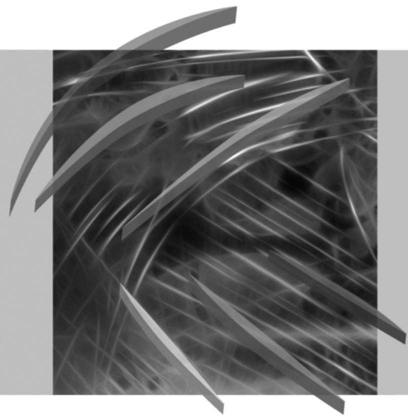
PRACE NAUKOWE
Uniwersytetu Ekonomicznego we Wrocławiu
RESEARCH PAPERS
of Wrocław University of Economics

232

Knowledge Acquisition and Management



edited by
Małgorzata Nycz
Mieczysław Lech Owoc

Publishing House of Wrocław University of Economics
Wrocław 2011

Reviewers: Grzegorz Bartoszewicz, Witold Chmielarz, Halina Kwaśnicka,
Antoni Ligęza, Stanisław Stanek

Copy-editing: Marcin Orszulak

Layout: Barbara Łopusiewicz

Proof-reading: Barbara Łopusiewicz

Typesetting: Beata Mazur

Cover design: Beata Dębska

This publication is available at www.ibuk.pl

Abstracts of published papers are available in the international database
The Central European Journal of Social Sciences and Humanities
http://cejsh.icm.edu.pl and in The Central and Eastern European Online Library
www.ceeol.com as well as in the annotated bibliography of economic issues BazEkon
http://kangur.uek.krakow.pl/bazy_ae/bazekon/nowy/index.php

Information on submitting and reviewing papers is available
on the Publishing House's website www.wydawnictwo.ue.wroc.pl

ISSN 1899-3192
ISBN 978-83-7695-200-0

The original version: printed

Printing: Printing House TOTEM

Contents

Preface 7

Iwona Chomiak-Orsa: Selected instruments of controlling used in the area of knowledge management 9
Roman V. Karpovich: Creating the portfolio of investment projects using fuzzy multiple-criteria decision-making 19
Jerzy Korczak, Marcin Iżykowski: Approach to clustering of intraday stock quotations 29
Antoni Ligęza: A note on a logical model of an inference process. From ARD and RBS to BPMN 41
Maria Mach: Analysing economic environment with temporal intelligent systems: the R-R-I-M architecture and the concept of quasi-objects 50
Alsqour Moh'd, Matouk Kamal, Mieczysław L. Owoc: Integrating business intelligence and theory of constraints approach 61
Eunika Mercier-Laurent: Future trends in knowledge management. Knowledge EcoInnovation 70
Małgorzata Nycz: Business intelligence in Enterprise 2.0 79
Mieczysław L. Owoc: Key factors of Knowledge Grid development 90
Maciej Pondel: Data mining with Microsoft SQL Server 2008 98
Maria Radziuk: Multi-agent systems for electronic auctions 108
Tatiana V. Solodukha, Boris A. Zhelezko: Developing a multi-agent system for e-commerce 117
Jerzy Surma: Case-based strategic decision-making 126
Paweł Weichbroth: The visualisation of association rules in market basket analysis as a supporting method in customer relationship management systems 136
Radosław Wójtowicz: Office online suits as a tool for supporting electronic document management 146
Radosław Zatoka, Cezary Hołub: Knowledge management in programming teams using agile methodologies 156

Presentations

Markus Helfert: Current und Future "Trends" in Knowledge Management – A management capability perspective 167
Eunika Mercier-Laurent: Knowledge EcoInnovation 181

