

# The Reflexive System Inference Engine: A Tool to Use Metaknowledge

**Yann BARLOY**

Institut Charles Delaunay

Université de Technologie de Troyes,  
12 Rue Marie Curie, BP 2060,  
10010 Troyes Cedex, FRANCE  
yann.barloy@utt.fr

**Jean-Marc NIGRO**

Institut Charles Delaunay

Université de Technologie de Troyes,  
12 Rue Marie Curie, BP 2060,  
10010 Troyes Cedex, FRANCE  
nigro@utt.fr

## Abstract

This article deals with how metaknowledge can improve rule-based system and presents a new Reflexive System Inference Engine (RSIE) which not only enables the activation of rules, making it belong to systems managing metaknowledge. The experimentation section shows a rule-based system named IDRES with a structure which has been modified to use metaknowledge.

## I. Introduction

The domain of metaknowledge was conceived in the 1970s and 80s (Hayes 1973) at the same time as the emergence of rule-based systems. A metaknowledge can be defined as being knowledge about knowledge.

There is no specific architecture (or programming language) for manipulating metaknowledge. However, rules-based systems have the advantage of restoring for the building of different levels of knowledge (Genesereth and Nilsson 1987)

Due to this context, the idea to design a new inference engine, called RSIE (Reflexive System Inference Engine) appeared. It allows the developer to build systems based on rules and meta-rules (a meta-rule is executed as a rule). For example he would be able to use rules which remove some rules in a convenient moment.

---

Copyright © 2007, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

Contrary to most systems which use meta-rules (Cazenave 2003) (Spreeuwenberg, Gerrits and Boekenoogen 2000) in a static way, the triggering of meta-rules can dynamically modify the rules structure during the session. Also, reflective systems can be made by the RSIE.

## II. The Reflexive System Inference Engine

Two types of inference engines are known: those based on a filter algorithm and those using an RETE network (Forgy 1982).

The use of a RETE network is much faster than a filtering technique. On the other hand, it has a more complex physical structure: various types of nodes and joints. If this implementation had been chosen, every execution of a meta-action (for example to modify or to delete a rule) would have entailed, every time, an expensive reorganization of the RETE network.

In order to benefit of the advantages of the two methods, we use a hybrid method. A RETE network is created for each rule of the rules base. The idea is to preserve the powerful aspect of RETE architecture while preventing that a meta-action entail a total rebuilding of the rules base but affects only concerned rules.

The system is written in Java because this language has some reflexive aspects which we could make use of.

### **III. Experiments and assessment**

RSIE was tested with a 3.00 GHz Pentium by executing the rule-based system IDRES (Nigro and Barloy 2006). IDRES system recognizes maneuvers made by a vehicle from a sequence of known data given by a set of proprioceptive sensors.

It comprises ten rules and for each 0.010 s eight facts are generated starting from the data of eight sensors.

Results shows that adding metarules to delete rules which became useless allow to match less rules and to gain in term of time of execution and memory space. Indeed, the system creates 821 facts in 1.100 s without the meta-level and only 405 in 0.580 s with this level. The use of metarules which create progressively relevant rules allow to give only a restricted number of initial rules to the system and optimize speed of execution (0.500 s).

The structure used to develop and to use RSIE allows the conception of meta-rules, meta-conditions or meta-actions as easily as one rule. It also authorizes the creation of reflexive metarules (which apply to themselves) without having to duplicate knowledge (Kornmann 1996). Another advantage is that the reflexive system inferences engine is able to execute these metarules during the execution. It authorizes the development of learning techniques in real time or in very dynamic domains environment (like assistance in driving cars).

### **IV. Conclusion**

The reflexive system inferences engine is still in development and this article shows the first results of the core around whose will be implemented the final version. It is useful for the development of metaknowledge. In the future, new functionalities will be added to RSIE to make it more efficient.

### **Acknowledgements**

Research supported in part by Champagne-Ardenne Regional Council (district grant) and the European Social Fund.

### **References**

- Cazenave, T. 2003. Metarules to Improve Tactical Go Knowlegde. *Information Sciences*, 154 (3-4):173-188.
- Forgy, C.L. 1982. Rete : A Fast Algorithme for the Many Pattern / Many Object Pattern Match Problem. *Artificial Intelligence*, 19.
- Genesereth, M.R., and Nilsson N.J. 1987. Logical Fundations of Artificial Intelligence, Los Altos, CA Morgan Kaufmann.
- Hayes, P.J. 1973. Computation and deduction. Proc. 2nd. Symposium on Mathematical Foundations of Computer Science, Czechoslovakian Academy of Sciences, 105-118.
- Kornman, S. 1996. Infinite Regress with self-monitoring. *Reflection'96 Conference* :221-233. San Francisco.
- Nigro, J.M., and Barloy, Y. 2006. The Meta Inferences Engine : a tool to use metaknowledge, IPMU 06, Paris.
- Spreeuwenberg, S., Gerrits, R., and Boekenoogen, M. 2000. VALENS: A Knowledge Based Tool to Validate and Verify an Aion Knowledge Base. *ECAI 2000*:731- 735.