AGENT-BASED MODELING TO SIMULATE TRAINING SCENARIOS

Zakaria Maamar
zakaria.maamar@zu.ac.ae
College of Information Systems
Zayed University
PO Box 19282, Dubai
United Arab Emirates

Paul Labbé
paul.labbe@drev.dnd.ca
Command & Control Information
Systems and Experimentation Section
Defence Research Establishment Valcartier
2459 Pie-XI Blvd. North, Val-Bélair,
QC, G3J 1X5, Canada

Abstract

This paper presents the SSAHLA (Simulation based on Software Agents and the High Level Architecture) project. It aims at designing a simulation environment for military-training scenarios, using software agents and the HLA. Agents are considered due to of their relevant characteristics to simulation, such as autonomy, adaptability, mobility, and sociability. The HLA is considered as a communication middleware between the participants who wish to be involved in training scenarios. In fact, each participant is associated with different agents that work for him and thus, perform operations on his behalf. Designing agents and running them on top of the HLA constitutes a novel approach to deal with military-training scenarios.

1. Introduction

Training is a key factor to the development and enhancement of military forces' background and to the success of the missions these forces will undertake. Training, as an asset, aims at specifying to the participants, e.g., commanders, operators, and soldiers, how they should behave and react in front of real situations. Furthermore, training allows achieving a high level of forces' preparedness. However, deploying military forces for training needs is complex, costly, and time consuming. It usually requires several weeks of preparation. Therefore, it becomes relevant to suggest facilities for training. Digitalizing battlefields seems to be among these facilities. Indeed, training will mostly consist of simulating the missions to be carried out. Simulation has several advantages, such as:

- Allowing participants to rehearsal their missions, before being engaged.
- Reducing costs by avoiding real deployment.
- Recording missions for learning purposes.
- Attempting different combinations for evaluation purposes, e.g. assessing decisions impact could be done several times.
- Investigating and gaining insights into the utility and feasibility of new concepts and advanced technologies.

There exist several research projects in simulation [6], particularly in the military domain [1,10]. In this domain, most of the projects aim at connecting military systems, called Command & Control Information Systems (C2ISs), to simulation applications. The High Level Architecture (HLA) is the middleware that is widely used for "feeding" these applications with data from the C2ISs. In our work, we intend at leveraging this connection by designing a collaborative environment on top of the different simulation applications. **S**imulation based on **S**oftware **A**gents and the **H**igh **L**evel **A**rchitecture (SSAHLA) denotes this environment. How to design and implement this environment constitutes our main concern? Our work applies to both situations: joint and coalition. Such situations demand that critical battlefields data should be shared. In coalitions, each partaking country could have its own simulation application that needs to be connected to the applications of other countries. Exchanging update messages should achieve the coherence between the different simulation applications.

With digitalization facilities, simulation allows a participant to take part to military missions, without being "physically" in the battlefield. In order to meet preparedness and promptness requirements, this participant should be aware of the events that could occur in his battlefield and in other battlefields as well. In addition, due to the rapid development of information technologies a participant is able to receive multiple types of information from multiple sources, such as radars, Web sites, and on-line news. We are convinced that our participant would be rapidly overwhelmed with a huge quantity of information and events that need to be dealt with in a short period of time. This puts too much pressure on our participant. To assist such participants, we suggest associating them with Software Agents (SAs) [2]. A SA is an autonomous entity having the abilities to assist users in performing their tasks, to collaborate with other SAs to jointly solve different problems, and to answer users' needs. One of the advantages of using agents is their ability to adapt their behavior at run-time. This permits reducing the need to foreseen all possible scenarios and changes a participant could come across. Consequently, the digitalized battlefield would consist of several types of agents, identifying for example the friendly troops, the enemy troops, the spatial entities, and the information sources. Agents would be running on top of this battlefield.

The remainder of this paper presents further details on the SSAHLA project. Section 2 defines the basic concepts that are used for the design of the SSAHLA environment. Section 3 suggests this environment's architecture and functioning. Section 4 specifies an agent in a simulated theatre. Finally, Section 5 concludes the paper.

2. Background

This section presents the concepts that are used in the SSAHLA environment. Details on C2ISs, SAs, and the HLA are given.

