Modelling CGFs for Tactical Air-to-Air Combat Training Motivation-Based behaviour and Machine Learning in a Common Architecture

Jan Joris Roessingh, Ph.D., Robbert-Jan Merk, M.Sc., Remco Meiland, M.Sc., Pieter Huibers, M.Sc., Carolynne Montijn, Ph.D.
National Aerospace Laboratory (NLR), Amsterdam
THE NETHERLANDS

Maj. Roel Rijken, M.Sc.
Royal Netherlands Air Force
THE NETHERLANDS

Capt. Tsu Kong Lue, M.Sc.
United States Air Force - Air Force Research Laboratory
USA

smartbandits@nlr.nl

ABSTRACT

The Smart Bandits project in the Netherlands aims at developing Computer Generated Forces (CGF) exhibiting realistic tactical behaviour so as to increase the value of simulation training for fighter pilots. Although the focus lies on demonstrating adversarial behaviour in air-to-air missions, the results are more widely applicable in the simulation domain.

CGF behaviour is traditionally governed by scripts that prescribe pre-determined actions upon a specific set of events. There are certain shortcomings attached to the use of scripts, for instance, the high complexity of scripts when considering full mission scenarios and the rigid and unrealistic behaviour that scripted CGFs tend to exhibit. To overcome these shortcomings, more sophisticated human behaviour models, combined with state-of-the-art Artificial Intelligence (AI) techniques are required. The Smart Bandits project explores the possibilities of applying these AI techniques.

This paper explains the principal architecture that bridges the gap between theoretical behaviour models and their practical implementation in CGFs for fighter training purposes. The training environment in which the CGF are tested consists of four networked F-16 fighter aircraft simulators. This setup is capable for providing experimental training to pilots for combat against enemy fighter formations (in the form of intelligent CGFs). The architecture is generic in the sense that it can cater for various human behaviour models, differing conceptually from each other in their use of AI techniques, the internal representation of their cognition, and their learning capabilities. Behaviour models based on cognitive theory (e.g. on theories of situational awareness, theory of mind, intuition and surprise) and behavioural models based on machine learning techniques are actually embedded into this architecture.

1.0 INTRODUCTION

Tactical training of fighter pilots in simulators is already widely used. An essential feature of the training of tactics is the presence of participants, other than the trainees. These participants may be team mates, e.g. other fighters in the formation, supporting forces, e.g. forward air controllers, neutral forces, e.g. civilians, or enemy forces, such as adversary fighters. In simulations, the roles of these participants can be performed by humans, Semi-Automated Forces (SAFs) or CGFs. SAFs have some functionality to perform role-related tasks, such that multiple virtual entities can be controlled by one human. However, the use of human experts to participate in tactical simulations may neither be cost-effective, nor operationally effective. First, these human participants are expensive assets. Second, as the simulation is not meant to provide training to them, they could be used somewhere else. Therefore, it is more effective to perform these roles by CGFs, insofar these CGFs are capable of performing these roles in an adequate manner.

However, the current state-of-the-art of CGFs is in many cases inadequate for tactical training purposes, because of their behavioural simplicity. Apart from aforementioned SAFs, four categories of CGF-behaviour can be distinguished (Roessingh, Merk & Montijn, 2011):

1) Non-responsive behaviour, in which the CGF behaves according to a pre-determined action sequence, with minimal capability to observe or react to the environment; Such a CGF is, for example, able to follow a route defined by waypoints.

2) Stimulus-Response (S-R) behaviour, in which the CGF, in response to a certain set of stimuli or inputs from the environment, always exhibits a consistent behaviour; Such a CGF is, for example, able to intercept an aircraft when the aircraft position can be observed continuously.

3) Delayed Response (DR) behaviour, in which the CGF not only takes into account a current set of stimuli from the environment, but also stimuli from previous moments, which are stored in the CGF’s memory. Such a CGF is, by means of remembering previous positions, able to intercept an aircraft, even though this aircraft is not continuously observable.