Streszczenia

Iwona Chomiak-Orsa: Wybrane instrumenty controllingu wykorzystywane w obszarze zarządzania wiedzą 18
Roman V. Karpovich: Tworzenie portfela projektów inwestycyjnych przy użyciu wielokryterialnych rozmytych metod podejmowania decyzji 28
Jerzy Korczak, Marcin Iżykowski: Próba klasteryzacji dziennych notowań giełdowych........ 40
Antoni Ligęza: Uwaga na temat logicznych modeli procesu wnioskowania. Od ARD i RBS do BPMN 49
Maria Mach: Analiza środowiska ekonomicznego przy pomocy inteligentnych systemów temporalnych – architektura R-R-I-M i koncepcja quasi-obiektów 60
Alsqour Moh'd, Matouk Kamal, Mieczysław L. Owoc: Integracja *business intelligence* z teorią ograniczeń 69
Eunika Mercier-Laurent: Przyszłe trendy w zarządzaniu wiedzą. Ekoinnowacje wiedzy 78
Małgorzata Nycz: *Business intelligence* w koncepcji *Enterprise* 2.0 89
Mieczysław L. Owoc: Kluczowe czynniki rozwoju *Knowledge Grid* 97
Maciej Pondel: Drążenie danych w MS SQL Server 2008 107
Maria Radziuk: Wieloagentowy system wspierający aukcje elektroniczne... 116
Tatiana V. Solodukha, Boris A. Zhelezko: Budowa systemów wieloagentowych na potrzeby handlu elektronicznego 125
Jerzy Surma: Podejmowanie strategicznych decyzji w oparciu o analizę przypadków........ 135
Paweł Weichbroth: Wizualizacja reguł asocjacyjnych w analizie koszykowej jako metoda wspierająca systemy klasy CRM 145
Radosław Wójtowicz: Pakiety biurowe *on-line* jako narzędzia wspierające zarządzanie dokumentami elektronicznymi 155
Radosław Zatoka, Cezary Hołub: Zarządzanie wiedzą w zespołach programistycznych przy użyciu metodyk zwinnych........ 164

PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU nr 232
RESEARCH PAPERS OF WROCŁAW UNIVERSITY OF ECONOMICS
Knowledge Acquisition and Management ISSN 1899-3192

Antoni Ligęza
AGH University of Science and Technology in Kraków

A NOTE ON A LOGICAL MODEL OF AN INFERENCE PROCESS. FROM ARD AND RBS TO BPMN[1]

Summary: An interesting question arises when solving the problem of the way of applying given knowledge in order to obtain a solution in an efficient way. Since the knowledge at hand is often formulated as a set of rules, a classical approach consists in applying a forward or backward chaining inference engine. Various techniques of inference control aimed at improving efficiency, ranging from simple rule ordering to advanced Rete-type engines are used. However, in the case of complex decision support or business rules such blind techniques may become inefficient. In this paper we argue that intelligent inference control is a key issue for efficient problem solving. The model of logical inference is studied in some details and some new aspects of its structure and components are put forward.

Keywords: rule-based systems, inference control, model of inference, BPMN.

1. Introduction

Intelligent reasoning has been a focus of research interest for ages. Traditionally, inference was studied in the domain of logic, but other sciences, ranging from philosophy and psychology to modern Artificial Intelligence (AI) and Knowledge Engineering (KE), also do investigate models of inference.

In traditional logic three basic inference paradigms are dominant; these are deduction, induction, and abduction. In classical Rule-Based Systems, forward or backward chaining is a prevailing strategy. However, in more complex systems, such as CYC or in the LarKC Projects, the repository of inference methods is wider. One can assume that in modern AI over twenty different inference paradigms are in use.

An interesting question arises when problem solving is *how to apply given knowledge* to the data and problem at hand in order to obtain the *right solution* in an *efficient way*. It can be observed that various people and computer systems, being provided the same knowledge and problems to be solved, yield different solutions in

[1] The research is carried out within Bimloq research project No. N N561 422338, supported by the Ministry of Science and Higher Education.

different processing time. The provided solution may different with respect to quality and level of detail.

A brillant example is the person of Sherlock Holmes, who given only few facts and observations was able to draw detailed, far going conclusions.[2] Since in most cases the knowledge at hand is often formulated as a set of rules, a classical approach consists in applying a forward or backward chaining inference engine. Various techniques of inference control aimed at improving efficiency, ranging from simple rule ordering to advanced Rete-type engines, are used. However, in the case of complex, decision support or Business Rules (BR) such blind techniques may become inefficient. In this paper we argue that intelligent inference control is a key issue for efficient problem solving. Elements of the model of logical inference are studied in some details and some new aspects of its structure and components are put forward.

