RECIBE

Revista electrónica
DE COMPUTACIÓN, INFORMÁTICA, BIOMÉDICA Y ELECTRÓNICA

Año 2 No. 2

Vol.2 No.2 | Jun-Nov 2013

Índice

Computación e Informática

ReCIBE, Year 2 No.2, Nov 2013

Towards a framework for threaded inference in rule-based systems

Luis Casillas Santillan
Computer Science Department
CUCEI – Universidad de Guadalajara
aluis.casillas@red.cucei.udg.mx

Miguel Guerrero Segura Ramirez
Computer Science Department
University of Guadalajara, Mexico
miguel.guerrero@cucei.udg.mx

Abstract: *Information and communication technologies have shown a significant advance and fast pace in their performance and pervasiveness. Knowledge has become a significant asset for organizations, which need to deal with large amounts of data and information to produce valuable knowledge. Dealing with knowledge is turning the axis for organizations in the new economy. One of the choices to gather the goal of knowledge managing is the use of rule-based systems. This kind of approach is the new chance for expert-systems' technology. Modern languages and cheap computing allow the implementation of concurrent systems for dealing huge volumes of information in organizations. The present work is aimed at proposing the use of contemporary programming elements, as easy to exploit threading, when implementing rule-based treatment over huge data volumes.*

Keywords: *inference, thread, rule, expert, knowledge, framework.*

1. Introduction

Information and communication technologies have shown a significant advance and fast pace in their performance and penetration. Knowledge has become a significant asset for organizations, which need to deal with large amounts of data and information to produce valuable knowledge.

There are different mechanisms enabling organizations to gather, discover and inference valuable knowledge: surveys, data-mining techniques, Bayesian networks, expert systems, neural-nets, and etcetera. The present work will focus in the rule-based systems, and the possibilities to implement some concurrency model using threads.

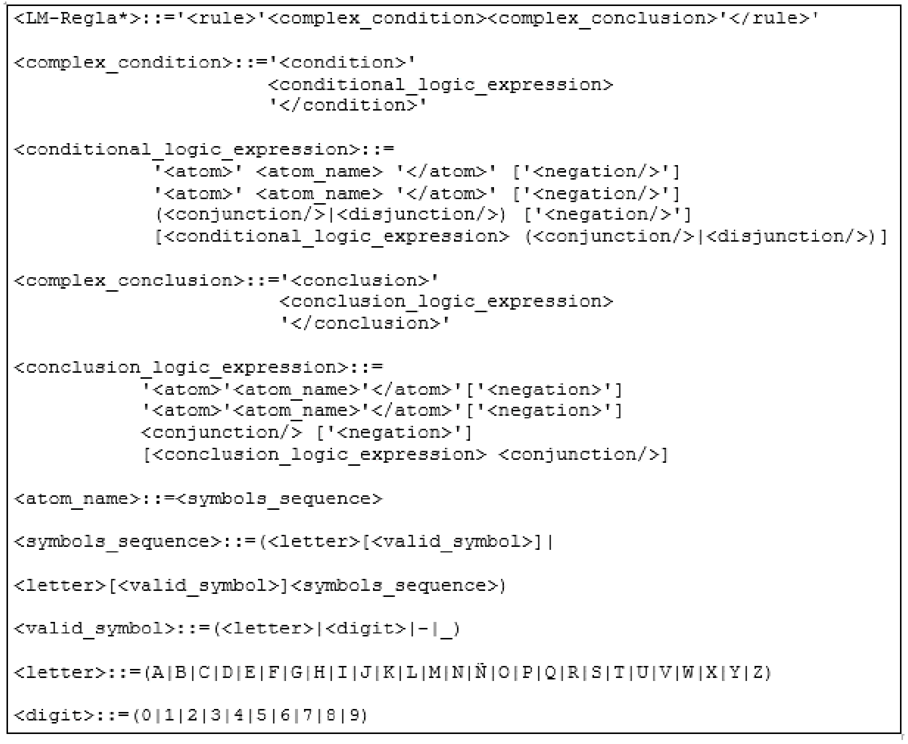
Traditional inference procedures undertake a one-thread mechanism to develop its work; forward-chaining and backward-chaining are the main methods for this approach (Giarratano and Riley, 1998). The Rete Algorithm (Forgy, 1982) presents an improved mechanism to match patterns at fast pace, by taking advantage of temporal common structures presented in the reasoning process and rules. Our proposal is inspired, precisely, in the Rete Algorithm; our goal is to implement a multi-threaded model capable of perform simultaneously pattern matching when no functional dependencies are present among involved rules for the current inference case.