4) motivation-based behaviour, which CGF combines S-R and DR behaviour but additionally takes into account its motivational states. These motivational states are the result of *internal* processes and may represent goals, assumptions, expectations, biological and emotive states. Such a CGF could, for example, make the assumption that a targeted aircraft is running low on fuel and that it will return to base. As a consequence, the CGF may decide to abort the interception. Alternatively, the CGF may anticipate the route change of the aircraft and decide to intercept the aircraft at a more favourable position.

A characteristic of the CGF that so far is not included in the discussion is learning behaviour or adaptive behaviour (in the sense of Russell and Norvig, 2003). CGFs that exhibit behaviour that is either S-R, DR or motivation-based, may be extended with the capability to adapt this behaviour on the basis of Machine Learning (ML). ML-techniques enable the development of CGFs that are better tailored to the expertise of the trainee. Also, ML-techniques prevent the painstaking development of a set of rules (for example *‘if-then* rules’) that need to be derived for each specific problem or situation to be resolved, based on the manual elicitation of operational expertise that is largely implicit and not simply explicated in terms of logical rules.

The goal of this paper is to illustrate the development of intelligent CGFs within the Smart Bandits project (2010-2013). This project seeks to implant humanlike intelligence into the CGFs that appear in simulated mission scenarios. With the project Smart Bandits, the Dutch National Aerospace Laboratory (NLR) and the Royal Netherlands Air Force (RNLAF) are aiming to take a significant step forward in the area of simulated tactical fighter pilot training. The central message of this paper is that cognitive modelling is a powerful means to create motivation-based behaviour in CGFs. However to mitigate drawbacks of cognitive modelling, we advocate the additional use of ML techniques. These techniques are essential to reduce knowledge elicitation efforts for the development of agents acting in complex domains. It is demonstrated how different approaches can be combined into hybrid models.

2.0 MODELLING MOTIVATION-BASED BEHAVIOUR

2.1 Smart Bandits models

One approach to generate intelligent behaviour is cognitive modelling. In this approach, computational models are designed to simulate human cognition. Within the Smart Bandits project, three cognitive models have been designed so far: a naturalistic decision making model, a surprise generation model and a situation awareness model. All three models have been evaluated using abstracted scenarios from the air combat domain.

2.1.1 Naturalistic Decision Making

As decision making is a crucial part in generating any intelligent behaviour, a naturalistic decision making model was developed early in the project. The model is inspired by Damasio's Somatic Marker Hypothesis. The Somatic Marker Hypothesis provides a theory on decision making which dedicates a central role to experienced emotions as an intuitive part of decision making while integrating this intuitive part with rational reasoning to form a two-stage decision making process. A description of this model is given in Hoogendoorn, Merk & Treur (2009).

2.1.2 Surprise generation

Surprise is considered a universally experienced human cognitive reaction to unexpected situations with recognisable impact on behaviour. However, there is little attention to the phenomenon of surprise in CGF research and few CGFs have human-like mechanisms for generating surprise intensity and surprised behaviour. This leads to impoverished and unrealistic behaviour of CGFs in situations where humans would react surprised. For air combat this forms a problem as the element of surprise is considered an important factor in military operations by many military experts.

For this reason, a model for generating surprise intensity and its impact on the behaviour has been developed (Merk, 2010). The model is based on various theories and empirical results from cognitive research on human surprise behaviour. Besides the unexpectedness of a situation, other cognitive factors such as the novelty of the situations are factored in.

2.1.3 Situation awareness

An important factor for effective decision making is Situation Awareness (SA). SA is especially important in work domains where the information flow can be quite high and poor decisions may lead to serious consequences. For this reason we designed a model based on Endsley's (1995) three levels of SA: (1) the perception of cues, (2) the comprehension and integration of information and (3) the projection of information into future events.