- **What is a C2IS?** Nowadays, information technologies are an inherent part of the commanders' decision-making process. Particularly, C2ISs help commanders to obtain a view of the situation in which they are involved. A C2IS consists of a structure, tasks, and functions [3]. The structure represents an assembly of facilities, arranged to meet the C2IS's objectives. To reach these objectives, the functions are triggered and the needed tasks are carried out. Different types of functions exist; they vary from planning and weather forecast to data fusion. It occurs that some functions receive messages from the external environment, e.g. sensors, through communication modules.
- **What is a SA?** In Distributed Artificial Intelligence, researchers have studied several issues related to the distribution and coordination of knowledge and actions in environments involving multiple entities, called agents. Agents can take different

forms depending on the nature of the environment in which they evolve. A particular type of agents, known as software agents, has recently attracted much attention [2]. A SA is an autonomous entity that acts on a user's behalf, takes initiatives, and makes decisions. For more details on agents, see [4].

– **What is the HLA?** The HLA is a software interoperability framework evolving under the guidance of the Defense Modeling and Simulation Office (DMSO) Architecture Management Group (AMG). The HLA provides a set of Application Programming Interfaces (APIs) for run-time data interchange services, pre-runtime templates and tools for reconciling data exchange details between applications, and rules for proper use of these services and tools. The HLA's APIs allow to applications that contribute to simulation to know what data is expected from them and in what form data will be delivered to them. According to [5], linking a C2IS to a simulation application pursues the following course of actions: a simulation application is identified, the required data flows between the C2IS and this application are analyzed, simulation data is mapped into message formats, an inter-process communication technique is selected and finally, the necessary software is implemented at the simulation side. We recall that the Object Management Group has adopted the HLA as the facility for Distributed Simulation Systems 1.0 in Nov. 98.

3. Presentation of the SSAHLA Environment

In this section, the architecture and functioning of the SSAHLA environment are described. The concepts this environment encompasses are also discussed.

In [13], the authors define several characteristics that should be part of a multi-agent simulation system. For instance, the system should allow unplanned events to occur. The purpose is to make simulated situations more realistic and applicable. To this end, a-priori observations should be conducted. Moreover, the system should support deterministic experimentation, at least to compare different attempts and retain the conclusions. Last but not least, the system should allow agents to act and interact, according to the events they detect and the environment they observe.

3.1 Architecture

Figure 1 presents the general architecture of the SSAHLA environment. Each participant, e.g. a country, is associated with multiple software agents that fulfill operations on his behalf. In fact, these agents are the basis of what we call the simulated theatre. Currently, we are considering two types of agents: event-agents and participant-agents. Event-agents monitor the C2ISs' behavior in term of updates and associate these updates with events. Next, event-agents identify the relevant participant-agents that will be in charge of fulfilling these events. Events and their outcomes could be advertised to participants of other theatres. These participants could have expressed their interests, through subscription mechanisms. The HLA is in charge of managing broadcasting and subscription. In the SSAHLA environment, it may occur that a participant either joins or leaves this environment. This requires from the SSAHLA architecture to be adaptable. Details on how this adaptability is achieved are explained later on.

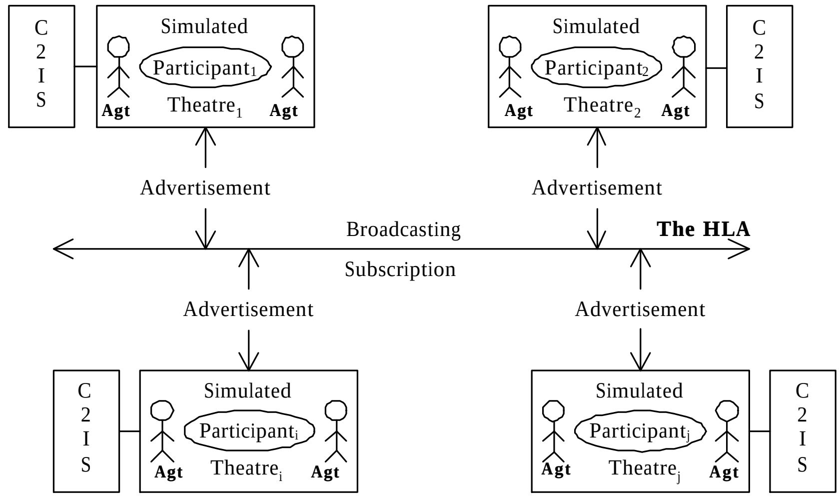


Figure 1 Architecture of the SSAHLA environment

In what follows we explain the concepts of the SSAHLA architecture, namely simulated theatre, event-agents, and participant-agents.

