Pluggable Brains: Separating the Intelligent Behavior from the Simulated Player

Philip Kerbusch, Willem van Doesburg
TNO Defense, Security and Safety
The Hague, Netherlands
philip.kerbusch@tno.nl, willem.vandoesburg@tno.nl

Roger Jansen, Peter Langeslag
TNO Defense, Security and Safety
The Hague, Netherlands
roger.jansen@tno.nl, peter.langeslag@tno.nl

ABSTRACT

The use of simulated environments in the military domain has increased significantly over the past years, and with it the demand for more realistically behaving autonomous systems. The autonomous systems become ever more complex and need to support advanced human-machine interactions when applied for training and instruction (virtual role players) or concept development and experimentation (e.g. adjustable autonomy of unmanned vehicles).

In order to remain flexible, cost-effective and maintainable, a re-evaluation is needed of the way in which these autonomous systems are integrated in the simulated environment.

In this paper we describe our approach of separating behavior components ('Artificial Intelligence' (AI) or 'Brains') from the simulation engine. To make this decoupling as efficient as possible, both the simulation engine and the AI need to provide a suitable interface. To enable maintainability and reusability the interface should support legacy simulation components and enable iterative development. We therefore developed the idea of using a double decoupling. In this approach the interface has a part that can be reused in other systems and a part that is specific for every system.

The selected data exchange mechanism between the simulator and the AI is the High Level Architecture (HLA IEEE1516). HLA is the widely used standard for coupling distributed simulators. The modular Federation Object Model (FOM) feature introduced by the latest HLA version, IEEE1516-2010, supports the idea of decoupling reusable and specific modules very well, and allows gradual development of extension modules and (scenario) specific modules.

The feasibility of our approach was demonstrated by two experiments, in which several commercial of the shelf (COTS) and proprietary tools and simulation components were integrated, in different formation, through HLA.

The paper will present the design and development of the 'Pluggable Brains' approach and discuss the initial results of the two experiments.

ABOUT THE AUTHORS

Philip Kerbusch, MSc is a research scientist at TNO Defense, Security and Safety. Philip holds a MSc. degree cum laude in Artificial Intelligence from Maastricht University (2007). He has a strong background in computer science and knowledge engineering. His current work involves various applications of artificial intelligence in the defense and safety domain, specializing in behavior modeling for serious games and simulation.

Willem van Doesburg, MSc is a research scientist at TNO Defense, Security and Safety. Willem holds a MSc. Degree in Philosophy Cognition and Artificial Intelligence from Utrecht University. His work has centered around the application of artificial intelligence techniques in training simulations, serious games and more recently in user interfaces.

Roger Jansen, MSc is a member of the scientific staff in the M&S department at TNO Defense, Security and Safety in the Netherlands. He holds an M.Sc. degree in Computing Science and a Master of Technological Design (MTD) degree in Software Technology, both from Eindhoven University of Technology, The Netherlands. He works in the field of distributed simulation and his research interests include distributed computing and simulation interoperability.

Peter Langeslag, MSc. is a member of the scientific staff in the Defense, Security and Safety Division at TNO. As a research engineer he has a broad experience in the architectural design of simulators and the procurement and development of training simulators for the Dutch army. Peter contributed to the NATO MSG-003 Feasibility Study on Modeling & Simulation Technology in Support of Simulation Based Acquisition (SBA) and was a member of the MSG-030 task group on Collaborative Working Environments (CWEs) for Simulation Based Acquisition (SBA). Peter has profound knowledge of simulation protocols like DIS and HLA. This was expressed by his work for the HLA working group NATO MSG-050.

Pluggable Brains: Separating the Intelligent Behavior from the Simulated Player

Philip Kerbusch, Willem van Doesburg
TNO Defense, Security and Safety
The Hague, Netherlands
philip.kerbusch@tno.nl, willem.vandoesburg@tno.nl

Roger Jansen, Peter Langeslag
TNO Defense, Security and Safety
The Hague, Netherlands
roger.jansen@tno.nl, peter.langeslag@tno.nl

INTRODUCTION

Over the past decade the military, in a broad sense, has embraced simulation. Today, simulation is being used for an ever greater variety of applications; ranging from supporting large scale collective mission rehearsal to cultural awareness training on personal handheld devices. This increased use and diversity of simulation poses new demands on simulated environments, both in terms of the features a user expects as well as the effort needed to cost-effectively deliver those features. One of the topics dealing with this challenge is the artificial intelligence used in the simulated environments.

Artificial intelligence (AI) has numerous applications in a simulated environment. For example, AI can be used to model virtual role-players and entire unmanned (autonomous) systems, but also be the enabling technology for an automated assessment framework. The complexity of these AI applications has increased, as users expect more, especially realism. From a modeling and simulation (M&S) point of view we have been able to deal with this increased complexity; for example, we were able to remain flexible and cost-effective by relying on systems engineering techniques and standardization. Also, a major contribution came from the adoption of the use of middleware components (intermediate software products aimed at a specific task such as AI) from a booming commercial entertainment industry.

However, in recent years two trends have developed that influence the 'traditional' use of AI in simulated environments. The first trend is the growing research into the subject of human-machine interaction (HMI) with respect to autonomous systems and platforms (e.g. unmanned aerial vehicles), the second trend is the use of commercial-off-the-shelf (COTS) available software and tools.

