Intelligent Agent Supported Training in Virtual Simulations

Dr. Annerieke Heuvelink, Dr. Karel van den Bosch,

Willem A. van Doesburg, M.Sc, Maaike Harbers, M.Sc.

TNO Defense, Security, and Safety

P.O. Box 23, 3769 ZG Soesterberg, the Netherlands.

Phone: +31 346 356 430

Fax: +31 346 353 977

{annerieke.heuvelink, karel.vandenbosch, willem.vandoesburg, maaike.harbers}@tno.nl

Abstract

Simulation-based training in military decision making often requires ample personnel for playing various roles (e.g. team mates, adversaries). Usually humans are used to play these roles to ensure varied behavior required for the training of such tasks. However, there is growing conviction and evidence that intelligent agents can also produce human-like, variable behavior. At the same time, it is known that goal-directed, systematic training is more effective than learning-by-doing only. To achieve goal-directed, effective training in (embedded) virtual simulations, events in the simulated environment as well as the behavior of these intelligent agents must be carefully controlled. We propose to do that by using a director agent (DA). A DA can be seen as a supervisor, capable of diagnosing task performance, instructing intelligent agents and steering the simulation. These capacities enable a DA to control a training scenario not only on the basis of an off-line scenario model, but also on its on-line assessment of the trainee's task performance. A DA can thus bring about a simulation-based training tailored to the needs of the trainee, enhancing his or her learning experience. In this paper, we explain and illustrate the concept of a DA in the context of simulation-based training in on-board fire fighting.

Intelligent Agent Supported Autonomous Training in Virtual Simulations

Military organizations tend to operate in highly uncertain and dynamic environments, and therefore require competent staff that acts adequately in any emerging situation. From the literature we know that acquiring expertise in complex tasks as faced during military missions is a matter of intensive, deliberate and reflective practice over time (Ericsson, Krample, & Tesch-Römer, 1993). Unfortunately, the very nature of military missions makes it hard to set up real-world training. In addition to practical issues such as the high level of danger, logistical issues play a role: mimicking a military mission in the real world requires many people and resources.

Scenario-based simulator training is considered appropriate for learning decision making in complex environments (Oser, 1999). An (embedded) virtual simulation enables trainees to experience the causal relations between actions, events and outcomes in the simulated environment. It thus gives access to experiential learning, e.g. by free-play practice. However, goal-directed, systematic training is more effective than learning-by-doing only (Blackmon & Polson, 2002). In order to make learning purposive and goal-directed, events in the simulation as well as the behavior of key players need to be carefully managed (Cannon-Bowers, Burns, Salas, & Pruitt, 1998; Fowlkes, Dwyer, Oser, & Salas, 1998). Players in the scenario should respond realistically to any situation emerging from the trainee's actions, and the responses should keep the scenario on track of the learning goals.

Common practice to realize this in simulation-training is to use Subject Matter Experts (SMEs) (usually staff members) to play the role of key players (van den Bosch & Riemersma, 2004). SMEs have the expertise to take the context into account when evaluating (on-line) the appropriateness of trainee behavior. They can also assess whether the scenario develops in the intended direction, and make adjustments, if necessary. Thus, SMEs make it possible to deliver training that represents reality in terms of dynamics and complexity, whilst tailoring the training to the performance of the trainee. However, the need for SMEs elevates costs of training, and staff is generally scarcely available. As a result, there are often (too) few opportunities to receive this type of training. The military acknowledges that developing expertise demands frequent, goal-directed, and intensive training. They are therefore looking for more flexible forms of simulation-training that require fewer organizational and logistic efforts.

A solution is to use virtual intelligent agents to play the required roles autonomously. If we can develop agents that in training scenarios produce intelligent and realistic behavior of the individual or entity that they represent, we would be able to make training more cost-efficient. However, in order to make such agent-based training also goal- and trainee-directed, we need an extra function. Like SMEs do, consideration should be given to which response will produce the best learning situation for the trainee. The agent should then act accordingly. For instance, an agent may deliberately act inaccurately because this enables the trainee to achieve the learning goal "detecting and correcting errors made by team mates". What we therefore need is management of agent behavior to ensure that the scenario develops in service of the learning goals.

One possibility to do this is to equip the virtual agents with didactical knowledge, thus enabling agents to take didactical considerations into account when deciding on how to act (van Doesburg & van den Bosch, 2005). However, we consider it important for agent development to separate domain-related knowledge required to generate task behavior from didactical knowledge required to exert control over the scenario. Therefore, we propose the concept of a "director agent" (DA). A DA can be seen as a supervisor, capable of diagnosing the trainee's task performance, instructing agents to perform certain behavior (thereby overruling what agents would otherwise do) and capable of steering the simulation (thereby overruling the specified chain of events). A DA can ensure that a training is and stays tailored to the needs of the trainee, thus creating an optimal learning experience in an (embedded) virtual simulation.

In this paper we report on the development of such a DA. First we will introduce the task domain for which we develop a virtual training environment. Next, we elaborate on the design of the DA and on its capacities to manage the scenario and to diagnose the trainee's task performance. Then we discuss how the DA can steer the simulation and agents to increase the trainee's learning experience. Last, we discuss the chosen approach and draw preliminary conclusions.

Task Domain

In this paper we describe the development of a desktop-simulation training that is equipped with virtual players that can act independently and intelligently, but whose responses can also be adjusted to create or utilize emerging learning opportunities. The domain is on-board fire fighting, and the task to be trained is that of the commanding officer, the Officer of the Watch (OW). The Royal Netherlands Navy (RNLN) currently provides training in on-board fire fighting using a high-fidelity simulation. Due to the rare availability of other trainees to play the role of team members, courses are organized infrequently and they contain few simulator sessions. On request of the RNLN we are developing an agent-based simulator that is more flexible and requires fewer personnel. Figure 1 shows an impression of the trainer. Within this trainer the trainee controls the avatar of the OW; we developed agents that play the team roles in an intelligent and autonomous fashion.