Rule-Based Systems (RBS) constitute a powerful tool for the specification of knowledge in design and implementation of knowledge-based systems (KBS) in applied Artificial Intelligence and Knowledge Engineering. They provide also a universal programming paradigm for domains such as system monitoring, intelligent control, decision support, situation classification, system diagnosis, and operational knowledge encoding. Apart from off-line expert systems and deductive data-bases, one of the most useful and successful applications consists in development of wide spectrum of control and decision support systems. Some features of modern rule-based systems decisive for success in sophisticated applications include:

- possibility of defining complex preconditions and conclusions (depending on a language in use);
- ability to specify dynamic shaping of knowledge in the knowledge-base (with use of the *retract* and *assert* predicates);
- incorporation of arbitrarily complex inference control mechanism;
- possibility of hierarchical knowledge encoding and operation, and last but not least;
- capability of automated verification of knowledge specification.

Furthermore, although the rule-based programming paradigm seems relatively conceptually simple, in the case of realistic systems it is a hard and tedious task to design. One of the crucial issues is how to control the inference having hundreds or thousands of rules at hand.

The main contribution of this paper is the proposal of introducing a *declarative inference control strategy specification*. Contrary to most of the current approaches, we assume that both domain-specific knowledge as well as the meta-knowledge on *how to use the domain knowledge* must be provided in a declarative way. In fact, a new level of specification of the way knowledge is to be used in contrast to purely mechanical, blind inference control engines/strategies is put forward.

[2] Note that in fact, from the logical point of view, contrary to the original claims by Sir Arthur Conan Doyle, Sherlock Holmes was mostly using *abduction* instead of *deduction*.

2. State-of-the-art in inference control

In order to build an intelligent system for problem solving, the following three basic problems must be solved:
- selection of appropriate *knowledge representation language*;
- selection of *inference paradigms*;
- efficient solution of *inference control*.

The first problem is typically solved by adapting some logical formalism. A most typical solution consists in using an Attribute Logic (AL) or a subset of First Order Logic (FOL). Some more recent trends consists in application of various Description Logics (DL). Static knowledge is represented as facts, while operational knowledge takes the form of rules. Many of the Knowledge-Based Systems follow the RBS paradigm.

The second problem – especially in the case of RBS – has two basic solution, i.e., applying *forward chaining* or *backward chaining paradigm*. In the case of forward chaining the logical bases are constituted by the *modus ponens* inference rule. In the case of backward chaining it can be the *resolution rule* (as it is in case of the Prolog language) or abduction-type inference (as in the case of some diagnostic systems). Moreover, some system are capable of performing mixed, combined forward and backward inference.

Having the Knowledge Base (KB), the main problem consists in an efficient application of the rules to generate the solution. An inference control strategy must be provided. One of the basic approaches consists in determining a set of applicable rules – the so-called *conflict set* – and finding in it the most preferred rule to be fired. For conflict resolution various strategies can be applied, e.g.:
- static *vs*. dynamic strategies; static strategies are based on criteria constant over time, while dynamic ones can take into account current context, time, number of (successful) repetitions of a rule, etc.;
- syntactic *vs*. semantic strategies; the first one is based on the “shape” of the rule preconditions, while the second ones may take into account the current context, desired goal, and evaluable user-specified criteria;
- direct *vs*. indirect strategies; the direct ones are based on simple comparison of rules and assigned to them “ordering factors”, e.g., priorities, while the indirect can be implemented with an auxiliary knowledge-based system;
- meta-rules and complex inference schemes;
- strategies based on simple, constant criteria vs. ones modifiable/adaptable;
- learning.

Some more advanced approaches try to avoid matching of all the rules during each cycle. A core idea is based on the assumptions that rule preconditions are structurally similar and during a cycle only a small part of the fact base changes. Hence efficient indexing and propagation of these changes can save much effort.

This is the basic idea behind algorithms such as Rete, Treat, or Gator. Such mechanism are used in modern RBS such as Clips, Jess, or Drools.

The main problem with current inference engines is that they are in fact blind. Some of the undesirable consequences are as follows:

- excessive number of unnecessary facts are produced;
- computation or inference can be unnecessarily repeated;
- the inference process can be directed into a blind alley or and infinite inference branch;
- in the case of inconsistent knowledge inconsistent conclusions can be drawn;
- in the case of indeterministic knowledge a random solution can be obtained.

In the eXtended Tabular Trees (XTT) paradigm, in order to avoid such problems and make the inference efficient, three inference control strategies are proposed:

- fixed-order of groups of rules;
- token-transfer approach;
- goal-driven approach.

