Low-level Battle Management Language

A. Alstad, O.M. Mevassvik, M.N. Nielsen, R.A. Løvlid
FFI - Norwegian Defence Research Establishment
PO Box 25, NO-2027 Kjeller, Norway
anders.alstad@ffi.no, ole-martin.mevassvik@ffi.no, martin-n.nielsen@ffi.no, rikke-amilde.lovlid@ffi.no

H.C. Henderson, R.E.J. Jansen, N.M. de Reus
TNO Defence, Security and Safety
PO Box 96864,
2509JG The Hague, The Netherlands
henk.henderson@tno.nl, roger.jansen@tno.nl, nico.dereus@tno.nl

Keywords:
Battle Management Language, Computer Generated Forces, Agent Modelling, Artificial Intelligence, Context-based Reasoning, Belief-desire-intention,

ABSTRACT: *TNO (The Netherlands) and FFI (Norway) are cooperating in extending a COTS Computer Generated Forces (CGF) tool with a Coalition Battle Management Language (C-BML) interface for executing C-BML orders and issuing reports. Due to the lack of satisfactory models for command and control (C2)/combat management in existing CGF tools, TNO and FFI have investigated the use of external agent frameworks. Two different modelling paradigms have been used: Belief-desire-intention (BDI) and Context-based Reasoning (CxBR).*

As part of this work a Low-level Battle Management Language (Low-level BML) has been created for communication between the C2/combat management agents and the CGF tool over High Level Architecture (HLA). The hierarchy of combat management agents decompose a C-BML order into Low-level BML commands and tasks understandable by a CGF tool. The agents also receive Low-level BML events reported by the CGF tool and make use of these for agent behaviour and C-BML reports.

This paper presents the structure of Low-level BML, how it is used and the rationale behind it.

1 Introduction

For the last two years TNO and FFI have investigated possible applications of Battle Management Language (BML) capable Computer Generated Forces (CGF) in the framework of the Anglo-Netherlands-Norwegian Cooperation Programme (ANNCP). The motivation is to enable autonomous simulation of orders created in Command and Control Information Systems (C2ISs). The combination of a C2IS and a BML capable CGF will facilitate training without a large exercise staff and support operational planning. The initial collaborative work is described in [1].

Existing COTS CGF tools covering the land domain are in general not capable of processing and simulating C-BML [2] orders. C-BML captures orders and requests in a command and control (C2) language that typically address units at company level and above. This requires that a C-BML compliant simulation system models military doctrine for higher level units.

The collaboration between TNO and FFI has focused on creating a C-BML capability based on a common COTS CGF tool. Each nation has developed a Multi-Agent System (MAS) that is used in conjunction with MÄK VR-Forces [3]. These MASs are able to process higher level orders (e.g. a battalion order) and decompose them into lower-level commands according to doctrine. To express such lower level commands and reports we have created what we call Low-level BML. The language has been designed to be communicated over existing standards used for communication with CGF tools. The main focus of this paper is to describe this Low-level BML and explain why we need it in addition to C-BML, Real-time Reference Platform (RPR) Federate Object Model (FOM) and Distributed Interactive Simulation (DIS).

There exists similar and related work done by others. Potts et al. have developed their own CGF interface layer which communicates with a CGF tool in a similar way as we do with Low-level BML [4]. However, that solution does not appear to try to create a reusable language for communication with CGF tools, which is our main objective. MyBehaviour for VR-Forces is an example of an extension of a CGF tool that represents a tight coupling between the CGF tool and the higher level reasoning module [5]. Pullen et al. [6] describe a bridging application between systems using Military Scenario Definition Language (MSDL)/C-BML and VR-Forces. This bridging application is able to convert MSDL and C-BML documents into calls to the C++ Remote Control Interface provided by VR-Forces. It appears this bridging solution does not address planning and executing tasks for higher level units.

