Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

TWO APPROACHES TO DEVELOPING A MULTI-AGENT SYSTEM FOR BATTLE COMMAND SIMULATION

Rikke Amilde Løvlid
Anders Alstad
Ole Martin Mevassvik

FFI - Norwegian Defence
Research Establishment
PO Box 25
NO-2027 Kjeller
NORWAY

Nico de Reus
Henk Henderson
Bob van der Vecht
Torec Luik

TNO - Netherlands Organisation for
Applied Scientific Research
PO Box 96864
2509JG The Hague
THE NETHERLANDS

ABSTRACT

In the military, Command and Control Information Systems (C2ISs) are used for issuing commands to subordinate units. In training or decision support, simulations are used instead of live military forces. The Coalition Battle Management Language (C-BML) is currently being developed as an interface language between C2ISs, simulations and robotic forces, and Norway (FFI) and the Netherlands (TNO) are working towards extending their national C2ISs and a COTS simulation system with a C-BML interface. One of the challenges encountered during this work is the fact that the orders issued by the C2IS are at company level and above, while most available simulation systems are designed to execute platoon and single platform tasks. Both FFI and TNO are investigating using a multi-agent system for decomposing orders to lower level task by simulating national tactics and doctrine. This paper presents and compares our development approaches and agent modeling paradigms.

1 INTRODUCTION

Current command and staff training uses simulation systems that consist of computer generated forces (CGF) in combination with human operators, so-called LOwer CONtrol operators (LOCONs). These LOCONs receive high level tasks, e.g. company level, by voice, paper or in electronic form, transform this into lower level tasks for subordinate units (platoon level and lower), and manually enter this more detailed set of instructions into the simulation system that simulates the combat (Cayirci 2009).

Training events are usually big events where a large number of LOCONs are required next to the instructor staff. Although the general idea is that these lower level operators also benefit from this work, it is usually difficult to train multiple levels because of the different training goals for these levels. If the number of LOCONs could be reduced by (partial) automation of their job, this could greatly enhance the number of training events and consequently mission readiness.

Simulation systems can also be used during operations in planning and mission rehearsal, which even more inhibit the use of a large simulation support staff. For instance, in the planning phase of an operation, simulation systems can be used to support what-if analysis or course of action analysis.

The military simulation applications mentioned above would benefit from a capability that interfaces the C2IS with simulations in a seamless way, minimizing the number of LOCONs necessary. In order to realize this seamless integration, an order made in a C2IS must be expressed in a standard, unambiguous language, which is interpretable by a simulation system, and the simulated forces must have sufficient knowledge about tactics and doctrine to carry out the order.

978-1-4799-2076-1/13/$31.00 ©2013 IEEE

Since 2005 FFI and TNO have participated in NATO research activities under the NATO Modelling and Simulation Group that has focused on assessing and exploiting the use of a standard Coalition Battle Management Language (C-BML) (Carey et al. 2001). C-BML is being developed by the Simulation Interoperability Standards Organization (SISO) with the aim to define a standardized language for exchanging orders, requests and reports between C2ISs, simulation systems and robotic forces. However, existing COTS CGF systems covering the land domain are in general not capable of processing and simulating C-BML orders and requests. C-BML captures orders and requests in a language that typically addresses units at company level and above. This requires that a C-BML compliant simulation systems models military tactics and doctrine for higher level units. In 2011 FFI and TNO started cooperating to create a C-BML capable CGF system in the framework of the Anglo-Netherlands-Norwegian Cooperation Program (ANNCP). The motivation was to enable autonomous simulated execution of orders created in a C2IS (de Reus et al. 2011).

The collaboration between TNO and FFI has focused on creating a C-BML capability based on a common COTS CGF system. Each research group has developed a Multi-Agent System (MAS) that is used in conjunction with VT MÄK VR-Forces. These MASs have been designed to process higher-level (battalion) orders and decompose them into lower-level commands according to military doctrine. These low-level commands have then been sent to and simulated by VR-Forces, which in return has provided simulation state and low-level reports back to the MAS. Based on the received simulation state and low-level reports, the MAS conducts tactical decision making, in addition to providing high-level C-BML reports back to the C2IS. VR-Forces combined with a MAS thus becomes a C-BML capable simulation system. Figure 1 illustrates the basic system architecture.

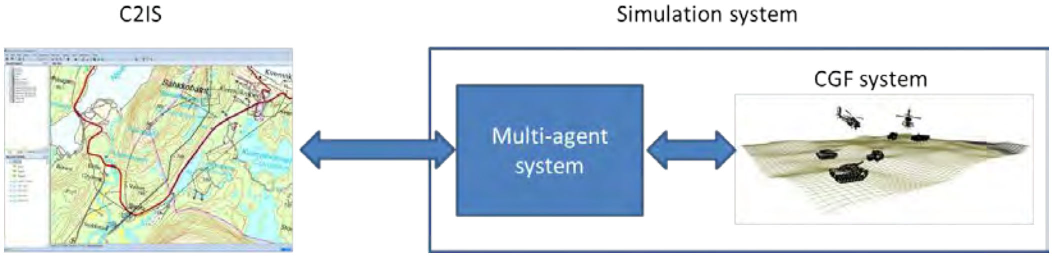

Figure 1: System Architecture.