In military practice, as well as in other sectors, there is not only a trend to rely more on technology, but also, the technology is becoming increasingly complex, especially with increased adoption of AI and autonomous systems. This trend is paired with a shift of "operators" becoming "decision makers", and as such users expect systems to provide information on which decisions can reliably be based. However, when the system supplying the information is no longer understandable by the user there is the risk of a "trust gap". For instance, trust in system advice was badly hurt when system weaknesses lead to the USS Vincennes mistakenly shooting down a commercial airliner (Klein, 1999). Recently, much research is dedicated to how human-machine interaction should be designed in order to better align human trust with the quality of information provided by complex systems. Human users have a tendency to rely on human social cues to establish trust. This tendency is so strong that social properties are attributed to machines (Lee & Nass, 2010). Simulation is used to develop and test (using human in the loop experiments) software concepts that use social cues to regulate trust in order to improve human-machine performance. However, there is currently no way to easily federate different combinations of simulations, AI systems and HMI designs.

The use of COTS tools and software components is a trend that breaks away from the more conservative approach of acquiring a one-of-a-kind simulation system (e.g. a command and staff training system). Instead of an expensive time consuming process of requirements analysis, development, and acceptance testing, a readily available product is bought; general consensus is that anticipated shortcomings do not outweigh the benefits of a less expensive, industry driven product.

From a research related M&S perspective the trend of using COTS tools and software components has had implications. As a one-of-a-kind simulation system cannot keep up with an industry driven product in terms of advancement and maintainability, research has adopted many of the same COTS available tools and software components to do concept development and experimentation (CD&E). For AI middleware, as well as other middleware components, this adoption of COTS tools does have drawbacks: for example, some COTS products may not be able to integrate middleware components at all and others may be hard to integrate because of closed source and propriety interface formats.

As a result of the two trends described above (using COTS tools and software components, and HMI research with respect to autonomous systems) the current M&S approaches of keeping simulated environments flexible, cost-effective, and maintainable are challenged. Middleware components (designed for integration) and closed source tools (designed for customer lock-in) do not mix well, nor do standardized and propriety interface formats. Therefore, M&S industry adopted and developed (Evertsz, Pedrotti, Busetta, Acar, & Ritter, 2009; Levesque, Cazzolato, & Martonosi, 2009; de Kraker, Kerbusch, & Borgers, 2009) a new approach of integrating AI tools and software components into simulated environments that is based on completely decoupling the AI from the simulation engine. We call this making the brains (AI) pluggable.

In the next section, we discuss the proposed 'Pluggable Brains' approach. The third section will cover HLA Evolved and how Federation Object Model (FOM) modules are used in our approach. The fourth section will present the current setup and the initial results that were achieved. Section five will discuss these results and the way forward for the 'Pluggable Brains'.

PLUGGABLE BRAINS

The creation of pluggable brains (and other pluggable components) has long been the desire in the M&S community and much work has been done in this area. What then prompted this desire? What pluggable solutions are already available? Are there standards that support pluggable brains? What is the yield of pluggable brains in terms of flexibility, reuse, maintainability and cost?

The desire for pluggable components has risen because of the large investments made in complex legacy systems. For instance, in the area of simulating fighter pilots much effort and time has gone into building models of fighter pilot behavior (Heinze, Papasimeon, Goss, Cross, & Connell, 2008). However, over time the technology employed grows outmoded. In order not to lose all accumulated knowledge and expertise captured in these models there is a desire to reuse the legacy systems in new architectures, i.e. making them pluggable.

This trend is not only visible within one research organization or research field but across a wide range of fields. Lebiere, Stocco, Reiter, and Juvina (2010) claim that in order to fully exploit the progress research into behavior modeling and AI in simulation it is necessary that local progress can be translated better into progress of the field in general. Because most "brains" are only researched and tested in the context of one simulation environment, little value is generated for other applications or simulation environments. They advise: "*Most fundamentally, a standard interface between cognitive models and task environments is needed to assure portability across tasks and models*" and "*If both tasks and models comply with the interface, both scientific (principled model comparison, separation between task and model) and technical (reusability, productivity) goals will be enhanced and the exponential growth associated with systems embracing common standards (e.g., the Internet, the Personal Computer) will then be possible*". They also draw our attention to the issue of maintainability; the problem of how computational models need to be updated and kept current. Developers should have incentives to maintain their code up-to-date to claim cumulative credit from models developed under previous versions of their framework. Current practice lacks such an incentive to keep models up to date because their dedication to one environment limits the number of stakeholders and there is no culture of crediting model developers.

How do non-defense sectors deal with the use of COTS systems and how do they efficiently and cost-effectively federate multiple COTS components for their needs? Boer (2005) has investigated this in the case of industrial simulation. COTS components are heavily utilized in this area. The most salient problem with the COTS approach, Boer reports, is the accessibility of simulation models. When federating closed systems it is unclear which (internal) model concepts map onto concepts of another simulation model and it is often not possible to have access to the specific data within the simulation in order to communicate it between systems. When interviewing COTS vendors a large portion indicated that they support distributed simulation architectures. However, upon deeper analysis it turns out that this support is very shallow and that the vendors are optimizing and competing on cost effectiveness of their models and, as a result, are more oriented to ever more specialization of their specific model rather than its integration in a larger whole. They view model accessibility as a risk because other vendors might start to compete on those model parts that are opened up.