The basic SA model (see figure 1) used for intelligent CGFs in Smart Bandits (see Hoogendoorn, van Lambalgen & Treur, 2011) consists of five components: (1) observations, (2/3) belief formation of current situation, (4) belief formation of future situation and (5) the mental model. The beliefs on current situations and future situations are activated (receive an activation value) through a threshold function, a technique adopted from the neurological domain. The SA model in figure 1 represents the knowledge of the domain that is used to form the beliefs. Humans use dedicated mental models which represent the relationships between various observations and the formation of beliefs about the environment, which, in turn, direct the further observations to be performed.

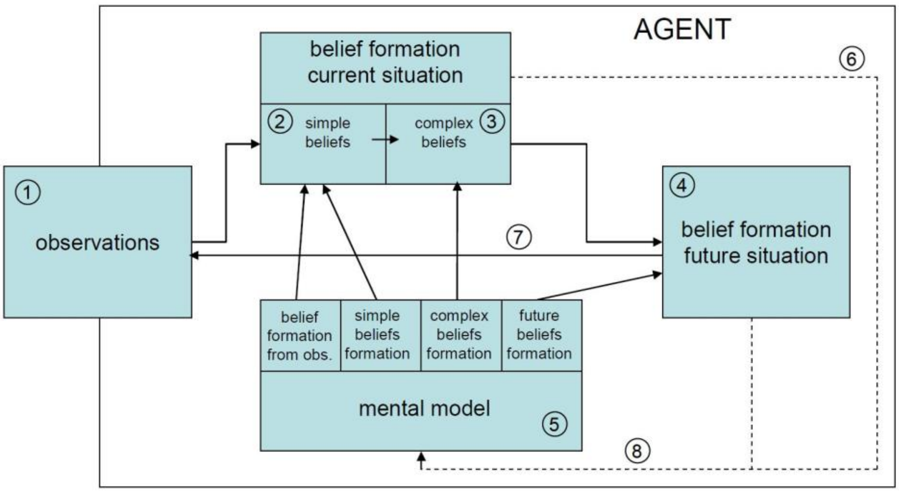
AGENT
belief formation current situation
2 simple beliefs
complex beliefs 3
6
1 observations
7
4 belief formation future situation
belief formation from obs.
simple beliefs formation
complex beliefs formation
future beliefs formation
mental model
5
8

Figure 1: Cognitive model for situation awareness: overview

Another important aspect is the degradation of SA that may arise in demanding circumstances. When time is limited, perception and the integration of cues is impaired leading to incomplete knowledge of the environment. In addition, humans will not always be able to make all necessary observations due to limitations in working memory. Depending on the amount of time available, knowledge on the situation can be further refined by considering less active beliefs. These characteristics are reflected in the behaviour of the intelligent CGFs. A detailed description of the above model can be found in Hoogendoorn, Lambalgen and Treur (2011).

3.0 MACHINE LEARNING

3.1 Reinforcement Learning

A common distinction in machine learning techniques is between supervised and unsupervised learning (e.g. Russel and Norvig, 2003). In supervised learning, after each trial, the agent is presented with the responses that should match the input presentation (also called input example) on which he was supposed to act. The difference between the actual response and the desired response is used to train the agent, just as a trainer or supervisor would make a student aware of the desired response. For example, the agent could learn to fly a manoeuvre by being presented with the correct responses. In unsupervised learning, the agent is merely presented with input examples. The agent has to find hidden structures in the presented examples. Since the examples given to the agent are not accompanied by the responses, there is no difference signal to train the agent. The agent could e.g. learn to distinguish between friendly and enemy tactics.

Reinforcement learning has elements of both aforementioned learning techniques. Rather than being presented with the correct response after each trial, the agent receives feedback from the environment *during* the execution of each trial. Although the feedback may not necessarily represent the correct response for each individual action, the learning technique aims at providing aggregated feedback for the complete trial and therewith reinforcing the correct responses on average. However, this does not guarantee convergence to the correct response. Technical implementation of reinforcement learning is explained in Sutton & Barto (1998).

Reinforcement learning is particularly suited for agent application in simulated environments, because in such environment the agent is able to explore the environment such that a large number of successful and less successful responses can be evaluated. Also, in complex environments, the desired responses, e.g. the best possible opponent engagement tactic, is often unknown. Reinforcement learning provides a technique to improve responses with each trial, therewith discovering better tactics.

