Proceedings of the International Symposium on Evolving Intelligent Systems, Plamen Angelov, Dimitar Filev, and Nikola Kasabov (Eds.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

Reactive-adaptive Methodology to encode Evolving Intelligent Agents in Serious Games

Mario Gongora
Centre for Computational Intelligence
De Montfort University
Leicester, UK
mgongora@dmu.ac.uk

David Irvine
Institute of Creative Technologies
De Montfort University
Leicester, UK
dirvine@dmu.ac.uk

Abstract—**This paper describes our proposed methodology for the creation of novel AI based Agents capable of evolving to self-improve. We use AI concepts which are well established in the implementation in character behavior for games, such as Fuzzy Logic, the Belief-Desire-Intention model, and Finite State Machines; and discuss their limitations. We present a novel combination of these techniques to optimize the process of self adaptation for evolving intelligent agents.**

Keywords: Evolvable Agents, Agent Behaviour, Hybrid Systems, Self Developing Systems

I. INTRODUCTION

In the areas of character simulation, such as serious gaming or populating virtual worlds, Intelligent Agents are a powerful tool to implement character behavior. Currently the implementation of these agents' intelligence relies on a primary AI technique, sometimes implemented using exclusively a single methodology. The mostly used techniques are Belief Desire Intention [1], Fuzzy Logic [2], or Neural Networks [3]. While these methods are adequate for basic character behavior in a prescribed environment (e.g. a standard game), using them to create complex Agents requires the particular methodology to be stretched to its limits by modifying it, creating a very complex implementation. In addition, if the agent is to be capable of self-adaption during its *life* (i.e. online execution), current methodologies pose additional restraints. We believe that a hybrid AI system with a well designed structure will result in a more efficient implementation for Agents that are required to behave in a semi-realistic manner and be capable of evolving during their *lifetime*.

In this paper we discuss current AI tools in the field of character behaviour simulation, their strengths and limitations. We present our hybrid system that uses a combination of AI methods, normally unsuitable for self adaptation, with a novel structure that makes it possible to apply constructive methods and enable an evolving Intelligent Agent.

This paper is organised as follows; Section II provides a background of the base techniques that we intend to combine; Section III presents our proposed hybrid complementary method, in Section IV we analyse our structural novelty, concluding in Section V and proposing further work in Section VI.

II. BACKGROUND

A number of well studied techniques within the field of AI are commonly used to implement the behaviour of semi-realistic Agents. On their own, these techniques often contain inherent limitations that make it complicated to achieve a fully featured behaviour simulation. The following sections analyse the main aspects of these techniques.

A. *Belief Desire Intention*

Belief/Desire/Intention (or BDI) is a method of modelling behaviour in rational agents. It can be seen as giving "mental attitudes of Belief, Desire and Intention, representing, respectively, the information, motivational, and deliberative states of the agent" [4]. The BDI model is based in the areas of both Philosophy and Cognitive Science, including work on Rational Agents [5]. Rational agents have bounded resources and usually incomplete knowledge about their environment. This incomplete knowledge is usually defined as the agents' belief network.

Knowledge is "that which is true in the world" while "beliefs are an agent's understanding about the world" [6]. Therefore agents' beliefs can be false or incorrect. Beliefs can also change over time as the agent receives more accurate information. Desires are what the agent wishes to accomplish in regards to its current objectives. These "objectives, or their priorities, are being generated instantaneously or functionally, and thus do not require any state representation" [4]. Desires are not necessarily consistent or compatible, and some cannot be achieved simultaneously. Goal deliberation over consistent desires results in the production of goals [7].

Plans describe the method with which agents achieve their goals and react to events, and can include sub goals. Other plans are executed to achieve these sub goals, creating a hierarchy of plans. When an agent has decided on a plan it commits itself to the plan, forming an intention [7].

BDI has been implemented in several research models, as well as applied in several situations such as Air Traffic management [4], military combat simulation [8], and has subsequently been implemented into frameworks such as the Procedural Reasoning System [9], Jadex [7] and JACK Intelligent Agents [10].

