ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (27-35), Month: October 2014 – March 2015, Available at: www.paperpublications.org

An Extended Reasoning Cycle Algorithm for BDI Agents

Donald Rodriguez-Ubeda[1], Dora-Luz Flores[2], Luis Palafox[3], Manuel Castanon-Puga[4], Carelia Gaxiola-Pacheco[5], Ricardo Rosales[6]

[1] Faculty of Chemical Sciences and Engineering, Autonomous University of Baja California, Department of Mathematics, University of Sonora.
[2,3,4,5,6] Faculty of Chemical Sciences and Engineering, Autonomous University of Baja, California.

Abstract: **Multi-agent systems and particularly BDI agents are mostly used in a wide range of projects, from agent-based simulations to air-traffic control. They all benefit from the autonomy and proactive behavior that provides agent-based architectures, as well as the characteristics of reasoning that are outlined by the BDI architecture. Therefore the Belief Desire Intention agent model and Agent Speak language have become a state-of-the-art and one of the challenging research subjects in the agent modeling and programming area.**

In particular the BDI architecture is frequently used in the development of agents that try to simulate certain aspects of human behavior, and precisely perception and formulation of beliefs are two of the elements of BDI agents that require special attention in the development of such agents. This work propose a way to extend the reasoning cycle algorithm on BDI agents, in a way that it allows to process inaccurate perceptions in the formulation of beliefs in such agents; it also shows an example implemented in Agent Speak as well as the results of its execution within the Jason interpreter.

Keywords: **Agent, Agent Speak, Beliefs, BDI, Fuzzy-BDI, Fuzzy Perceptions, Simulation.**

I. INTRODUCTION

Many of the systems we need to build in practice have a reactive characteristic, in the sense that they have to maintain a long-term, ongoing interaction with their environment; they do not simply compute some function processing an input, generating an output and then terminate.

Reactive software systems are systems that cannot adequately be described by the relational or functional view. The relational view regards programs as functions starting from an initial state following a set of state transitions until they reach a terminal state, so we always can predict the function's output for a given input, which is known as deterministic. These plain systems also have the characteristic that can be paused and resumed without any data loss, because they don't have a close and a long term interaction with the environment.

On the other side, the main role of reactive systems is to maintain a frequent interaction with their environment, accordingly, must be described and specified in terms of their on-going behaviour. The best way to study a concurrent system is through the analysis of the behaviour exhibited. This is because each individual module in a concurrent system is a reactive subsystem, interacting with its own environment which consists of the other modules [1].

Examples of reactive systems include computer operating systems, process control systems, on-line banking systems, web servers, and the like. Many of those systems are developed using the agent-oriented paradigm.

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (27-35), Month: October 2014 – March 2015, Available at: www.paperpublications.org

A. Agents

An agent is a reactive system that exhibits some degree of autonomy in the sense that we can delegate some task to it, and the system itself will determine the best way to achieve it. These systems are named agents because they are perceived as active entities, a kind of purposeful producers of actions; it is possible to send them out into the environment to achieve some goals, and they will actively pursue them, figuring out for themselves the best way to accomplish such goals, so they don't need any human guide that tell them in a low-level detail how to do it. We can imagine such agents being delegated a goal like booking a flight and hotel for us, bidding on our behalf in an on-line auction, or cleaning our office space for us if they are robotic agents [2].

As mentioned above, agents are systems situated within some environment which maintain a frequent interaction with it. Therefore, agents need to perceive their environment (using sensors), and must have a repertoire of possible actions that they can perform (via effectors or actuators) in order to interact with (or modify) such environment.

The key question facing the agent behaviour is how to go from sensor input to action output: decide what to do based on the information obtained via sensors. There are many approaches trying to explain and solve that question; such approaches are known as cognitive models or agent architectures, like SOAR [3], [4], ACT-R [5] and BDI [6], [7].

B. The BDI Agent Architecture

Agent systems are becoming increasingly popular for solving a wide range of complex problems. BDI agent systems have a substantial base in theory as well as a number of medium-to-large-scale implemented systems that are used for challenging applications such as air-traffic control, space systems and human decision making simulation [8].