We will start by describing the background for our work in section 2. Section 3 will discuss why we are developing Low-level BML, and the design of the language is explained in section 4. The experiments conducted by FFI and TNO are presented in section 5. Section 6 contains conclusions and future work.

2 Background

TNO and FFI have participated in NATO Modelling and Simulation Group (MSG) research activities on interoperability between C2ISs and simulation systems since 2005. During NATO MSG-048 "C-BML" [7]-[9] the focus of both the Netherlands and Norway was on C-BML interface to their national C2ISs [10]. The successor NATO MSG-085 "Standardization for C2 to Simulation Interoperability" [11]-[13] is working to advance the C2-simulation interoperability domain towards an operational capability. MSG-085 also investigates how MSDL [14] can be used in concert with C-BML for simulation system initialization.

During MSG-048 both TNO and FFI realized that they needed a C-BML capable simulation system implementing national doctrine and tactics. A national C2-simulation capability was needed both for use in MSG-085 experiments and to explore how a C-BML and MSDL capability can be used for training and decision support for the Netherlands and Norwegian Army.

Since fall 2010 TNO and FFI have collaborated on implementing a national BML capable CGF under the umbrella of ANNCP. The main activities of this collaboration are:

1. To share knowledge on the experience of using different agent modelling paradigms for developing a MAS.
2. To develop a common interface between a CGF tool and a MAS and share the software implementing this interface.

This paper describes the results of the second joint activity, the Low-level BML.

TNO and FFI are currently also involved in NATO MSG-106, which amongst other activities are extending the NATO Education and Training Network (NETN) FOM. The NETN FOM is based on the RPR FOM and includes many HLA Evolved compliant FOM modules. One of the modules being developed is a C-BML FOM module.

3 Autonomous Control of a CGF

C2ISs use MSDL for initialization and C-BML for assigning tasks and receiving reports from higher level units like battalions and companies. As illustrated in figure 1, future CGF tools should directly provide simulation services to C2ISs by implementing C-BML and MSDL capabilities.

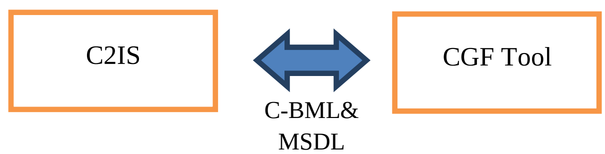

Figure 1. Data exchange between a C2IS and a CGF tool capable of C-BML & MSDL

COTS CGF tools like MÄK VR-Forces [3] and CAE STAGE [15] currently do not provide C-BML or MSDL capabilities. There are however exceptions, MASA SWORD [16] is an example of an aggregate entity CGF tool with BML support. The main reason most CGF tools lack support for C-BML is of course that C-BML is a standard under development, but also that most available CGF tools have no or little modelling of military doctrine for higher level units. A complicating factor is that high level behaviour typically is dependent of a nationally specific doctrine.

In case of a battalion order a BML capable CGF tool must for example be able to execute a company task to occupy a certain area. Managing such a task involves modelling what it means for the individual platoons, vehicles and humans in the company. Most CGF tools are not able to decompose such high level C-BML tasks, but need simple tasks for each individual entity. The other way around, individual entities cannot send reports at the organization level (e.g. platoon) the C2IS requires.

A C-BML compliant simulation should fulfil the following set of requirements:

- Receive C-BML orders and requests from a C-BML infrastructure.
- Model military doctrine for higher echelon units in a military organization reflecting the purpose of the simulation. This includes planning and executing received orders and requests for any unit in the organization hierarchy.
- Publish C-BML reports to a C-BML infrastructure of the perceived truth and ground truth of the simulated forces. It should be configurable which organization level reports should be produced at.

The chosen approach has been to create a hierarchical Multi-Agent System (MAS) that models the doctrine for the leaders in military organizations. Both the MAS developed by TNO and the MAS developed by FFI have one agent for every unit in the organization, i.e. one agent for each platoon, company, etc. Each agent represents the leader for the corresponding unit. The systems receive incoming C-BML orders and translate them into agent tasks that are assigned to the tasked agents. Each agent then plans its tasks and executes them by assigning tasks to subordinates. The lowest level agents command units represented in the CGF tool. Figure 2 shows a MAS in between the C2IS and the CGF tool.