Figure 1. Impression of the agent-based virtual simulation environment for on-board fire-fighting training.[1]

[1] Courtesy of VSTEP (www.vstep.nl), the company that developed the virtual simulation.

The general course of events in naval on-board fire fighting is as follows. If aboard a navy frigate a fire breaks out, the Officer of the Watch (OW) is in charge of handling the incident. When the alarm sounds, the OW hastens himself to the Machinery Control Room (MCR) of the ship. From there, he contacts his team, develops a plan to contend the incident, gives orders, monitors the events, and adjusts plans if necessary. The Officer of the Watch communicates with four other officers: Chief of the Watch (CW), Machinery Control Room Operator (MCRO), Confinement Team Leader (CTL), and the Attack Team Leader (ATL). The first two are also situated in the MCR, the last two are at or near the location of the incident.

Several phases can typically be distinguished when contending an incident. Upon the alarm signal, the OW immediately orders initiating actions (e.g. stopping ventilation, checking water pressure, checking for wounded or missing persons) and broadcasts the incident across the ship. He then develops a confinement plan (e.g., cooling compartments adjacent to the fire; switching off power in areas at risk) and an attack plan (attack route; passage bans; escape route). Plans are then issued as orders. When the fire is extinguished, a plan for safe removal of smoke and gasses is executed. Finally, restoring and cleaning activities are initiated.

The task of the OW is a typical example of decision making in a complex environment. There are, of course, procedures for handling a fire accident. However, the OW also has to anticipate on possible complications, needs to respond to unforeseen actions, has to adjust plans when events require him to do so, and so on.

Agent-Based Training Simulation

The Simulation

The system under development is a stand-alone low-cost desktop simulation trainer (see Figure 1), to be used by a single trainee who is playing the role of OW. All four other players involved are played by intelligent agents. The avatar of the trainee is situated in the MCR of the ship throughout the training (as the OW is in reality). All equipment that is normally used is simulated and available to the trainee (damage control board, information panels, communication equipment, etc). In reality, team members communicate by speech. Our simulation has no speech recognition facilities, however. If an agent is the sender, it uses pre-recorded speech expressions. The trainee uses context-sensitive menus to send communication to the agents (see Figure 1).

The Training

In a broad sense, the goal of training is to learn and practice the assessments, procedures and decisions fundamental to fire-command. Instructors from the Navy school translated the abstract training goals into *learning objectives*, defined in terms of observable behavior. For instance: "trainee selects alternative attack route if circumstances require him to do so (e.g. due to blocked passage)". Instructors then formulated scenarios. Scenarios contain fixed elements, representing phases in the attack of an on-board fire (see the previous section). For each phase it is formulated which behavior of the OW would be correct. In addition, one or more states are formulated within these phases that will enable trainees to achieve a learning goal (e.g. a blocked passage on the logical attack route). What events may bring about those states is described, e.g. a particular event or situation (aisles are filled with laundry bags) or specific agent behavior (an agent "forgets" to close a door through which smoke enters the attack route).

Of course, we deal here not with independent, but with interactive elements of training. For example, certain events can only happen if the trainee has not taken precautionary measures earlier. Moreover, the agents and trainee are constrained by a set of actions possible in the simulated environment. For example, in our system one can contact other persons and make compartments voltage free, but one cannot navigate the ship. The simulation environment is developed in such a way that it allows for those actions that makes the trainee experience autonomy and control with respect to the task being trained.

As we can not know in advance what the trainee will do or not, we need a form of control to select what events must be released or prevented to bring about the desired states for learning. In the next sections we explain how we handle this problem.

The Agents

The intelligent agents in our virtual training simulation are modeled as experts, implying that they are able to autonomously perform expert behavior in all possible situations. Note that, as in real life, this involves more than blindly following the trainee's commands. For instance, an agent could be of the opinion that the trainee's plan involves unacceptable risks, and could thus propose an alternative plan.

We use the Belief Desire Intention (BDI) framework (Bratman, 1987; Rao & Georgeff, 1991) to develop the team agents. The BDI paradigm stems from folk psychology, i.e. the way people *think* that they reason (Norling, 2004). Humans usually describe their reasoning and explain their actions in terms of *beliefs*, *desires* and *intentions*. The BDI paradigm is based on these three mental concepts. As a rule, a BDI agent has beliefs, goals (desires), and intentions (goals to which it commits itself). Usually, BDI agents also have a plan library containing a set of *plans*. A plan is a recipe for achieving a goal, given particular preconditions. The plan library may contain multiple plans for the achievement of one goal. An intention is the commitment of the agent to execute the sequence of steps making up the plan. A step can be an executable action, or a sub-goal for which a new plan should be selected from the plan library.

It has been demonstrated that BDI agents can provide virtual players with believable behavior in computer games (Norling, 2003), and in virtual training (van den Bosch & van Doesburg, 2005). To generate such behavior, the agents require domain knowledge, which can be acquired from experts. Because experts tend to explain their actions in terms of beliefs, goals and intentions, expert knowledge can be easily translated to a BDI model (Norling, 2004). Furthermore, decision making in fire fighting is often procedural in nature: plans for achieving goals under given conditions are thus available. Some goals may be achieved in more than one way, which can be incorporated in the BDI model by defining multiple plans for one goal. As an example, see Figure 2 for an overview of the goals, plans and sub goals required for the most simple team agent, the Machinery Control Room Operator, to perform its task during fire fighting. The models are implemented in Jadex, an agent architecture based on the BDI framework, which allows for programming intelligent software agents in XML and Java.