The BDI model (called the belief-desire-intention model or architecture) originated in the work of the Rational Agency project at Stanford Research Institute in the mid-1980s. The spirit of the model is located within the theory of human practical reasoning developed by the philosopher Michael Bratman [6], which focuses particularly on the role of intentions in practical reasoning.

One of the strengths of the BDI based systems is their strong link to theoretical work, in particular that of Rao and Georgeff [9], Bratman et al. [7] and Wooldridge [10]. Although the theory is not implemented directly in the systems, it does inform and guide the implementations [11].

The concepts of the BDI paradigm allow the use of a programming language to describe human reasoning and actions in everyday life [12]. Because of this straightforward representation, the BDI paradigm can easily map extracted human knowledge into its framework. The conceptual framework of the BDI model is described in [7], which also describes a specific BDI agent architecture called IRMA.

Therefore, BDI is a software model developed for programming intelligent agents, which is characterized by the implementation of an agent's beliefs, desires and intentions, it actually uses these concepts to solve a particular problem in agent programming. BDI also provides a mechanism for separating the activity of selecting a plan (from a plan library or an external planner application) from the execution of currently active plans. To achieve this separation, the BDI software model implements the principal aspects of Michael Bratman's theory of human practical reasoning [6], [7]. In other words, BDI implements the notions of belief, desire and intention, in a manner inspired by Bratman.

Intention and desire are both pro-attitudes, in fact, both of them are mental attitudes concerned with action, but intention is distinguished as a conduct-controlling pro-attitude. Commitment is the distinguishing factor between desire and intention, noting that it leads to temporal persistence in plans and further plans, being made on the basis of those to which it is already committed.

The hierarchical nature of plans is easily implemented: a plan consists of a number of steps, some of which may invoke other plans. The hierarchical definition of plans itself implements the kind of temporal persistence mentioned above since the overarching plan remains in effect while subsidiary plans are being executed.

However, it should be observed that before an agent can achieve his goals, first must formulate a course of actions (the plans). Those actions successfully will be executed as long as his beliefs about the state of the world around him are true.

In fact, the trustworthy of the agent's beliefs is a consequence of his skills to perceive (with some degree of confidence) the changes in the environment.

It is noteworthy that the BDI architecture does not provide a method so that agents can formulate their beliefs, only states [2] that can be obtained in several ways, of course suggests that perceptions influence the process of formulation of beliefs, but is left open for everyone to solve it. In this paper, we show how a notion of belief like inaccurate (or vague) perceptions can be integrated into the BDI logic of Rao and Georgeff [9], preserving the features of the logic while adding to it in ways that complements and aids in the building of BDI agents that have the skills to construct their own beliefs taking as input inaccurate perceptions. We argue that our method allows those BDI agents to get a more real picture of the world where they live than those BDI agents which perceptions are limited to crisps values.

II. RELATED WORKS

This work has been inspired by the works of various authors who have made significant contributions in review of beliefs from the perspective of logic [13], [14] and artificial intelligence [15], [16]. Its focus is to understand how an agent should change his belief in the light of new information.

In their paper Kacprzak and Kosinski [17] propose a new approach in which ordered fuzzy numbers (OFN) are applied. They focus on multi-agent systems and concentrate on agents' beliefs. In particular, they want: first to make agents able to use ordered fuzzy numbers in their thinking and making decisions. Second used ordered fuzzy numbers for evaluating agents' beliefs about their beliefs.

In the logic above, the beliefs are all formulated from crisp perceptions. However in real life, it is not always the case [18], as Jing and Luo said [19] our belief is often fuzzy. For example, "I think the temperature will be high tomorrow". Here, "high" is a fuzzy conception because there is not any crisp division between "high" and "low".

Moreover, we believe that the perceptions of the agents cannot be crisp because in the real world everything is perceived diffusely; for example is very usual to hear the expression "I feel that the water's temperature is warm", nobody says "I feel that the water's temperature is 123 degrees". Even if we use a sensor to measure it, that sensor will measure it with some degree of imprecision, and we would have to formulate a belief from that measurement.