The focus in this paper is to describe and compare the two approaches for building a MAS capable of simulating battle command. TNO has used JADEX, a Belief-Desire-Intention (BDI) reasoning engine for intelligent agents (Rao and Georgeff 1995), and has modeled the behavior based on combat instruction sets (CISs) developed by the Netherlands Army. FFI on the other hand, has built their MAS from scratch based on the Context-Based Reasoning (CxBR) paradigm (Gonzales et al. 2008) and used an example order to develop behavioral models with support from Subject Matter Experts (SMEs).

Both the BDI and the CxBR approach seems suitable for modeling battle command agents. BDI is a well-known paradigm for agent behavior modeling, and with several available frameworks it appears easy to start with. CxBR on the other hand is more modular, which might make it easier to understand the whole behavioral model and avoid inconsistencies when the action space grows large.

We will start by describing the two agent behavioral modeling paradigms and continue in section 3 with how they were applied to the battle command agents. We discuss our experiences in section 4, and end with conclusions and further work in section 5.

2 BACKGROUND THEORY

An *agent* is an autonomous entity that observes through sensors and acts upon its environment using actuators in order to meet its design objectives. To be called intelligent, an agent also has to be reactive, proactive and social; meaning it must be able to react to changes in the environment, pursue goals and be able to communicate with other agents (Wooldridge 2004).

A MAS contains a number of agents that interact with one another through communication. Each agent will influence different parts of the environment, and the agents are linked by some kind of organizational relationship.

We use a hierarchy of intelligent agents to decompose orders from company level tasks into low level CGF commands. The MASs consist of one agent for the battalion, one for each company in the battalion and one for each platoon. The agents represent the commanders with staff of these military units, and we model the most important decisions that these staffs make in certain situations.

The idea is that the agent paradigm makes the design intuitive and easily understandable by subject matter exerts, which is crucial for the users of such a system (Cares 2002). Two agent behavior modeling paradigms were applied, CxBR and BDI. These are explained next.

2.1 Context-Based Reasoning Agents

CxBR is a reasoning paradigm for representation of tactical behavior in agents (Gonzales and Ahlers 1998; Gallagher et al. 2000; Gonzales et al. 2008). The motivation behind CxBR is the realization that people only use a fraction of their knowledge at any given time. The idea is to divide the knowledge into contexts in order to limit the number of possibilities for the action selection process. For example, an agent representing a military platoon will require a different set of capabilities and knowledge when it is performing an attack versus when moving along a road.

The contexts are organized in a context hierarchy consisting of a mission context, major and minor contexts. The *mission context* is a purely descriptive context, and does not control the agent. It contains information defining the current overall mission of the agent. Such information includes the objective and a plan for reaching it together with parameters like objective area, phase-line, route etc., and a context map, where all possible transitions between major contexts are defined.

Major contexts are the primary element of agent control and represent the major situations an agent can face. There is only one major context in control of the agent at any time, called the active context. A major context basically contains three kinds of knowledge: *action knowledge*, *transition knowledge* and *declarative knowledge*.

Action knowledge is knowledge about how the agent should behave in the context. Since our agents are battle command agents, the actions are commands to the subordinates, reports to the superior and possibly reports and/or requests to other agents at the same level. If a part of the behavior is shared with other major contexts, this behavior should be expressed as a *minor context,* which controls the agent for a short period of time. There can be unlimited levels of minor contexts, but one or zero should be sufficient.

As the situation and environment for an agent change, the agent will transition accordingly between contexts. Knowledge of when to switch to another context is collected in the transition knowledge. This includes recognition of a situation leading to deactivation of the active context and activation of a better suited context. This knowledge can be contained in transition rules, with criteria for when the agent makes the transitions defined in the context map. The transition rules consist of both general, doctrinal reactions and scenario specific, planned transitions, and should include transition to a default context when no other context is applicable.

Declarative knowledge includes context specific parameters and possible minor contexts. The contexts can also access all information available to the agent, which includes a global fact base, with information visible to all agents, and a local fact base containing information that only the particular agent is aware of.

2.2 Belief-Desire-Intention Agents

BDI is a well-known paradigm for agent behavior modeling with its roots in folk-psychology (Rao and Georgeff 1995). Using beliefs about the environment and desires about what should be achieved, BDI provides the means for agents to be modeled in a way related to human thinking. An abstract representation of reasoning in a BDI agent is illustrated in Figure 2.