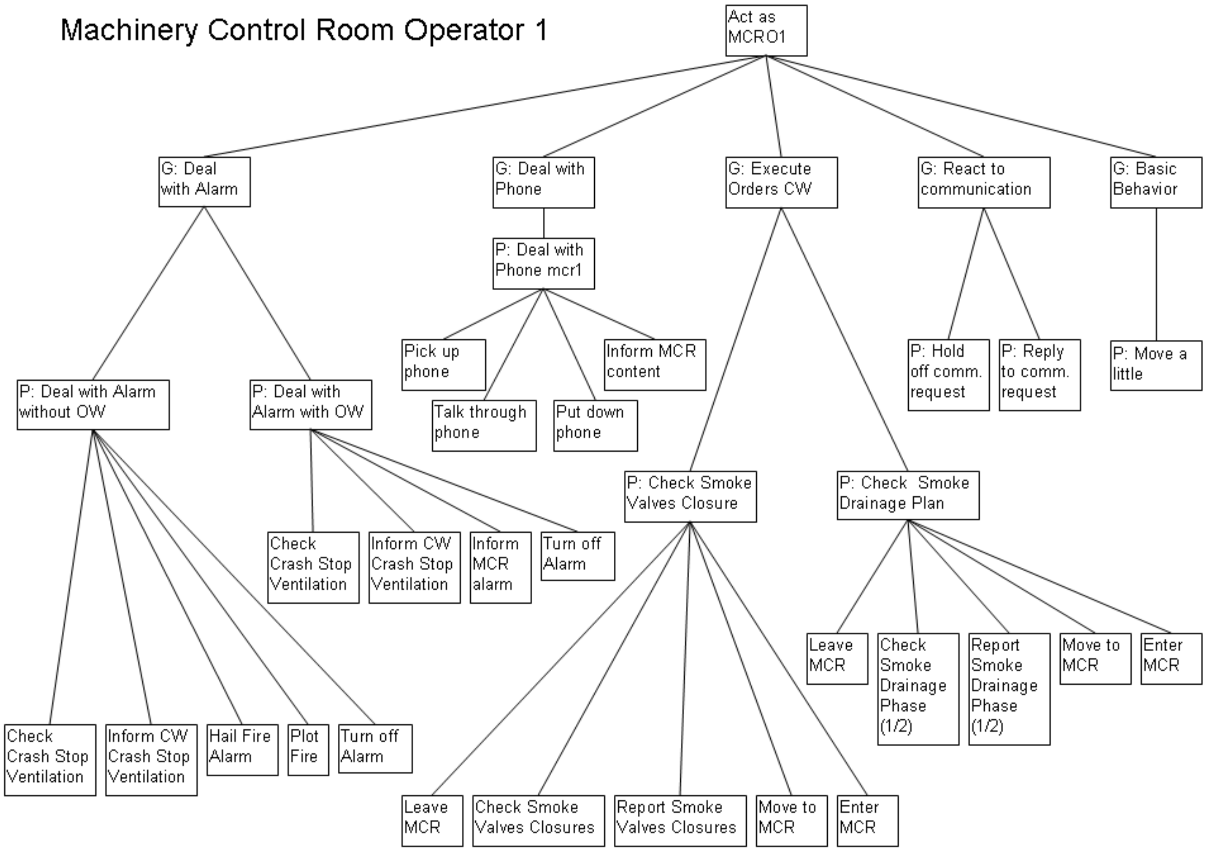


Figure 2. Overview of the goals, tasks and sub goals of the MCRO for on-board fire-fighting.

Director Agent

Agent-based simulation training generally contains the elements introduced in the previous section: a trainee (here: the OW); autonomous agents (here: team members); and the simulation environment. In addition, frequently a human instructor is involved that selects the scenario to train with (in our case: a certain type of fire in a specific compartment of the ship), and specifies - before the training starts - specific events or states (e.g. the presence of an injured person, or that the ventilation is not crash stopped) that will bring about a situation that enables trainees to achieve a learning goal(s) (e.g., keeping track of ship crew, or checking the automatic ship systems).

Once the training has started, the behavior of the trainee in interaction with the agents and the simulation environment determines how the scenario develops. The interaction between the various autonomous elements makes it difficult to predict the course and outcome of a scenario. Of course, the human instructor tries to bring about certain learning situations by specifying specific events and states. But whether or not the aspired situation will in fact occur is not sure because during the session, the instructor is unable to exert influence. Therefore, in addition to the scenario and states chosen by the instructor off-line, we need a manner to control the scenario on-line.

We advocate the use of a *director agent* (DA) to control the course of the scenario. A DA can be considered as an agent 'behind the scene'. The concept originates from studies into interactive narratives where story directors or drama managers are used (Riedl & Stern, 2006). In contrast to an intelligent tutor, a DA does not explicitly provide feedback or intervene an exercise (Riedl, Lane, Hill, & Swartout, 2005). See Figure 3 for the design of the agent-based virtual training simulation.