- **What is a simulated theatre?** It is a digitalization of a participant's battlefield. A theatre is divided into several regions (two types of regions are considered: intervention and influence, cf. Section 4.1). Agents populate regions and specific symbols are used to represent them within these regions. In addition, each theatre has topographic symbols that illustrate for example rivers, roads, and buildings. In the SSAHLA architecture, each simulated theatre maintains a local picture and a shared picture as well. The local picture represents the operations that are carried out within a participant's theatre. However, the shared picture is "fed" by the data other simulated theatres have decided to push to another theatre through the HLA.
- **What is an event-agent?** It monitors the updates that occur at the C2IS level. In case there is an update, e.g. troops movement; it is, firstly, materialized in the simulated theatre using a specific symbol. An event-agent is in charge of this operation. Secondly, this event-agent associates this update with an event. Finally, the same agent triggers the Condition-Action template that corresponds to this event.
- **What is a participant-agent?** It implements the actions that belong to the events that have been detected by the event-agent. It may occur that several participant-agents are needed, in order to deal with an event. In that case, the event-agent designates a supervisor participant-agent that will be in charge of coordinating the operations of its group of participant-agents. We define a group as a set of self-interested agents that agree upon cooperation to achieve common missions. The principal motivation behind grouping participant-agents and making them to cooperate is to share responsibility and use abilities of specific agents. Samples of the actions that participant-agents could implement are attacking enemy troops and moving a camp to another position. Usually, undertaking actions either succeeds or fails according to their outcomes. These outcomes update the C2IS's content. In the SSAHLA environment, we aim at embodying participant-agents with capabilities that permit them to answer: under what terms they carried out actions, why they carried out these actions, how they carried out these actions, and when they carried out these actions.

3.2 Functioning

The functioning of the SSAHLA environment consists of five (5) steps (cf. Figure 2). Events are the backbone of this functioning. Indeed, events force agents to act in response, on the basis of the knowledge they possess and the operations in-progress. In what follows, numbers between brackets correspond to the chronology of operations in Figure 2.

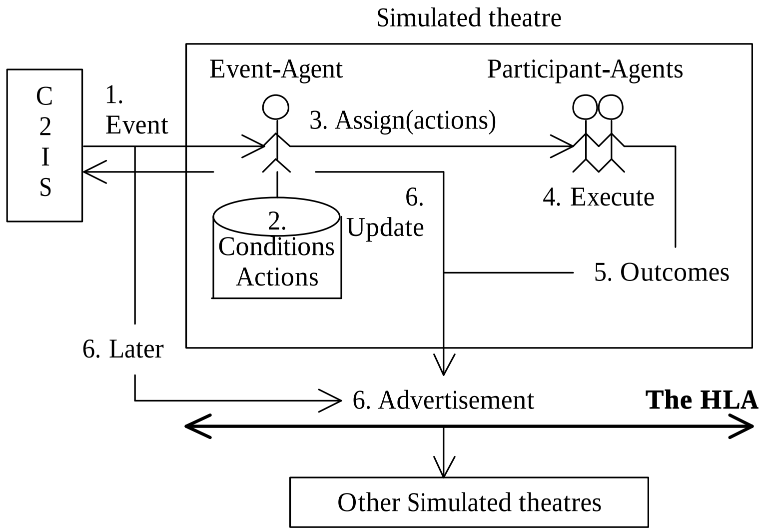


Figure 2 Functioning of the SSAHLA environment

- Step 1: it consists of setting up the simulated theatre and its agents and connecting the C2IS to this theatre. Event-agents will be mandated to monitor the C2IS. For participant-agents, a few of them will be tasked to interact with participants in order to identify their interests regarding other theatres. These interests will be transmitted to these theatres, through the HLA.
- Step 2: The HLA launches an exchange process of interests between the participant-agents of the different theatres. All participants acknowledge theatre's rules of engagement (ROEs). Thus, each theatre would be alert to its interests as well as to the interests of other theatres. In addition, this exchange allows the HLA to be aware of who is interested in what for broadcasting purposes.
- Step 3: as soon as an event-agent detects a modification in the C2IS's behavior (1), this modification is associated with an event. Next, the Condition-Action template of this event is identified and then, triggered (2). If conditions are valid, the event-agent selects relevant participant-agents and entrusts them the actions to be fulfilled. Among these agents, a supervisor participant-agent is designated. When carrying out actions, participant-agents are in constant interactions. Indeed, an action performed by a participant-agent could have results on another one. In the SAAHLA environment, influence and commitment relationships illustrate these interactions. They are detailed in Section 4.
- Step 4: after forwarding the actions to the participant-agent supervisor who puts together its group of participant-agents (3), actions are assigned and then executed (4). Agents record each new course of actions (COAs) and create generalized models for future COAs. Among these actions, the event that is currently under consideration could be forwarded to other theatres, through the HLA. The outcomes of actions influence the shape of theatres, for example a combat unit moves to another position, the space of a combat zone shrinks, etc.
- Step 5: as soon as the actions are completed (5), the participant-agent supervisor