Defense simulation is traditionally more focused on simulating higher levels of aggregation (total theater of war) and therefore there is less room for very advanced specialization of components. Hence, the comparison with industrial simulation only goes so far. The two sectors do share the pressure to realize systems ever more cost-effectively. So, there is clearly a need to devise ways to easily federate legacy and COTS components to simulate the theater of war and to obtain more leverage from existing "brains" to increase progress in behavior modeling and AI.

Along with the desire for pluggable components many approaches for distributed architectures have been developed over time. What progress is reported in the application of these architectures to defense-simulation?

While we found many proposals for efficiently and cost-effectively federating simulations and brains using distributed architectures, very few report on experience and actual applications.

BinSubaih, Maddock, and Romano (2004) report the ability to reuse domain knowledge by decoupling domain knowledge from simulation logic. Cap, Heuvelink, van den Bosch, and van Doesburg (2011) report a training application in which the non-player characters are controlled by completely decoupled AI. The decoupling is done in a non-standardized way. They report that this approach yields a maintainable and cost-effective system; however, no comparison with other systems was made. Lees, Logan, and Theodoropoulos (2006) report their experience in using HLA to decouple AI from simulation. One of the benefits they claim is that HLA democratizes development of "brains", potentially allowing any developer to access a common simulation arena. As a drawback, they report a slight (fixed) delay in communications between federates. However, the application reported is an academic problem and not a military simulation.

From the above experience reported only Lees *et al.* used HLA (an existing standard for federating systems). During our search no reports have been found about applications that use the latest HLA standard, HLA Evolved.

Obviously the decoupling of brains from simulation is a new area in which little experience is available to draw upon. This is confirmed by Zacharias, MacMillan, & van Hemel (2008) who state that one of the challenges facing the M&S community is to combine components and federate models. How then do we know what the benefit of such a decoupling will be? We will have to rely on conceptual arguments about benefits in terms such as flexibility, maintainability, and cost effectiveness.

Cohen, Ritter, and Haynes (2010) discuss how to make the development of brains for simulation more cost effective. They suggest: use maintenance-oriented development environments and focus on software reuse. They present a development environment that supports these dimensions. However, in our experience one of the most recurring and expensive parts of reusing brains is integrating them in COTS and legacy systems. The authors don't discuss this issue.

The benefits of using a standard decoupling method like HLA are hard to project because of the lack of experience reported. Additionally Boer (2005) reports for industrial simulation that:

1. HLA is rarely applied in industry when integrating COTS simulation packages.
2. When HLA is applied to integrate COTS simulation models, this occurs mostly in defence oriented projects.
3. COTS simulation packages are rarely applied for distributed defence oriented simulation models.
4. The COTS vendors and their customers prefer simple homespun solutions over distributed simulation in industry.

Underlining that we are in the beginning phase of a new trend and that there might be a high threshold (because of the high learning curve of HLA) to use standard decoupling methods. It seems that not many parties are willing to pay the costs before reaping the benefits.

In our view, decoupling simulation and brain brings both cost and benefits. The cost is the initial investment needed to adopt the use of a distributed architecture. The benefit is that the basis of decoupling is a shared ontology, a data-exchange agreement, describing the information types on which federates agree to communicate. This agreement is enforced between brain and simulation, forcing the developer to explicitly formulate the information produced and expected by both simulation and brain. This makes both brain and simulation easier to integrate with other federates. Another benefit is improved maintainability. Changes in the brain or simulation (when the shared ontology is not affected) can happen completely independently of each other. Flexibility is increased by making input/output relations explicit. This enables developers to determine whether a component can be replaced by a different component as long as it supports the input/output relationship. So when better or more efficient components come along, the application can adopt these more easily. Cost effectiveness is in our view improved over a longer term, and consists mostly out of two parts. First, return on investment is improved when a component is built once and used many times. Second, using a distributed approach to separate brain from simulation makes it easier to scale-up the total system.

HLA IMPLEMENTATION

Using the pluggable brains approach described in the previous section results in a setup in which multiple tools and software components collaborate. In M&S terminology the subject of making these components communicate and share information is referred to as simulation interoperability. The High-Level Architecture (HLA) is a widely accepted and adopted IEEE standard that can be used to achieve simulation interoperability.

We chose to use HLA for various reasons. First, HLA is a *shared* data-exchange mechanism, as opposed to point-to-point mechanisms as for example web services. A shared mechanism enables composability (Möller, Morse, Lightner, Little, & Lutz, 2008) because different components need not have detailed knowledge about *each other*, but rather about the information being exchanged. Second, HLA is a widely accepted standard and many COTS simulation components natively support HLA simulation interoperability. This should make integrating such a component into a pluggable brains setup easier. Third, a lot of HLA supporting tools are available, that aid in FOM development by code generation and logging.