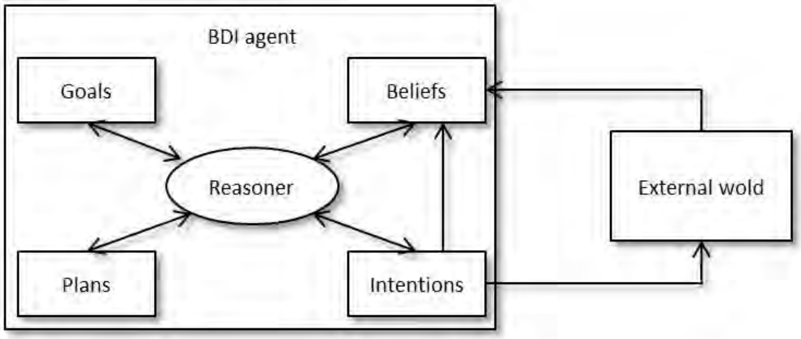

Figure 2: BDI reasoning paradigm.

Knowledge gained by the agent from reasoning or sensing are stored as *beliefs*, contrary to CxBR, in BDI the beliefs are only local: these are the facts the agent subjectively believes to be true. An agent can respond to new beliefs, e.g. by performing actions. Thus beliefs allow for the agent to react to changes in its environment.

However, the agent can also act in a goal-driven fashion because of the agent's *desires*, which are the goals that the agent wants to achieve. These desires might be defined in the form of multiple sub-goals to be achieved or basic actions to be performed. Desires can also be triggered or deactivated based on certain beliefs and they are generally accomplished by executing an action or adjusting a belief.

The third element of the BDI framework is the *intention*, which is the so-called deliberative state of the agent, and refer to the goal the agent is currently pursuing.

The plan database contains plans. A plan is a sequence of actions that aims for a certain goal, and is tailored to a specific situation by specifying beliefs. A general notation is: <Goal> ← <Beliefs | Actions>, which should be understood as: if <Goal> matches the agent’s goals, and <Beliefs> matches the agent’s beliefs, then the <Actions> can be executed. If a plan is selected and will be executed, the goal is removed from the agent’s goal base as it is assumed to be achieved.

In the BDI reasoning process the agent decides upon the actions to execute by analyzing which plans fit to the current goals and beliefs. A typical reasoning process consists of the following steps: 1) Select the plans matching the Goals, 2) select the plans matching the beliefs, 3) select the plan with highest priority / highest utility which then becomes the intention, and 4) execute the action sequence in the plan.

The execution of the action sequence can lead to belief updates or new goals as well as to the execution of actions in the physical world. We distinguish internal and external actions, where internal actions are belief updates or dispatch a sub-goal, and external actions includes sending a message or executing a ‘physical’ action in the environment, such as *move.*

By looping over the four steps from the reasoning process the agent tries to reach its goals in the way that fits best to the current situation, until it gets to the point where there are no more actions to execute. Then it waits until an external event triggers a new goal or a plan. Such external event is typically an observation of some aspects in the environment, or an incoming message from another agent.

BDI has been used in military applications numerous times already (Mcilroy et al. 1996; Rao et al. 1992; Tidhar et al. 1999; van Doesburg et al. 2005), and there are many extensions available.

2.3 Similarities between CxBR and BDI

Both CxBR and BDI are intuitive paradigms for modeling human behavior, and the same behavior can be expressed with both. Barrett and Gonzales compare the two paradigms, and explain what the beliefs, desires and intentions correspond to in the CxBR paradigm (Barrett 2007; Barrett and Gonzales 2011).

CxBR is designed to represent an agent’s knowledge through mission contexts, major contexts and sub contexts. In BDI, this knowledge is in the *Beliefs* of the agent, including what is stored in the global and local fact bases that are accessible in the contexts. The agents’ *Desires* corresponds to the goals of the agent. The mission context includes the highest level goal for an agent. Sub-goals, primarily concerned with accomplishing the missions goal are either implicitly or explicitly contained within the major and minor contexts. During a scenario the mission is carried out with a sequence of major contexts. Therefore the agent's *Intentions* are determined by the agent's active context.

3 MODELING BATTLE COMMAND WITH CXBR AND BDI

In this section we first present the two approaches for modeling and implementing the MAS based on CxBR and BDI separately, before we compare the two solutions in section 3.3.

3.1 Battle Command Agent Modeling using CxBR

The research group at FFI modeled the behavior of the battle command agents using CxBR and implemented a custom multi-agent framework. An example scenario that encompasses an offensive battalion operation was used to limit the behaviors that were modeled and implemented.

Using scenarios in system design is a common approach with several benefits and applications. Since a scenario is a coherent and concrete vision, not an abstract goal or list of requirements, it can for example serve as a common medium for discussing system requirements among different project participants with various backgrounds, generate test cases for validation purposes and help developing more usable computer system by envisioning user patterns (Carroll 1999; Go and Carroll 2004). In the current project the scenario-based approach helped subject matter experts and developers understand each other by providing a clear example to use during discussions about the execution of different tasks. Also, it helped analyzing the requirements of the system and resulted in a prototype which could be used to demonstrate autonomous execution of a realistic order.

The example scenario is part of a larger offensive operation, and the goal of the entire operation is to attack an enemy that is situated further to the north–east of the map show in Figure 3. The plan is to approach the enemy along the road that exits the map near the upper right corner. There is an enemy vanguard at area 102, and this area must be seized before the friendly forces can continue towards the main enemy position. Our example scenario considers this first part of the operation, to seize area 102.

