots \vdots$	$\vdots \ddots \vdots$	$\vdots \ddots \vdots$	$\vdots \ddots \vdots$	$\vdots$	$\vdots$
k	ψ	$t_{k1} \dots t_{kn}$	$b_{k1} \dots b_{kb}$	$c_{ik} \dots c_{kc}$	$h_{k1} \dots h_{kh}$	g_k	e_k

found in [Neg02, Lig05]. They are defi ned with four XT specifi cations (Tab. 2-5).

Table 2: An XT for rules 1 and 2. *Context 1:* none.

Info	Prec	Retract	Assert	Decision	Ctrl	
I	aDD	aTD	aTD	H	N	E
1	sWD	–	wd		2.3	1.2
2	sWK	–	wk		2.6	1.1

Table 3: An XT for rules 3–6. *Context 2:* $aTD \in \{wd, wk\}$.

Info	Prec		Retract	Assert	Decision	Ctrl	
I	aTD	aTM	aOP	aOP	H	N	E
3	wd	[9:00, 17:00]	–	dbh		3.7	2.4
4	wd	[00:00, 09:00]	–	$ndbh$		3.7	2.5
5	wd	[17:00, 24:00]	–	$ndbh$		3.7	2.6
6	wk	–	–	$ndbh$		3.7	2.3

Table 4: An XT for rules 7–10. *Context 3:* none.

Info	Prec	Retract	Assert	Decision	Ctrl	
I	aMO	aSE	aSE	H	N	E
7	$sSUM$	–	$summer$		4.13	3.8
8	$sAUT$	–	$autumn$		4.15	3.9
9	$sWIN$	–	$winter$		4.17	3.10
10	$sSPR$	–	$spring$		4.11	3.7

The tables incorporate the following attributes: aDD – day, aTD – today, aTM – time, aOP – operation, aMO – month, aSE – season, $aTHS$ – thermostat_setting.

Further, the following attributes values were used (the name of each set value starts with 's'), $sWD = \{Mon \dots Fri\}$, $sWK = \{Sat, Sun\}$; $sSUM = \{Jan, Feb, Mar\}$, $sAUT =$

Table 5: An XT for rules 11–18. *Context 4:* $aSE \in \{spr, sum, aut, win\} \wedge aOP \in \{dbh, ndbh\}$.

Info	Prec		Retract	Assert	Decision	Ctrl	
I	aSE	aOP			$aTHS$	N	E
11	spr	dbh			20	1.1	4.12
12	spr	$ndbh$			15	1.1	4.13
13	sum	dbh			24	1.1	4.14
14	sum	$ndbh$			17	1.1	4.15
15	aut	dbh			20	1.1	4.16
16	aut	$ndbh$			16	1.1	4.17
17	win	dbh			18	1.1	4.18
18	win	$ndbh$			14	1.1	1.1

$\{Mar, Apr, May\}$, $sWIN$ = $\{Jun, Jul, Aug\}$, $sSPR$ =$\{Sep, Oct, Nov\}$; wd = 'workday', wk = 'weekend' dbh = 'during business hours', $ndbh$ = 'not during business hours'.

For intuition, rule number 13 in Tab. 5, can be read as: if the season (*aSE*) is Summer (*sum*) and the operation (*aOP*) is during business hours (*dbh*), the temperature should be set to 24, and next rule to be executes is the first one in table 1 (1.1); moreover, if rule 13 fails, the next rule to be examined is the one having number 14 in table 4. For simplicity, the original enumeration of the eighteen rules was kept.

Basing on the idea of XT, the *eXtended Tabular Tree (XTT)* knowledge representation has been developed ([Nal04]). It allows for a hierarchical visual representation of the XT tables linked into tree-like structure, according to the control specification provided in the *Ctrl* column. XTT as a design and knowledge representation method offers transparent, high density knowledge representation as well as formally defined logical, PROLOG-based interpretation, while preserving flexibility with respect to knowledge manipulation.

