Intelligent Agents for Training On-board Fire Fighting

Karel van den Bosch, Maaike Harbers,
Annerieke Heuvelink, Willem van Doesburg

TNO Defence, Security, and Safety
PO Box 23, 3769 ZG Soesterberg, the Netherlands
{karel.vandenbosch, maaike.harbers, annerieke.heuvelink, willem.vandoesburg}@tno.nl

Abstract. Simulation-based training in complex decision making often requires ample personnel for playing various roles (e.g. team mates, adversaries). Using intelligent agents may diminish the need for staff. However, to achieve goal-directed training, events in the simulation as well as the behavior of key players 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 instructing agents and steering the simulation. We explain and illustrate the concept in the context of training in on-board fire fighting.

Keywords. Virtual Training, Intelligent Agents, Simulation, Director Agent, Scenario Based Training

1. Introduction

Modern society has ample systems where one decision maker controls the safety of many. For example, the safety of aircraft passengers depend on the decisions of the pilot; a military commander's decision may entail danger or protection for soldiers and civilians; the fate of fire-fighters, bystanders and victims largely depend on decisions of the fire officer. It is evident that for such safety critical systems we need competent and experienced decision makers. From the literature we know that acquiring expertise for complex tasks is a matter of intensive, deliberate and reflective practice over time [1].

Scenario-based simulator training is considered appropriate for learning decision making in complex environments [2]. A 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 [3]. 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 [4, 5]. 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 [6]. 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 maintaining a high level of control. 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. Organizations acknowledge 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 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-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.

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.

The trainee controls the avatar of the OW. We developed agents that play the team roles in an intelligent and autonomous fashion. As we argued, autonomous agents are not sufficient to achieve goal-directed training. A form of control is needed to maintain the scenario on track of the learning goals. One possibility is to expand cognitive models with didactical knowledge, thus enabling agents to take didactical considerations into account when deciding on action [7]. 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. We therefore propose a "director agent" (DA). The proposed DA can be seen as a supervisor, capable of instructing agents to perform certain behavior (thereby overruling what agents would otherwise do) and capable of steering the simulation (thereby overruling the chain of events specified in the simulation model).

Fig. 1. Impression of the agent-based simulator for on-board fire-fighting training.[1]

The DA needs to have a rule set that defines the relations between learning goals, scenario states, and interventions (pertaining to both simulation and agents). With these, it imposes some constraints upon the autonomy of simulation and agents to the benefit of maintaining control over training.

2 The virtual training system

In this section we will first briefly describe the organization of fire-fighting aboard a ship from the perspective of the officer in charge. Then we will point out the structure of our agent-based simulation training.

2.1 On-board fire fighting

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 Technical Centre (TC) 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, Technical Centre Operator, Leader Confinement Team, and the Leader Attack Team. The first two are also situated in the TC, the last two are at or near the incident scene.

Several phases can typically be distinguished when contending an incident. Upon the alarm signal, the OW immediately orders initiating actions (e.g. stopping

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

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, setting smoke borders; 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.

2.2 The Agent-based Simulation Training

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 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 are built up from scenes, each representing a phase in the attack of an on-board fire (see 2.1). Each scene contains one or more desired states: states that 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 precisely described in the scenario, e.g. a particular event (aisles are filled with laundry bags), agent behavior (an agent “forgets” to close a door through which smoke enters the attack route), and trainee behavior (waits too long to order fire attack, and fire has spread out to the route). Of course, we deal here not with independent, but with interactive elements of training. For example, the simulation can only cause an event to happen if the trainee has not taken precautionary measures earlier. 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 Simulation: The avatar of the trainee is situated in the TC 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 board, information panels, communication equipment, etc).

The Agents: We use the BDI-framework [8] to develop intelligent and autonomous team agents. 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).

3. Elements of Virtual Training

As mentioned in the introduction, one difficulty of simulation-based training is to balance the players' freedom of action (of both agents and trainee) on the one hand, and control on the scenario on the other. In this section we describe how we organize the elements of virtual training in order to achieve scenarios that are experienced as realistic by trainees, but are also sufficiently controlled to ensure proper learning.