B. *Fuzzy Logic*

As noted in [11] among many others, fuzzy logic has been used in robotic behaviour systems as it provides a method that can cope with the uncertainty inherent to natural environments. Fuzzy logic has the capacity to represent uncertainty with fuzzy set theory [12]. Fuzzy sets are the most relevant to this work, as is their use that can make a behavior more realistic and adequate to handle real world scenarios. Fuzzy sets can also be powerful when dealing with linguistic variables. And language can be a primary interface between agents or between agents and human users of a system.

In agent behavior simulations, the main use of fuzzy sets is the implementation of sensorial input. In a simulated environment, agents have an artificially intrinsic capability of seeing the whole world and the complete set of variables that represent its environment, including what it would *detect* while exploring this environment. By using fuzzy sets, in combination with simpler variable hiding, a more realistic *view* of the world can be achieved. In [2] the authors combined fuzzy logic with BDI in the analysis of the surroundings, enabling the Agents to group data into *sets* and then use these sets to make decisions with their BDI systems. However the BDI required a number of explicit changes to deal with certain situations; "Commitment" to Intentions must be added to prevent an Agent from constantly changing its Intention, and Desires must be weighted to ensure more urgent Desires are met faster than less urgent Desires.

C. *Finite State Machines*

Finite State Machines are by far the most frequently used in today's computer games as a method for generating AI behaviour. In most cases, non-player characters (NPC) behaviour can be modelled in terms of mental states, and hence, FSMs are a natural choice when designing NPCs [13].

Each state that an Agent can enter into has to be foreseen and predefined by the programmer and then entered into the system. When the Agent enters this state it follows the rules laid down by the programmer. The Agent can enter an alternative state and follow another set of rules, or can remain in the same state if the rules fail to change it. If well thought out, this approach can lead to convincing AI that can carry out tasks with some degree of success, however, as the amount of states increases, so does the complexity of the FSM. For complex behaviour implementation, FSMs are not scalable.

Code complexity in a FSM increases significantly as the system scales up [14]. Iterating through all the state options may become too computationally expensive for a system that needs to run in real-time. By definition, a FSM is made up of a finite number of states; this number is implicitly fixed, as a variation at run-time would potentially violate the *finite* principle. Therefore if a state is not predefined by the programmer this cannot be handled or considered by the system at any time once deployed.

Still, FSMs are a powerful and efficient tool in the correct context, and therefore we aim to use them to deal with very specific states that we expect our agents to encounter, where it becomes necessary to control their actions directly.

III. PROPOSED HYBRID METHOD

The techniques presented briefly in the previous section are all capable of producing intelligent Agents, but it is often the case that a substantial amount of adjustment of these techniques is necessary to achieve all aspects of a required behavior, even if simplified. A BDI system often spends a high proportion of time in deliberation, and needs careful adjustments to ensure it has Commitment to an Intention, while preventing it from constantly changing which Intention it acts upon.

Finite state machines can become far too large for a complex AI simulation; and are intrinsically designed to be fixed for the entire *lifetime* of the agent. An attempt to insert or delete states at run-time can be catastrophic or it can become an unfeasible or impractical validation problem from the algorithmic point of view.

But by combining these two technologies in a very specific way, we formulate a methodology to enable the creation of complex semi-realistic agents more efficiently. In addition we can enable the structure to be adapted at run-time so that the Agent can evolve during its *lifetime*. In [14] the authors propose that research should be undertaken to determine if hybrid models of finite state machines and a BDI Agents' can reduce the design and coding while increasing the performance of a system. There appears to have been no subsequent reports of such research undertaken in depth. In this work we show how we can not only address this, but provide the necessary structure for an evolving system.

We take our inspiration from how many creatures, including human beings, behave; at our core, we have instincts, as all creatures do. We have then built upon these instincts through our evolution to create a method of reasoning for many other problems that require reasoning rather than reactions. When we take this type of behaviour and convert it to the closest matching AI technologies, we are presented with the following method:

Our Agents are primarily controlled by a BDI system; this is our emulation of human reasoning. The virtual world in which our Agents exist is simulated in a crisp fashion, therefore it is necessary to simulate the belief system of our Agents, adding uncertainty and imperfection to their beliefs; this will be addressed in the following section. We pass these beliefs on to the BDI system to enable our Agents to have a more realistic view of the virtual world. A FSM is given overriding control of the BDI system for situations where direct action must be taken with no deliberation, providing our emulation of instincts. In this way, we define a reaction-reasoning hierarchical system. Later we will show how we convert the reasoning system into an adaptive system which results in our reactive-adaptive methodology.