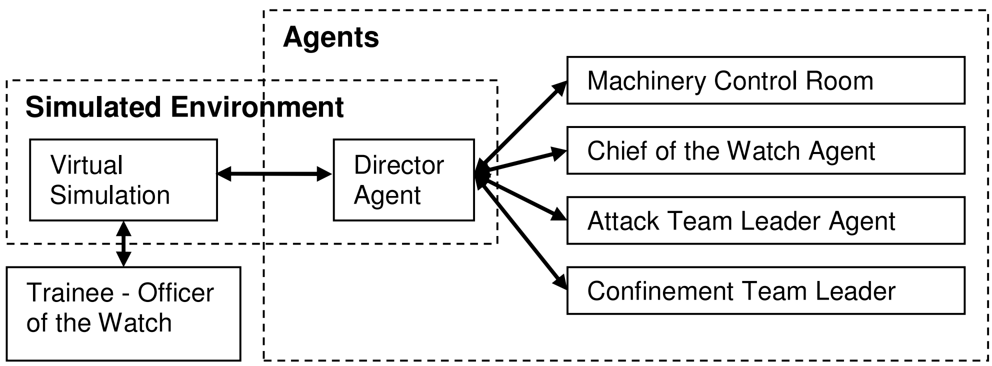


Figure 3. Design of the agent-based virtual training simulation.

As can be seen in Figure 3, the simulated environment is shared between the virtual simulation and the DA. This is due to the fact that the virtual simulation only simulates that part of the world visible to the trainee: the MCR and everything that happens there, see Figure 1. The DA simulates those parts of the world that are not handled by the virtual simulation, for example the events at the location of the fire. This entails that the trainee can only observe those events that take place in the virtual simulation (the visualized MCR), and has no direct access to the non-visualized events outside it, which are handled by the DA. From the perspective of the agents however, there is no difference between interaction with the visualized and non-visualized environment.

Scenario Management

The first goal of the DA is to ensure that the training scenario can develop as intended by the human instructor off-line. This role is called **Scenario Management** and contains three main functions.

To correctly execute its role, the DA has to have an overview of all the events that happen in the visualized and non-visualized environment. Therefore, all information from the virtual simulation is channeled to the DA and taken as input by the first function of the Scenario Management role: *the maintenance of the world state*. This function updates the world state of the DA based on all events that happened, either visualized in the virtual simulation or only simulated within the DA. To ensure that the agents only receive information from the virtual simulation that is consistent with the world view of the DA, the agents to not directly receive information from the virtual simulation. Instead, this function of the DA that maintains the world state also sends new information to the agents that should receive it.

Events can originate from one of three sources: the trainee may bring about an event, the team member agents may do so, and last the DA may initiate events in the simulated environment. Events stemming from the DA come from the second function of the Scenario Management role: *the control of events in the simulated environment*. This function helps the scenario to unfold according to the intentions set out by the instructor at the beginning of the training. For example, the DA can start a specific type of fire at a specific location, and decides whether the ventilation is or is not crash stopping. In addition, this function makes inferences from the world state that may lead to new events, e.g., a smoke alarm due to the presence of a fire.

The third function, *the handling of agent plan executions*, simulates the plan execution of the agents that are not present in the MCR, and are therefore not visualized. This function ensures that those plans take a certain time, and have certain effects in the environment.

Tutoring

The second important goal of the DA is to on-line pursue that the scenario stays in service of the learning objectives. This role is called **Tutoring** and embeds three functions.

The first function is *the recording of the trainee behavior*. This function updates the DA's model of the behavior of the trainee, e.g. the communicative act that he or she selected, or the plan of attack that was drawn on the virtual damage control board.

The second function of the Tutoring role is *the evaluation of the student behavior*. Based on its knowledge about the behavior of the trainee, the DA determines the quality of the trainee's actions. It does so by using expert knowledge formulated in the form of constraints that determine whether or not an action is correct (e.g., the agent should hail the fire alarm within 30 seconds after it sounds).

The third function, *tailoring the scenario to the trainee*, uses the outcome of the trainee evaluation to determine whether and how the training scenario can be adjusted to ensure the trainee reaches it training objectives. From the literature it is known that a trainee stays focused and motivated when a training is not too easy, but neither too hard (Bransford, Brown, & Cocking, 2000). If the DA observes, for example, that the trainee frequently executes the required actions too late, he may decide to activate the "Guide" function of the intelligent agents. We will elaborate on this and other support functions in the next section. The DA may also notice that the trainee performs extremely well throughout the scenario. The DA may then decide to add additional challenges, e.g., an agent that forgets to close a smoke valve, resulting in an additional fire alarm. Various of these scenario adjustment opportunities are defined, and can be brought about in several ways, which will be discussed in the next section. To determine which intervention to perform, this function of the DA requires knowledge about the relation between learning objectives and scenario interventions. This is implemented as a set of rules relating learning objectives to possible adjustments of events in the simulation environment and behaviors of the agents.

Tailoring the Training to the Trainee

In the previous section we introduced the DA and its function to tailor the scenario according to the performance of the trainee. To create an optimal learning experience, the difficulty level of the training should suit the skill level of the trainee. Instructors mostly try to make the scenario a bit more difficult than the trainee's current level can handle. From this optimal situation two deviations exist: either the difficulty level is too high given the level of the trainee, or it is too low. Both these situations result in an non-effective training situation. In this section we introduce the interventions that the DA can do to adjust the difficulty of training, and what this entails for the design of the BDI-agents.

Supporting the Trainee

Remember that our BDI agents are designed as expert agents, entailing that they know how to handle a fire incident. In addition, we made them realistic team members of the OW and equipped them with the inclination to advise and possibly correct the trainee in its task execution. For a specific example see Figure 4. Figure 4 depicts a small part of the hierarchical task analysis of the CTL. In particular, it specifies the plans and sub goals the CTL forms in order to reach its goal to receive an initial Boundary Management Plan (BMP) of the OW ("Get First BMP").