III. DISCUSSION

The BDI (Beliefs-Desires-Intentions) architecture is one of the most studied architectures for cognitive agents. In the area of agent-oriented programming languages in particular, AgentSpeak(L) is one of the best known languages based on the BDI architecture.

Agent Speak (L) is an abstract logic-based agent-oriented programming language introduced by Rao [20]. In such language practical BDI agents are implemented as reactive planning systems: they run continuously, reacting to events (e.g., perceived changes in the environment) by executing plans given by the programmer. Plans are courses of actions that agents commit to execute so as achieve their goals. The pro-active behaviour of agents is possible through the notion of goals (desired states of the world) that are also part of the language in which plans are written.

Jason is a software framework that interprets an extension of the Agent Speak language; Bordini et al. [2] argue in their book that the interpretation of the agent program effectively determines the agent's reasoning cycle. An agent constantly perceives the environment, reasoning about how to act so as to achieve its goals, then acting so as to change the environment. The (practical) reasoning part of the agent's cyclic behaviour, in an Agent Speak agent, is done according to the plans that the agent has in its plan library. Initially, this library is formed by the plans the programmer writes as an Agent Speak program.

We modified the algorithm proposed by [2] as the reasoning cycle for a BDI agent, in a way that it includes a perception mechanism that allows handling vague perceptions and processing them to formulate the agent beliefs. The changes are showed on Figure 1, we introduced a knowledge base (K), because any agent that need to perceive the environment must have the necessary knowledge to perform some processing on the perceptions.

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (27-35), Month: October 2014 – March 2015, Available at: www.paperpublications.org

The perceptions or signals are obtained from the agent's sensors, and passed to the Belief Construction Process (BCP), such process starts by converting the signals to a fuzzy representation, then it infers over those signals and the knowledge base (K), the result is a new belief (B') which is introduced to the agent's belief base by the Belief Registration Process (BRP). The algorithm for the BCP is showed on Figure 2.

```
B ← B0 ;    /* B 0 are initial beliefs */
I ← I 0 ;    /* I 0 are initial intentions */
K←K0;    /* K0 is the initial Knowledge base */
while true do
            /*get next percept ρ via sensors; */
   get input signals from sensors;
   B← BCP(signals);
   B ← brf (B, ρ) ;
   D ← options(B, I) ;
   I ← filter(B, D, I) ;
   π ← plan(B, I, Ac) ; /* Ac is the set of actions */
   while not (empty(π ) or succeeded(I, B) or impossible(I, B)) do
      α ← first element of π ;
      execute(α) ;
      π ← tail of π ;
            /* observe environment to get next percept ρ ; */
      get input signals from sensors;
      B← BCP(signals);
      B ← brf (B, ρ) ;
      if reconsider(I, B) then
         D ← options(B, I) ;
         I ← filter(B, D, I) ;
      end-if
      if not sound(π , I, B) then
         π ← plan(B, I, Ac)
      end-if
   end-while
end-while
```

Fig. 1: Algorithm for a BDI practical reasoning agent.

```
FuzzySets← fuzzifier(signals);
B' ← FIS(FuzzySets, K);
B←BRP(B');
return(B);
```

Fig. 2: Belief Construction Process Algorithm for a BDI agent.

A. A BDI Agent with Fuzzy Perceptions Example

Below we show an example to demonstrate our proposal to extend BDI architecture as well as the reasoning cycle on Jason's agents to provide such agents with the ability to process inaccurate perceptions and use them to build their own beliefs.

The example consists of a BDI agent (adjust water temperature agent) which aims to regulate the combination of water on a sprinkler to obtain the ideal temperature for a person. Table 1 describes the agent of this example.

TABLE I: Agent Properties

Beliefs	Hot(water), Cold(water), Warm(water)
Desires	Get water with a warm temperature
Intentions	Open cold water, Open hot water

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (27-35), Month: October 2014 – March 2015, Available at: www.paperpublications.org

To reach its desire the agent formulates a course of actions, which is headed by the intention to open the hot sprinkler, as can be observed on Figure 3.