A. *Belief Simulation*

It is important to remember that an autonomous agent's decisions and plans are influenced by the information it receives about the world in which it is operating [8]. For the purposes of attempting to accurately simulate real-world behaviour, a purely computational agent is at an immediate advantage; without the additional stage of filtering data that the agent receives from the Virtual Environment, the agent is automatically given a cognition of a situation that no real-world creature is capable of obtaining – our Agents automatically "see and understand everything", and as such we must inhibit this ability.

Work on simulating belief systems of agents has already been undertaken by [8]. They suggest that when simulations are used for real-world decision support tools, it is important that the agent's belief processes are also properly simulated, i.e. based on realistic observation and detection techniques.

The authors in [8] also suggest "conceptualizing the agent's activities in terms of three main steps: detection, measurement and interpretation". This idea has been carried over to the belief simulation in our work. As the Agent picks up (detects) new information it should be measured in such a way that the level of accuracy is inversely proportional to the distance.

Once a level of accuracy has been determined we are able to affect the quality of information that the agent receives. This can be defined as the "interpretation" stage. Given a set of imperfect data the agent must make a decision on how to interpret it. If an object is visible in the distance to the Agent, it is less likely to interpret it correctly as opposed to a nearby object. By determining how imperfect data it has received is, the Agent is capable of making a decision of what to do with the data. This decision making process is handled by our BDI implementation. As this is based on Fuzzy Systems then we can enable evolvable membership functions to adapt towards a realistic simulation; therefore there are no issues here.

B. Planning and decision making using BDI

Decision making via a BDI system enables us to control our Agent's higher level functions, such as social interaction, complex problem solving, and other behaviours that are not otherwise covered by human instinct.

The authors in [4] proposed an abstract interpreter for BDI:

```
BDI-interpreter
Initialize-state();
repeat
  options := option-generator(event-queue);
  selected-options := deliberate(options);
  update-intentions(selected-options);
  execute();
  get-new-external-events();
  drop-successful-attitudes();
  drop-impossible-attitudes();
end repeat
```

This shows the basic interpreter cycle for a BDI agent. The agent updates its beliefs, and then updates its available options. Once it has done this it examines its desires, matches them against these options and proceeds to deliberate over them, before updating its intentions and executing the plans to facilitate those intentions. In the final steps, the interpreter removes successful and impossible goals and intentions.

By expanding on this abstract interpreter we are able to form a method that allows our Agents to make decisions about the virtual world based on imperfect values it receives, and then form plans to determine its actions in this virtual world.

Methods to expand the abstract interpreter already exist in several implementations, including JADEX [7], METATEM [15] and AIL [16]. These existing implementations provide a software framework for the creation of goal-oriented agents [7], and as such could be used to implement BDI in our hybrid system.

BDI Agents can suffer in the aspect of realism due to their rational nature. If not carefully controlled, a rational Agent can, in some cases, continue a behaviour that could not be considered rational in real terms. In a situation where an Agent is trying to achieve a Desire with a high priority, it may ignore a situation which would naturally have a higher priority; thereby causing what appears to be irrational behaviour. While this issue can be overcome with a suitably well developed implementation, it can be avoided entirely by extracting behaviours from the BDI Agent that could be considered "priority behaviours", or, "instinctual behaviours". By handling these behaviours outside of BDI, we can avoid the potential problem of the Agent ignoring them.

C. Instinct emulation using FSMs

Instinct is often an overriding factor in behaviour. In situations that are life threatening, instinct comes into play and determines a behaviour to act to the situation with little reasoning or thought for consequences. While instincts can be controlled, it is more common (and expected) to let an instinct take hold and dictate the actions.