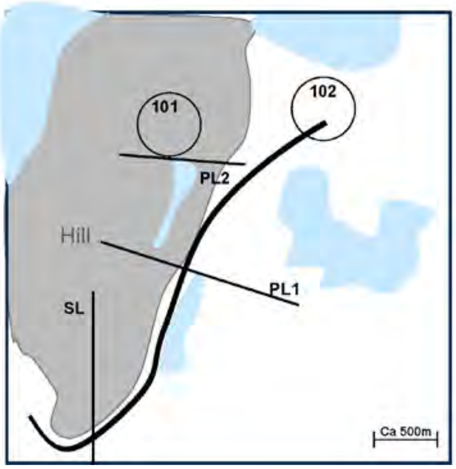

Figure 3: Map of the operation area.

The operation is to be carried out by one battalion consisting of the companies RecceCoy1, MechInfCoy3 and MechInfCoy 4. The battalion operation with task for each of these companies is shown in Table 1. SMEs helped describing how the tasks in the order should be carried out, and how the agents should handle possible events not described in the order.

Table 1: Synchronization matrix.

Company	**Task**		
RecceCoy1	**Reconnoiter** axis from SL to 102	Keep 102 under surveillance	
MechInfCoy3		**Seize** Objective area 101	**Support by fire** MechInfCoy4 (towards 102)
MechInfCoy4			**Seize** Objective area 102

All task verbs in the operational order got a corresponding mission context in CxBR terminology. The mission contexts were planned as a sequence of major contexts by the agent who received the mission. This planning was done in three steps: 1) retrieve a basic plan for the current mission context, 2) adapt the basic plan to the current situation by removing redundant major contexts and specifying when a context in the plan is completed, and 3) replace the general major contexts from the basic plan with more specific versions.

A basic plan was defined for each mission context. For example, for mission context Seize the basic plan was “*Move→Move Cautiously→Attack*”. When adapting the basic plan to the current situation it might change to “*Move Cautioulsy* -When at distance x from observed enemy→ *Attack”*. In the third step the major context *Attack* could for example be replaced by *Fix-Attack* or *Attack-with-follow-and-support*, indicating that the subordinates will get different roles.

Figure 4 illustrates the context map for mission context Seize, including all possible major contexts and both planned transitions (dotted in gray) and transition rules for reactive behavior. Action rules in major context *Regroup* are included to illustrate what can be the behavior inside a major context.

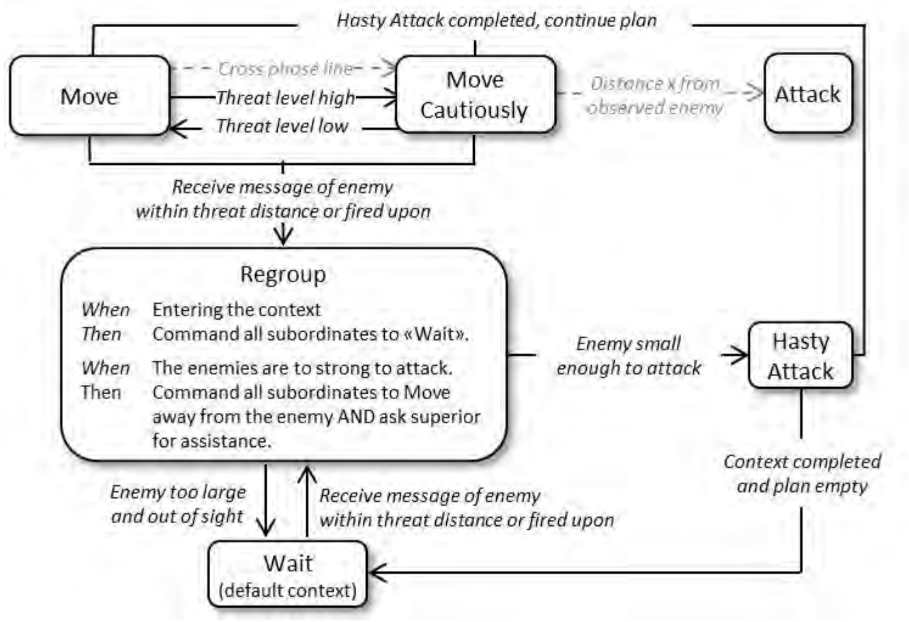
Hasty Attack completed, continue plan
Move
Cross phase line
Threat level high
Threat level low
Move Cautiously
Distance x from observed enemy
Attack
Receive message of enemy within threat distance or fired upon
Regroup
When Entering the context
Then Command all subordinates to «Wait».
When The enemies are to strong to attack.
Then Command all subordinates to Move away from the enemy AND ask superior for assistance.
Enemy small enough to attack
Hasty Attack
Context completed and plan empty
Enemy too large and out of sight
Receive message of enemy within threat distance or fired upon
Wait
(default context)

Figure 4: Context map for mission context *Seize*.