A common problem with reinforcement learning is that it requires a large amount of memory to store intermediate calculated values (responses combined with states of the agent in its environment, e.g. its position, speed and heading). In a realistic tactical environment this practically translates to an infinite amount of response-state combinations (‘state-action-space’). In the Smart Bandits project, air-to-air engagements were simulated between two friendly aircraft and two enemy aircraft, the latter two represented by learning agents. In these engagements, the learning agents could only respond in four ways (left, right, forward and shoot). In this example, we stored the state-action-space in a table, which after an acceptable number of learning trials took in the order of 2 gigabytes of memory. Such memory-demand scales-up exponentially with additional parameters. The outrageous memory demands can be diminished by approximating the state-action-space, rather than keeping all the exact values. One way of approximating a large state-action space is by using Neural Networks (NNs), as will be explained in the next section.

3.2 Reinforcement Learning & Neural Networks

In a general sense, a NN (Haykin, 1998) can be considered as a network that can model any mathematical function. In this case, we use NNs to approximate the aforementioned state-action-space. The input for the NN is the current state of the agent in its environment. The output of the NN is a value for each possible action of the agent. The output of the NN is optimized on the basis of the data that is generated by the RL algorithm. The data of the RL algorithm does no longer need to be stored. In fact, the NN is trained using the data that becomes available from the RL algorithm. Where previously we needed 2 gigabytes of memory for resolving a relatively simple air-to-air problem, we now only require approximately 10 kilobytes of data to store the NN knowledge for this problem. This knowledge is represented by the weight-values of the NN. Also, the memory demand does no longer scale up exponentially with complexity of the problem, but only linearly. For this purpose, relatively simple NNs, of the feed-forward type, can be used, rather than recurrent NNs. However, we identified two reasons to develop alternative ML techniques for the type of agents that are needed to act in complex tactical scenarios:

1) Unlike domains, such as resolving problems in games like chess, where the optimal next action is completely determined by the current state of the world, the resolution of tactical problems is characterised by the need to use previous world states. For example, an air-to-air opponent may disappear for some time and may pop-up at a different position, which must be taken into account by the agent. In other words, tactical problems are characterized by imperfect or incomplete knowledge of the environment[1]. RL techniques are known for not being overly robust for these types of problems and we have indeed experienced divergence from the correct response of our agents when confronted with more complex problems.

[1] In more formal terms, the solutions to these problems do not possess the so called Markov Property: the next state *s*' depends on the current state *s* and the decision maker's action *a*. But given *s* and *a*, it is conditionally independent of all previous states and actions.

2) Some realistic tactical problems require memory of the previous states to be taken into account in current decisions. Because of this, RL-based agents are not well suited for realistic tactical problems. For applications in which Delayed Response behaviour or Motivation Based behaviour is required (see chapter 1), RL may not be the preferred technique.

For more advanced problems in the air-to-air domain, *evolutionary techniques* are investigated as an alternative to RL in the next section.

3.3 Evolutionary Techniques & Neural Networks

Artificial autonomous systems are expected to survive and operate in dynamic, complex environments. The specific abilities of an agent, necessary to perform in such an environment, are hard to predict a priori, let alone to specify in detail. Artificial evolution of autonomous systems enables agents to optimize their behaviour in complex, dynamic environments, without the use of detailed prior knowledge of domain experts. Where RL-techniques assume solutions to the problem to possess the Markov Property (see footnote, earlier), evolutionary techniques (Bäck, Fogel, Michalewicz, 1997) are not bound by this constraint and are applicable to a larger set of problems.

Evolutionary techniques use an iterative process to search the fitness landscape in a population of solutions, in this case the solutions to a tactical problem. More successful instances in the populations are selected in a guided[2] random search using parallel processing to achieve the desired solution. Such processes are often inspired by biological mechanisms of evolution, such as mutation and cross-over. Many experiments in evolutionary techniques use neural networks to control the agent. Neural networks offer a smooth search space, are robust to noise, provide generalisation and allow scalability (see Nolfi and Floreano, 2000). Furthermore, network architectures can be evolved or optimised to allow Delayed Responsive behaviour. These characteristics, combined with an evolutionary method to optimise the network, provide an interesting research area for complex, dynamic domains. As an example one could update the weights of the connection strengths of the SA model (see section 2.3) using an evolutionary technique in Smart Bandits.