While a BDI system is capable of simulating instinct using pre-programmed Plans, due to the nature of BDI it is often costly to run this simulation with respect to the rational situations. When combined with Commitment to a Plan, BDI reasoning can often override instinct even in situations where instinct would more readily override decision making in the real world. This shortcoming can be mitigated by fine tuning of the BDI system and an Agent's commitment to certain plans, but this is often time consuming. A FSM is more efficient at handling predefined behaviours, equivalent to instinct.

Finite State Machines are capable of making an Agent act a certain way in a finite amount of states with little processing requirement, therefore making them very fast when the list of states is small. The larger a FSM becomes, the less manageable it is for the systems programmer, and the more states the Agent must iterate through to determine its behaviour. For this reason, in our methodology, a FSM should only be responsible for the states that we define as being "instinctual". Our structure will contain a small set of basic instincts, dealing with very specific cases; basically we will define a single *rational* state and a small set of *instinctual* states.

After the Agent receives data from the belief simulation, this is passed to the BDI system for interpretation. Once evaluated, the FSM is able to judge the state that the Agent is currently in. If the Agents' state is recognised as "instinctual", the FSM seizes control from the BDI. Once the Agent's state

Proceedings of the International Symposium on Evolving Intelligent Systems, Plamen Angelov, Dimitar Filev, and Nikola Kasabov (Eds.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

has changed to rational, the belief simulation system generates a new set of beliefs and acts upon them.

IV. Analysis of Reaction-Adaptive method

When analysing our hybrid FSM-BDI system we see that by combining these methods in the correct way we can produce a technique which is more efficient computationally, as well as more effective in simulating intelligent behavior and enables the system to evolve during its life.

We will define a test scenario in which to place a group of Agents to illustrate and evaluate our novel methodology.

A. Test Scenario

For this analysis, a number of Agents will be placed at random in a virtual environment; each of these Agents has a basic need for social interaction. In addition to this basic need, they follow the instinctual desire to avoid outbreaks of fire. Each Agent is given 4 traits (Wealth, Education, Authority and Fitness), which will each be assigned a value between 0 and 1.

At this stage we will simplify the Agents' traits to this [0-1] range rather than formal fuzzy sets as this is a known technique we have borrowed from other research and does not form part of our novel approach. In actual implementations we can formalize this fuzziness and the evolving capabilities will remain the same in principle.

By varying the degree of *simplified-membership* of each trait that every Agent has, we are able to create Agents that have different degrees of compatibility with each other. When created, each of the Agents is free to move about within the virtual environment at a certain speed. As the Agents move they look in front of them with a limited visual radius and focal length. Any other Agents that they encounter within this visual range they should approach and "interact" with. This interaction allows for a calculation of how compatible each Agent is with each other. This is a simple calculation based on comparing the degree of each trait each Agent has:

```
Compatibility = {                                    (1)
                  1 if δTraits ≤ 1
                  0 if δTraits > 1
                }
                Where:
δWealth     = Agent1.Wealth - Agent2.Wealth
δEducation  = Agent1.Education - Agent2.Education
δAuthority  = Agent1.Authority - Agent2.Authority
δFitness    = Agent1.Fitness - Agent2.Fitness
δTraits     =
  δWealth+δEducation+δAuthority+δFitness
```

If the other Agent is "compatible", this Agent remains stationary as it has completed its task of finding a compatible Agent. If the other Agent is not compatible, this Agent moves off in search of other Agents.

At randomly spaced intervals a fire breaks out at a random location in the virtual world. Any Agent near to the fire should then move away from it. Once they have moved away they return to their default searching behavior.

B. Modelling of Test Scenario

As an initial test, we can model the different behaviour in a classical way using each one of the FSM or BDI techniques.

1) FSM Based Model.

This model demonstrates all the possible states that our Agent can enter in our test scenario, as well as determining the transitions that can occur to move the Agent between states. In every state it is possible to transition to the "Escaping Fire" state, and it should be noted that the addition of any further states would cause the model to become increasingly complex, as those states would also need to transition to "Escaping Fire". It is important to consider as well that we may not have included all the possible states that the Agent should be able to enter; without repeated tests of the scenario, not all possibilities may have presented themselves. Fig. 1 shows such a basic static behavioural model.