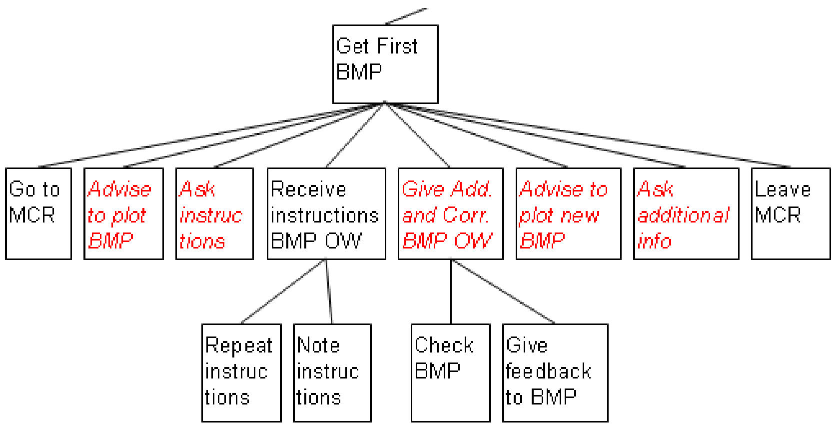


Figure 4. Overview of the tasks and sub goals of the CTL to reach its goal of receiving an initial BMP of the OW.

Some plans and sub goals of the “Get First BMP” are written in italics, denoting that they are not required for the main task execution, but concern support functions. Three types of support functions exist: explicitly advise, implicitly advise, and correct functions.

Explicitly advise plans guide the trainee through the scenario by suggesting the correct action, e.g., “I suggest that you plot a boundary management plan”. To trigger these plans, two conditions should be met. First, a certain time has to be elapsed from the moment that the action became appropriate. Second, the agent should be in the **Guide** mode.

Implicitly advise plans help the trainee by asking a question that may spur him or her to do a correct action. For example, the CTL’s question “Do you have any additional information for me?” after it receives the BMP, is intended to trigger the OW to ask the CW whether compartments adjacent to the fire are voltage free. To trigger these plans the agent should be in the **Suggest** mode, and again a certain time has to be gone by from the moment the correct action started to become appropriate.

Whether or not the *correct* plans are activated has the largest influence on the course of the scenario. If the agent is in **Correction** mode it will correct mistakes the trainee makes, thereby keeping him or her on track of the scenario. If the trainee does not correct certain mistakes, e.g., a wrong BMP, the trainee will be faced with the consequences, in this case a spreading fire.

Which support functions should be selected is, among others, a matter of the phase of training. For a novice, it might be good to put the agents in the Guide mode so they can inform the novice about the required steps in fire fighting by advising the trainee to do them. In addition, the correction of mistakes in Correction mode will give the novice insight in the correct actions. For advanced trainees the Guide mode could be turned of, while the Suggest mode might still help them to think about the required steps. Switching support modes is thus an easy generic way to change the difficulty level of training.

Interfering with the Development of the Scenario

In the previous section we explained that for each scenario several scenario adjustment opportunities are specified that can help achieving a specific learning objective. Some of these adjustments can be brought about by an event in the simulated environment that is triggered or inhibited by the DA, e.g., the starting of a second fire. Other adjustments are caused by agent behavior that deviates from the expert norm. To have the agents display non expert behavior, three interventions are possible that vary in their level of intrusiveness. Which intervention is required depends on the level of the agent behavior the DA wants to change (e.g., forgetting of a step in a plan vs. giving a wrong task priority).

First, the DA may wrongly execute an action that an agent wants to execute. All the actions an agent wants to perform are first sent to the DA that may then either forward them to the virtual simulation (for actions visualized in the virtual training simulation) or simulate them itself (when they take place in the non-visualized environment). The actions of the agents visualized in the MCR generally consist of low-level, atomic actions. An example of a mistake that a DA can simulated there is the pressing of the wrong button on a console. The non-visualized agent actions that are simulated by the DA are usually less atomic and might for example be “Check Smoke Valves Closures”. An error that the DA can let occur here is that an open valve is overlooked.

A second way for the DA to have an agent execute non expert behavior, is to send false world information to the agent, possibly resulting in behavior that is wrong given the actual circumstances. Important to realize for this type of intervention is that this false information may soon be overwritten by correct information received from the simulated environment. At this moment we do not consider the halting of this correct information upon sending false information, but this could be added.

Third, the DA may be able to intervene in the actual functioning of the agent by authorizing it to change the team agents’ plan and goal bases. An intervention could then entail sending a (false) plan or goal to the agent. When it is assumed that the plan or goal received from the DA receives priority over all the other goals and plans, the sending of a goal can simulate the error of giving priority to a less important goal. By giving an agent a false plan, the execution of a wrong procedure can be simulated. Unfortunately, due to difficulties with sending and adopting new goals or plans from one Jadex agent (the DA) to another (one of the team agents), we did not yet implement this option.

Discussion and Conclusion

Becoming an experienced and competent staff member deployable in military missions is a long-lasting undertaking. Good training requires frequent and deliberate practice in situation assessment and decision making. This is not only taxing for the trainee, but also for the organization responsible for delivering training. It requires ample staff to realize the environments that students need to acquire domain-specific knowledge and to practice assessment and decision making skills. Recent developments in simulator- and agent technology open opportunities to improve this situation. Modern computers are capable of generating highly realistic, dynamic, interactive (embedded) simulations. Advances in agent technology can be used to generate the behavior of human entities in such (embedded) virtual simulations.