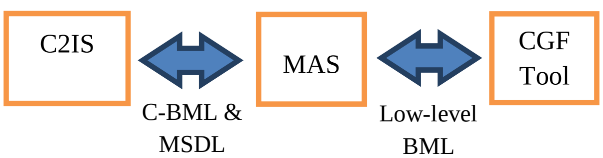

Figure 2 . Data exchange between a C2IS and a CGF tool using MAS

CGF tools simulate the physical representation of units. Platforms simulated in a CGF tool are referred to as "entities", while platoons and other higher level units are referred to as "aggregated entities".

Currently both TNO and FFI have chosen to have platoons as the lowest level of units in the MAS. This is because the capabilities provided by VR-Forces make it more suitable to task platoon aggregates than single entities. To support the agent decision making, the platoon aggregates in VR-Forces produce reports that are sent to the corresponding platoon agents in the MAS. The platoon agents report to their superiors and so on.

In order to have a generic and reusable way for agents in a MAS to communicate with CGF (aggregated) entities, Low-level BML was created. The requirements that guided the design of Low-level BML were:

- Low-level BML should be a language that reflects the capabilities commonly found in COTS CGF tools.
- Low-level BML should be independent of one specific COTS CGF tool and one specific agent framework or agent modelling paradigm.
- Low-level BML should contain compact low-level tasks and commands that easily can be interpreted and executed by CGF tools. This is in contrast to C-BML orders that typically require more advanced processing to allow planning and collaboration according to doctrine.
- Low-level BML should be independent of any specific doctrine or tactics.
- Low-level BML should define entity status and logistic reports. These reports are necessary for the agent decision making and for producing C-BML reports.
- Low-level BML should support HLA and/or DIS. This because HLA and DIS is the most generic and standardized ways of communicating within simulation systems.

4 Low-level BML

Low-level BML is designed to enable effective and compact communication between agents in a MAS and (aggregated) entities in a CGF tool. The language is created to be independent of both the MAS and the CGF tool. As Low-level BML is meant to control CGF entities, it can also be considered as a generic remote control language for CGF tools.

4.1 Language Constructs

The language consists of three main parts:

1. commands used by the MAS to instruct CGF entities,
2. reports from the CGF entities related to task status and status of the tactical environment used by the agents to perform higher level reasoning and
3. scenario management functions used by the MAS to initialize the CGF tool.

All commands are sent from agents to CGF entities (table 1), scenario management functions are sent to the CGF tool (table 3), while events flow from CGF entities to the agents (table 2). The following tables describe the different Low-level BML constructs.

Table 1. Commands from agents to CGF entities

Command	Description
Move to	Tasks a (aggregated) entity to move to a predefined point, line or area
Move to location	Tasks a (aggregated) entity to move to the specified location coordinates
Move along route	Tasks a (aggregated) entity to follow a predefined route
Move into formation	Tasks an aggregate entity to move into the given formation at the given location coordinates, with the given heading
Follow entity	Tasks a (aggregated) entity to follow another entity
Fire at	Tasks a (aggregated) entity to fire at another specified entity
Wait	Tasks a (aggregated) entity to halt whatever it is doing
Set camouflage on/off	Turn the camouflage on/off
Set rules of engagement	Change the current rules of engagement
Subscribe to events	Control which events and reports the (aggregated) entity should generate

Table 2. Reports sent from CGF entities to agents

Report	Description
Spot report	The entity has spotted an unknown, neutral or enemy entity
Entity in area	The entity has entered or exited an area
Entity crossed line	The entity has crossed from one side of a line to the other
Under fire	The entity is under fire
Task completed report	The entity has completed the last task assigned to it
Entity fuel	The entity's current amount of fuel
Entity ammunition	The entity's current amount of ammunition