These strategies are aimed at improving inference efficiency and assuring that the inference process will stop with producing the solution.

A current graphical method for specification of decision processes in business is the BPMN graphical language [Juric, Pant 2008]. It seems that it has some potential not only for graphical definition of business procedures, but it can be expanded to provide possibilities of declarative specification of inference control for complex processes. As the BPMN can be translated to BPEL (Business Process Execution Language), the formal aspects of control specification seems to be solved in a satisfactory way.

3. Basic assumptions

The basic assumption follow from the observation of human flexibility in problem solving and the examination of several AI inference paradigms. To a certain degree, they are also inspired by the analysis of problems encountered by classical inference engines.

The very first observation is that in practical systems there are in fact very different types of rules. Such rules should be used in different ways. A simple taxonomy for rules in use in Decision Support Systems (DSS) presented there was as follows:

- deductive rules: rules for production of new facts;
- extension rules: rules specifying universal properties of facts (a kind of ontology);
- numerical rules: rules specifying computational dependencies;
- testing rules: rules for refining the KB and detection of inconsistency.

Moreover, reasoning control rules were proposed for guiding the inference process, depending on current task, status, and mode of work.

In contemporary BR systems the taxonomy of rules typically includes:
- facts: rules defining true statement (with no conditional part);
- definition rules: for defining terms and notions in use;
- integrity rules: rules defining integrity constraints;
- production rules: for derivation of new facts;
- reaction rules: rules triggered by events, reactive rules, or ECA rules;
- transformation rules: rules defining possible transformations, term-rewriting rules; they may include numerical recipe rules;
- control rules: in fact meta rules used for inference process control.

For each type of rule, a specific way of using it should be defined.

A second important observation is that rules should be fired only if they are necessary to achieve a goal or enable firing rules that lead to achieving the current goal. Hence, a planning process for reasonable way of arriving at the goal should take place. This can be done by hand with use of a graphical language, such as BPMN, decision trees, decision graphs, flow charts, etc. or by a search procedures, such as DF, ID, A*, etc.

Third, rules are usually fired within a *specific context*. Rules of similar application area are to be grouped together forming a kind of decision component with a clearly defined context of work, input data, and output data. In XTT, for example, we have in fact extended attributive logic decision tables; related rules are placed together. For each such group specific inference control must be defined.

Fourth, the same knowledge can be used in different ways. This means also that rules can be used in parallel. One must define possible splits and their properties, and possible meets and way of amalgamating the results.

Five, certain procedures can require cyclic operation. Hence, loops must be possible to define, as well as exit conditions must be specified.

Six, various auxiliary inference modes, such as search, optimisation, case-based reasoning, etc. must be served in an appropriate way.

Finally, seven, time factor must be taken into account. Both absolute time (timestamps) and relative time (time delays) should be possible to specify and serve.

4. Towards development of logical inference model

In order to develop an efficient inference process, its model must be defined so that it offers solutions to the aforementioned problems. With respect to the most common types of rules, an outline of the recipes for use may be as follows:
- facts: true fact are stored within a fact base; negative fact, if present, are stored as well and internal consistency mechanism is activated whenever a new fact is added;
- definition rules: they are activated only when a check (for qualification) is necessary;

– integrity rules: they are used when new facts are generated; in the case of inconsistency detection backtracking must be enforced;
– production rules: these are the main rules; declarative specification of inference should be provided;
– reaction rules: a monitoring system (watch-dog) must be supplemented;
– transformation rules: they are used when matching of incompatible objects is necessary (a kind of a subprocess);
– control rules: they can be implemented as part of the inference control mechanism.

As for the observation that rules should be fired only if they are necessary to achieve the goal or enable firing rules that lead to achieving a current goal, two approaches are suggested. The first consists in using a graphical language, such as BPMN; in fact, we start from Attribute-Relationship Diagrams (ARD) and then develop BPMN diagram. Alternatively, some automated search procedures, such as DF, ID, A* can be applied.

Rules of similar application area are grouped together forming a kind of decision component with clearly defined context of work, input data and output data. We can use XTT, for example, for each such group specific inference control must be defined, i.e.:
– Is the table scanned once, several times, or is it scanned repeatedly until a stopping condition becomes true?
– During a single scan, is it only the first rule with satisfied preconditions to be fired, or are all such rules fired?
– what to do if no rule during a scan was fired: stop, exit with no added knowledge or perhaps some predefined default values, or repeat the scan?
– What to do when the last rule in a table was examined and fired (or not)?