HLA Evolved: FOM Modules

A fourth, important, reason for choosing HLA is that the most recent version of the HLA standard, IEEE 1516-2010 or HLA Evolved (IEEE, 2010), introduces the concept of Federation Object Model (FOM) *modules*. In HLA a FOM is used to unambiguously describe the information that is exchanged via HLA. It is evident that a well-designed FOM enables reuse. Not solely in terms of the effort needed to develop a FOM, but also in terms of being able to reuse tools and software components that already are compatible with the FOM. The FOM modules introduced with HLA Evolved enable the composition of the FOM from a set of separate modules. Modules can be entirely independent, for example describing different domains, or modules can extend other modules, adding new features or functionality to already existing parts.

By dividing and extending FOM modules a more flexible, standardized, and reusable FOMs can be achieved. Also, an existing composition of FOM modules can easily be extended by adding or replacing a module, without breaking the integration of all components that are using the old FOM modules.

Reusable FOM Modules

In order to achieve reuse of FOM modules Möller, Gustavson, Lutz, and Löfstrand (2007) suggest using a FOM development process in which Base-Object Models (BOM) and FOM modules are worked with. A BOM describes the relations and interplay between the objects that are being modeled on a conceptual level, whereas a (combination of) FOM module is an instantiation of this conceptual model tied closely to HLA. By having the BOMs and FOM modules evolve simultaneously when new functionality is required, traceability from conceptual model to FOM modules can be maintained. This traceability makes it easier to identify FOM modules that can be reused, or that have the potential of being standardized for a set of application domains.

Decoupling among FOM Modules

In our approach we propose at least a double decoupling between brains and simulation, see Figure 1.

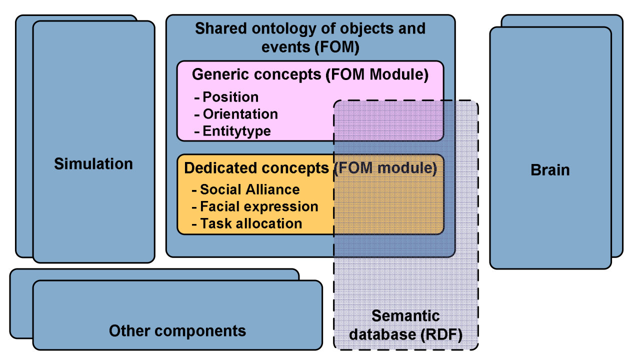

Figure 1. Overview of decoupling between brains, simulation, and other components.

First, placing the brain outside the simulation and thus decoupling these systems allows the developer to choose from available AI middleware. *Secondly,* in many situations much of the shared concepts between brains and simulation can be reused; for instance, concepts like position, orientation, entity types. It serves to encapsulate such concepts in a FOM module for easy reuse. All other concepts that are unique to a specific combination of brains and simulation should be encapsulated in one or more dedicated FOM modules. In our project, a *third* decoupling is added, but this is not strictly necessary to create pluggable brains. For our purposes data needs to be searched for using semantic queries using the SPARQL Protocol and RDF Query Language (SPARQL). We have created a database in which some of the concepts in both the generic and dedicated FOM modules are posted. Other components (such as mission support tools) use this database to create specific views on the available data.

We have developed several FOM modules with federation agreements on how to use them. At this moment we use, besides the standard RPR FOM and its extensions for the simulation part, an AI module for general AI and one for task (scenario) specific AI. Further research and experience will be needed to iteratively develop an ideal set of modules.

CURRENT SETUP AND INITIAL RESULTS

The current pluggable brains simulation environment has several tools and software components incorporated. This set of components consists of, COTS simulation software, AI middleware, and HMI components. The first subsection will describe these components and how they are integrated into the simulation environment. The second subsection will present several use-cases in which this simulation environment was used. Also, an evaluation of these use-cases is given.

Current Environment

We have developed a concept CD&E environment using the pluggable brains approach. It is composed of several tools and software components. Each component and its integration into the simulation environment is described below.

Virtual Battlespace 2
Virtual Battlespace 2 (VBS2), shown in Figure 2, is a COTS infantry training simulator that is used by many armed forces around the world, including the Royal Netherlands Army. VBS2 originates from a commercial computer entertainment game, and was brought to the military domain by Bohemia Interactive. VBS2 is relatively open, as it provides various options for users and developers to extend its functionality. As a result, VBS2 is nowadays being used for a variety of training applications and research projects, although primarily intended as an infantry training solution.

Figure 2. A screenshot demonstrating the first-person point of view of VBS2.

VBS2 natively supports HLA, for which Calytrix' LVC Game is used. LVC Game supports the most commonly used reference FOM modules. Recently, LVC Game has been made HLA Evolved compliant, which resulted in a succesful participation of VBS2 in the MSG-068 demonstration at the end of 2010.

However, LVC Game is a closed source solution, which means that LVC Game cannot be easily extended to support specific or extended FOM modules. To be able to work with VBS2 in a pluggable brains environment we build a second HLA plugin for VBS2 using VBS2's Application Scripting Interface (ASI), as depicted in Figure 3.