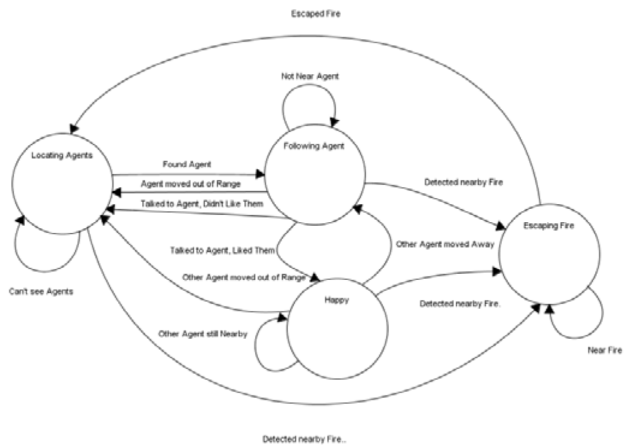

Figure 1. FSM model for Implementing our test scenario.

2) BDI Based Model.

To implement our test scenario using BDI the Agent would need to have the following BDI components:

<u>Beliefs</u>
- Location of other Agents
- Compatibility with other Agents
- Location of Fire
  - Proximity to Fire

<u>Desires</u>
- Socialise with others
- Avoid Fire

<u>Intentions</u>
- Move to another Agent
- Talk to/determine compatibility with other Agents
- Move away from Fire

This model does not deal with the complexities of adding values of "Commitment" to each Intention. It should also be noted that the Intentions listed here are conceptual in basis, as a BDI Agent only has one Intention at any time, selected from its Desires. We also do not deal with the priority of any Desire; without this an Agent can follow a Desire blindly when another of its Desires should clearly take precedent.

While this model is not complex, implementing it highlights at least one of the potential limitations of such a system; namely that the computational cost of each BDI Agent is significantly higher than for a FSM implementation [14].

3) Self-adaptation.

From the FSM model presented in 1) we can see that adapting to an additional state becomes a major problem. Even if the ethos of FSM is ignored, the inclusion of an additional state becomes a matter of exhaustively analysing the relation and transitions between all states with each other.

This presents two main problems for an evolvable system:

- Scalability becomes a serious issue and with every new *attempt* to adapt, the computational load increases exponentially
- Analysis of relations and transitions, as this is a very complex process, normally done by a human designer. It would require a whole *Intelligent Analysis tool* to pursue this task!

The BDI model presented in 2) presents its own set of limitations in this respect. On the positive side, as the evaluation of every layer of B, D and I are sequential, real-time adaptation increases the computational cost only linearly; even if the computational cost itself is heavier than that for FSMs. This is a clear advantage over FSM.

On the other hand, as every component of a layer is a parallel condition, and independent of the condition in the other layers, evolving to a new condition does not require a very thorough analysis of the holistic interaction of every component... so far an additional advantage over FSM.

But this is only valid as long as the prioritization relationships between behaviours, and between desires relative to behaviours, and between intentions relative to desires are ALL non-critical. If this is not the case (as it would always be in a BDI only implementation), then there is a serious complexity issue in terms of analysing these relationships at an online adaptation stage.

Therefore, with any of the alternatives in 1) and 2) we have not only scalability and performance issues, but it becomes impractical to enable an evolvable system.

4) Proposed Evovable Reactive-Adaptive Model.

Our hybrid model will combine the most appropriate components from both FSM and BDI in order to reduce the overall complexity of the system as well as reducing the computational cost of running the test. In addition, if the structure is carefully formed, we can enable as well an evolvable system. Fig. 2 shows the FSM component of the same model presented in 1) and 2) earlier in this section.

This FSM component deals with situations where we believe an "instinctual behaviour" is necessary to resolve the situation. In this case the need for an Agent to avoid nearby fires has been defined in this category. The "Escaping Fire" is an instinctual state; the "Not Escaping" is *the* rational state. At this point is it important to clarify that there can be more than one instinctual state, while these all gravitate around a single rational state, therefore *the* rational state could be just called that.