The major contexts represent tactics; that is, how to organize the subordinate agents, which tasks to give them and how they should be synchronized. The behaviors modeled inside each of the contexts were relatively simple with few action rules. For example, when entering the context Regroup, the agent first ordered all subordinates to stop and wait. If the observed enemy is too large to attack, the agent calculates a route away from the enemy and orders all subordinates to move along that route. In addition the agent notifies its superior, which could then transit to the Regroup context, if this is indicated in the superiors context map. If the superior is strong enough relative to the enemy, it could then attack.

While developing the model, we strove to keep major contexts and basic plans for mission contexts equal for platoon- and company agents. SMEs confirmed that the behaviors and tactics are basically the same at these two levels. Keeping the contexts the same also made the decomposition of the ordered tasks through the agent hierarchy, from company task to low level CGF tasks relatively straight forward.

To test the capabilities of the behavioral model, size and behavior of the enemy were varied. The model should handle both strong and weak, expected and unexpected enemy forces and act accordingly. The enemy forces were created in the CGF system and their behavior were scripted. In the future a separate MAS may be used to simulate the enemy forces.

3.2 Battle Command Agent Modeling using BDI

TNO has been involved in modeling the behavior of the battle command agents since 2008 (Borgers et al. 2008). Since 2010 the JADEX framework (JADEX 2011) is used for this. JADEX is one of the available frameworks for BDI agents.

In order to start to implement BDI for Battle Command, a limited set of military tasks were chosen for implementation. In the initial work (de Reus et al. 2011) the agents merely served as an ordered task distributor, providing simplified tasks from the commander to the units in the CGF. The tasks described in that work were *move*, *attack*, *defend* and *seize*, and the transformation was from company level down to single units (tanks). Since there was a need for more elaborate, organized and cooperative task handling, this was modeled later (Luik 2012). In this later work the tasks were only transformed one level down, so from company to platoon where the platoon tasks were send directly to the CGF.

The modeling started with the selection of the agents. Four types of agents have been modeled, namely the Tank Battalion Agent, Tank Company Agent, Tank Platoon Agent and HLAmanager, visualized in Figure 5, where the HLAmanager agent handles the reception of C-BML orders via HLA. The agent system is initialized with the correct numbers of units using the order of battle available from an MSDL file.

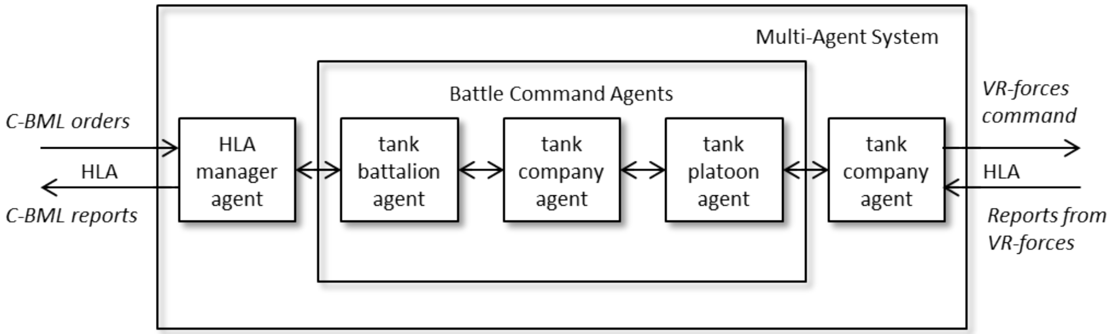

Figure 5: BDI agents and their connections.

So called Combat Instruction Sets (CISs) were used, which are low level tactical drills describing how to perform certain tasks. These real life drills, which were available at company level as well as platoon level, can also contain sub-CISs. They were translated in the past into detailed descriptions to be used in the Netherlands' tactical indoor simulator, and these descriptions proved useful for the agent modeling. In this section, as an example, one of the CISs namely *Assault on enemy position* is described and the implementation using BDI elaborated. The CIS can be described by the different phases: *Preparation*, *Fire&Movement*, *Attack Position* and *Consolidate&Reorganize*, as has been depicted in Figure 6.

The main part of the agent implementation is the plan base. The part of the plan base relevant for processing an Assault command looks as in Table 2. If the <Goal> or <Belief> field are kept empty, the plan always matches the goal or belief constraints. The ID in this table reflects priority: if two plans match, both beliefs and goals, the plan with the highest ID will be chosen.