updates the content of its C2IS according to these actions' outcomes (6). Then, it broadcasts these outcomes to the participant-agents of other theatres (6). We recall that the HLA is aware of who is interested in what. According to these outcomes, these participant-agents start acting in response. These agents should be able to know if outcomes have either positive or negative impact on them. For instance, if a participant-agent knows that the friendly troops from another theatre have been defeated then this participant-agent should know that it has been influenced negatively. Thus, it should behave consequently for example by expecting attacks from the enemy troops.

In the SSAHLA environment, participant-agents exhibit both behaviors: reactive and cognitive. Acting reactively, participant-agents receive from event-agents the actions they will perform. Interesting are situations in which participant-agents act cognitively. In the next section, we underline that agents influence each other either positively or negatively. Depending on how they are influenced, participant-agents commit themselves and adapt their behaviors.

4. Agent Specification in a Simulated Theatre

In a simulated theatre, it is important to be aware of the places, i.e. regions, in which participants associated with their event-agents and participant-agents are authorized to perform. In what follows, we suggest an approach for specifying agents in a theatre.

4.1 Specification Approach

In the SSAHLA environment, agents are defined at three levels: intrinsic, organisational, and social. Currently, the suggested levels are more relevant to participant-agents than to event-agents.

1. The intrinsic level is related to the agent and has the following properties:
   - Identifier.
   - Role.
   - Services (capabilities to fulfill, e.g. handling an event, looking for participant-agents, coordinating participant-agents, etc.).
   - Knowledge (in terms of goals, beliefs, intentions, and plans).
2. The organisational level is related to the management of the agent according to the theatre in which it evolves. This level has the following properties:
   - Supervisor: (yes/no) ? identifier (if an agent has a supervisor).
   - Subordinates: (yes/no) (if an agent has subordinates with their identifiers).
     - If yes ? Group of subordinates - {identifiers}.
   - Intervention regions (places in which an agent executes its services).
   - Influence regions (an agent influences other agents that are within specific places, known as influence regions).
     - Commitment type ? Strong/Weak (relationships between agents, based on influence type. Commitment applies only to agents within influence regions).
3. The social level is related to the behavior of the agent towards other agents. This level has the following properties:
   - Message ? interaction pattern (how agent should behave, according to the messages it receives).
   - Event ? Condition-Action template (applies only to event-agents).

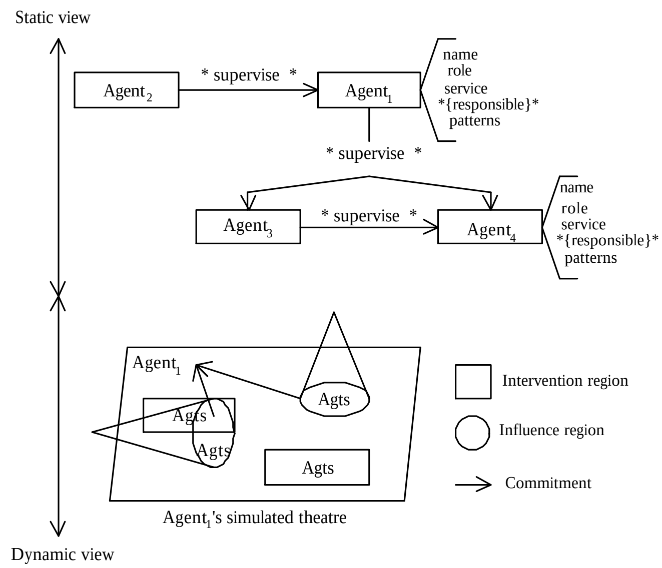


Figure 3 Agent representation in a simulated theatre

Figure 3 represents the three levels that specify an agent in a simulated theatre. This representation has a static view and a dynamic view as well. The static view consists of all the intrinsic level's properties, a part of the organisational-level's properties (agent-supervisor and agent-subordinates) and finally, all the social level's properties. The dynamic view consists of the properties that are left from the organisational level (intervention region, influence region, and commitment type). The dynamic view illustrates how a simulated theatre adjusts its shape, on the basis of the outcomes of the actions to be fulfilled. For example an intervention region could absorb an influence region. In the SSAHLA environment, the following assumptions are made regarding the dynamic view:

- An intervention region of an agent could be empty, i.e. it doesn't contain agents.
- An intervention region and an influence region could overlap.
- Agents within intervention regions or influence regions could be either friendly or hostile.
- Influence regions should contain at least an agent. If there are no agents, these regions disappear.

How to manipulate intervention regions?

In the SSAHLA environment, each intervention region has a supervisor-agent[1]. The mechanisms that operate this region correspond to the services that will embody this agent. Among these services, we cite:

- Region cleaning: its purpose is to purge a region from agents. In fact, the cleaning initializes a region.
- Execution authorization: agents of an intervention region request permissions from the supervisor-agent, before performing their services.
- Region management: depending on the outcomes of actions, an intervention region

[1] This type of agents has not been introduced in Section 2.

could either shrink or expand. In both cases, potential overlapping with other intervention regions as well as with other influence regions could occur. Such overlapping should be detected and traced.

– Content identification: its purpose is to be aware of the content of a region, in terms of types of agents, their numbers, their origins, etc. This allows an agent to know who are its acquaintances in an intervention region.

How to manipulate influence regions?

In the SSAHLA environment, each influence region has a supervisor-agent. The mechanisms that operate this region correspond to the services that will embody this agent. Among these services, we cite:

– Region cleaning: its purpose is to purge a region from its agents. In fact, the cleaning occurs when an influence region is completely annexed to an intervention region.
– Content identification: its purpose is to be aware of the content of a region, in terms of types of agents, their numbers, their origins, etc. This allows an agent to know who is currently evolving in an influence region.

How to perform services?

In the SSAHLA environment, we recall that agents' services are invoked by the actions these agents have been assigned (cf. Figure 2). In order to carry out its services, an agent requires authorizations from whether its supervisor (cf. the organisational level), the supervisor of an intervention region, or both. Table 1 provides examples of service specification. For illustration, we assume that $agent_1$ offers $service_1$ and $service_2$. To perform $service_1$ in $region_1$, $agent_1$ requires authorizations from $agent_2$. $Agent_2$ is $agent_1$ supervisor (link: Agent-Supervisor). However, $agent_1$ does not require authorizations for $service_1$ in case it is executed in $region_2$. Regarding $service_2$, $agent_1$ requires authorizations for $region_1$ and $region_2$. These authorizations are given by $agent_2$ that supervises $agent_1$ (link: Agent-Supervisor), $agent_4$ that supervises $region_1$ (link: Region-Supervisor), and $agent_3$ that supervises $region_2$ (link: Region-Supervisor).

Table 1 Examples of service specification

Service	Intervention Regions	Authorization	Agent	Link
$Service_1$	$Region_1$	Yes	$Agent_2$	Agent-Supervisor
	$Region_2$	No	-	-
$Service_2$	$Region_1$	Yes	$Agent_2$ $Agent_4$	Agent-Supervisor Region-Supervisor
	$Region_2$	Yes	$Agent_3$	Region-Supervisor

4.2 Why participant-agents need to be coordinated?

Participant-agents either from the same simulated theatre or from different simulated theatres have to communicate. In fact, they have to exchange information on different matters, such as the progress of the operations they are in charge. However, in military situations communication could be impeded for several reasons, among them network's states, interception risk, etc. In the SSAHLA environment, in order to prevent such situations and reduce communication workload we suggest associating each participant-agent with an acquaintance model. This model consists of details on the agent itself and on other agents as well. In addition, we suggest decomposing this model into twofold: internal and external. Each part will consist of static and dynamic information.