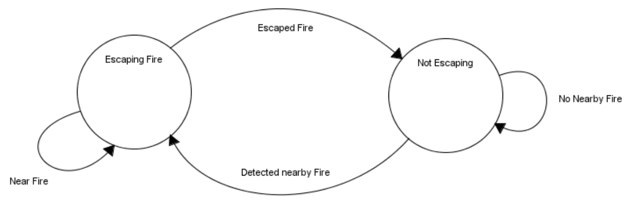

Figure 2. FSM model for Implementing our hybrid test scenario.

Following from the rational state, in this model, the BDI system handles any other situations when the FSM is in the "Not Escaping" state. At this point, ALL the BDI components fall into the rational category and their relations are non-critical across all the layers.

<u>Beliefs</u>

- Location of other Agents
- Compatibility with other Agents

<u>Desires</u>

- Socialise with other Agents

<u>Intentions</u>

- Move to other Agents
- Talk to/determine compatibility with other Agents

C. *Analysis of Hybrid Model Effectiveness*

Runtime performance for a FSM system compares favourably to that of a BDI system, especially when the amount of Agents is significant within the simulation environment. Our BDI model maintains a low complexity when compared to the FSM design, and adding more behaviours increases the complexity in a linear fashion [14]. However, it is a complex task to include information regarding the Agents commitment values or its priority values for plans. Without this planning, Agents using the BDI model are subject to situations where they react in an unrealistic manner. The inclusion of this planning, apart from being unreliable, makes the adaptation a prohibitively complex problem.

In our combined model, the FSM method becomes significantly less complex, requiring only the minimum number of states to handle "instinctive behaviour"; in this case we have only 2 states from the original 4. More critically, the number of transitions, where the complexity grows exponentially, has been reduced from 13 to 4. When the FSM is in its rational state, it passes control to the BDI system, which means only one of the models is effectively active at any one time. The BDI model itself is also less complex as it has fewer Desires; we have in total 5 out of the original 9 elements. By keeping control of instinctive behaviours in the FSM, we are able to avoid the higher computation cost of a BDI Agent and react more quickly to situations where it is deemed necessary, without the requirement of deliberating over a BDI system's Desires and checking commitment and priority values.

Moreover, as the BDI system now has absolutely no need to consider priority values for plans, the task of evolving the system for online adaptation becomes feasible using established constructive techniques. The authors have applied these principles to other problems [17], [18] and many others explored by the research community.

The BDI system in our Reactive-Adaptive model has been relieved of the following constrains:

- There are no prioritization relationships between behaviours as the FSM takes over this task at an external point to the entire BDI system
- There are no prioritization relationships between desires relative to behaviours as these are now always independent with no need for a hierarchical structure
- The prioritization relationships between intentions relative to desires are non-critical as all the desired have equal validity and have no hierarchies to follow from the behaviour level.

These characteristics make our BDI section of the structure consistent with a single independent layer of rules, which can now enable an evolvable system as it can support a *flexible* rule based model [19] or an online reward feedback function [17]. These techniques can now be applied without invalidating the decision making process; and without requiring an exhaustive consistency check every time an automatic adaptation occurs.

V. CONCLUSIONS

While existing AI methodologies are capable of creating Agents that behave in a semi-realistic manner, these techniques are often very complex to implement or result in behaviours that can be clearly unrealistic. In addition, their structure and limitations make it impractical to activate a reasonable or effective evolvable adaptation.

We have shown how using a novel hybrid and hierarchical combination of these methods we created a technique that brings the best features of existing and proven methodologies while attempting to avoid their limitations. By analysing these widely used AI methodologies in our test scenario we have been able to present a case that demonstrates how our combination of techniques creates a simpler implementation that results in a more efficient and realistic simulation than using techniques individually.

The main achievement is that a coherent structure suitable for self-adaptation can be enabled using techniques that normally have serious intrinsic limitations to achieve this.

VI. FURTHER WORK

For the purpose of testing both the computational cost and realism of behaviour produced, it will be necessary in the future to implement our hybrid technique in a more complex scenario while providing an automated measure of realism.