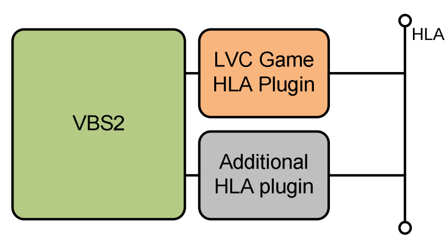

Figure 3. Two HLA plugins integrate VBS2 with the pluggable brains approach.

This second HLA plugin is designed to handle only those FOM modules that LVC Game cannot handle. Although this solution works well in most cases it requires some degree of extra management code; e.g. to keep track of entity identifiers. Another solution would be to fully discard the existing LVC Game plugin and have our own HLA plugin handle all FOM modules.

However, this would require a lot more work to create and maintain.

VR-Forces

VR-Forces, by VT MÄK, shown in Figure 4, is commonly referred to as a Computer Generated Forces (CGF) tool. This COTS simulator can be used to design and simulate large scenarios. VR-Forces, therefore, has high fidelity models of both land and sea based as well as airborne entities.

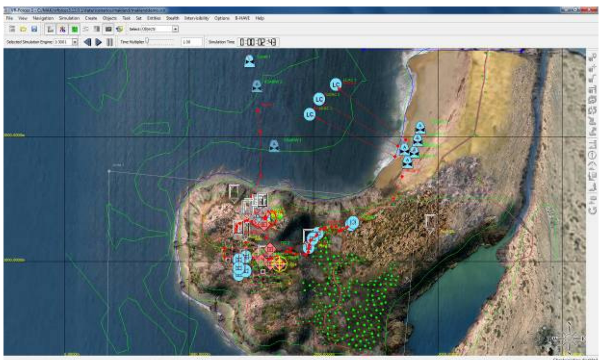

Figure 4. A screenshot of the VR-Forces scenario editor showing a large number of entities.

VR-Forces is used in the pluggable brains environment when a large quantity of entities needs to be simulated or large areas need to be populated (e.g. with civilians). VR-Forces is also well suited to modify the scenario at runtime; experimentation staff can change the course of the scenario when unexpected situations occur (e.g. due to man-in-the-loop involvement).

VR-Forces inherently supports HLA trough its VR-Link framework that is integrated into VR-Forces. VR-Link can be extended to handle specific and extended FOMs. However, at the time of writing, VR-Link does not yet support HLA 1516-2010, thus FOM modules cannot be used. Therefore, we used a HLA 1516 to HLA 1516-2010 (Evolved) adaptor to integrate VR-Forces into the pluggable brains approach, as shown in Figure 5.

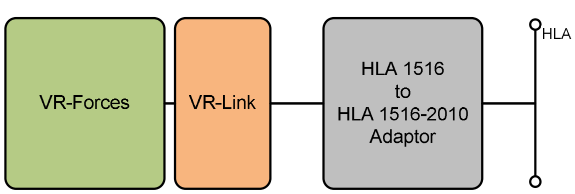

Figure 5. Integration of VR-Forces into the pluggable brains approach.

We expect VT MAK to deliver native HLA 1516-2010 support for VR-Link later this year. At that time, the adaptor can simply be discarded.

TASTE

TNO has developed the TActical sensor network TEst bed (TASTE) (van Dorp, Bekman, & Sandbrink, 2008). TASTE is a simulation platform that can be utilized to simulate novel sensors, as shown in Figure 6. TASTE started out as a software tool for specifying and deploying unattended ground sensors (UGS) in a composition which the commander assumes will suit his needs the best. Within TASTE, different sensor types such as acoustic, magnetic, seismic, radar and IR imaging sensors can be deployed virtually and their individual and combined performances analyzed.

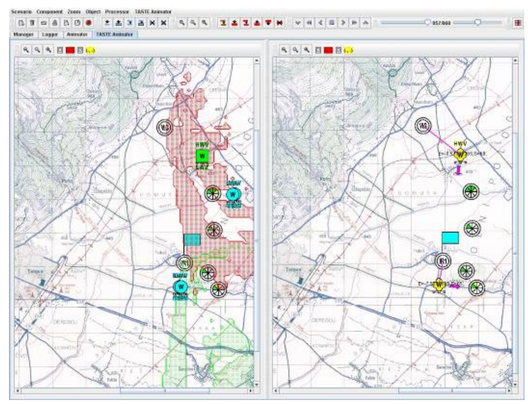

Figure 6. TASTE showing part of a sensor suite simulation.

For the pluggable brains environment, TASTE discloses its capability of providing high-quality observed truth data that can be used by for example intelligent agents and user interfaces. On the other hand, as sensors have often been developed and evaluated in isolation, integration of TASTE in the pluggable brains approach creates a powerful platform to investigate sensor networks in a broader setup.

HLA coupling of TASTE is very straightforward; we used Java generated-code for producing compliant data.

JADEX

JADEX is an open source agent based AI framework that can be used to construct complex behavior models. JADEX is based on the Beliefs-Desires-Intention (BDI) paradigm. We have successfully used BDI and JADEX for several experimental training solutions; for example CARIM (van den Bosch, Harbers, Heuvelink, & van Doesburg, 2009).