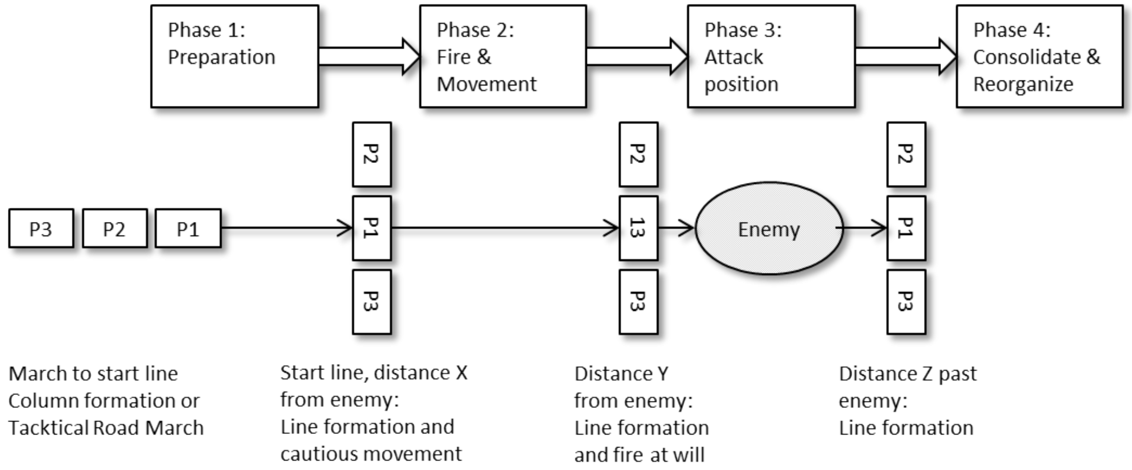

Figure 6: The different phases of the Assault on enemy position CIS.

Table 2: A part of the plan base relevant for handling the Assault on enemy position CIS.

ID	**<Goal>**	**<Events / Beliefs>**	**<Actions>**		
			Belief update	**Dispatch sub-goal**	**External action**
1		Event(ReceiveAssaultOrder)	Assaulting=true	ProcessAssault	
2		Event(StatusCompleteMsg) Assaulting=True		ProcessAssault	

3	Process Assault		AssaultPrep=true
			Marching=true
				March(X,Y)	getLocation(X,Y)
					getDistanceToLocation(Dist)

4	Process Assault	AssaultPrep=true	Marching=false	March(X,Y)	
5	Process Assault	AssaultMarch=true	AssaultMarch=false AssaultFireAndMovement=true		vrfCommand (AssaultFireAndMovement())
6	Process Assault	AssaultFireAndMovement=true	AssaultFireAndMovement=false AssaultFinal=true		vrfCommand (PlanAssaultFinalPhase)
7	Process Assault	AssaultFinal=true	AssaultFinal=false AssaultConsol=true		vrfCommand(Consolidate)
8	Process Assault	AssaultConsol=true	AssaultConsol=false Assaulting=false		
9	March (X,Y)				vrfCommand(March(X,Y))
10	March (X,Y)	Marching=False	AssaultPrep=false, AssaultMarch=true	MoveIntoFormation	
11	Move IntoFormation				vrfCommand (MoveIntoFormation)

In every time step, the agent loops the four steps from the reasoning cycle 1) Select the plans matching the Goals, 2) select the plans matching the beliefs, 3) select the plan with highest priority / highest utility which then becomes the intention, and 4) execute the action sequence in the plan until no more actions can be executed. This means that at first the agent does nothing, as none of the plans match both the goals and the beliefs of the agent. The chain is started by a ReceiveAssaultOrder-event. This event triggers the plan 1, which sets the belief 'Assaulting=true' and dispatches the goal 'ProcessAssault'. Now, immediately plan 3 is selected because of the goal-match. This means that the goal 'ProcessAssault' is removed from the goal base and the action sequence in the plan is executed. In the action sequence of plan 3 actions are executed, beliefs are set, and a new goal, March(X,Y), is dispatched. Then plan 9 is triggered, which results in a VR-Forces command. At that point, the goal base of the agent is empty and the belief base consists of (Assaulting=true, AssaultPrep=true), which means that no more plans can be executed, so the agent gets into waiting mode. A StatusCompleteMsg-event starts a new cycle, which triggers plan 2, 4, 10 and 11 subsequently until the agent gets into waiting mode again (note that if two plans match the goals and beliefs, the one with the highest ID is chosen). New StatusCompleteMsg-events lead the agent through all ProcessAssault phases, until the belief Assaulting is set to false in plan 8.

Currently the agents only respond to C-BML orders with tasks for one unit, the tasking of multiple (company) units with detailed timing is postponed for future work.

3.3 Comparing the two Approaches

In the previous sections, two approaches for modeling battle command and implement a multi-agent system used as an intermediary between the C2IS and the CGF system. In this section we will try to compare the two approaches.

First, the basis for the behavior modeling was different. TNO had access to formalized CISs, which described how to carry out various military task. These could relatively easily be translated to BDI-plans, and a selection of these CISs was implemented in the BDI framework. FFI on the other hand had to make these formalized models based on conversations with SMEs. A scenario based approach was chosen for capturing the expert knowledge.

In the CxBR approach one mission context was defined for each military task. The mission context includes a basic plan consisting of a sequence of major contexts, a context map including all possible major contexts and both planned transitions and transition rules for reactive behavior. The transition rules representing reactive behavior were the same in all mission contexts.

The BDI approach was based on CISs, which described the behavior for each military task in the order. Each CIS was translated to a set of BDI plans. Only behavior described in a CIS was implemented, meaning possible reactive behavior had to be part of a CIS. Note that general reactive behavior could have been implemented in the BDI approach as well by having high priority plans without goals, which match when certain events take place.

