HeaRT rule inference engine in intelligent systems

Szymon Bobek
AGH University of Science and Technology, Department of Automatics

Abstract: Rules are one of the most important knowledge representation methods. Rule-based expert systems proved to be a successful AI technology in a number of areas. Building such systems requires creating a rule-base, as well as providing an effective inference mechanism that fires rules appropriate in a given context. Building and maintaining such a rule-base is a very difficult task and can be supported by a sepcially designed tools. Due to the fact that inference engines usually are parts of such tools it is difficult to integrate such expert systems with external software. In this paper a custom inference engine was presented, that implements three inference algorithms and can be easily integrated with other systems. Its usage in selected intelligent systems was also described.

Keywords: inference engine, rules, rule-based systems

Due to the fact that rules are very simple and easy to understand, they became a very powerful and popular method for knowledge representation [14]. Rules constitute a cardinal concept of the rule-based expert systems (RBS for short) [3, 9] which are widely used artificial intelligence systems [13]. Every rule-based expert system consist of at least two parts: 1) knowledge base (KB), where rules are stored and 2) inference engine, which is responsible for reasoning tasks. Language for representing KB and the inference algorithms are provided by an environment caled *shell*, that is designed to create, maintain and run expert systems.

The most popular expert systems shells are: Clips, Jess, Drools [1]. Although KB languages are different in that tools, all of them use unformalized production rules as a knowledge representation, and Rete based algorithms for reasoning tasks.

In this paper an overview of custom inference engine called HeaRT is presented. It is an open source solution that provides formal KB representation, different inference algorithms, and ca be embedded within other intelligent systems like Semantic Wikis.

1. Rule-based expert system shells

CLIPS is an expert system tool that is based on Rete algorithm that has been developed by NASA. It provides its own programming language that supports rule-based, procedural and object-oriented programming [3]. Thanks to this variety of programming paradigms implemented in CLIPS, there are three ways to represent knowledge in it:

- rules, which are primarily intended for heuristic knowledge based on experience,
- functions, which are primarily intended for procedural knowledge,
- object-oriented programming, also primarily intended for procedural knowledge.

CLIPS has been written in C language. This makes the tool very efficient and platform independent. However, the integration with other existing systems is not easy. Moreover it supports only forward chaining and very simple modularisation Jess is a rule engine and scripting environment written entirely in Sun's Java language by Ernest Friedman-Hill [2] that derives form CLIPS (*C Language Integrated Production System*).

Jess supports both *forward-chaining* and *backward chaining*. However designing and maintaining systems with CLIPS or Jess is very difficult and inefficient. Modularisation of KB in Jess does not improve efficiency. Moreover neither CLIPS nor Jess provides visual tools for designing expert systems. Drools 5 introduces the Business Logic integration Platform which provides a unified and integrated platform for Rules, Workflow and Event Processing. It uses Drools Expert inference engine that implements the Rete-based algorithm called ReteOO. Its main advantages is a visual representation of the inference process and easy integration with Java. The drawbacks of Drools are: not formal rule representation, no visual KB representation.

As an answer to the limitations of existing expert system shells a custom inference engine called HeaRT was developed.

2. HeaRT inference engine

HeKatE RunTime (HeaRT) [8] is a lightweight embeddable rule inference engine built as a part of the HeKatE project (See `http://hekate.ia.agh.edu.pl`.) [10]. The distinctive features of the HeaRT engine are the following:

- support for an expressive rule language (XTT2) that has a complete formal definition in the ALSV(FD) logic [7, 11],
- part of an expert system development platform that provides visual knowledge representation [4, 5]
- three custom inference algorithms: forward chaining, backward chaining and token driven inference
- rule base verification mechanisms called HalVA that allows for checking for logical completeness, and redundancy in the rule base, and
- lightweight and embeddable implementation using a fast Prolog compiler.

2.1. HeaRT in intelligent systems

HeaRT inference engine was used in a couple of project that aim was to developed an intelligent system based on rule-based knowledge. In this section those projects are briefly described.