A prototype CASE tool for the XTT method called MIRELLA [Nal04] has been developed. It supports XTT-based visual design methodology, with an integrated, incremental design and implementation process, providing the possibility of the on-line, incremental, verification of formal properties. Logical specification is directly translated into PROLOG-based representation providing an executable prototype, so that system operational semantics is well-defined. In MIRELLA the specification looks as in Fig. 1.

3 Verification of XTT Components

Within the proposed approach verification of the following theoretical properties is performed:

redundancy – subsumption of rules,

indeterminism – overlapping rules,

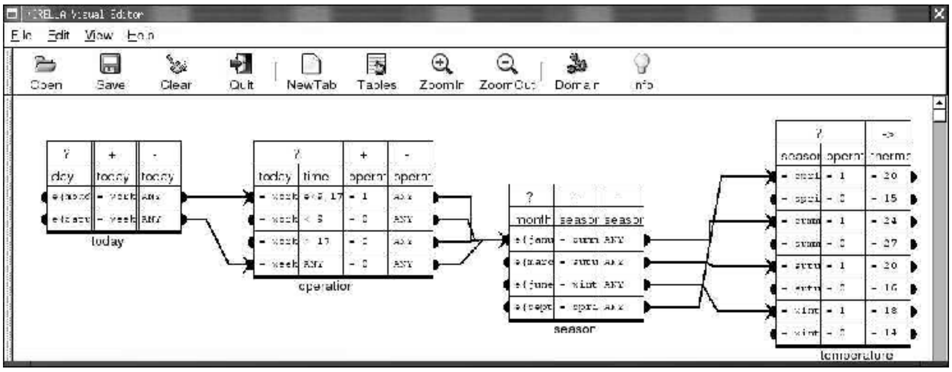

Figure 1: Thermostat system design in MIRELLA

completeness – missing rules.

Additionally, the components are checked if they are minimal and reduction possibilities are suggested. Reduction of an XT component is performed by gluing two (or more) rules having identical conclusions.

The analysis of subsumption is performed as follows. Consider two rules, r and r' given below (with a simplified XAT scheme):

$rule$	A_1	A_2	$\ldots$	A_j	$\ldots$	A_n	H
r	t_1	t_2	$\ldots$	t_j	$\ldots$	t_n	h
r'	t'_1	t'_2	$\ldots$	t'_j	$\ldots$	t'_n	h'

The condition for subsumption in case of tabular rule format takes the algebraic form $t'_j \subseteq t_j$, for $j = 1, 2, \ldots, n$ and $h' \subseteq h$. If it holds, then rule r' can be eliminated leaving the more general rule:

$rule$	A_1	A_2	$\ldots$	A_j	$\ldots$	A_n	H
r	t_1	t_2	$\ldots$	t_j	$\ldots$	t_n	h

For example, in the following tabular system the first rule subsumes the second one:

$rule$	A_1	A_2	A_3	A_4	H
r	7	$[2, 9]$	$[3, 5]$	$\{r, g, b\}$	$\{a, b, c\}$
r'	7	$[3, 5]$	4	$\{b, r\}$	$\{a, c\}$

Hence, rule r' can be eliminated. In the example case of the Thermostat specification there are no subsumed rules.

Note that the definition of subsumption is quite a general one. Checking for subsumption, defined as above, covers also detection and elimination of identical rules and equivalent rules; moreover, it is performed with purely algebraic means. More details can be found in [Lig05].

The analysis of indeterminism in case of XTT is also almost straightforward; in order to have two rules applicable in the same context, their preconditions must have non-empty intersection. In case of tabular systems this can be expressed as follows. For any attribute A_j there is an atom of the form $A_j = t_j$ in r and $A_j = t'_j$ in r', $i = 1, 2, \ldots, n$. Now, one has to fi nd the intersection of t_j and t'_j — if at least one of them is empty (e.g. two different values; more generally $t_{1,j} \cap t_{2,j} = \emptyset$), then the preconditions are disjoint and thus the rules are deterministic. The check is to be performed for any pair of rules. In the example case of the Thermostat specifi cation there are no indeterministic rules.

In design of knowledge rule-based systems one can encounter further theoretical problems; two most important ones following from the lack of determinism are *conflict* when two (or more) rules are applicable to the same input situation but the results are conflicting (under the assumed interpretation) and *inconsistency* when purely logical inconsistency occurs.