In this paper we report current work on the design of such an advanced training system. We have argued that in order to make such training goal-directed and systematic, a DA can be used that exerts control over the simulation and over the intelligent team agents. The DA can do so by using a rule set defining the relations between learning goals, scenario states, and interventions (pertaining to both simulation and agents). In our concept, the DA imposes constraints upon the autonomy of simulation and agents to the benefit of maintaining control over the scenario, adjusting it to the level of the trainee.

The question is then: will the proposed DA indeed achieve the desired control? From earlier work we learned that the main difficulty is endowing an agent with the capabilities to detect that a scenario is going off-course, and to diagnose the nature of digression. We have been able to develop an agent that successfully diagnosed student errors, by combining both outcome- and process measures of student task performance (Heuvelink & Mioch, 2008). This diagnosis was subsequently used to present feedback to the trainee. Similarities with the present work are obvious. Rather than using a diagnosis for selecting feedback, our DA selects an intervention aimed to bring about the desired states for learning. We are therefore confident that the approach will work.

Another question is: will interventions yield the scenarios that we hope for? Most likely it will not succeed always. We see the same when team members are played by human instructors. Instructors often make ‘smart moves’ to create challenging learning situations, but they too do not always succeed. Likewise, our DA may fail some of the times. One possibility is that the DA’s rule set contains no intervention that may bring the actual scenario state into a desired scenario state. It is also possible that an applied intervention fails to produce the desired state (e.g. because it elicits the trainee to take an action that blocks the potential effect of the intervention). We want to emphasize here that our goal of using a DA is not to achieve full and total control over a scenario. This would very likely harm the trainee’s sense of autonomy and the experienced realism of the scenarios. The DA must thus be considered as a tool to advance from free-play to more deliberate, goal-directed form of training.

A third question is: is the concept of DA appropriate to achieve control? An alternative way of exercising control is to build didactical considerations into the playing agents. In this way, didactical considerations are decentralized. But this can harm control too. If each agent has its own set of behavioral instructions, then several agents at once may act trying to achieve a desired scenario state. This may lead the scenario further astray. Another disadvantage of decentralization is the issue of reusability. For a simulation training it is best if the models underlying the agents can be used for many scenarios. Didactic considerations, however, tend to be scenario and trainee specific. What is desirable for achieving a particular learning goal doesn’t necessarily need to be desirable for another. We consider it therefore important to separate domain-related knowledge required to generate task behavior from didactical knowledge required to exert control over the scenario.

Concluding, recent developments in embedded virtual simulations and intelligent agent technology promise better opportunities for autonomous training in decision making for military tasks. Our approach presented here should be able to add learning value to that promise.

Acknowledgments

This research has been supported by the Netherlands Department of Defense (032.13359) and by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

References

Blackmon, M.H., & Polson, P.G. (2002). Combining Two Technologies to Improve Aviation Training Design. In *Proceedings of Human Computer Interaction* (HCI) (pp. 24-29). AAAI Press.

Bosch, K. van den, & Doesburg, W.A. van. (2005). Training Tactical Decision Making Using Cognitive Models. In *Proceedings of the Seventh International NDM Conference*. Amsterdam, the Netherlands.

Bosch, K. van den, & Riemersma, J.B.J. (2004). Reflections on scenario-based training in tactical command. In S.G. Schiflett, L.R. Elliott, E. Salas, & M.D. Coovert (Eds.), *Scaled Worlds: Development, Validation and Applications* (pp. 1-21). Ashgate, Aldershot.

Bratman, M. (1987). *Intention, Plans and Practical Reason*. Cambridge, MA: Harvard University Press.

Bransford, J.D., Brown, A.L. & Cocking, R.R. (Eds.) (2000). How people learn: Brain, Mind, Experience, and School. Washington, D.C.: National Academic Press.

Cannon-Bowers, J.A., Burns, J.J., Salas, E., & Pruitt, J.S. (1998). Advanced technology in scenario-based training. In J.A. Cannon-Bowers, & E. Salas (Eds.), *Making decisions under stress: implications for individual and team training* (pp. 365-374). Washington, DC: APA.

Doesburg, W.A., & Bosch, K. van den (2005). Cognitive Model Supported Tactical Training Simulation. In *Proceedings of the Conference on Behavioral Representation in Modeling and Simulation* (BRIMS) (pp. 313-320). Universal city, CA.

Ericsson, K.A., Krampe, R.Th., & Tesch-Römer, C. (1993). The Role of Deliberate Practice in the Acquisition of Expert Performance. *Psychological Review*, *100(3)*, 363-406.

Fowlkes, J., Dwyer, D.J., Oser, R.L., & Salas, E. (1998). Event-based approach to training (EBAT). *The International Journal of Aviation Psychology*, *8(3)*, 209-222.

Heuvelink, A., & Mioch, T. (2008). FeGA: A cognitive Feedback Generating Agent. In *Proceedings of the 7th IEEE/WIC/ACM International Conference on Intelligent Agent Technology* (IAT 2008), (pp. 567-572). IEEE Computer Society Press.

Norling, E. (2003). Capturing the Quake Player: Using a BDI Agent to Model Human Behaviour. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems* (AAMAS), (pp. 1080-1081). Melbourne, Australia.

Norling, E. J. (2004). Folk psychology for human modelling: Extending the BDI paradigm. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems* (AAMAS 2004) (pp. 202-209). Washington, DC: IEEE Computer Society.

Oser, R.L. (1999). A structured approach for scenario-based training. In *Proceedings of the 43rd Annual meeting of the Human Factors and Ergonomics Society* (pp. 1138—1142). Houston, TX.

Rao, A., & Georgeff, M. (1991). Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, & E. Sandewall (Eds.), *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning* (pp. 473-484). San Mateo, CA: Morgan Kaufmann publishers Inc.