Since cognitive models like the SA model usually have a large set of interrelated parameters, the determination of their (initial) value, using Subject Matter Experts, is cumbersome, speculative and labour intensive. This creates the need to use evolutionary learning techniques for the appropriate weights for the connections between the aforementioned observations, simple beliefs, complex beliefs and future beliefs. A simplified example of a network representation of the SA model mentioned in section 2.3 is given in figure 2 (taken from Hoogendoorn, van Lambalgen & Treur, 2011).

[2] in the sense of evaluation of a solution by a fitness function.

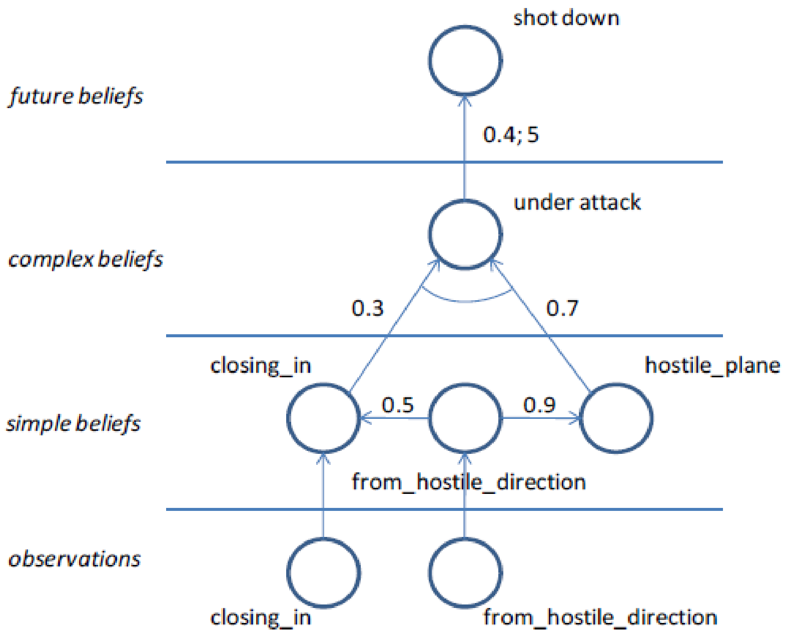

Figure 2: Example model for situation awareness (Hoogendoorn, van Lambalgen & Treur, 2011)

In order to learn the connection weights of the network in figure 2, two different approaches have been utilised (Gini, Hoogendoorn & van Lambalgen, 2011), namely a genetic algorithm application and a dedicated approach based upon the importance of the weights. The latter approach is called the 'Sensitivity Based' approach. Both approaches utilised a fitness function, which expressed how well a solution complied with the desired state. In this case, the fitness could be measured by the difference between actual activation levels and the activation levels estimated by a subject matter expert. The genetic algorithm performed significantly better than the sensitivity-based approach.

3.5 Complicating aspects of ML in multi-agent systems

A Multi-Agent Systems (MASs) falls into one of two categories: centralised or decentralized control based systems. Centralised control systems consist of agents that have a certain degree of autonomy but the overall system is controlled by a unifying strategy, approach or agent to achieve a specific goal. However, despite the overall unifying strategy, an individual agent does not know what the other agents are doing, so the team strategy usually conflicts with the individual agent's strategy at various points within the task. This issue[3] has become the quintessential hurdle for the implementation of MASs in complex settings. Decentralised systems differ from their centralised counterparts by having agents with a higher degree of autonomy, but lack a pre-existing strategy that guides all of the agents. They typically have some form of communication system that allows the agents to develop the needed overall strategy while exploring their environment. The challenge of developing intelligent CGFs, capable of air-to-air tactics, falls straight into the centralised category of MAS environments. As such, the individual agents must be trained together within the same environment. This however inflates the state space by multiples of the number of agents present in the environment. This is a consequence of each agent maintaining its own unique view of the environment, which is captured within its own state space. Nonetheless, there are valid arguments for pursuing the multi-agent approach, particularly for modelling domain-related issues where different flight members may have different, possibly conflicting, goals and incomplete situation awareness.