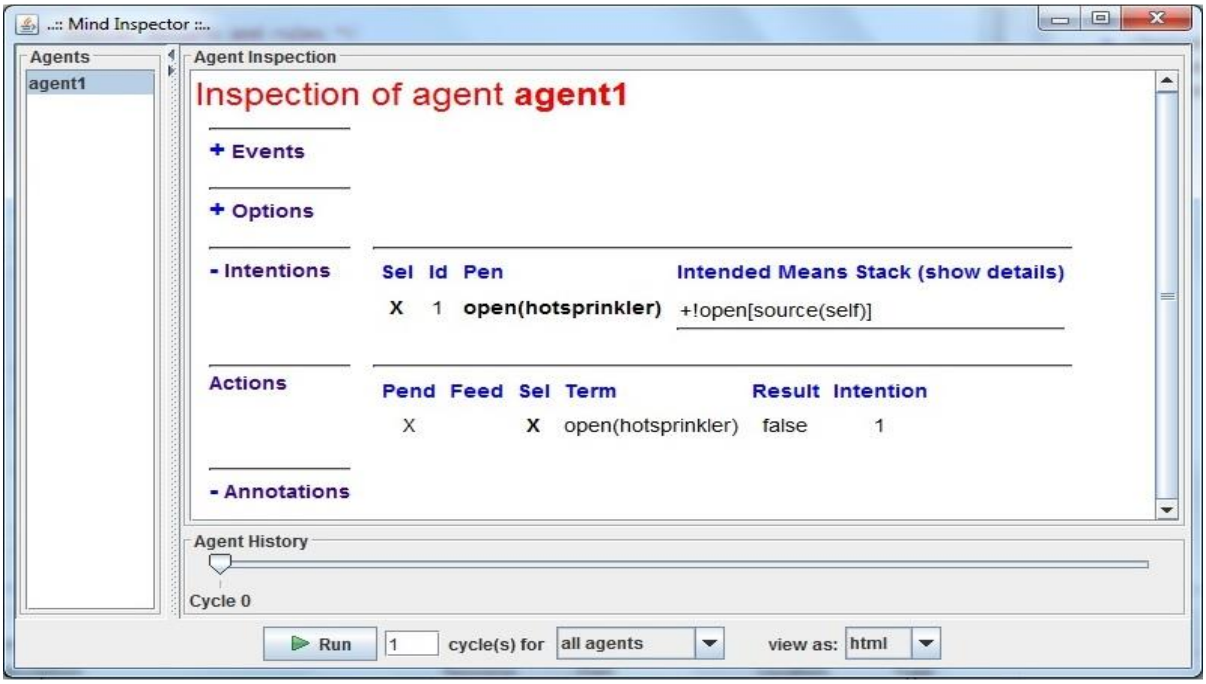

Fig. 3: Mind Inspector showing the agent Intentions (open the hot sprinkler).

After opening the hot water sprinkler, the agent starts perceiving the temperature of the water and formulating its beliefs, it will be mixing hot and cold water until it reach the warm temperature. The agent's knowledge base is described by three fuzzy membership functions (see Figure 4), which are passed to the BCP algorithm to build the agent's beliefs. The BCP algorithm was implemented in java and used a Fuzzy Inference Library developed at our University [21] to evaluate the agent's input signals.