Riedl, M.O., Stern,O. (2006). Believable agents and intelligent scenario direction for social and cultural leadership training. In *Proceedings of the 15th Conference on Behavior Representation in Modeling and Simulation* (BRIMS). Darmstadt, Germany.

Riedl, M.O., Lane, H.C., Hill, R., & Swartout, W. (2005). Automated Story Direction and Intelligent Tutoring: Towards a Unifying Architecture. In *Papers of the Narrative Learning Environments Workshop at the 12th International Conference on Artificial Intelligence* (AIED05), Amsterdam, The Netherlands.

Figure Captions

Figure 2. Impression of the agent-based virtual simulation environment for on-board fire-fighting training.[2]

Figure 2. Overview of the goals, tasks and sub goals of the MCRO for on-board fire-fighting.

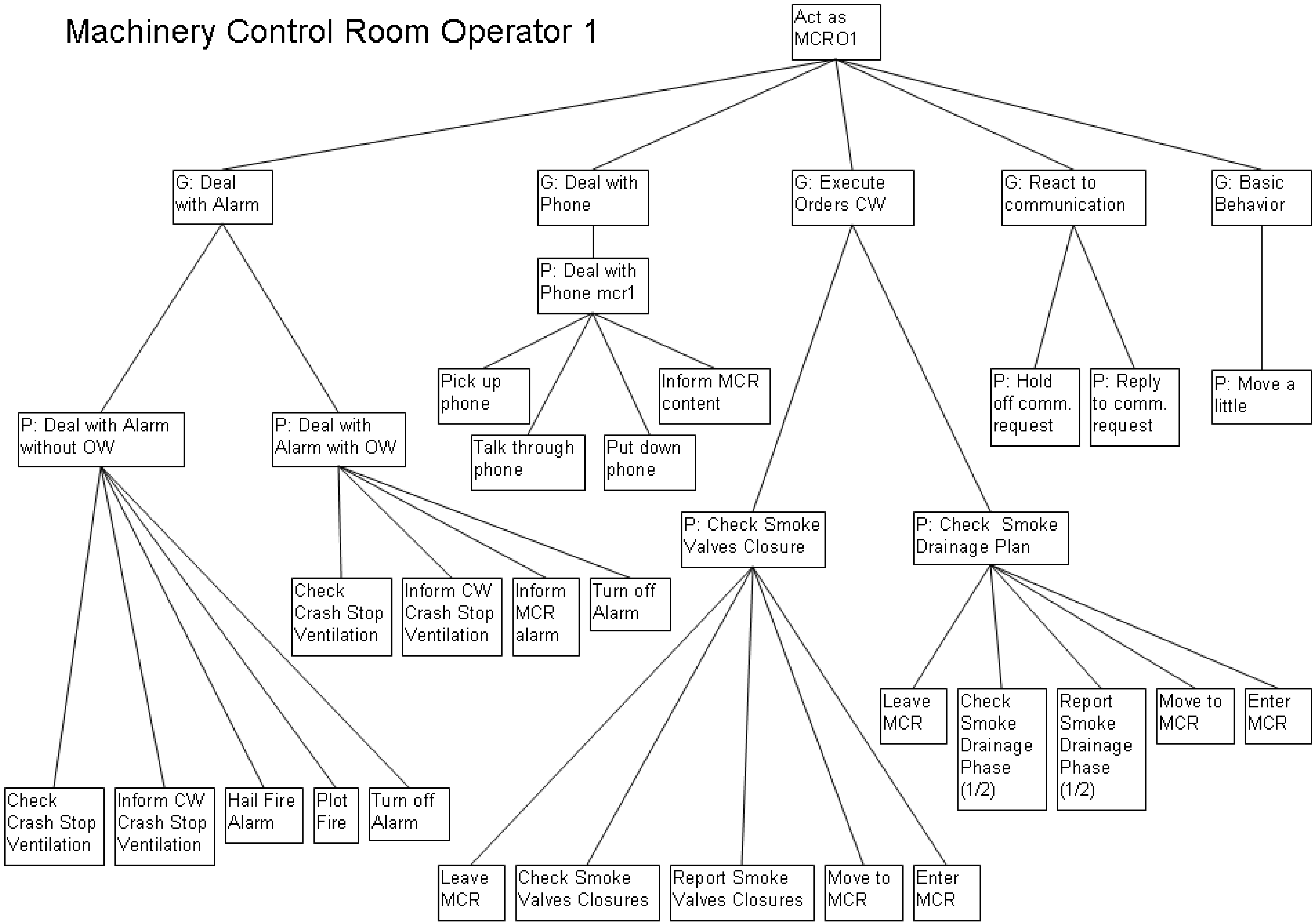
Figure 3. Setup of the agent-based virtual training simulation environment.