Agent-based simulation training generally contains the following elements: a scenario writer; a trainee (here: the OW); autonomous agents (here: team members); and the simulation. All influence the course of the scenario. The *scenario writer* selects one of more learning goals and specifies -in advance- which main events will bring about a situation that enables trainees to achieve the learning goal(s) (e.g. there is a fire in compartment X of the ship). The *trainee* has to deal with the situation as he thinks is right (he may, for instance, issue a fire attack plan and give commands to his team agents). The *agents* respond autonomously to the trainee. 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 it proposes an alternative plan. Finally, the *simulation* processes events and actions realistically (e.g. it lets a fire extinguish if it is deprived from oxygen).

All elements comprising the virtual training system have a certain degree of freedom, but as they interact, they also influence each other. For instance, the scenario writer determines which events occur in the simulation. The agents and trainee can only execute those actions that are supported by the simulation environment. The interaction between the various autonomous elements makes it difficult to predict the course and outcome of a scenario. Of course, the scenario writer tries to bring certain learning situations about by specifying the appropriate events. But whether or not the aspired situation will in fact occur is not sure because during the session, the scenario writer is unable to exert influence. Therefore, in addition to the specification formulated by the scenario-writer, 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 [9]. In contrast to an intelligent tutor, a DA does not explicitly provide feedback or intervene an exercise [10]. Figure 2 shows our setup of the elements of virtual training. Black arrows represent active influence relations (e.g. the writer influences off-line the way the DA should act, while agents and trainee on-line influence the situation in the environment); dashed arrows represent passive influence relations (e.g. the trainee is influenced, but not controlled by the simulation). In the remainder of this section we discuss this model.

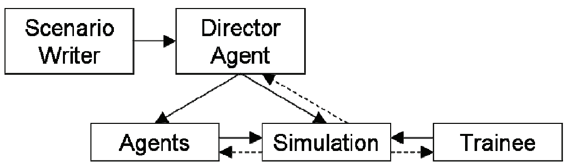

Fig. 2. The elements of virtual training and their relations.

The *scenario writer* defines a scenario using a specification language that is interpretable by the DA. The specification language should allow the representation of a sequence of events in the scenario, containing the time of occurrence, nature and location, and factors that complicate solving the problem (e.g. defect equipment). Moreover, it should provide the possibility to specify which learning goals are the focus in this (scene of) the scenario (e.g. communication or situation assessment). Indirectly, the writer is constrained in the kinds of scenarios it can write by the simulation and the agents. For instance, in our virtual training system only the TC can be visualized, and therefore the scenario should not involve situations in which Officer of the Watch has to walk out of the TC. After writing such a specification, the writer completely delegates his control over the scenario to the DA.

The *DA* directs the simulation and agents according to the scenario specification. The DA provides the simulation with events that must occur (e.g. a fire has to start). Most of the time, it does not pursue control over the agents; the agents autonomously generate behavior and execute actions in the environment. However, if the scenario requires specific behavior from an agent in order to bring about a desired state, the agent receives instructions from the DA to do so. We discuss the relation between the DA and team agents in more detail in the next section.

The *simulation* processes the events assigned by the DA. Events are specified at a high level of detail. The simulation processes these autonomously to lower-level consequences. For example, the simulation interprets a specification of the event “fire” at a particular location to effects like smoke, limited vision, etc. Under certain circumstances the DA may assign certain events to the simulation that were not specified by the scenario writer (e.g. to bring the scenario back on course after it has been led astray by erroneous or unexpected behavior of the trainee). We will not discuss this further here.