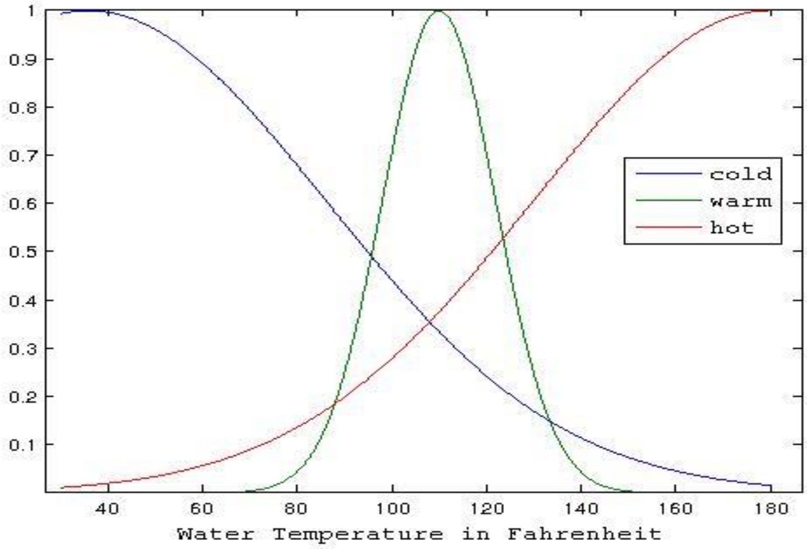

Fig. 4: Knowledge Base for the Agent (K).

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (27-35), Month: October 2014 – March 2015, Available at: www.paperpublications.org

The Figure 5 show the "hot(water)" agent's belief, as a result of the agent's processing on its perception. Such processing was made using the signal obtained from the agent's temperature sensor and it's knowledge base.

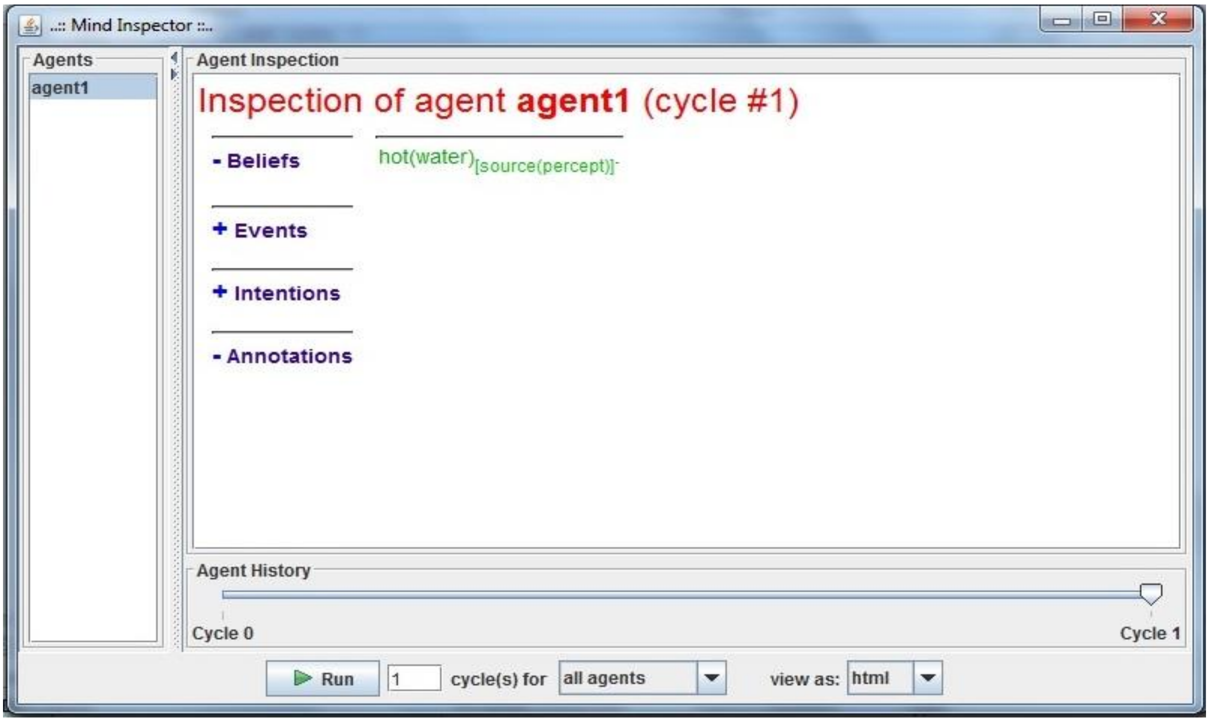

Fig. 5: Mind Inspector showing agent Beliefs (hot water).

As the results from it's belief the agent reacts formulating the corresponding intention to get more cold water on the mix, as it shows on Figure 6.