Note that problems of *conflicting* and *inconsistent* rules are specifi c cases of indeterminism. Detection of indeterminism is a necessary condition for eliminating conflict and inconsistency. Moreover, in tabular systems with no explicit negation purely logical inconsistency cannot occur; it always follows from the intended interpretation and thus it falls into the class of conflicts.

Reduction of XT is performed through gluing rules having identical conclusion part. Several rules can be glued to a single, equivalent rule according to the following scheme:

$rule$	A_1	A_2	$\ldots$	A_j	$\ldots$	A_n	H
r^1	t_1	t_2	$\ldots$	t_{1j}	$\ldots$	t_n	h
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$
r^k	t_1	t_2	$\ldots$	t_{kj}	$\ldots$	t_n	h
$rule$	A_1	A_2	$\ldots$	A_j	$\ldots$	A_n	H
r	t_1	t_2	$\ldots$	T	$\ldots$	t_n	h

provided that $t_{1j} \cup t_{2j} \cup \ldots \cup t_{kj} = T$. If T is equal to the complete domain, then $T = \_$. Of course, the rules $r^1, r^2, \ldots, r^k$ are just some selected rows of the original table containing all the rules. The logical foundations for reduction are covered in [Lig05]. In the example system, rules 4 and 5 of table 2 can be glued, provided that the time specifi cation can be expressed with non-convex intervals (i.e. [00:00-09:00]∪[17:00-24:00]). Further, rules 11 and 15 of table 4 can be glued to a single rule (in this case the preconditions would read $aSE \in \{spr, sum\} \wedge aOP = dbh$).

Completeness verifi cation in the XTT-based systems can be viewed as a two-stage procedure. First some maximal reduction is performed on the precondition part of a selected table. In the ideal case an empty table (full logical completeness) is confi rmed; in fact, any set of input values will be served. In other case one has to check which input specifi - cations are not covered. Here, thanks to allowing for non-atomic values of attributes it is not necessary to go through the list of all possible atomic combinations, i.e. performing the so-called *exhaustive enumeration check*, i.e. analysis of all the elements of Cartesian product of the precondition attribute domains. In the proposed approach the attribute domains can be divided into subsets (granularized) corresponding to the values occurring in the table; hence the check is performed in a more abstract level and with increased effi -

ciency. Uncovered input specifications define the potentially missing rule preconditions. The system is complete in the sense that there are no admissible (correct) inputs which are uncovered.

In the presented tool all of the above checks are implemented as transparent PROLOG [CNV97] modules and are used as plug-ins for the main system [Nal04]. Since verification of the above characteristics can be performed for partially specified tables, verification can be integrated with the design procedure and performed on-line whenever necessary.

4 Integration of Design and Verification

The proposed XTT approach and the visual tool introduces strong structurization of the design process. Further, at any stage of partially designed system any XTT component can be verified and corrected. In fact, integration of design and on-line verification is one of the crucial novel ideas of the presented approach. Simultaneously, this is a top-down approach, which allows for incorporating hierarchical design. Using XTT as a core, in [Nal04] an integrated design process, covering the following phases has been presented:

1. *Conceptual modeling*, in which system attributes and their functional relationships are identified;
2. *Logical design with on-line verification*, during which system structure is represented as XTT hierarchy, which can be instantly analyzed, verified (and corrected, if necessary) and even optimized on-line; and
3. *Physical design*, in which a preliminary Prolog-based *implementation* is carried out.

Using the predefined XTT translation it is possible to automatically build a prototype. It uses PROLOG-based meta-language [CNV97] for representing XTT knowledge base and rule inference (also referred to as XTT-PROLOG).

One of the most important features of this approach is the *separation of logical and physical design*, which also allows for a transparent, *hierarchical* design process. The hierarchical conceptual model is mapped to modular logical structure. The approach addresses three main problems: visual representation, functional dependency and logical structure, and machine readable representation with automated code generation.