Finally, actions of the *trainee* and *agents* are processed by the simulation. For instance, if the trainee wants to open a door, it sends a message to the simulation and the door is opened. Communication between agents and trainee is also mediated by the simulation; agents and trainee send communication messages to the simulation, which passes it on to the indicated receiver. The agents and trainee are constrained by a set of actions possible in the simulation. For example, in our system one can contact other persons and make compartments voltage free, but one cannot navigate the ship. The simulation is developed in such a way that it allows for those actions that makes the trainee experience autonomy and control with respect to the training task.

4. Control of Agent Behavior

In this section, we discuss how the behavior of the agents representing the trainee’s team members should be controlled by the DA. We first discuss our approach of developing intelligent, autonomous agents. Subsequently, we explain how the DA can pursue control over these agents, without completely taking over their behavior.

4.1 Autonomous BDI-agents

The intelligent agents in our virtual training systems are modeled as experts, implying that they are able to autonomously perform expert behavior in all possible situations. They are developed according to the Belief Desire Intention (BDI) paradigm which stems from folk psychology, i.e. the way people *think* that they reason [10]. Usually, humans 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. A typical BDI execution cycle contains the following steps: i) observe the world and update the agent's internal beliefs and goals accordingly, ii) select applicable plans based on the current goals and beliefs, and add them to the intention stack, iii) select an intention and iv) perform the intention if it is an atomic action, or select a new plan if it is a sub-goal.

It has been demonstrated that BDI agents can provide virtual players with believable behavior in computer games [12], and in virtual training [13]. 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 [14]. Furthermore, decision making in safety critical situations is often highly 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.

4.2 Director Agent

In the previous section we shortly described our approach to developing autonomous agents. The resulting BDI agents are experts in their task domain, but know nothing about training. In other words, they know how to handle an incident even if others make errors (e.g. a trainee), but not how they can support a trainee in his learning process. The responsibility for this second aspect is completely delegated to the DA. To accomplish the desired support, the DA needs an expert model of the role that the trainee is playing, which in our case is the OW. Moreover, it requires didactical knowledge about the relation between learning goals and scenario interventions. The first can be implemented as an expert BDI-model, and the second as a set of rules relating learning goals to possible directions to simulation and agents.

The DA knows which learning goals are active, and assesses on-line the current situation emerging in the scenario. In the rule set it is specified which interventions create situations to train the specified learning goals. For example, a possible learning goal is: "check whether initiating measures are taken"; and an intervention is: "prevent other agents from taking initiating measures". Thus, if the scenario allows for it, the DA selects an intervention from the rule set to still bring about the desired learning situation. An intervention either releases or inhibits an event (e.g. an alarm) in the simulation, or instructs an agent to adopt or drop a goal specified in the rule set (e.g. checking whether it is safe to enter a room).

We adhere to the position that to train a specific skill, situations requiring that skill should be created. This means that the right complications need to be introduced. We distinguish two strategies to accomplish such situations. First, the DA can order the team agents not to correct or support the trainee when he or she is making an error. Second, the DA can order team agents to make a mistake on purpose.

In order to make team agents sensitive for instructions from the DA for the first strategy, we need to model the team agents in such a way that they have a notion of when the trainee performs suboptimal behavior. We will illustrate this with an example. The model of team agent A may contain a rule that the trainee should inform team agent B about a particular event. If agent A obtains the belief that the trainee failed to do this, it will automatically adopt the goal to bring that inform action about (e.g. by reminding the trainee to inform agent B, or by reminding agent B itself). However, the scenario writer may have specified that the learning goal is 'communicate situation update to team members'. The DA translates this requirement to a desired state in the scenario that makes the trainee experience the consequences of negligence. In our example, the DA could bring about the intention of the scenario writer by instructing team agent A to withhold his goal to bring the trainee's missed inform action about. Thus, the reasoning rule that makes a team agent correct an omission of the trainee is applied under the right conditions in the scenario, but *only* if the DA does not issue a withhold instruction.