Table 3. Scenario management functions sent from a MAS to a CGF tool

Scenario management	Description
Create entity	Create a CGF entity
Create aggregate	Create an CGF aggregated entity
Create area	Create an area in the CGF tool
Create phase line	Create an phase line in the CGF tool
Create route	Create a route in the CGF tool

4.2 Comparison to C-BML

While C-BML typically will be communicated between C2ISs and simulation tools or robotic forces, Low-level BML is designed to be communicated between models of command and control/combat management and CGF-entities. As C-BML can be used to control single entities in a simulation (e.g. a UAV) and robotic forces, there is a partial overlap between the two languages. The major differences are found in the granularity of the languages and that Low-level BML does not require the receiving entity to have knowledge of military doctrine.

Figure 3 shows a sequence diagram of how a task in a C-BML order might be decomposed into multiple Low-level BML commands. While some of the illustrated Low-level BML commands (e.g. "move to area 102") could have been represented with C-BML, many of them cannot. An example of difference is that formation is not defined in C-BML.

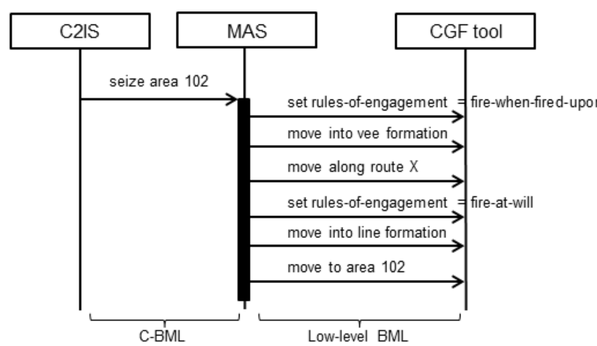

Figure 3. Differences in granularities between C-BML orders and Low-level BML tasks

In contrast to tasks in C-BML orders, a Low-level BML task does not include why- and when-elements. This is because Low-level BML tasks currently are designed to be executed immediately by the receiving (aggregated) entity. The temporal relations between tasks are controlled by the MAS.

The why-element is not needed by the CGF entities, as the task intent should be interpreted by the MAS. The MAS monitors that the CGF entities execute the task to fulfil the higher level goals.

4.3 Representation

As mentioned in the problem description, it should be possible to communicate Low-level BML interactions over HLA and/or DIS.

DIS defines both a wire format and an information exchange model. To be able to use standards compliant DIS, Low-level BML has to be embedded in existing DIS protocol data units (PDU). It is possible to extend the DIS information exchange model with Low-level BML. This will however break compatibility with the standard.

The basis for the Low-level BML FOM module is the RPR FOM 2.0 draft 17 [17]. The RPR FOM is compatible with DIS and is extendable. Extensions will not break compatibility with the HLA standard or with existing federates. However, the extensions will break compatibility with DIS.

TNO and FFI first chose to extend the RPR FOM with custom Low-level BML interactions. This is the preferred way of adding extensions to HLA based FOMs. Later FFI chose to embed Low-level BML within the existing RPR FOM ApplicationSpecificRadioSignal-interaction. Both methods worked satisfactory.

FFI used Google Protocol Buffers (ProtoBuf) [18] to serialize the Low-level BML interactions and wrapped them in the mentioned radio signals. This was done for compatibility with DIS and because FFI found it easier and quicker to modify and implement Low-level BML this way than making a RPR FOM extension.

5 Proof of Concept Implementations

As described in chapter 2, TNO and FFI have investigated different paradigms for the modelling of C2/combat management: BDI [19] and CxBR [20]-[21]. This has resulted in two different solutions. However both are based on the use of a MAS which exchange Low-level BML messages with VR-Forces. The following sections describe these two solutions and the experiments that have been conducted with them.