1. Internal part: reserved to the agent and allows answering the following questions:
    a. Static: What is the agent able to do? And how to perform what the agent is able to do?
    b. Dynamic: What is the agent currently doing?
2. External part: reserved to other agents and allows answering the following questions:
    a. Static: Who are the collaborators of the agent? What are the collaborators able to do? And how to find out the collaborators?
    b. Dynamic: Who are currently collaborating with the agent? What are the collaborators currently doing for the agent? What is the status of what the collaborators are currently doing for the agent? And, what are the collaborators expecting from the agent?

In the SSAHLA environment, each participant-agent could fulfill three roles: a supervisor leading a group of participant-agents, a supervised belonging to a group of participant-agents, or both i.e. supervisor/supervised (cf. Section 4.1, organisational level). In all these situations, conflicts could rise:

1. Between the participant-agents of the same group of the same simulated theatre.
2. Between the participant-agents of different groups of the same simulated theatre.
3. Finally, between the participant-agents of different groups of different simulated theatres.

Conflicts could be of several types, namely doctrine, goal, resource, and commitment (adapted from [8]).

- Doctrine conflict: doctrine specifies how a participant should behave according to the situation to which it takes part, such as peace keeping. Since a military operation could involve participants from different origins, conflicts at the doctrine level mainly deal with who is supposed to obey to whom?
- Goal conflict: this conflict could occur when participants do not agree on the operations to be performed first. In fact, each participant has its priorities. Participants need to identify the goals to pursue otherwise their efforts could vanish.
- Resource conflict: participants require resources in order to carry out their operations. However, it may happen that different operations need to use the same resource. Conflicts at the resource level should deal with who is supposed to use what?
- Commitment conflict: a participant could make decisions based on the commitment he got from other participants. In case a commitment is not respected, this participant should review his decisions and probably cancel his operations.

4.3 Intervention regions vs. influence regions

In the SSAHLA environment, a simulated theatre consists of two types of regions: intervention and influence (cf. Figure 3). In what follows, we enumerate agents' positions regarding both regions. For illustration, we assume the existence of $agent_1$ and $agent_2$ (cf. Figure 4). We recall that both agents have their own intervention regions.

1. $Agent_2$ is within the intervention region of $agent_1$. When $agent_1$ influences $agent_2$, two

situations take place:

a) The influence region that contains $agent_2$ is totally included in the intervention region of $agent_1$ (cf. Figure 4 – 1.a).

b) The influence region that contains $agent_2$ is partially included in the intervention region of $agent_1$ (cf. Figure 4 – 1.b).

2. $Agent_2$ is not in the intervention region of $agent_1$. When $agent_1$ influences $agent_2$, two situations take place:

   a) The influence region that contains $agent_2$ is totally included in the intervention region of $agent_2$ (cf. Figure 4 – 2.a).

   b) The influence region that contains $agent_2$ is partially included in the intervention region of $agent_2$ (cf. Figure 4 – 2.b).

   c) Note: another situation, which is not discussed here, could exist. The influence region that contains $agent_2$ could be included, either totally or partially in the intervention region of another $agent_{i,\ i\#1,2}$.

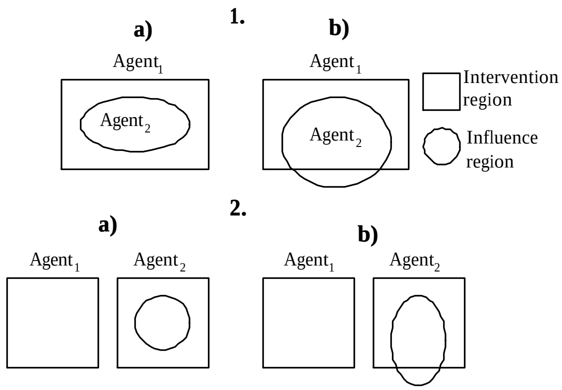


Figure 4 Intervention region *vs.* influence region

4.4 Influence Analysis

When different agents are set up together, they usually interact and hence, influence each other. This is the case in military situations where different combat units need to communicate to each other their intentions and actions for different reasons: identify the right target, avoid efforts duplication, etc. In what follows, we explain how the SSAHLA environment deals with influence.

Specification

In the SSAHLA environment, an agent could influence another agent at three different levels, namely goal, resource, and task. Since influence could be either positive or negative, the following combinations are obtained (cf.

Table 2). We assume that $agent_1$ influences $agent_2$.