In the XTT-based approach the verification can be performed on-line, as an internal part of the design process. At any design stage any XTT component (extended table) can be analyzed. The analysis of a selected property is performed by external PROLOG-based plugins. The results of the analysis can be instantly used to improve the design of the given XTT table. In the current version of the MIRELLA system the verification modules operate taking as the input the state of the knowledge-base and return the diagnosis in the form of a report.

In the current version of MIRELLA the designed system is verified against the anomalies discussed in Section 3: subsumption of rules, indeterminism and incompleteness. In fact, these issues are generic and cover a number of more specific problems, such as rule equivalence, inconsistent rules, etc. Moreover, reduction to minimal form through gluing of table rows with *backward dual resolution* [Lig05] is supported.

5 Related Work

Numerous research have been undertaken in the domain of verification of rule-based systems. Some best known results are recapitulated in a comprehensive book edited by J. Liebowitz [Lie98]. A recent book focused on verification and validation of knowledge-based systems is [VC99]. A number of tools is listed in [Lig05].

A classical work on verifying logically specified rule-based systems are the one of Preece [PBS92]; a more recent paper concerns a method for structure-based testing of rule-based systems [PGCR98]. Some interesting recent report on the VALENS system are presented in [GS00]; it is one of rare tools incorporating verification capability; however, it follows the classical approach where the verification is performed off-line, for a completed knowledge base. No details on the language and its expressive power nor about the technical aspects of verification were reported in [GS00].

Some recent interest in the so-called business rules and their applications are presented in [KSGJ04], [Wag98]. State-of-the-art concerning an interesting project related to rule-based systems for web technologies *REWERSE*[1] is summarized in [Bea04].

With respect to the above-mentioned research, as well as many other approaches ([Lie98, Hop01, Jac99, HS02, BM04]), the distinguished features of the presented in this paper proposal are:

- a precisely defined algebraic language for knowledge representation based on attributive logic with set values,
- a very powerful knowledge representation tool in the form of XTT structures for really *visual knowledge representation*, enabling visual design,
- a clear solution for efficient control of forward-chaining rules,
- verification integrated with the design process,
- intrinsically hierarchical and modular approach.

MIRELLA is an experimental tool of open architecture aimed at illustrating the advanced and innovative features of the proposed approach.

[1]See: `http://rewerse.net`

6 Concluding Remarks

The paper presents an outline of a new approach to design and verifi cation of rule-based systems. It is argued that a reasonable solution should consists in an integrated design and verifi cation procedure allowing for on-line, verifi cation of partially designed system.

The presented concept of tabular systems (XTT) seems to provide a new quality in knowledge representation. Simultaneously, it constitutes a step on the way to the *algebraization* of knowledge which seems important both for effi ciency reasons and making the approach close to engineering practice. XTT offers possibility of *visual knowledge representation* which seem very important for practical applications. It also incorporates possibility of *hierarchical* knowledge representation and *hierarchical development* of a rule-based systems. Finally, it enables interleaving the stages of *verification* and *design*, so that a possibly correct system is designed and developed. The proposed experimental tool integrates these features in a single system and enables far going support of development of rule-based systems.

A presentation of the XTT approach covering a simple yet non-trivial example should serve as illustration of the proposed ideas. An experimental version of the MIRELLA tool was implemented and tested on several examples ([Nal04]) of wide classes of RBS, including decision support and web security systems ([NL05]).

It seems that further studies can be directed towards further automatization of construction of rule-based systems, especially combining development of such systems with learning and automatic knowledge acquisition techniques. An interesting application areas are the one of semantic web and development of 'intelligent electronic documents' incorporating rule-based component for knowledge specifi cation and execution and capable of automatic knowledge processing.Possible application of the rule-based system methods and technology to knowledge management and development of more elaborated knowledge representation and processing formalism, e.g. ones based on the concept of granularity can also be considered. Finally, extension of the knowledge representation formalism towards covering fuzzy specifi cation of knowledge is planned.

References

[Bea04] Piero A. Bonatti and et al. Rule-based policy specification - state of the art and future work. Technical report, REWERSE, 2004.

[BM04] Valerie Barr and Zdravko Markov, editors. *Proceedings of the Seventeenth International Florida Artifi cial Intelligence Research Society Conference, Miami Beach, Florida, USA*. AAAI Press, 2004.