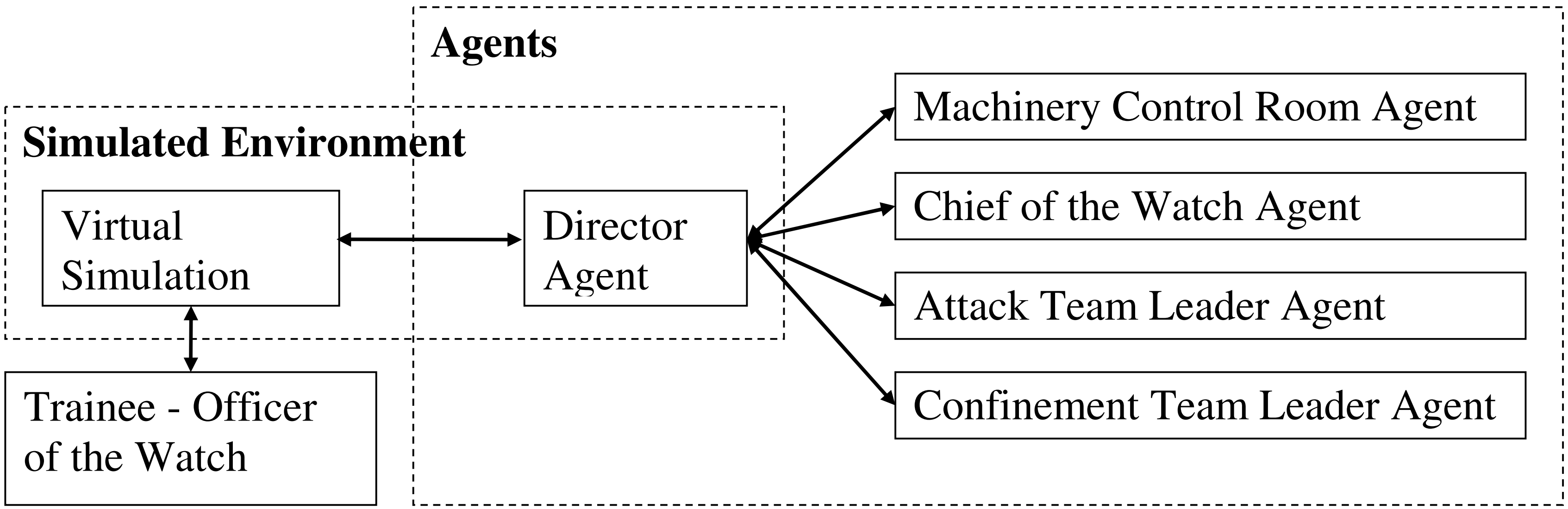
Figure 4. Overview of the tasks and sub goals of the CTL to reach its goal of receiving an initial BMP of the OW.

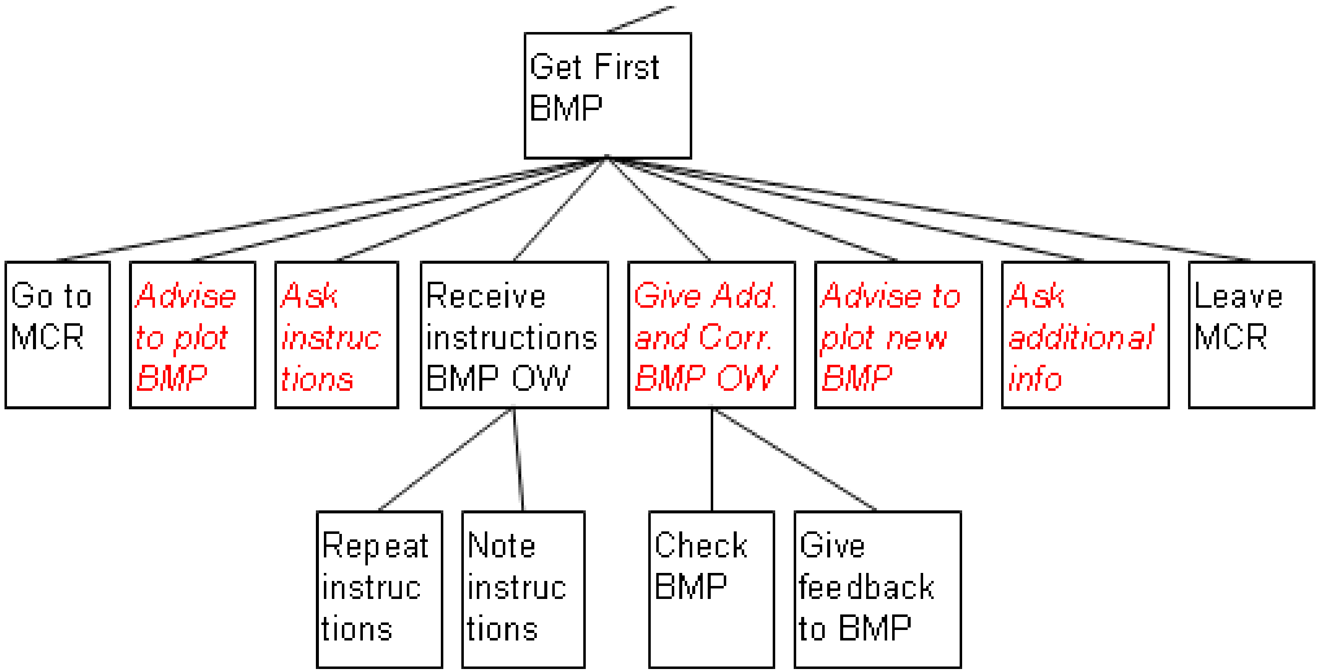
[2] Courtesy of VSTEP (www.vstep.nl), the company that developed the virtual simulation.

BARCO
RGD 651
Situation Report
Aanvalsploeg:
Gewonden: -
Vermisten: -
Chef van de Wacht
V Zijn de brandbluspompen bijgezet?
V Is de ventilatie gecrashstopt?

Machinery Control Room Operator 1




Agents
Simulated Environment
Virtual Simulation
Director Agent
Machinery Control Room Agent
Chief of the Watch Agent
Attack Team Leader Agent
Confinement Team Leader Agent
Trainee - Officer of the Watch


Get First BMP
Go to MCR
Advise to plot BMP
Ask instruc tions
Receive instructions BMP OW
Give Add. and Corr. BMP OW
Advise to plot new BMP
Ask additional info
Leave MCR
Repeat instruc tions
Note instruc tions
Check BMP
Give feedback to BMP