The intelligent, autonomous agents are built on top of the JADEX middleware, which provides a message mechanism for the communication between the agents. In this sense, JADEX is comparable to HLA, but we want to take benefit from both the reasoning engine of JADEX and the simulation interoperability of HLA. A special agent, the HLA Agent, lives both in the JADEX and the HLA world and takes care for the integration of JADEX into the pluggable brains environment, as shown in Figure 7.

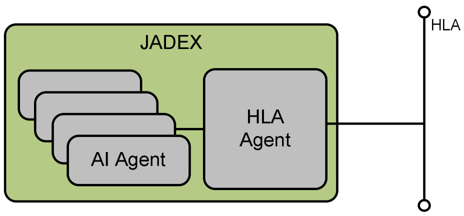

Figure 7. The JADEX HLA Agent integrates JADEX into the pluggable brains environment.

Man-Machine collaboration HMI-tool

Experimenting with collaboration in man-machine teams often means (re)considering the workload division between man and machine. A very effective research tool is the ability to change such workload divisions across experimental conditions. However, simply changing the workload division means that the user will have a different interaction with the machine. When the workload division changes, the interface should adapt itself to the new type of interaction that results. So there are two main requirements; 1) the workload division should be configurable (most likely by the researcher), and 2) the user interface must adapt to a new workload division.

For the collaboration, a three stage user interface has been developed. The simplest first stage collaboration interface is a simple pie chart that shows the amount of support that the operator receives from the system. The second stage collaboration interface shows the tasks that need to be performed by the operator and the available support for those tasks. The third stage collaboration interface, shown in Figure 8, is a fully fledged virtual assistant that can notify and explain the work division options. The virtual assistant is voice controlled and the operator can speak his work division desires and the assistant will execute what is ordered.

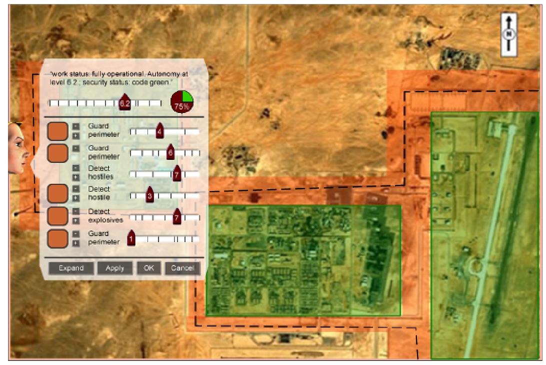

Figure 8. Screenshot of the third stage collaboration interface, showing a map overview as well as assistant options (left).

To make this HMI HLA compliant we developed a template HLA connector in C# and .NET, as shown in Figure 9. This template allowed us to quickly integrate these three different HMI designs into the pluggable brains environment.

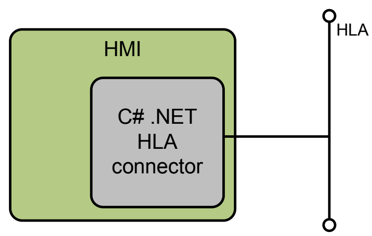

Figure 9. The C# .NET connector integrates the HMI tool into the pluggable brains environment.

Sesame Repository

Semantic interoperability is an issue that is getting more attention because components from different domains are being federated more often, each relying on their own terminology. TNO is developing a shared data pool that includes semantic services to aid in semantic interoperability among components.

In order to ensure that the information exchange between different components goes well, there is a language needed that all can understand. We are using ontologies to achieve this. Essentially, an ontology is a contract between at least two components prototypes such that the information they share with each other is understood. To increase flexibility in the federation it is possible that several pairs of systems with different ontologies share information.

In order to enable components to effectively use ontology enriched data, we integrated a queryable data store service in the pluggable brains environment. The data store service is build around an Apache Sesame RDF repository, and provides semantic services and a Java generated HLA connection, as shown in Figure 10.

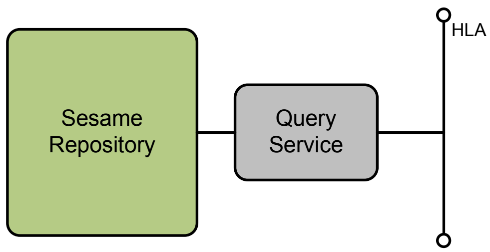

Figure 10. Integration of the Sesame RDF repository into the pluggable brains environment.

Use-Cases

In this section, we will present the two use-cases in which our CD&E environment was used. Results and findings will be presented per use-case as well.

2010 M4E Experiment

In the 2010 Man-Man-Machine-Machine Etiquette (M4E) project, we have developed a pluggable brains environment to test advanced man-machine collaboration with two levels of support for the user in high and low workload conditions in a compound protection scenario (van Doesburg, Heuvelink, Langeslag, & de Kraker, 2010). The setup of the pluggable brains environment for this experiment, showing the components that were used, is depicted in Figure 11.

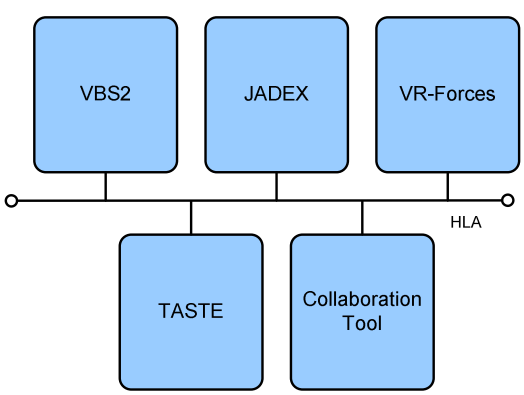

Figure 11. The pluggable brains environment of the 2010 M4E Experiment.