This is not an untreated effort, i.e. Krause et al. (2007) provide a framework to deal with incomplete context information through the use of alternative context construction trees by developing diverse inference branches following Bayesian net templates. Besides, a series of works have considered the possibility of implementing models with implicit or explicit parallelism for rule-based systems under Rete’s philosophy (Gupta et al., 1986; Gupta et al., 1989)(Tambe et al., 1988). For these works, parallelism is achieved through multi-processor computers using shared memory. Our proposal is not intended to be implemented over pervasive parallel systems; we rather seek the development of such approach over common computers using application threads, which are already available in most of modern programming tools. We are also looking for developing this approach in a local environment as an own technology, instead of using canned technology coming from abroad. Such goal is perfectly valid in the academic scope, giving students the chance to know and deal with technology development throughout open access to own technology, without copyrighting constraints.

Hence, this work is aimed at discovering an easy-to-implement approach, taking advantage of the capacity of modern languages to exploit the power of multi-threading. We believe that it is possible to construct an object oriented framework: providing a set of classes which are ready to inherit or for being instantiated. Current proposal is not aimed at improving the performance achieved in previous works (Forgy, 1982)(Gupta et al., 1986; Gupta et al., 1989)(Tambe et al., 1988), but indeed it is oriented at presenting a contemporary treatment for the problem of parallel reasoning in rule-based systems. Contemporary platforms are then enabled to make use of the framework, as said before, by inheriting or instantiating from available classes.

2. Framework elements

When standing in the object oriented paradigm, building a framework consist in developing a set of classes representing the different objects from reality and a set of relations representing all the binds distinguished among the objects of interest. Additional considerations must be attended; such as re-usability, defensive programming, security, results-reliability and etcetera. The current study agrees with the object orientation; hence, the proposed framework will be built under the object oriented considerations.



```
<LM-Regla*>::='<rule>'<complex_condition><complex_conclusion>'</rule>'

<complex_condition>::='<condition>'
                      <conditional_logic_expression>
                      '</condition>'

<conditional_logic_expression>::=
            '<atom>' <atom_name> '</atom>' ['<negation/>']
            '<atom>' <atom_name> '</atom>' ['<negation/>']
            (<conjunction/>|<disjunction/>) ['<negation/>']
            [<conditional_logic_expression> (<conjunction/>|<disjunction/>)]

<complex_conclusion>::='<conclusion>'
                       <conclusion_logic_expression>
                       '</conclusion>'

<conclusion_logic_expression>::=
           '<atom>'<atom_name>'</atom>'['<negation>']
           '<atom>'<atom_name>'</atom>'['<negation>']
           <conjunction/> ['<negation>']
           [<conclusion_logic_expression> <conjunction/>]

<atom_name>::=<symbols_sequence>

<symbols_sequence>::=(<letter>[<valid_symbol>]|

<letter>[<valid_symbol>]<symbols_sequence>)

<valid_symbol>::=(<letter>|<digit>|-|_)

<letter>::=(A|B|C|D|E|F|G|H|I|J|K|L|M|N|Ñ|O|P|Q|R|S|T|U|V|W|X|Y|Z)

<digit>::=(0|1|2|3|4|5|6|7|8|9)
```

Table 1.*BNF for the LM-Regla Language*

The main framework elements are the rule and the atom; figure 1 sketch key components - considering the roles played in the whole rule-based model. These models represent a pragmatic treatment for the task of representing knowledge inside a rule-based system.