REFERENCES

[1] Patel, Purvag and Hexmoor, Henry. Baltimore, “Designing BOTs with BDI agents.” International Symposium on Collaborative Technologies and Systems, 2009. pp. 180-186.

[2] Li, Yifan, Musilek, Petr and Wyard-Scott, L.. “Fuzzy logic in agent-based game design.” Proceedings of the 2004 Annual Meeting of the North American Fuzzy Information Processing Society. pp. 734-739.

[3] Rhalibi, Abdennour El and Merabti, Madjid. “A Hybrid Fuzzy ANN System for Agent Adaptation in a First Person Shooter.”. International Journal of Computer Games Technology, Volume 2008 (2008), Article ID 432365, 18 pages, doi:10.1155/2008/432365.

[4] Rao, Anand S and Georgeff, P Michael, “BDI Agents: From Theory to Practice.” Proceedings of the First International Conference on Multi-Agent Systems, San Francisco, 1995.

[5] Bratman, Michael E. “Intentions, Plans, and Practical Reason”, Massachusetts : Harvard University Press, 1987.

[6] Genesereth, M R and Nilsson, N J. “Logical foundations of artificial intelligence”, Los Altos : Morgan Kaufman Publishers, Inc., 1987.

[7] Braubach, L., Pokahr, A. and Lamersdorf, W. “Jadex: A BDI Reasoning Engine”, Multi-Agent Programming Volume 15. s.l. : Springer US, 2005, pp. 149-174.

[8] Bhargava, Hemant K and Branley, Jr., William C. “Simulating belief systems of autonomous agents” 1995, Decision Support Systems 14, pp. 329-348.

[9] Ingrand, Francois F, Georgeff, Michael P and Rao, Anand S. “An architecture for Real-Time Reasoning and System Control”, 1992, IEEE Expert: Intelligent Systems and their Applications, 1992 pp. 34 – 44.

[10] Busetta, Paolo, P Busetta, R Ronnquist, A Hodgson, A Lucas, “JACK - Components for intelligent agents in JAVA”, AgentLink News Letter. 2 Jan 1999, pp. 2-5.

[11] Saffiotti, A. “The uses of fuzzy logic in autonomous robot navigation”, 1997, Soft Computing - A Fusion of Foundations, Methodologies and Applications, pp. 180-197.

[12] Zadeh, L A and Bellman, R E. “Decision-Making in a Fuzzy Environment” 1970, Management Science, pp. 141-164.

[13] Yue, Billy and de-Byl, Penny., “The state of the art in game AI standardisation”, Proceedings of the 2006 international conference on Game research and development, Perth : Murdoch University, 2006, pp. 41-46.

[14] Bartish, Arran and Thevathayan, Charles. “BDI Agents for Game Development”, Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2. ACM, Bologna, Italy, 2002, pp. 668 – 669.

[15] Fisher, Michael. Heidelberg, “A survey of concurrent MetateM — The language and its applications”, Temporal Logic—Proceedings of the First International Conference (LNAI Volume 827). Springer-Verlag, 1994, pp. 480-505.

[16] Dennis, L A., Farwer, B., Bordini, R. H., & Fisher, M., “A Common Semantic Basis for BDI Languages”, . In *Programming Multi-Agent Systems*. Springer Berlin / Heidelberg, Springer, 2008, (pp. 124-139).

[17] A. Huemer, D. Elizondo, M. Gongora, “A Reward-Value Based Constructive Method for the Autonomous Creation of Machine Controllers”, Lecture Notes in Computer Science, Vera Kurkova, Roman Neruda and Jan Koutnik Eds., Vol. 5164, 2008, pp. 773-782.

[18] M. Gongora, M. Rodas, “Analysis of Organized Asymmetry Development Using Artificial Cellular Differentiation Models”, Proceedings of the 2009 IEEE Workshop on Evolving and Self-Developing Intelligent Systems (ESDIS 2009), Nashville, TN, USA, April 2009.

[19] Angelov, P. P. “Evolving rule-based models: a tool for design of flexible adaptive systems”, Springer, 2002, ISBN: 978-3-7908-1457-6.