[CNV97] Michael A. Covington, Donald Nute, and André Vellino. *Prolog programming in depth.* Prentice-Hall, 1997.

[GS00] Rik Gerrits and Silvie Spreeuwenberg. VALENS: A Knowledge Based Tool to Validate and Verify an Aion Knowledge Base. In *ECAI*, pages 731–738, 2000.

[Hop01] Adrain A. Hopgood. *Intelligent Systems for Engineers and Scientists*. CRC Press, Boca Raton London New York Washington, D.C., 2nd edition, 2001.

[HS02] Susan M. Haller and Gene Simmons, editors. *Proceedings of the Fifteenth International Florida Artifi cial Intelligence Research Society Conference, May 14-16, 2002, Pensacola Beach, Florida, USA*. AAAI Press, 2002.

[Jac99] Peter Jackson. *Introduction to Expert Systems*. Addison–Wesley, 3rd edition, 1999.

[KSGJ04] Rainer Knauf, Silvie Spreeuwenberg, Rik Gerrits, and Martin Jendreck. A Step out of the Ivory Tower: Experiences with Adapting a Test Case Generation Idea to Business Rules. In *FLAIRS Conference*, 2004.

[Lie98] Jay Liebowitz, editor. *The Handbook of Applied Expert Systems*. CRC Press, Boca Raton, 1998.

[Lig96] Antoni Ligęza. Logical support for design of rule-based systems. Reliability and quality issues. In M.C. Rousset, editor, *ECAI-96 Workshop on Validation, Verifi cation and Refi nment of Knowledge-based Systems*, volume W2. ECAI'96, Budapest, 1996.

[Lig05] Antoni Ligęza. *Logical Foundations for Rule-Based Systems*. AGH University of Science and Technology Press, Kraków, 2005.

[LWN01] Antoni Ligęza, Igor Wojnicki, and Grzegorz J. Nalepa. Tab-Trees: a CASE tool for design of extended tabular systems. In H.C. Mayr et al., editor, *Database and Expert Systems Applications*, volume LNCS 2113 of *Lecture Notes in Computer Sciences*, pages 422–431. Springer-Verlag, Berlin, 2001.

[Nal04] Grzegorz J. Nalepa. *Meta-Level Approach to Integrated Process of Design and Implementation of Rule-Based Systems*. PhD thesis, AGH University of Science and Technology, AGH Institute of Automatics, Cracow, Poland, September 2004.

[Neg02] Michael Negnevitsky. *Artifi cial Intelligence. A Guide to Intelligent Systems*. Addison-Wesley, Harlow, England; London; New York, 2002.

[NL05] Grzegorz J. Nalepa and Antoni Ligęza. Security Systems Design and Analysis Using an Integrated Rule-Based Systems Approach. In Piotr Szczepaniak, Janusz Kacprzyk, and Adam Niewiadomski, editors, *Advances in Web Intelligence: 3rd Atlantic Web Intelligence Conference AWIC 2005*, volume LNAI 3528, pages 334–340. Springer-Verlag, 2005.

[PBS92] Alun D. Preece, R. D. Bell, and Ching Y. Suen. Verifying Knowledge-Based Systems Using the COVER Tool. In *IFIP Congress (3)*, pages 231–237, 1992.

[PGCR98] Alun D. Preece, Clifford Grossner, P. Gokul Chander, and Thiruvengadam Radhakrishnan. Structure-Based Validation of Rule-Based Systems. *Data Knowl. Eng.*, 26(2):161–189, 1998.

[SG02] Silvie Spreeuwenberg and Rik Gerrits. Requirements for Successful Verification in Practice. In *FLAIRS Conference*, pages 221–225, 2002.

[VC99] Anca Vermesan and Frans Coenen, editors. *Validation and Verifi cation of Knowledge Based Systems. Theory, Tools and Practice*. Kluwer Academic Publisher, Boston, 1999.

[Wag98] Gerd Wagner. *Foundations of Knowledge Systems with Applications to Databases and Agents*. Kluwer Academic Publishers, 1998.