The main difference between the BDI and the CxBR approach is that in BDI all the behavior is stored in one plan base, whereas in CxBR the behavior is distributed between the different contexts. The BDI plans correspond to both the action knowledge and the transition knowledge within the CxBR major contexts. Although the CxBR paradigm does not set any restriction on how this action knowledge or transition knowledge are to be represented (rules, neural networks, etc.), the CxBR approach described here used rules, similar to the BDI plans. One could therefore say that in the CxBR approaches, the BDI plan base was divided into different contexts.

The challenge in CxBR is how to divide the behavior into appropriate contexts. One can for example put all BDI plans which corresponds to the same goal or a sub-goal of that goal in one context. BDI plans that leads to new goals will then become transition rules, whereas plans resulting in belief updates and/or external actions would be considered action knowledge. In the description of the CxBR approach an even higher grade of division was suggested. For example, all BDI-plans matching the goal "find enemy" might be stored in one context. However, there might be different strategies for finding the enemy, and these could be put in different contexts, even though they all try to achieve the same goal. Which one to choose would then depend on both the goal and the beliefs of the agent.

A major difference between the two approaches for modeling battle command was that in the CxBR approach the company and platoon agents were kept alike, with the same mission and major contexts. In the BDI approach, on the other hand, the behavior of the company and platoon agents were based on different CISs. This decision had nothing to do with the two modeling paradigms, and it is an open question what will work best as the complexity of the simulation system grows.

4 EXPERIENCES

Both the BDI approach and the CxBR approach seems suitable for modeling battle command agents. Since the two implementations did not include equivalent behavior and were not tested in the same way, it is hard to make any conclusions about pros and cons for each relative to the other. Instead we will discuss our experiences with the two approaches, starting with how to capture the knowledge the be modeled.

4.1 Capturing Expert Knowledge

In order to model the execution of military tasks and doctrinal behavior, access to expert knowledge is crucial. The research group from FFI had to gather this knowledge themselves through conversations with SMEs. This resulted in the developers gaining valuable insight into the military decision making process. However, the approach was very time consuming, and foremost applicable for prototyping.

TNO had access to formalized descriptions of military behavior in the form of CISs, which meant someone had done the expert knowledge capturing for them. The availability of CISs was very helpful. However, this led to the straightforward use of the quite detailed CIS descriptions without the explanations of why the task were to be carried out that way, possible variations, and the connections between the behavior at different levels in the unit hierarchy.

Potts et al. present an alternative solution, where system developers only make the framework and have subject matter experts add the behavior through a user interface (Potts et al. 2010). This might be a more scalable approach.

4.2 Agent Modeling Paradigm

The experience with using CxBR for battle command agents was positive. First, applying CxBR made it easy to explain the behavioral models to non-experts, and it is expected that organizing the behavior in this way will make it easy to implement a user interface which illustrates and explains the decisions the agents make along the way.

The main challenge with applying CxBR was that it is a relatively new idea, which means contrary to BDI there is not much previous experience to rely on or frameworks to give you a kick start. There are not many guidelines for what is a good number of contexts, how to do the planning, how to use CxBR in a multi-agent setting with communication and coordination among agents, etc. Still, the paradigm is very easy to understand and the modularity facilitates a gradual increase of behavior complexity.

With BDI it is relatively straightforward to figure out for a specific problem what should be the Beliefs, Desires and Intentions, and having examples and frameworks to build on facilitates the development of military applications with BDI agents. However, what seemed to be lacking was a firm software development process facilitating a seamless development from problem statement. So, basically the same question as was posed for CxBR applies here, there are not many guidelines for the (number and type of) beliefs, desires and intentions making the implementation of our BDI agents ad-hoc and an art rather than a science.

A disadvantage with BDI that was encountered was the fact that although the actions can be mapped fairly well from the expert (in the BDI case the CIS) to programming code, the understandability for the expert of the whole BDI agent with all its states and goals is not straightforward. One reason for this is the fact that no good visualization of BDI agents seems to exist.

5 CONCLUSION

In order to enable the simulation of higher (battalion or company) level C-BML orders for training and decision support applications, the use of a COTS CGF in combination with a Multi Agent System (MAS) has been studied. This paper presented the effort and approach taken in a comparative way and described how a part of the national doctrines has been implemented.

Two agent modeling paradigms were studied, BDI and CxBR. Both seems suitable for developing battle command agents. BDI is a well know paradigm for agent modeling, with several available frameworks, which makes it easy to start with. CxBR on the other is more modular, which might make it easier to understand the whole behavioral model and avoid inconsistencies as the behavior model grows larger.

REFERENCES

Barrett, G. 2007. “Collaborative Context-based Reasoning.” Ph.D. thesis, University of Central Florida.

Barrett, G., and A. J. Gonzales 2011. “Effective Agent Collaboration through Improved Communication by Means of Contextual Reasoning.” *International Journal of Intelligent Systems* 26 (2): 129-157.