Table 2 Types of influences between agents

Influence	Type		Description
Goal	Pos.	**(+)**	$Agent_1$ generates a new goal that will support $agent_2$ in achieving its goals. $Agent_1$ will be in charge of satisfying this new goal for the benefit of $agent_2$. *Facilitate relationship between goals*.
	Neg.	**(-)**	$Agent_1$ generates a new goal that will delay $agent_2$ in achieving its goals. In fact, $agent_2$ will be in charge of satisfying this new goal for the benefit of $agent_1$. *Hinder relationship between goals*.
Task	Pos.	**(+)**	$Agent_1$ carries out some of $agent_2$'s tasks on its behalf. *Conduct relationship between agents and tasks*.
	Neg.	**(-)**	$Agent_1$ entrusts some of its tasks to $agent_2$, in addition to the tasks $agent_2$ already has. *Work for relationship between agents and tasks*.
Resource	Pos.	**(+)**	$Agent_1$ offers some of its resources to $agent_2$. This helps $agent_2$ in carrying out its tasks as well as in achieving its goals. *Offer relationship between agents and resources*.
	Neg.	**(-)**	$Agent_1$ takes over some of $agent_2$'s resources. Therefore, $agent_2$ will lack resources for carrying out its tasks as well as for achieving its goals. *Take over relationship between agents and resources*.

Representation

Figure 5 suggests the symbols that are used in representing the different types of influences.

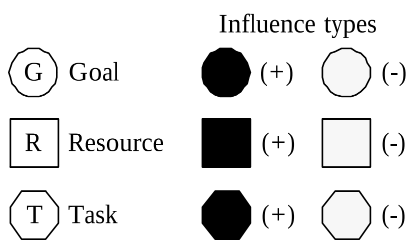
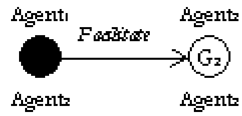
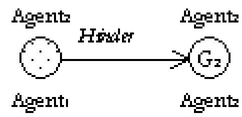
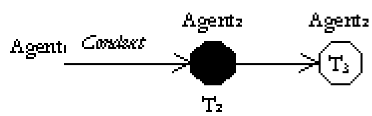
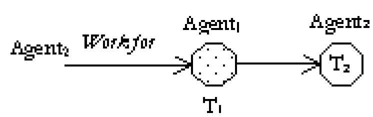


Figure 5 Symbols for influence representation

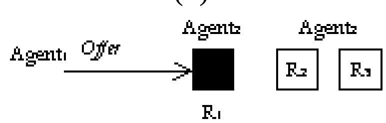
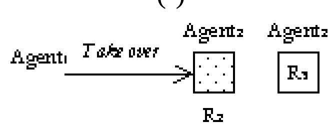
Goal influence

T	Agent$_2$ works towards achieving G_2 goal.	
T+1	Agent$_1$ influences agent$_2$	**(+)**  Facilitate(new_goal,G_2) Agent$_1$ generates a new goal, filled circle, for the benefit of agent$_2$.
		(-)  Hinder(new_goal,G_2) Agent$_2$ carries out the new goal, dotted circle, for the benefit of agent$_1$.

Task influence

T	Agent$_2$ carries out T_2 and T_3 tasks.	
T+1	Agent$_1$ influences agent$_2$	(+)  Conduct(Agent$_1$,T_2,Agent$_2$) Agent$_1$ conducts T_2 task, filled octagon, for the benefit of agent$_1$.
		(-)  Work-for(Agent$_2$,T_1,Agent$_1$) Agent$_2$ works for agent$_1$ regarding T_1 task, dotted octagon. T_1 task will precede T_2 and T_3 tasks.

Resource influence

T:	Agent$_2$ manages R_2 and R_3 resources.	
T+1	Agent$_1$ influences agent$_2$	(+)  Offer(Agent$_1$,R_1,Agent$_2$) Agent$_1$ offers R_1 resource, filled square, to agent$_2$.
		(-)  Take-over(Agent$_1$,R_2,Agent$_2$) Agent$_1$ takes over R_2 resource, dotted square, of agent$_2$.

Commitment vs. Influence

Usually, a commitment is a relationship that allows an agent or a group of agents to rely on the services another agent or group of agents will provide. Based on this relationship, an agent plans its operations considering the fact that it could use these services, when needed. A commitment is mainly viewed as a promise. In the SSAHLA environment, commitment is decomposed into two types: strong and weak. When an agent influences another one, either positively or negatively, a commitment is set up between both agents: from the agent that influences to the agent that is influenced, and *vice-versa*. As discussed in

Table 2, commitment will mainly be identified at resource and task levels. Therefore, the commitment type, i.e. strong and weak, will depend on the influence type, i.e. positive and negative. In what follows, we define commitment in opposition to influence. For illustration purposes, we assume the existence of $agent_1$ and $agent_2$.