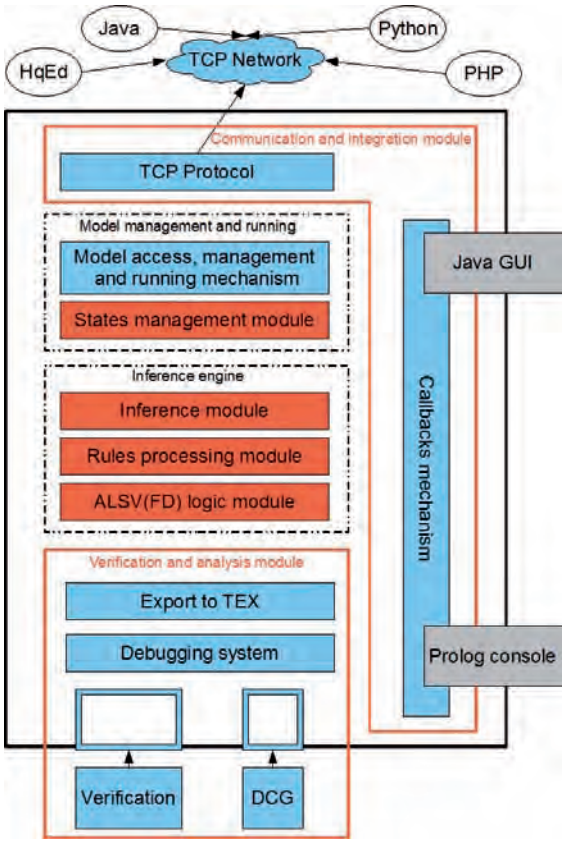

Fig. 1. Architecture of HeaRT
Rys. 1. Architektura silnika wnioskującego HeaRT

2.1.1. HeaRT in HeKatE

The main aims of the HeKatE project was to provide:

- an integrated design and implementation process,
- a visual representation of knowledge with a XTT decision tables,
- an on-line formal analysis of the design of rule-based systems.

HeaRT was used as an inference engine and verification tool that could work both: as a standalone application or as a inference server. An architecture of HeaRT is presented in Fig. 1.

2.1.2. HeaRT as a verification tool in Rebit

Rebit system that was developed to enhance business rules and processes management. It provides visual editors and several different methods of representing knowledge. A plugin to HeaRT, called HalVA was used in this project. It is a part of the inference engine that is responsible for verifying knowledge base in terms of logical completeness, contradiction and redundancy. The process of verification of the kowledge base is presented in Fig. 2.

2.1.3. HeaRT in Loki

Loki is a knowledge-based semantic wiki that combines flexible knowledge representation with strong reasoning

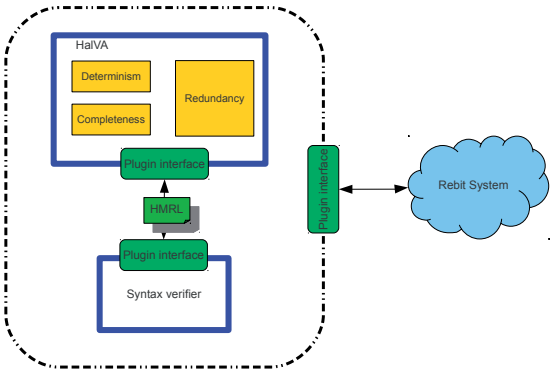

Fig. 2. Verification system in REBIT
Rys. 2. System weryfikacji w REBIT

mechanisms available due to the usage of HeaRT inference engine embedded withi it. On one hand, it is compatible with popular semantic annotations [12].On the other, it is integrated with a rule engine able to operate in various modes over modularized rule bases. The unified underlying representation makes it possible to process knowledge acquired in different ways. The process of rendering a

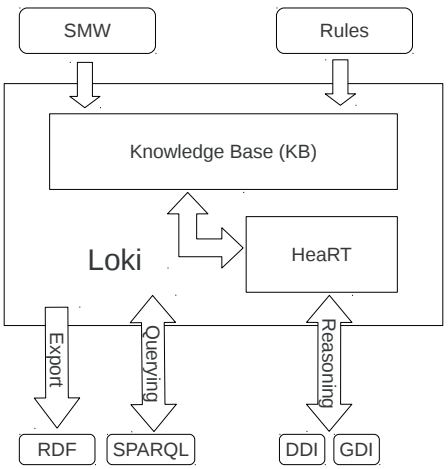

Fig. 3. Architecture of Loki with HeaRT
Rys. 3. Architektura Loki i HeaRT

wiki page in PlWiki with HeaRT looks as follows (see Fig. 3):

1) Wiki engine parses the Wiki page and extracts rules and reasoning queries (goals) for HeaRT.
2) Depending on a scope defined in the goal, PlWiki merges the HMR code from wiki pages in a given scope and passes it to HeaRT.
3) HeaRT performs the reasoning process and returns results to PlWiki engine.
4) PlWiki renders complete wiki page with previously parsed regular text and an answer to a given query (goal) produced by HeaRT.

3. Future work

HeaRT inference engine is still developed. Current focus is to embed HeaRT inference language in a BPMN design tool that would allow for running simulation of business

processes and verification of selected properties of such system [6].