Borgers, E., M. R. Hieb, R. Bonse, M. Spaans, J. M. Voogd. 2008. “Using a Command and Control Language to simulate Operations in a Multi-Agent Environment” In Proceedings of the 13th ICCRTS on : "C2 for Complex Endeavors"

Cares, J. R. 2002. “Agent Modelling: the Use of Agent-Based Models in Military Concept Development.” In *Proceedings of the 2002 Winter Simulation Conference,* edited by J. L. Snowdon and J. M. Charnes, 935-939. San Diego, California, USA: Association for Computing Machinery.

Carey, S., M. S. Kleiner, M. R. Hieb, and R. Brown. 2001. “Standardizing Battle Management Language - A Vital Move Towards the Army Transformation.” In *Proceedings of Fall Simulation Interoperability Workshop*.

Carroll, J. M. 1999. “Five Reasons for Scenrio-Based Design.” In *Proceedings of the 32nd Hawaii International Conferece on System Sciences.*

Cayirci, E. 2009. “Multi-Resolution Federations in Support of Operational and Higher Level Combinde/Joint Computer Assisted Exercises.” In *Proceedings of the 2009 Winter Simulations Conference,* edited by A. Dunkin, R. G. Ingalls, E. Yücesan, M. D. Rossetti, R. Hill, and B. Johansson, 1787-1797. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

de Reus, N. M., H. C. Henderson, R. Bronkers, O. M. Mevassvik, A. Alstad, and G. Skogsrud. 2011. “Battle Management Language capable Computer Generated Forces”. In *Proceedings of European Simulation Interoperability Workshop*.

Gallagher, A., Gonzalez, A. J., and DeMara, R. 2000 “Modeling Platform Behaviors Under Degraded States Using Context-Based Reasoning”. In *Proceedings of the 2000 Interservice/Industry Training, Simulation and Education Conference (I/ITSEC).*

Go, K., and Carroll, J.M. 2004. “The Blind Men and the Elephant: Views of Scenario-Based System Design.” *Interactions* 11(6):44-53.

Gonzales, A. J., and Ahlers, R. 1998 “Context-based representation of intelligent behaviour in training simulations.” *Trans Soc Comp Simul* 15(4): 153-166.

Gonzales, A. J., B. S. Stensrud, and G. Barrett. 2008. “Formalizing Context-Based Reasoning - A Modeling Paradigm for Representing Tactical Human Behavior.” *International Journal of Intelligent Systems* 23 (7): 822-847.

JADEX. 2011. Distributed Systems and Information Systems Group at the University of Hamburg.

Luik, T. 2012. “Using Agent-based Simulation in a Decision Support System for Military Command & Control.” Master's Thesis, Vrije Universiteit Amsterdam.

Mcilroy, S. D., D. Mcilroy, and C. Heinze. 1996. “Air combat tactics implementation in the smart whole air mission model.” In *Proceedings of the First International SimTecT Conference.*

Norlander, L. 1999. “A Framework for Efficient Implementation of Context-Based Reasoning in Intelligent Simulation.” Master's Thesis, University of Central Florida.

Potts, J. R., T. Griffith, J. J. Sharp, and D. Allison. 2010. “Subject Matter Expert-Driven Behavior

Modeling Within Simulation.” In *Proceedings of the 2010 Interservice/Industry Training, Simulation and Education Conference (I/ITSEC).*

Rao, A. S., A. Lucas, D. Morley, M. Selvestrel, and G. Murray. 1992. “Agent-oriented architecture for air combat simulation.” In *Proceedings of Future Directions in Simulation Systems Workshop.*

Rao, A. S., and M. P. Georgeff. 1995. “BDI agents: From theory to practice.” In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95).*

Tidhar, G., C. Heinze, S. Gross, G. Murray, D. Appla, and I. Lloyd. 1999. “Using intelligent agents in military simulation or using agents intelligently.” In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference, AAAI'99/IAAI'99.*

van Doesburg, W. A., A. Heuvelink, and E. L. van den Broek. 2005. “Tacop: a cognitive agent for a naval training simulation environment.” In *Proceedings of the fourth international joint conference on autonomous agents and multiagent systems (AAMAS'05).*

Wooldridge, M. 2004. “MultiAgent Systems.” John Wiley & Sons.

AUTHOR BIOGRAPHIES

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. Her email address is rikke-amilde.lovlid@ffi.no.

ANDERS ALSTAD is a Senior Scientist at FFI. His work is within the field of modelling and simulation. His interests are in design and development of distributed information systems. His email address is anders.alstad@ffi.no.

OLE MARTIN MEVASSVIK is a Principal Scientist at the FFI. His research interest is within the area of modelling and simulation, with emphasis on simulation systems interoperability. His email address is ole-martin.mevassvik@ffi.no.

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. His email address is nico.dereus@tno.nl.

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. His email address is henk.henderson@tno.nl.

BOB VAN DER VECHT is a member of the scientific staff in the technology area Networked Organisations, working in the Defence, Safety and Security theme in the Netherlands’ TNO. His work focuses on the Artificial Intelligence. His email address is bob.vandervecht@tno.nl.

TOREC LUIK was a master student at the time of the study specializing in Artificial Intelligence at Defence, Safety and Security theme in the Netherlands’ TNO. His email address is t_luik@hotmail.com.