The scenario concerned a crowd near the fence of a compound. A small number of people behave suspiciously in the sense that they enter the forbidden area. These people are detected by sensors. A threat detector support system (agent) signals the suspicious behavior and informs a command support system (agent) and the human operator (via a HMI). The human operator, supported by the command agent, takes action and, also via the HMI, commands a small quick reaction force (QRF) to arrest the suspicious people. The QRF is simulated and its behavior is controlled by a behavior simulation (agent).

In this case, two levels of operator support are compared. We offered a low level of support in which the workload of protecting the compound rests mainly with the operator. Alternatively, in the condition of high level support, the operator was supported in threat detection and resource assignment.

Even with a relative simple case as the compound scenario case, the amount and types of data flowing across the man-machine network becomes complex quickly. This underscores the helpfulness of developer tools in creating such networks.

With the aid of the tools described in the previous section we were able to integrate these (mainly legacy and COTS tools) in a period of approximately 3 man-months. With all the specific demands of this application and the tailoring to its needs this is a remarkable result. During development of this environment, HLA 1.3 was used and not HLA evolved; yet, we expect that HLA evolved allows us to further compartmentalize functionality and will in the future help shorten integration time even further.

Urban Short Range Interaction

Urban Short Range Interaction (USRI) is an ongoing research project at TNO that investigates novel techniques for enriching interaction in an integrated Live, Virtual, and Constructive (LVC) environment. The current research is aiming to enhance a 'Live' training shoot-house with virtual target representation (Muller, van den Bosch, & Kerbusch, 2011), for which techniques as artificial intelligence, motion tracking (Microsoft Kinect) and voice recognition are brought together. The virtual target is projected on a wall of the shoot-house using VBS2, as shown in Figure 12.

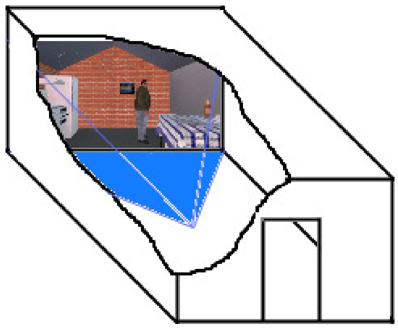

Figure 12. Global layout of the USRI shoot-house. VBS2 is used to visualize the virtual target.

Within this short range, interaction quickly becomes more social and cultural of nature, instead of only physical. In order to be able to evaluate behavioral models that bring these aspects into the virtual targets in the shoot-house, we used the pluggable brains environment, as depicted in Figure 13. The target is visualized using VBS2; however, the intelligent behavior of the target is represented in a JADEX agent. This agent uses social and cultural models for both interpreting the actions of the trainee and deciding how to respond.

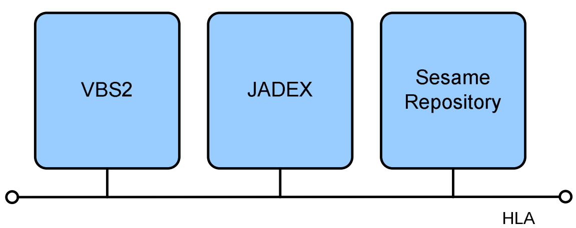

Figure 13. The pluggable brains environment of the USRI project. The Sesame Repository was added in a later experiment.

Using the pluggable brains approach allows us to separately develop the required sensor techniques and iteratively upgrade the behavioral, social, and cultural components of the virtual target representations. In later stages of the project it will be easier to reuse components (e.g. cultural models) or replace components (e.g. VBS2 visualization).

The USRI project is the first to reuse components (VBS2, JADEX) from earlier experiments. Although a transition was made from an older HLA version to HLA Evolved, integration of components took as little as two man-weeks. A repeated experiment, in which the Sesame Repository was added, took as little as one day to set up.

Since the USRI project is the first using the HLA Evolved pluggable brains, and thus the first using FOM modules, we didn't have a well thought strategy for designing reusable FOM modules at the beginning of the project. However, it quickly became clear that designing FOMs can be a challenging task; and time consuming if it needs to be done over and over again, because all components that use a particular FOM module need to be updated if the FOM module is changed.

DISCUSSION & CONCLUSIONS

The experience gained during the two use-cases described forms the evidence for the pluggable brains approach. In the case of the M4E experiment we were able to federate simulations, HMI and agents quite flexibly within a rather short timeframe. This use case is oriented to research and development, as such the approach was very effective in enabling the researchers to experiment with various man-machine collaboration concepts. The second, USRI, use-case proved that we are able to efficiently reuse components from previous experiments and integrate more components into the existing environment. We also upgraded the HLA infrastructure to its newest version and gained insight in the importance of FOM module design to facilitate better reuse.