[3] Otherwise known as the tragedy of the commons (Hardin, 1968).

4.0 ARCHITECTURE

4.1 Simulation Environment

The simulation environment that was used for CGFs in the Smart Bandits project is STAGE ™, a scenario generation and CGF software suite. As a basic scenario tool, STAGE provides us with a level of fidelity and abstraction which is well suited for the tactical air-to-air combat simulations that are currently considered. When a higher level of fidelity in platform, sensor or weapon models is required, the basic functionality provided by STAGE is extended. This ability to extend the basic functionality of the CGF environment is one of the reasons that STAGE was chosen as the main CGF software suite in Smart Bandits.

4.1.1 Middleware (Mediator)

Traditionally, Stimulus-Response (S-R) behaviour (see chapter 1) in agents can be realised in CGF software through the use of scripting and/or basic conditional statements. Combining these simple building blocks often provides a level of credibility to CGF behaviour, which may be adequate for many simulation training exercises. However, for more advanced problems and the associated agent behaviour, including learning behaviour, as described in sections 2 and 3, this method will not suffice. As argued in the previous sections, a wide array of techniques exists for developing CGF behaviour and controlling CGF in a simulation environment. A standard CGF platform does not cater for implementing these different techniques.

In order to use STAGE as the CGF platform in Smart Bandits while delegating the control of the CGFs to external software (i.e. specific software, built using a programming language of choice), an interface was developed through which external software can receive observations from any CGF in STAGE and can command the CGF to perform actions in the simulation environment. This middleware layer (the so-called Mediator in figure 3) communicates in real-time with STAGE through a specific protocol (nCom, Presagis proprietary) and can send and receive the aforementioned observations and actions to and from different agents (possibly distributed over different computers). In order to communicate with the Mediator, external software uses a specific interface, defined in a library, which can easily be linked to the software, e.g. in Java or C++.

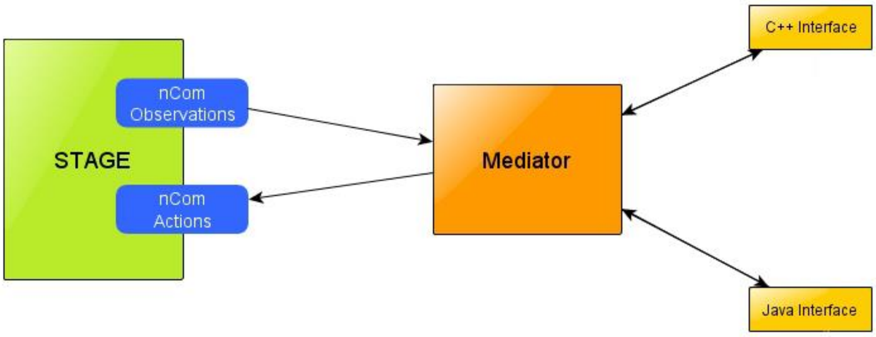

Figure 3: Architecture for including intelligent agents in a Commercial Off-The-Shelf CGF package (STAGE™), intelligent agents can use the C++ or Java interface to communicate with STAGE via the Mediator

5.0 CONCLUSIONS & DISCUSSIONS

A technique for cognitive modelling and various Machine Learning techniques have been presented in this paper. Unfortunately, there does not seem to be one single technique to resolve all emergent tactical problems of intelligent CGFs engaged in air-to-air missions.

Cognitive modelling is a powerful means to create motivation-based behaviour in CGFs. However to mitigate drawbacks of cognitive modelling, we advocate the additional use of Machine Learning techniques. Machine learning techniques are essential to reduce knowledge elicitation efforts for the development of CGFs acting in complex domains. This paper recommends combining different approaches into hybrid models.