The second strategy, ordering team agents to make a mistake, is also a powerful didactic instrument. In order to do this, we authorize the DA to change the team agents' goal-, belief-, and plan bases. The requested mistakes can generally be specified in advance by the scenario writer. If at a particular -prespecified- point in the scenario a certain mistake is necessary, the DA assigns sub-optimal or incorrect beliefs, goals or plans onto the team agent(s). Goals and plans that an agent receives from the DA always receive priority over all other plans and goals in the current intention stack or goal base of the agent, respectively. By giving an agent a false belief, the error of unjustly assuming that a condition is true can be simulated. By giving an agent a false goal, the error of giving priority to a less important goal can be simulated. And, finally, by giving an agent a false plan, the execution of a wrong procedure can be simulated.

5. Discussion

Becoming an experienced commander of a safety critical system 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 student, 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 and interactive simulations. Advances in agent technology can be used to generate the behavior of human entities in the simulation.

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 playing 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.

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 [15]. This diagnosis was subsequently used to present feedback to the student. 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 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 simulation, cognitive modeling, and agent technology promise better opportunities for autonomous training in decision making. 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 Defence (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

1. Ericsson, K.A., Krampe, R.Th., Tesch-Römer, C.: The Role of Deliberate Practice in the Acquisition of Expert Performance. Psychological Review, 100(3), 363--406 (1993)
2. Oser, R.L.: A structured approach for scenario-based training. In: 43rd Annual meeting of the Human Factors and Ergonomics society, 1138--1142. Houston, TX (1999)
3. Blackmon, M.H., Polson, P.G.: Combining Two Technologies to Improve Aviation Training Design. In: proc. of Human Computer Interaction (HCI), 24--29. AAAI (2002)
4. Cannon-Bowers, J.A., Burns, J.J., Salas, E., Pruitt, J.S.: Advanced technology in scenario-based training. In: Cannon-Bowers, J.A., Salas, E. (eds.), Making decisions under stress: implications for individual and team training. 365--374. APA, Washington, DC (1998)
5 Fowlkes, J., Dwyer, D.J., Oser, R.L., Salas, E.: Event-based approach to training (EBAT). The International Journal of Aviation Psychology, 8(3), 209--222 (1998)
6 Bosch, K. van den, Riemersma, J.B.J.: Reflections on scenario-based training in tactical command. In: Schiflett, S.G., Elliott, L.R., Salas, E., Coovert, M.D. (eds.), Scaled Worlds: Development, Validation and Applications. 1--21. Ashgate, Aldershot (2004)
7 Doesburg, W.A., Bosch, K. van den: Cognitive Model Supported Tactical Training Simulation. In: Conference on Behavioral Representation in Modeling and Simulation (BRIMS), 313--320. Universal city, CA (2005)
8 Rao, A., Georgeff, M.: Modeling rational agents within a BDI-architecture. In: Allen, J., Fikes, R., Sandewall, E. (eds.), Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, 473--484. Morgan Kaufmann publishers Inc., San Mateo, CA, USA (1991)
9 Riedl, M.O., Stern,O.: Believable agents and intelligent scenario direction for social and cultural leadership training. In: Proc. of the 15th Conference on Behavior Representation in Modeling and Simulation. Darmstadt, Germany (2006)
10. Riedl, M.O., Lane, H.C., Hill, R., Swartout, W.: 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 (2005)
11. Bratman, M.: Intention, Plans and Practical Reason. Harvard University Press, Cambridge, Massachusets (1987)
12. Norling, E.: 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, 1080--1081. Melbourne, Australia (2003)
13. Bosch, K. van den, Doesburg, W.A. van: Training Tactical Decision Making Using Cognitive Models. In: Proceedings of the Seventh International NDM Conference. Amsterdam, the Netherlands (2005)
14. Norling, E.: Folk Psychology for Human Modelling: Extending the BDI Paradigm. In: Third International Joint Conference on Autonomous Agents & Multi Agent Systems, 202--209. New York (2004)
15. Heuvelink, A., Mioch, T.: FeGA: A cognitive Feedback Generating Agent. In: Proceedings of the 7th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2008). IEEE Computer Society Press, *to appear* (2008)