The current CD&E environment is an amalgam of quite diverse tools. Ranging from legacy systems to COTS tools to quickly developed prototypes. It consists of systems developed in various languages and with varying degrees of distribution (from monolithic to completely distributed). With the pluggable approach and with the support for HLA in the form of code generators, templates and JADEX HLA-Agents we are able to quickly exploit this toolset to its full extent.

We hope to have sketched some of the advantages that were observed in our initial experience with the pluggable brains approach. It creates focus on the interfaces between systems. This approach requires a developer to think carefully what functionality to place where in the federation and furthermore what data-exchange agreements can be reused and which need to be compartmentalized in a separate FOM module. By creating this focus our approach yields the benefits of cost-effectiveness, flexibility and maintainability as argued. However, we have also experienced the consequences of this approach of which new adopters should be aware. With the increase of the number of federates participating in a system there is a growing risk in data-exchange agreements getting out of sync or overlapping. Therefore, we advise that federation developers appoint a data-exchange supervisor that is responsible for keeping the data-exchange agreements up to date and making sure that any changes are distributed to all the separate federates.

With this paper we aim to contribute to the experience reported on using distributed standards to create federated systems; in our case simulations and brains. Furthermore, we propose to standardize FOM modules that are generic so that they can be reused and shared within the M&S community.

REFERENCES

BinSubaih, A., Maddock, S.C., & Romano, D. (2004). An Architecture For Domain-Independent Collaborative Virtual Environments. *Proceedings GAME-ON 2004, 5th Annual Conference on Simulation and AI in computer Games*.

Boer, C.A. (2005). *Distributed simulation in industry*. Rotterdam: Erasmus University.

van den Bosch, K., Harbers, M., Heuvelink, A., & van Doesburg, W. (2009). Intelligent agents for training on-board fire fighting. Digital Human Modeling, 463–472.

Cap, M., Heuvelink, A., van den Bosch, K., & van Doesburg, W. (2011). Using Agent Technology to Build a Real-World Training Application. *Agents for Games and Simulations II*, 132–147.

Cohen, M.A., Ritter, F.E., & Haynes, S.R. (2010). Applying software engineering to agent development. *AI Magazine*, 31(2), 25.

van Doesburg, W.A., Heuvelink, A., Langeslag, P., & de Kraker, K.J. (2010). Man-Man-Machine-Machine Etiquette: Towards an Environment to Investigate Behavioral Rules for Dynamic Man-Machine Teams. Presented at the RTO-MP-HFM-202 - Human Modelling for Military Application, NATO Research and Technology Organisation.

van Dorp, I.P., Bekman, H.H., & Sandbrink, R.D.J. (2008). TActical sensor network TEst bed (TASTE). In Proceedings of SPIE (Vol. 7112, p. 711206).

Evertsz, R., Pedrotti, M., Busetta, P., Acar, H., & Ritter, F.E. (2009). Populating VBS2 with Realistic Virtual Actors. *Proceedings of the 18th Conference on Behavior Representation in Modeling and Simulation*, 1-8.

IEEE (2010). 1516-2010 IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. www.ieee.org.

Heinze, C., Papasimeon, M., Goss, S., Cross, M., & Connell, R. (2008). Simulating fighter pilots. *Defence Industry Applications of Autonomous Agents and Multi-Agent Systems*, 113–130.

Klein, G.A. (1999). *Sources of power: How people make decisions.* Cambridge, MA: The MIT Press.

de Kraker, K.J., Kerbusch, P., & Borgers, E. (2009). Re-usable behavior specifications for tactical doctrine. *Proceedings of the 18th Conference on Behavior Representation in Modeling and Simulation*, 15-22.

Lebiere, C., Stocco, A., Reitter, D., & Juvina, I. (2010). Scaling Up High-Fidelity Cognitive Modeling to Real-World Applications. Presented at the *RTO-MP-HFM-202 - Human Modelling for Military Application,* NATO Research and Technology Organization.

Lee, J.E.R., & Nass, C.I. (2010). Trust in Computers: The Computers-Are-Social-Actors (CASA). *Trust and Technology in a Ubiquitous Modern Environment: Theoretical and Methodological Perspectives, 1.*

Lees, M., Logan, B., & Theodoropoulos, G.K. (2006). Agents, games and HLA. *Simulation Modelling Practice and Theory, 14* (6), 752-767.

Levesque, J., Cazzolato, F., & Martonosi, J. (2009). CAMX: Civilian Activity Modelling for eXercises and eXperimentation. Technical Memorandum DRDC-CORA-TM-2009-065.

Möller, B., Gustavson, P., Lutz, B., & Löfstrand, B., (2007). Making Your BOMs and FOM Modules Play Together. *Fall Simulation Interoperability Workshop.*

Möller, B., Morse, K.L., Lightner, M., Little, R., & Lutz, R. (2008). HLA Evolved – A Summary of Major Technical Improvements. *Fall Simulation Interoperability Workshop*.

Muller, T., van den Bosch, K., & Kerbusch, P. (2011). LVC Training in Urban Operation Skills, *Spring Simulation Interoperability Workshop.*

Zacharias, G., MacMillan, J., & van Hemel, S. B. (Eds.). (2008). *Behavioral modeling and simulation: From individuals to societies*. Washington, DC: National Academy Press.