The goal of the principal architecture presented here is three-fold:

- decoupling the intelligent CGF models from the tactical fighter simulation,
- facilitating the process of linking models of human behaviour to the aforementioned simulation,
- enabling the distribution of intelligent CGF models at different clients.

Together, these three characteristics enable pursuing the hybrid method.

Within the Smart Bandits project, behaviour and design of intelligent CGFs must be tailored to the training objectives of the tactical training on hand. In this paper we have not dealt explicitly with training requirements. However, implicit in this paper we have assumed that required CGF behaviour for tactical training of operational fighter pilots comprises such aspects as the ability to surprise the human opponent, seemingly random behaviour, i.e. not repetitive in its responses, and realistic from a weapon platform perspective. The intelligent CGFs that have been created so far will be validated against training requirements in the coming project phase (2012/2013). Hence, the two main items for future work within the Smart Bandits project are:

- the implementation of hybrid models, in which cognitive modelling and ML are combined and
- tailoring the behaviour of intelligent CGFs to specific learning objectives or competencies.

6.0 REFERENCES

[1] Bäck, T, Fogel, D.B. & Michalewicz, Z. (1997), *Handbook of Evolutionary Computation*, Oxford University Press, USA, Edition April 17, 1997.

[2] Endsley, M.R., (1995). *Toward a theory of Situation Awareness in dynamic systems*. Human Factors 37(1), 32-64.

[3] Gini, M.L., Hoogendoorn, M., Lambalgen, R.M. van, (2011) *Learning Belief Connections in a Model for Situation Awareness*, In: Kinny, D., Hsu, D. (eds.), Proceedings of the 14th International Conference on Principles and Practice of Multi-Agent Systems, PRIMA'11. Lecture Notes in Artificial Intelligence, Springer Verlag, 2011, to appear.

[4] Hardin, G. (1968), *The Tragedy of the Commons* , Science, 162:1243-1248, 1968

[5] Haykin, S. (1998), *Neural Networks: A Comprehensive Foundation*, Prentice Hall/ IEEE.

[6] Hoogendoorn, M. & Lambalgen, R. van & Treur, J., (2011) *An Integrated Agent Model Addressing Situation Awareness and Functional State in Decision Making*. In: Kinny, D., Hsu, D. (eds.), Proceedings of the 14th International Conference on Principles and Practice of Multi-Agent Systems, PRIMA'11. Lecture Notes in Artificial Intelligence, vol. 7047, pp. 385–397. Springer-Verlag, Berlin Heidelberg, 2011.

[7] Hoogendoorn, M., Merk, R.J., & Treur, J., (2009), *A Decision Making Model Based on Damasio's Somatic Marker Hypothesis*. Presented at the 9th International Conference on Cognitive Modeling, ICCM'09, 2009.

[8] Merk, R.J. (2010) *A computational model on surprise and its effects on agent behaviour in simulated environments*. In: Demazeau, Y., et al. (eds.), Proceedings of the 8th International Conference on Practical Applications of Agents and Multi-Agent Systems, PAAMS'10. Advances in Intelligent and Soft Computing Series, Springer Verlag, 2010.

[9] Nolfi S. & Floreano D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, MA: MIT Press/Bradford Books.

[10] Roessingh, J.J., Merk, R.J. & Montijn, C. (2011*)*, *Modeling the Fighter Pilot's Situation Awareness and Decision Making Ability in Simulated Air-to-Air Engagements*. Proceedings of the 16th International Symposium of Aviation Psychology, Dayton, Ohio, 2-5 May, 2011.

[11] Russell, S.J., Norvig, P. (2003), *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2.

[12] Stone, P. & Veloso, M. (2000), *Multiagent Systems: A Survey from a Machine Learning Perspective*, Autonomous Robots, Volume 8, Number 3,345-383, 2000.

[13] Sutton, R. S. & Barto, A.G. (1998), *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.