Consider a single step of its operation consisting of examining a certain rule, firing it if its preconditions are satisfied, and passing the control to the next rule or somwhere else.

When examining the current rule, the following knowledge must be available:
– if the preconditions (LHS) are satisfied (yes) or not (no);
– if this is the last rule (LAST) in the table (yes) or some middle rule (no);
– if the next table(s)/rule(s) (NEXT) are specified (yes) explicitly or no explicit specification is provided (no).

Further to this, the following two meta-issues concerning properties of the whole table rather than sigle rules must be resolved:
– if only single, first rule with satisfied preconditions should be fired (FIRST) or all of them having satisfied preconditions (ALL);
– if the table is subject to a single scan (SINGLE) or repeated one (REPEAT); in the latter case an leaving condition must be satisfied (UNTIL).

Note that in fact we have as many as 32 potential possibilities.

As the same knowledge can be used in different ways, one must take care of the rules to be used in parallel. Possible splits (of the type AND, OR, ONE-OF) must be defined and ways of joining and amalgamating the results.

The problem of loops and auxiliary inference modes in not discussed here. In general, one can imagine specific components with counters or conditional switching for loops and black-box components for independent implementation of auxiliary modes of inference.

The time factor can be taken into account in two dimensions. With definite, absolute time we consider the time-stamp as a special attribute (of composed values; as in Relational databases). With respect to time delays, we need to specify explicite time delays and precedence relation.

5. An example

In order to illustrate the ideas, consider an example of a rule based system. This is the thermostat for temperature setting depending on the time of year and working hours [Negnevitsky 2002]. Full specification of 18 rules is given below.

Rule 1. If the day is Monday or the day is Tuesday or the day is Wednesday or the day is Thursday or the day is Friday, then today is a workday.

Rule 2. If the day is Saturday or the day is Sunday, then today is the weekend.

Rule 3. If today is workday and the time is “between 9 am and 5 pm”, then operation is “during business hours”.

Rule 4. If today is workday and the time is “before 9 am”, then operation is “not during business hours”.

Rule 5. If today is workday and the time is “after 5 pm”, then operation is “not during business hours”.

Rule 6. If today is weekend, then operation is “not during business hours”.

Rule 7. If the month is January or the month is February or the month is December, then the season is summer.

Rule 8. If the month is March or the month is April or the month is May, then the season is autumn.

Rule 9. If the month is June or the month is July or the month is August, then the season is winter.

Rule 10. If the month is September or the month is October or the month is November, then the season is spring.

Rule 11. If the season is spring and operation is “during business hours”, then thermostat_setting is “20 degrees”.

Rule 12. If the season is spring and operation is “not during business hours”, then thermostat_setting is “15 degrees”.

Rule 13. If the season is summer and operation is “during business hours”, then thermostat_setting is “24 degrees”.

Rule 14. If the season is summer and operation is “not during business hours”, then thermostat_setting is “27 degrees”.

Rule 15. If the season is autumn and operation is “during business hours”, then thermostat_setting is “20 degrees”.

Rule 16. If the season is autumn and operation is “not during business hours”, then thermostat_setting is “16 degrees”.

Rule 17. If the season is winter and operation is “during business hours”, then thermostat_setting is “18 degrees”.

Rule 18. If the season is winter and operation is “not during business hours”, then thermostat_setting is “14 degrees”.

Note that the rules are grouped into four groups of specific rules. In each group we have similar rules. For each group only a single rules should be fired. The precedence of inference is defined with the ARD diagram shown in Figure 1.

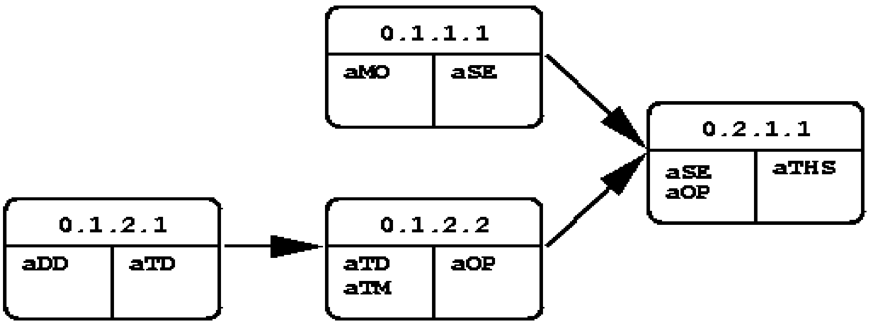


Figure 1. The ARD diagram for the termostat setting system

Source: author’s own study.