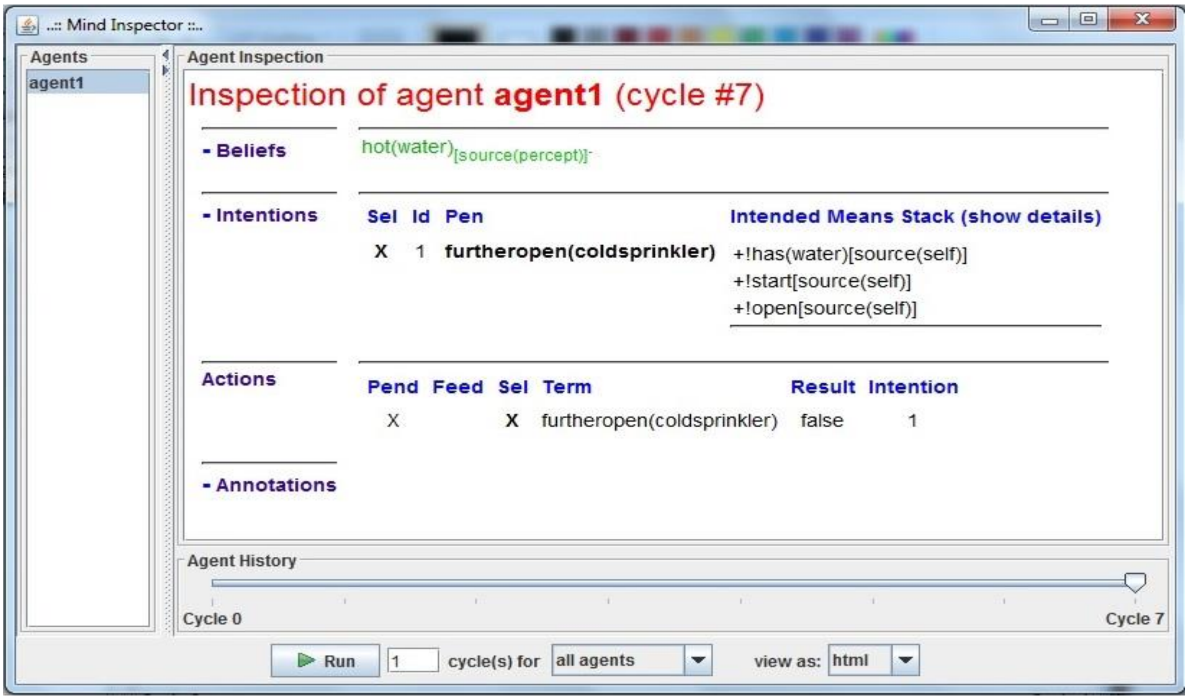

Fig. 6: Mind Inspector showing the agent Intentions (open the cold sprinkler).

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 1, Issue 2, pp: (27-35), Month: October 2014 – March 2015, Available at: www.paperpublications.org

The agent keeps perceiving the environment and reacting until it reaches the desire of getting "warm(water)", that is confirmed when it formulates the corresponding belief (See Figure 7).