5.1 FFI Experiment

FFI chose to create an agent framework from scratch using CxBR to model the agents' behaviour. The architecture used for the Norwegian experiment is illustrated in figure 4.

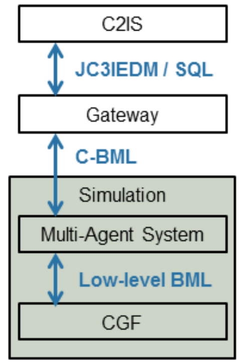

Figure 4. The architecture utilized by FFI.

In the Norwegian experiment the combination of NORTaC-C2IS and FFI C2-gateway was used to initialize the simulation, to send orders to the simulation and to receive reports from the simulation. NORTaC-C2IS stores its data in a Joint Consultation, Command and Control Information Exchange Data Model (JC3IEDM) compliant database. The FFI C2-gateway uses SQL to extract and generate MSDL documents and C-BML order documents from that database. The MSDL and C-BML documents were sent to the MAS through a FFI-developed message broker. For this experiment we configured the MAS to produce C-BML position reports for each simulated company. These C-BML reports were sent to the FFI C2-gateway through the message broker and inserted into the JC3IEDM database through SQL.

The MSDL documents generated by the FFI C2-gateway contained the friendly order of battle and the initial position for each friendly platoon. These MSDL documents were used by the MAS to both instantiate the agents and to create and position aggregated entities in VR-Forces. The aggregated entity creation was done through Low-level BML. The C-BML and MSDL capability is described in [10]-[12]. Figure 5 shows a systems view of the experiment setup.

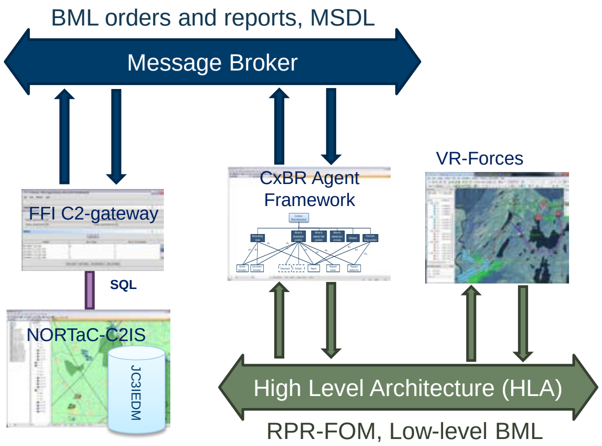

Figure 5. Systems view for the Norwegian setup.

In order to synchronize the agent behaviour model with the simulated entities, a custom tick-interaction was sent from VR-Forces to the MAS. In the future we plan to use HLA time management instead of a custom tick-interaction.

As mentioned earlier, FFI used the RPR FOM ApplicationSpecificRadioSignal-interaction to send ProtoBuf-encoded Low-level BML messages. The VR-Forces integration for sending and receiving such Low-level BML messages was done as a VR-Forces plug-in. As VR-Forces has been used through the whole project, Low-level BML mapped well to existing tasks and reports in VR-Forces.

5.2 TNO Experiment

TNO has conducted an experiment with a MAS based on the Belief-Desire-Intention (BDI) paradigm. This paradigm is based on the theory of human practical reasoning where Beliefs and Desires are mental attitudes concerned with action and Intention is a conduct-controlling attitude dealing with commitment. It was developed by Bratman [19]. This paradigm is used to model agents which need to have a form of human behavior representation.

TNO has built their MAS on the BDI-framework JADEX. JADEX is a software framework that facilitates easy intelligent agent construction [22].

The system architecture used in the TNO experiment is illustrated in figure 6.

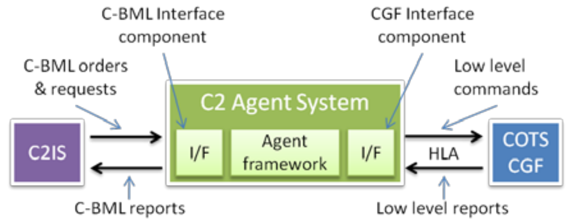

Figure 6. The system architecture utilized by TNO.