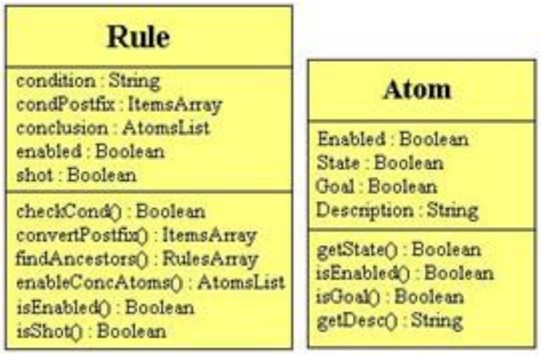


Figure 1.*The components for Rule and Atom*

Regarding the logics expressions inside the rules, it has been developed an adaptation of the algorithm for assembling reverse polish representations (Lukasiewics, 1956). This kind of representation releases computing-efforts when solving expressions and the conversion is made only during rules-production, then semantic networks are built with the rule's components and these data-structures are used during system's deployment.

3. Concurrent inference

In order to achieve the concurrent inference, every rule implements a thread of control. Hence, every rule is verifying all the time if its input atoms are enabled; when enabled, condition is verified. In those cases in which conditions' verification returns true, rules are triggered; once a rule is triggered, all conclusion atoms for that rule are enabled too.

Atoms are already in the working-memory, they are enabled when user verifies its condition in the real world or when some rule implying them is triggered. Therefore, the other rules waiting for its atoms-enabling will also verify their conditions. This mechanism generates a chain-reaction in the inference process. The name that we have used for that effect is "inference avalanche", and if fact it is the nickname used to call this inference system. The figure 2 shows the primitive mechanism supporting this effect of "avalanche".

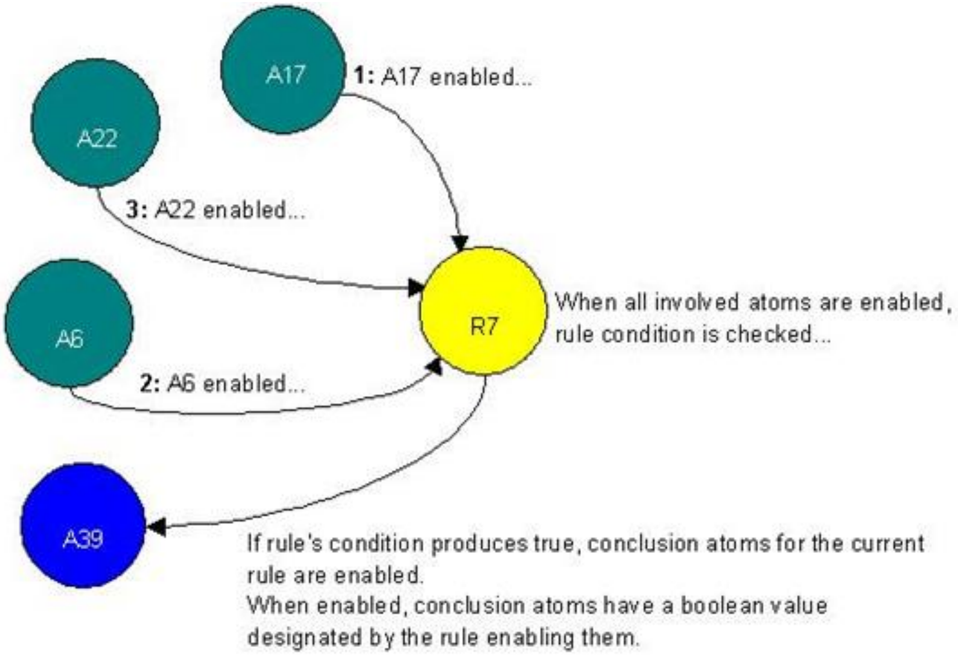

A17
1: A17 enabled...
A22
3: A22 enabled...
A6
2: A6 enabled...
R7
When all involved atoms are enabled, rule condition is checked...
A39
If rule's condition produces true, conclusion atoms for the current rule are enabled.
When enabled, conclusion atoms have a boolean value designated by the rule enabling them.