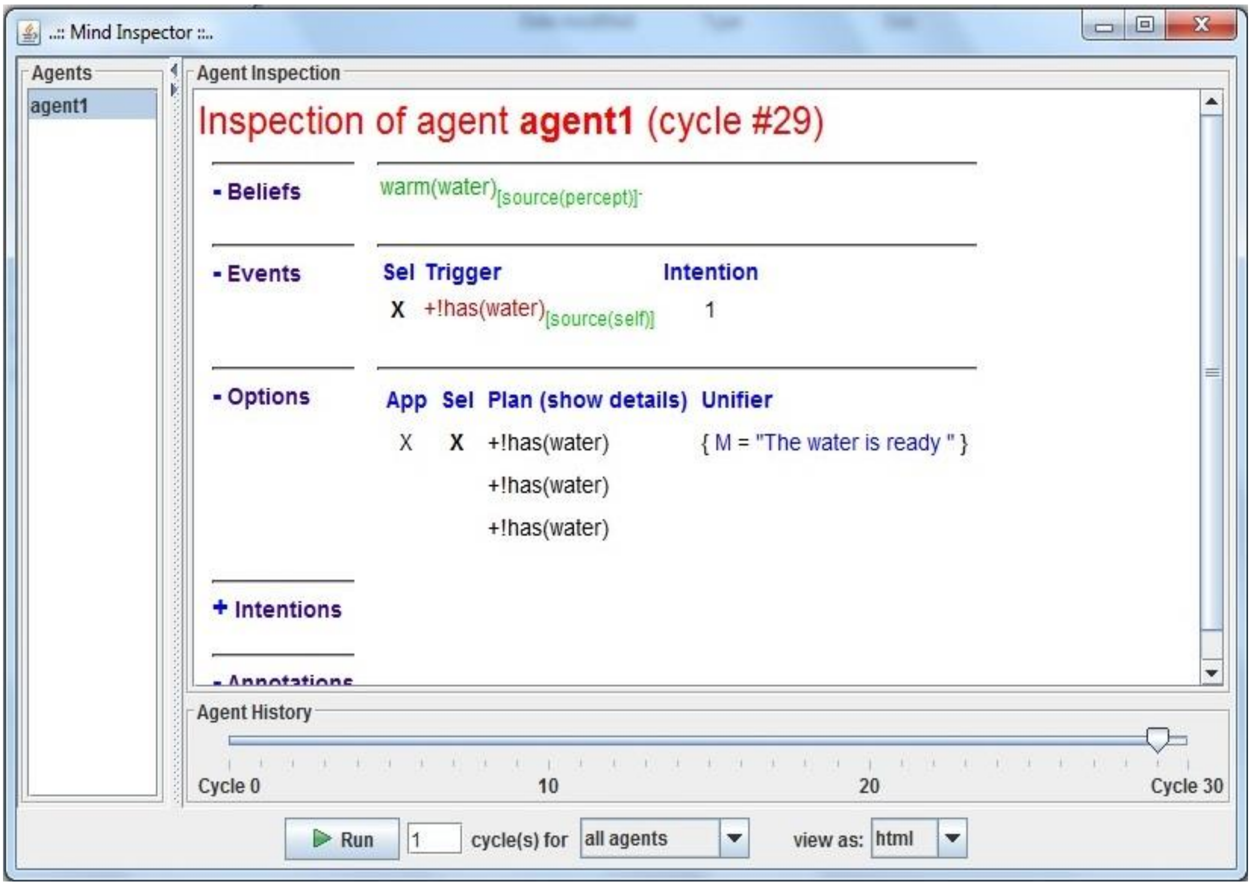

Fig. 7: Mind Inspector showing the agent Beliefs "warm (water)" and the message "The water is ready".

IV. CONCLUSION

After having performed the relevant tests, we have enough evidence to affirm that our algorithm allows to incorporate fuzzy perceivers in BDI agents in a way that it does not interferes with the reasoning cycle of the BDI agents. Therefore it is of great benefit, because our method allows the development of BDI agents whose perceptions may be inaccurate, but regardless of them such agents have the ability to build their beliefs without affecting the rest of his reasoning logic.

Additionally we believe that our method allows us to develop software agents whose perceptions become more like the way the humans perceive since they can perceive the world as vaguely the humans perceive it.

Another benefit of our method is that it allows performing the analysis and processing of fuzzy perceptions at an earlier stage, before the agent starts to perform any kind of reasoning about the beliefs that it has about the environment. In other words, all the numerical processing of fuzzy logic is done in java and its output are the beliefs, which are inserted into the (BDI) agent's beliefs database, who takes them to perform further processing according to the logic of reasoning that has been implemented in Agent Speak, for the execution within the Jason interpreter.

The progress of this work will allow us to continue with our projects related to agent-based simulation, including simulators to assist researchers focused on social simulation as well as other researchers who study human behaviour, which require agent-based simulators, whose behaviour resembles that of humans.

ACKNOWLEDGEMENT

This work was supported by the National Council of Science and Technology from Mexico, the Autonomous University of Baja California and the University of Sonora.