In the experiment the Integrated Staff Information System (ISIS) was used as the C2IS and VR-Forces as the COTS CGF tool. A VR-Forces plugin handles the low-level commands and reports. A description of the experiment of 2011 can be found in [1].

In 2012 the MAS and the VR-Forces plugin were extended to handle most of the commands and reports as described in section 4.1. MSDL import for the MAS was implemented, comparable to the FFI approach.

6 Conclusions

The use of C-BML demands simulation systems to be able to simulate higher level orders. One approach to alleviate this challenge is to extend COTS CGF tools with models of C2/combat management. FFI and TNO have chosen to integrate a COTS CGF tool with a MAS. In order to allow different CGF tools and MASs to operate together, FFI and TNO created Low-level BML.

Low-level BML defines communication constructs at a lower granularity than what is covered by C-BML & MSDL today. Low-level BML can be seen as a generic remote control language that uses the same granularity as most CGF tools. For the translation from MSDL and C-BML to Low-level BML a MAS is needed. In this paper two different methods are presented that illustrates how a MAS can communicate Low-level BML with a CGF tool.

6.1 Future Work

The current set of language constructs described in section 4.1 contains some constructs which are not being used by neither TNO nor FFI. Future work might trim away unused constructs and add new constructs when found necessary. The current language has been used to control ground forces. FFI is planning to also use Low-level BML to control maritime forces and will therefore probably add maritime specific commands and reports to the language.

The only CGF tool used so far is VR-Forces. In future experiments we might test if Low-level BML also works well with other CGF tools.

In NATO MSG-106 probably two new FOM modules related to C-BML will be developed: one that encapsulates an entire C-BML document in an object or interaction (figure 7) and another for the use of C-BML for lower echelons (units, platoons) with explicit objects or interactions representing specific tasks, requests, and reports. These new FOM modules will be part of the NETN FOM module set. This approach has many similarities to Low-level BML and is still being studied in MSG-106.

Figure 7. Example C-BML FOM module.

Furthermore, TNO will investigate alternative data formats and protocols for Low-Level BML in case the CGF tool is not HLA or DIS compliant.

References

[1] N. de Reus et al., “Battle Management Language Capable Computer Generated Forces”, in *Proceedings of the Euro Simulation Interoperability Workshop 2011*, 2011.

[2] Simulation Interoperability Standards Organization, “Standard for: Coalition Battle Management Language (C-BML)”, SISO-STD-011-2012-DRAFT, 4 April 2012

[3] VT MÄK, “Computer Generated Forces – VR-Forces”, http://www.mak.com/products/simulate/computer-generated-forces.html.

[4] J. R. Potts et al., “Subject Matter Expert-Driven Behaviour Modeling Within Simulation”, in *Interservice/Industry Training, Simulation, and Eduction Conference (I/ITSEC) 2010*, 2010.

[5] Antycip, “MyBehavior, Reusable Military Doctrines”, http://www.antycipsimulation.com/solutions/mybehaviour.

[6] M. J. Pullen et al., “An Open Source MSDL/C-BML Interface to VR-Forces”, *in Proceedings of Fall Simulation Interoperability Workshop 2012*, 2012.

[7] M. J. Pullen et al., "NATO MSG-048 Coalition Battle Management Initial Demonstration Lessons Learned and Way Forward", in *Proceedings of Spring Simulation Interoperability Workshop 2008*, 2008.

[8] M. J. Pullen et al., "Adding Reports to Coalition Battle Management Language for NATO MSG-048", in *Joint SISO/SCS European Multi-conference*, 2009.

[9] K. Heffner et al., "NATO MSG-048 C-BML Final Report Summary", in *Proceedings of Fall Simulation Interoperability Workshop 2010*, 2010

[10] N. de Reus et al., "BML-enabling national C2 systems for coupling to Simulation", in *Proceedings of Spring Simulation Interoperability Workshop 2008*, 2008.