Here aMO defines the current month; on the basis of it the current season, aSE is inferred with rules 7-10. The aDD attribute defined current day, and on the basis of it, it is inferred if we have working day or a holiday with rules 1-2. Attributes aTD and TM (time of day) are used to infer whether we have working hours or not with

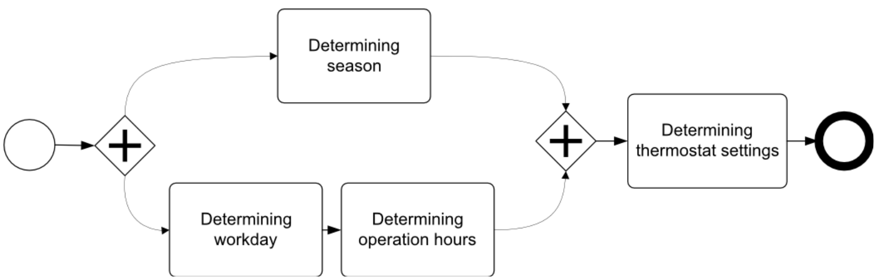


Figure 2. A BPMN inference control diagram for declarative specification of inference control for the thermostat setting system

Source: author’s own study.

rules 3-6. Finally, aSE (season) and aOP (operating hours) are used to define the exact set point.

Now, using the BPMN notation, we can specify the inference control diagram as in Figure 2.

6. Concluding remarks

The paper discusses inference problems in rule-based systems. The main idea consists in introducing declarative inference control specification and execution mechanism. Various details to be solved are mentioned and solution proposals are outlined. The BPMN language is proposed as a specification language form of inference control. The main focus of this paper is on the proposal of introducing a *declarative inference control strategy specification*. Contrary to most of the current approaches, we assume that both domain-specific knowledge as well as the meta-knowledge on *how to use the domain knowledge* must be provided in a declarative way.

References

Bratko I. (2001), *Prolog Programming for Artificial Intelligence*, Addison-Wesley.

Giarratano J., Riley G. (2005), *Expert Systems Principles and Programming*, Thomson Course Methodology.

Juric M.B., Pant K. (2008), *Business Process Driven SOA using BPMN and BPEL. From Business Process Modeling to Orchestration and Service Oriented Architecture*, PACKT Publishing.

Merrit D. (2000), *Building Expert Systems in Prolog*, Amzi! Inc.

Negnevitsky M. (2002), *Artificial Intellignece. A Guide to Intelligent Systems*, Addison Wesley.

Russel S., Norvig P. (2009), *Artificial Intellignece. A Modern Approach*, Pearson Eductaion International.

UWAGA NA TEMAT LOGICZNYCH MODELI PROCESU WNIOSKOWANIA. OD ARD I RBS DO BPMN

Streszczenie: Ciekawe pytanie powstaje, gdy rozwiązujemy problem, jak zastosować daną wiedzę w celu uzyskania rozwiązania w sposób efektywny. Ponieważ wiedza, którą mamy w zasięgu ręki, jest często formułowana jako zbiór reguł, klasyczne podejście opiera się na zastosowaniu silnika wnioskowania wprzód lub wstecz. Są stosowane różne techniki kontroli wnioskowania mające na celu poprawę efektywności, począwszy od prostej *Rule ordering* do zaawansowanych *Rete-type*. Jednak w złożonych przypadkach, wsparcia podejmowania decyzji na potrzeby reguł biznesowych takie ślepe techniki mogą stać się nieefektywne. W niniejszej pracy autorzy uważają, że inteligentna kontrola wnioskowania jest kluczem do efektywnego rozwiązywania problemów. Model logicznego wnioskowania jest badany w szczegółach i są przedstawione pewne nowe aspekty jego struktury i elementów.

Słowa kluczowe: systemy regułowe, kontrola wnioskowania, model wnioskowania, BPMN.