Figure 2. *Primitive mechanism for concurrent inference*

Isolated rules are connected through common atoms in *working memory* to the other rules waiting for its*atoms-enabling*. When isolated rules are triggered, their conclusion atoms are enabled, enabling new rules. The whole phenomenon is controlled by functional dependencies. At the first approach, most of the rules may be triggered simultaneously;

nevertheless, there are usually many dependencies. Hence, most of the rules have to wait for its input atoms-enabling for long periods. There is no way to improve such functional dependences, due to knowledge nature and it is not possible to improve this performance at this stage; *big concepts are made with small concepts and they keep inherent bounds which are hard to break*. At first, small elements must be satisfied. To accomplish this task without punishing the performance; a backtracking procedure is run in advance [during rules-loading], thus functional dependencies are established. These dependencies give order to the concurrent inference (see figure 3).

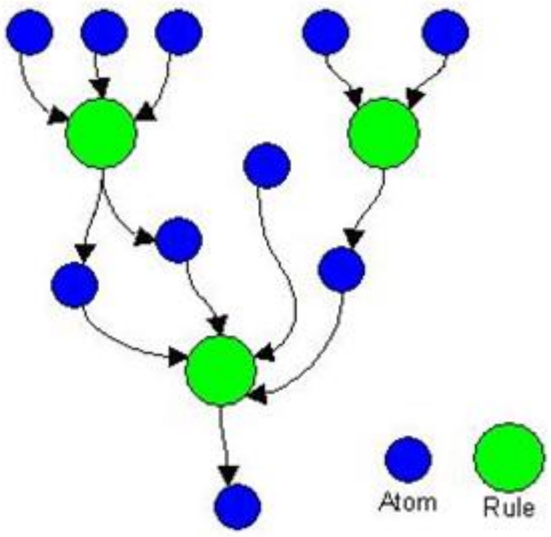


Figure 3. *Functional dependencies through the atoms*

All the rules are connected throughout a network of atoms. This organizational network obeys the hierarchical structure defined by the functional dependencies discovered during the rules-loading [using backtracking]. Figure 4 sketches this idea.

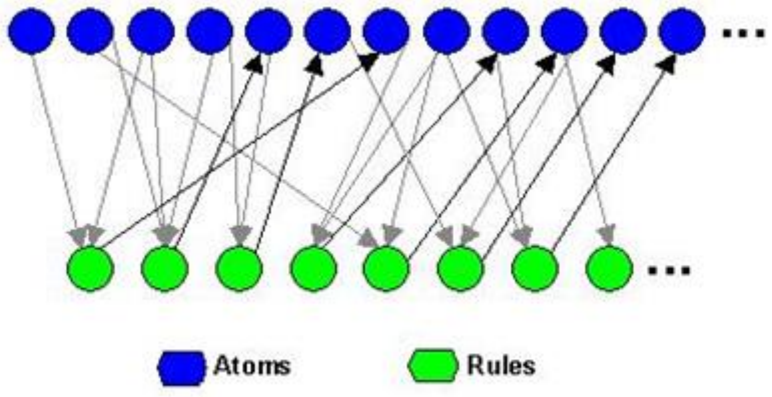


Figure 4. *Network of atoms and rules*

According to the proposed structure and functioning, those "free" (as in freedom) atoms (usually related to small concepts) can be verified at first by user's inputs, sensors' inputs or querying other systems for feeding the system. These will start the mentioned chain-reaction. The concurrent inference process will not finish until a Goal-Atom is *enabled*; although extended executions can be ran, trying to gather all possible goals for one single situation analyzed by the rule-based system.