- **If** $agent_1$ influences positively $agent_2$ //promise comes from $agent_1$.
  Then ($agent_2$ commits weakly to $agent_1$) **Or** ($agent_1$ commits strongly to $agent_2$).
- **If** $agent_1$ influences negatively $agent_2$ //promise comes from $agent_2$.
  Then ($agent_2$ commits strongly to $agent_1$) **Or** ($agent_1$ commits weakly to $agent_2$).

5. Conclusion

Our national and international military operations are complex and expensive. Both factors require sufficient training to offer the driving nations military operations capabilities to face securely and cost effectively unknown future threats and disasters. Due to high cost of such training, alternatives to "live exercises" are sought. The 21st century marks the beginning of the demonstration of efficient and cost-effective distributed collaborative activities with synthetic environments and autonomous software agents that support human agent activities for various businesses.

In this paper we have shown that the SSAHLA (Simulation based on Software Agents and the High Level Architecture) project aims at designing a simulation environment for military-training scenarios, using software agents and the HLA. Agents are considered due to of their relevant characteristics to simulation, such as autonomy, adaptability, mobility, and sociability. The HLA is considered as a communication middleware between the participants who wish to be involved in training scenarios. In fact, each participant is associated with different agents that work for him and thus, perform operations on his behalf. Designing agents and running them on top of the HLA constitutes a novel approach to deal with military-training scenarios.

The advantages of using this approach includes:

a- exploring the domain space of interactions in several military operations,
b- assessing artificial intelligence value for some of the interactions,
c- developing concept of operations (CONOPS) for systems before their operational deployment, and
d- training military for current and future systems and CONOPS at a fraction of the cost.

References

[1] Simulation Interoperability Workshops. http://www.sisostds.org/ visited December 2000.

[2] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1) pp. 7-38, 1998.

[3] S. Malerud, E.H. Feet, and U. Thorsen. A method for analysing command and control systems. Norwegian Defence Research Establishment (FFI) N-220 Kjeller, Norway.

[4] S. Franklin and A. Gaesser. Is it an agent, or just a program? A taxonomy for autonomous agents. in *Proc. of 3rd International Workshop on Agent Theories, Architecture and Language (ATAL'96)*. Springer-Verlag, LNAI, 1996.

[5] R. Lutz, M. Salisbury, and G. Bidwell. A Demonstration of C2I System-to-Simulation Interoperability: The NSS/GCCS-M Federation. in *Simulation Interoperability Workshop* (SIS'99). Florida, 1999.

[6] Multi-agent systems and Agent-Based Simulation Workshops. 1998, 2000.

[7] F. Bousquet and D. Gautier. Comparaison de deux approches de modélisation des dynamiques spatiales par simulation multi-agents : les approches spatiales et acteurs. *Cybergeo*, 89. 1999.

[8] S. El hadouaj, A. Drogoul, and S. Espié. To combine reactivity and anticipation: the case of conflicts resolution in a simulated road traffic. in *The Second Workshop on Multi-Agent Based Simulation*, Boston, MA, July 2000.

[9] V. Marik, M. Pechoucek, O. Stepankova, and J. Lazansky. ProPlanT: multi-agent system for production planning. *International Journal of Applied Artificial Intelligence*. September 2000.

[10] International Command and Control Research and Technology Symposium, Modeling & Simulation TrackV. http://www.dodccrp.org/ visited January 2001.

[11] S. G Person. Interoperability through education, training and exercises- simulations as a means for regional cooperation with PfP simulation network as an example. *in* The 5th International Command and Control Research and Technology Symposium, Modeling & Simulation Track. http://www.dodccrp.org/ visited January 2001.

[12] J. Debenham. Supporting knowledge-Driven processes in a multiagent process management system. in *proceedings Twentieth International Conference on Knowledge Based Systems and Applied Artificial Intelligence, ES'2000: Research and Development in Intelligent Systems XV*. Cambridge UK, 2000.

[13] M. E. P. Hanks and P. R. Cohen. Benchmarks, testbeds, controlled experimentation, and the design of agent architecture. *AI Magazine*, 14(4): pp. 17-42, 1993. Winter issue.