REFERENCES

[1] Pnueli, "Specication and development of reactive systems," Information Processing 86, pp. 845-858. Elsevier Science, Amsterdam, 1986.

[2] R. H. Bordini, J. F. Hübner and M. Wooldridge, "Programming Multi-Agent Systems in AgentSpeak using Jason," West Sussex: Wiley, 2007.

[3] J. E. Laird, A. Newell and S. Rosenbloom, "SOAR: An Architecture for General Intelligence," Carnegie-Mellon University Technical Report AIP-9, 1987.

[4] J. E. Laird, "The Soar Cognitive Architecture," MIT Press, ISBN 978-0-262-12296-2, 2012.

[5] J. R. Anderson, "How can the human mind occur in the physical universe?," New York, NY: Oxford University Press. ISBN 0-19-532425-0, 2007.

[6] M. E. Bratman, "Intention, plans, and practical reason," Cambridge, MA: Harvard University Press, 1987.

[7] M. E. Bratman, D. J. Israel and M. E. Pollack, "Plans and resource-bounded practical reasoning," Computational Intelligence 4, pp. 349-355, 1988.

[8] S. Rao and M. Georgeff, "BDI agents: From theory to practice," In: L. Gasser and V. Lesser, eds. Proceedings of the First International Conference on Multi-Agent Systems - ICMAS 95, San Francisco, USA, 1995, pp. 312-319.

[9] S. Rao and M. P. Georgeff, "Modeling rational agents within a BDI-architecture," In: J. Allen, R. Fikes and E. Sandewall, eds., Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[10] M. Wooldridge (2001), "Reasoning about rational agents," Robotica, 4: 459-462.

[11] S. Rao and M. Georgeff, "An abstract architecture for rational agents," Morgan Kaufmann, eds., In Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference - KR 92, California, USA, 1992, pp. 439-449.

[12] S. Lee, Y. J. Son and J. Jin, "An integrated human decision making model for evacuation scenarios under a BDI framework, " ACM Transactions on Modeling and Computer Simulation, 2010. 20(4):1-24.

[13] G. Aucher, "A combined system for update logic and belief revision," In: Intelligent Agents and Multi-Agent Systems, Intelligent Agents and Multi-Agent Systems, 2006. 3371(132):1-18.

[14] S. Shapiro, M. Pagnucco, Y. Lesperance, et al. "Iterated belief change in the situation calculus," Artificial Intelligence, 2011. 175(1):165-192.

[15] S. Sardina, L. Padgham, "A BDI agent programming language with failure handling, declarative goals, and planning," Autonomous Agents and Multi-Agent Systems, 2011. 23(1):18-70.

[16] Casalia, L. Godob and C. Sierra, "A graded BDI agent model to represent and reason about preferences," Artificial Intelligence, 2011. 175(7-8): 1468-1478.

[17] M. Kacprzak and W. Kosinski, "Modelling fuzzy beliefs of agents, " Zeszyty Naukowe Politechniki Biatostockiej, 2012. 9: 45-60.

[18] X. Wu and J. Zhang, "Rough set models based on random fuzzy sets and belief function of fuzzy sets," International Journal of General Systems, 2012. 41(2): 123-141.

[19] X. Jing and X. Luo, "A fuzzy dynamic belief logic, " In: Nguyen N T, eds. ICAART 2013 - International Conference on Agents and Artificial Intelligence, Barcelona, Spain, 2013, pp. 289-294.

[20] S. Rao, "AgentSpeak(L): BDI agents speak out in a logical computable language," In: Van de Velde W, Perram J W, eds. Springer-Verlag New York, Inc.,Secaucus, NJ, USA, Proceedings of the 7th European workshop on Modelling autonomous agents in a multi-agent world: agents breaking away: agents breaking away, Einhoven, The Netherlands. 1996, pp. 42-55.

[21] M. Castanon-Puga, J. Castro, J. Flores-Parra, "JT2FIS a java type-2 fuzzy inference systems class library for building object-oriented intelligent applications," In: L. N. in Computer Science (Ed.), Advances in Soft Computing and Its Applications, 2013, pp. 204-215.