4. Functionality of the framework

This inference model is used in the academic field to teach expert and rule-based systems in our University. The framework becomes the axis for explaining novel inference-mechanisms, just after explaining traditional inference mechanisms. It has been implemented using C#.

The students get the framework as a set of classes (see figure 5), and they develop an expert system around this basic structure. By inheriting and instantiating, they tailor an inference-mechanism adapted for a specific knowledge-field. They must undertake a research in a selected knowledge-field [every student selects by him/her self the field]; which implies interviews, observation, documentary research and even participating in the environment. The final product is a system capable to evaluate, decide and recommend in the selected knowledge-field.

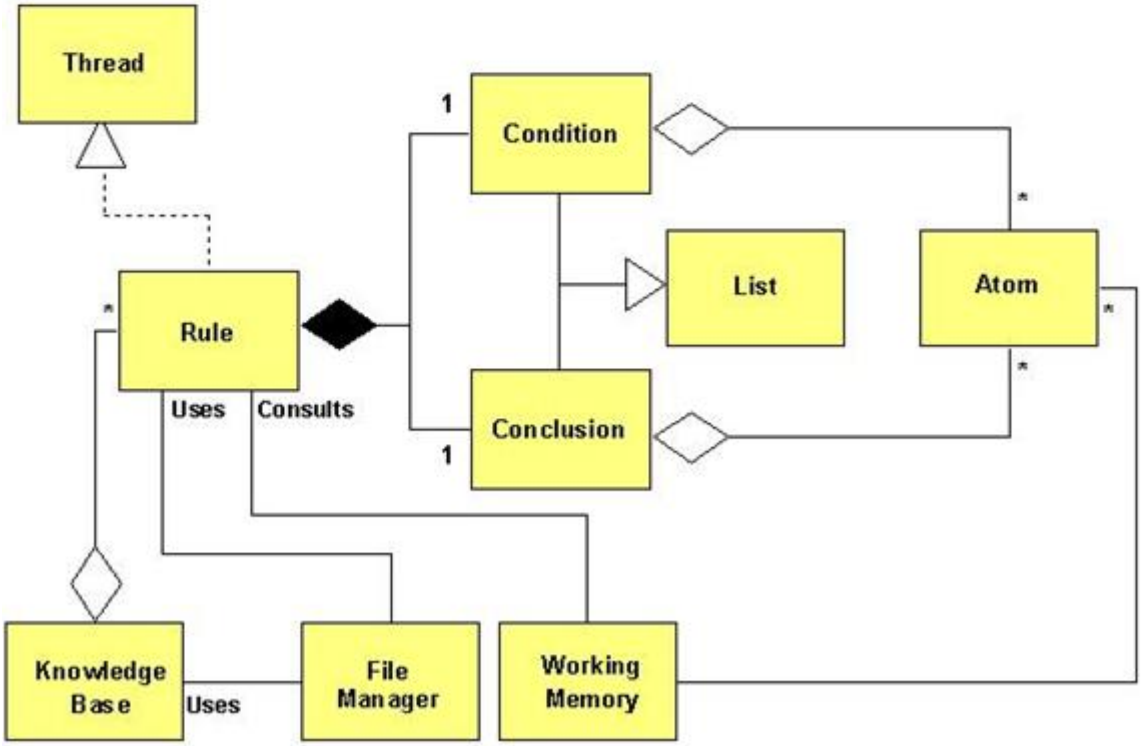


Figure 5. *Main structure for the framework, notice the absence of an inference-engine; its responsibility has been delegated in every concurrent rule. During the class loading, the knowledge base establishes the functional dependencies and the rules do the rest of the work under a concurrency model.*

5. Conclusions

Modern Information and Communications Technologies enabled the development of new approaches for traditional treatment of knowledge [symbolic]. Through the use of these modern elements, it had been possible to develop a new framework for building rule-based systems. Such systems provide renewed expectancy in current knowledge-management scenarios.