Bibliography

1. Bobek S., Kaczor K., Nalepa G. J., *Overview of Rule Inference Algorithms for Structured Rule Bases*,.
2. Friedman-Hill E. (2003): *Jess in Action, Rule Based Systems in Java*, Manning.
3. Giarratano J. C., Riley G. D. (2005): *Expert Systems*, Thomson.
4. Kaczor K., Nalepa G. J. (2008): *Design and Implementation of HQEd, the Visual Editor for the XTT+ Rule Design Method*, Technical Report CSLTR 02/2008, AGH University of Science and Technology.
5. Kaczor K., Nalepa G. J. (2009): *HaDEs – Presentation of the HeKatE Design Environment*, [in]: Baumeister J., Nalepa G. J. (Eds.), 5th Workshop on Knowledge Engineering and Software Engineering (KESE2009) at the 32nd German conference on Artificial Intelligence: September 15, 2009, Paderborn, Germany, 57–62, Paderborn, Germany.
6. Kluza K., Maślanka T., Nalepa G. J., Ligęza A. (2011): *Representing BPMN Diagrams with XTT2-based Business Rules Proposal*, [in]: Brazier F. M., Nieuwenhuis K., Pavlin G., Warnier M., Badica C. (Eds.), Intelligent Distributed Computing V, Studies in Computational Intelligence, Springer-Verlag, in press.
7. Nalepa G., Ligęza A., Kaczor K. (2011a): *Overview of Knowledge Formalization with XTT2 Rules*, [in]: Bassiliades N., Governatori G., Paschke A. (Eds.), Rule-Based Reasoning, Programming, and Applications, *Lecture Notes in Computer Science*, vol. 6826, 329–336, Springer Berlin / Heidelberg.
8. Nalepa G. J. (2010): *Architecture of the HeaRT Hybrid Rule Engine*, [in]: Rutkowski L., [et al.] (Eds.), Artificial Intelligence and Soft Computing: 10th International Conference, ICAISC 2010: Zakopane, Poland, June 13–17, 2010, Pt. II, *Lecture Notes in Artificial Intelligence*, vol. 6114, 598–605, Springer.
9. Nalepa G. J. (2011): *Semantic Knowledge Engineering. A Rule-Based Approach*, Wydawnictwa AGH, Kraków.
10. Nalepa G. J., Ligęza A., *HeKatE Methodology, Hybrid Engineering of Intelligent Systems*,.
11. Nalepa G. J., Ligęza A., Kaczor K., *Formalization and Modeling of Rules Using the XTT2 Method*, In press.
12. Noga M., Kaczor K., Nalepa G. J., *Lightweight reasoning methods in selected Semantic Wikis*,.
13. Tadeusiewicz R. (2011): *Introduction to intelligent systems*, [in]: Wilamowski B. M., Irwin J. D. (Eds.), Intelligent systems, The Electrical Engineering Handbook Series. The Industrial Electronics Handbook, 1–1–1–12, Boca Raton; London; New York: CRC Press Taylor & Francis Group, second edition edition.
14. van Harmelen F., Lifschitz V., Porter B. (Eds.) (2007): *Handbook of Knowledge Representation*, Elsevier Science. ■

Wykorzystanie regułowego silnika wnioskującego HeaRT w systemach inteligentnych

Streszczenie: Reguły są jedną z najważniejszych metod reprezentacji wiedzy. Systemy ekspertowe oparte na regułowej bazie wiedzy są istotną technologią w zastosowaniach sztucznej inteligencji w różnych dziedzinach. Budowanie takich systemów wymaga stworzenia bazy wiedzy, jak również dostarczenia wydajnego mechanizmu pozwalającego na uruchamianie reguł i wnioskowanie. Budowanie bazy wiedzy i zarządzanie nią jest bardzo trudnym zadaniem. Może ono być wspierane poprzez narzędzia przeznaczone do tego celu, jednak ponieważ silniki wnioskujące zazwyczaj stanowa integralną część takich narzędzi, integracja ich z zewnętrznym oprogramowanie stanowi jeszcze trudniejsze zadanie. W niniejszym artykule opisany został nowy regułowy silnik wnioskujący, pozwalający na stosowanie kilku algorytmów wnioskowania i łatwą integrację z innymi systemami. Przedstawione zostało również jego zastosowanie w wybranych inteligentnych systemach komputerowych.

Słowa kluczowe: silniki wnioskowania, reguły, systemy regułowe

Szymon Bobek, MSc

Szymon Bobek, MSc holds a position of a research assistant at the AGH UST in Krakow, Poland, Department of Automatics. Since 2008 he has been actively involved in the international project HeKatE, where he was responsible for a design and implementation of a runtime for XTT2 rule representation method. This was also a core subject of his master thesis. After graduation from the university in 2009 he has continued his education and research as a member of the GEIST team.
e-mail: szymon.bobek@agh.edu.pl