[11] M. J. Pullen et al., "MSDL and C-BML Working Toghether for NATO MSG-085", in *Proceedings of Spring Simulation Interoperability Workshop 2012*, 2012.

[12] M. J. Pullen et al., "Technical and Operational Issues in Combining MSDL and C-BML Standards for C2-Simulation Interoperation in MSG-085”, in *Proceedings of MSG-094 Symposium “Transforming Defence through Modelling and Simulation – Opportunities and Challenges”*, 2012.

[13] B. Gautreau et al., “Lessons learned from NMSG-085 CIG Land Operation demonstration”, in *Proceedings of Spring Simulation Interoperability Workshop 2013*, 2013.

[14] Simulation Interoperability Standards Organization, “Standard for: Military Scenario Definition Language”, SISO-STD-007-2008, 2008

[15] Presagis, “STAGE”, http://www.presagis.com/products_services/products/modeling-simulation/simulation/stage/

[16] MASA Group, “MASA SWORD: Automated, aggregated constructive simulation for efficient training and analysis”, http://www.masagroup.net/products/masa-sword/.

[17] IEEE, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification," IEEE,*IEEE Std 1516.1-2000*, 2000.

[18] Google, Protocol Buffers: Google’s Data Interchange Format, http://code.google.com/p/protobuf/.

[19] M. E. Bratman, “Intention, Plans and Practical Reason”, 1997, Harvard University Press, Cambridge, MA

[20] A. J. Gonzales et al., “Formalizing Context-Based Reasoning: A Modeling Paradigm for Representing Tactical Human Behavior”, International Journal of Intelligent Systems, vol. 23, p.822-847 (2008)

[21] A. J. Gonzalez and R. Ahlers, “Context-based representation of intelligent behaviour in training simulations”, *Trans Soc Comp Simul 1998; 15(4): 153-166* (1998)
[22] “Distributed Systems and Information Systems Group”, Jadex - http://jadex-agents.informatik.uni-hamburg.de [09.04.2011]
[23] N. Abdellaoui et al., “Comparative Analysis of Computer Generated Forces’ Artificial Intelligence “, in *Symposium Proceedings RTO-MP-MSG-069 2009*, 2009

Author Biographies

ANDERS ALSTAD is a Senior Scientist at the FFI (Norwegian Defence Research Establishment). His work is within the field of modelling and simulation. His interests are in design and development of distributed information systems.

RIKKE AMILDE LØVLID is a Scientist at FFI (Norwegian Defence Research Establishment), where she works with modelling and simulation. Her research interests include artificial intelligence, machine learning and advanced algorithms.

MARTIN NORMANN NIELSEN is a Scientist at FFI (Norwegian Defence Research Establishment), where he is working with network simulators, distributed simulation technologies and software engineering in general. He completed his M.Sc. in Computer Science at the University of Oslo in 2006. His research interests include modelling and simulation and design and development of distributed systems.

OLE MARTIN MEVASSVIK is a Principal Scientist at the FFI (Norwegian Defence Research Establishment). His research interest is within the area of modelling and simulation, with emphasis on simulation systems interoperability.

HENK HENDERSON is a member of the scientific staff in the technology area Modelling & Simulation and Gaming, working in the Defence, Safety and Security theme in the Netherlands’ TNO. His work focuses on the software engineering aspects of Modelling & Simulation.

ROGER JANSEN is a member of the scientific staff in the Modeling, Simulation and Gaming department at TNO, The Netherlands. He works in the field of distributed simulation and his research interests include distributed computing and simulation interoperability.

NICO DE REUS is a member of the scientific staff in the technology area Modelling & Simulation and Gaming, working in the Defence, Safety and Security theme in the Netherlands’ TNO. His work focuses on the application of Modelling & Simulation and on C2-Simulation interoperability.