Data and information volumes requiring some kind of processing seem to be bigger every time. The current pervasive computing also seems to deal more significant roles in present societies. Finally, the complexity of the knowledge that should be managed by present computer overloads common data processing algorithms. The availability of modern tools for dealing with the current manifestations of information and communication technologies is valuable. Societies demand knowledge to the organizations [government, business, non-profit and etcetera]. Rule-based systems are an interesting alternative. Throughout the use of new software development technologies, high performance concurrent solutions are easy-to-implement. This approach would, ideally, fold performance shown by one-thread inference systems.

Finally, a new versatile approach for *working-memory* has been developed; which is not a passive actor for rule-based systems anymore, now it would perform a dialog with rules, trying to help them to trigger during the inference process. This contribution includes the fact that *working-memory* has a *priori* all of the system atoms stored and connects the rules as a network using atoms as the amalgam. Besides, rules perform the inference process by themselves as threads.

Our proposal remains under construction and we are working in the final stages related to solve the functional dependences. The already finished stages of this effort are: grammatical handling for *LM-Regla*, knowledge base and working memory infrastructure, forward and backward chaining inference, knowledge and explanation modules, and a common inference engine capable of follow one of different inference approaches and also capable of handling multi-conclusion in rules (a not seen contribution in another inference engines). Presenting our proposal and our advances in this journal are intended as a scientific newsletter.

References

Forgy, C. (1982) “Rete: A fast algorithm for the many pattern/many object pattern match problem”. *Artificial Intelligence* Volume 19, number 1, 1982 pp 17-37.

Giarratano, J.; Riley, G. (1998) *Expert System : Principles and Programming*. International Thomson Editors, USA 1998.

AGupta, A.; Forgy, B.; Newell, A. (1989) “High-speed implementations of rule-based systems”. *ACM Transactions on Computer Systems (TOCS)*. Volume 7 , Issue 2, 1989 pp 119 – 146.

Gupta, A.; Forgy, B.; Newell, A.; Wedig, R. (1986) “Parallel algorithms and architectures for rule-based systems”.*International Conference on Computer Architecture*; Proceedings of the 13th annual international symposium on Computer architecture; Tokyo, Japan 1986 pp 28-37.

Krause, M.; Linnhoff-Popien, C.; Strassberger, M. (2007) “Concurrent Inference on High Level Context Using Alternative Context Construction Trees”. Third International Conference on Autonomic and Autonomous Systems, ICAS07.

Lukasiewics, J. (1956) “Investigations into the sentential calculus”. *Logic and semantic metamatemathics*; Oxford, England 1956.

Tambe, M.; Kalp, D.; Gupta, A.; Forgy, C.; Milnes, B.; Newell, A. (1988) “Soar/PSM-E: investigating match parallelism in a learning production system”. *Symposium on Principles and Practice of Parallel Programming*; Proceedings of the ACM/SIGPLAN conference on Parallel programming; United States 1988 pp 146 – 160.

Biographical notes:

Luis Casillas Santillan holds a Ph.D. and a master's degree in Information and Knowledge Societies from the Open University of Catalonia, Spain; as well as a master's degree in Information Systems and a B.Sc. in Informatics from the University of Guadalajara (UG), Mexico. He has been working as full-time professor for 18 years in the Computer Science Department from the UG and published various papers and scientific chapters in diverse journals and books. He serves as member of the editorial board for the Journal of Knowledge Engineering: EXSY and as reviewer for diverse journals and other scientific publications in knowledge engineering, computers science and ICT in education. His research interests are: knowledge gathering and representation, bio-inspired systems, expert systems, complex networks analysis, and soft-computing.

Miguel Guerrero Segura Ramirez holds a master's degree in Information Systems and a B.Sc. in Informatics from the University of Guadalajara (UG), Mexico. He has been working in expert systems development since 1994. His master's degree thesis regards an expert system for designing and configuring computers' networks. He has been working as full-time technical professor for 17 years in the Computer Science Department from the University of Guadalajara .