Smallpox Over San Diego: Joint Real-Time Federations of Distributed Simulations and Simulation Users under a Common Scenario

John M. Linebarger, Michael E. Goldsby, Daniel Fellig, Marilyn F. Hawley, Patrick C. Moore, and Timothy J. Sa
Sandia National Laboratories
{jmlineb,megolds,dfellig,hawley,pcmoore,tjsa}@sandia.gov

Abstract

A joint project between the California and New Mexico branches of Sandia National Laboratories has demonstrated the formation of joint real-time federations of both distributed simulations and distributed simulation users under a common scenario. Two software integration frameworks were used to achieve the real-time federations. The IDSim framework, developed by Georgia Tech University and Sandia National Laboratories, was used to create the real-time federation of distributed simulations, in this case the BioDAC WMD simulation and the N-ABLE™ agent-based microeconomic simulation (more properly, because of the impact of hurricanes Katrina and Rita, an N-ABLE™ emulator). The GroupMeld™ multimedia synchronous collaboration framework, developed by Sandia, was used to create the real-time federation of simulation users and simulation analysis communities. The common scenario was the release of smallpox over San Diego, California, and the operating hypothesis was that the economy itself dampens the spread of a pathogen. In addition, a small pilot experiment using the joint federations allowed a greater range of crisis management options to be performed and evaluated than would have been possible without the use of the integration frameworks.

1. Introduction

Linking distributed simulations together in real-time offers numerous advantages over separate or serialized execution, and several frameworks exist to create federations of distributed simulations. One of the earliest was the High Level Architecture (HLA), which was sponsored by the US Department of Defense (see [1], *inter alia*). Some recent work in this area has focused on linking simulations with different processing models together; see [2] for an example of a real-time link between a system dynamics simulation and a discrete event simulation. However, these integrating middleware architectures have invariably focused on real-time linkage between simulation codes, not between simulation users or the simulation analysis communities associated with the linked simulation codes.

The purpose of a year-long Lab-Directed Research and Development (LDRD) project at Sandia National Laboratories was to create and demonstrate integrating software architectures that enabled real-time federations of both distributed simulations and distributed simulation analysis communities. A unifying scenario (a smallpox release in the San Diego area) was used as the demonstration vehicle. The first simulation was BioDAC [3], a crisis management simulation for biological weapons of mass destruction, which ran in Sandia California. The second was N-ABLE™ [4], an agent-based microeconomic simulation, which ran in Sandia New Mexico. The IDSim distributed simulation framework [5] was used to create a unified federation of these two simulations. The two simulation analysis communities were those associated with the BioDAC and N-ABLE™ simulations, respectively. The GroupMeld™ synchronous multimedia collaboration framework ([6] and [7]) was used to create a collaborative federation of these two simulation analysis communities. The simulations were initially developed through the Weapons of Mass Destruction Decision Analysis Center (WMD-DAC), sited at Sandia California, and the National Infrastructure Simulation & Analysis Center (NISAC), sited at Sandia New Mexico.

The two federations are schematically depicted in Figure 1. Such a combination of joint real-time federations—simulations and simulation analysis communities—is extremely rare, at least in the non-military, crisis management response arena. The most closely related work of which we are aware is a series of exercises staged by the U.S. Joint Forces Command, beginning with the Millennium Challenge in 2002 [8] and extending most recently to the Multinational Experiment 4 in 2006 [9]. However, their collaboration environment (InfoWorkSpace from Ezenia [10]) is a dedicated collaboration application instead of a library of collaboration services programmatically embedded in a simulation application.

21st International Workshop on
Principles of Advanced and Distributed Simulation (PADS'07)
0-7695-2898-8/07 $20.00 © 2007 IEEE

In the pages that follow, the two simulation codes (BioDAC and N-ABLE™) will be briefly described; the integrating architectures that create the federations (IDSim and GroupMeld™) will be detailed; simulation integration issues will be elaborated; the results of a small pilot experiment using the unified scenario will be presented; and future work will be outlined.

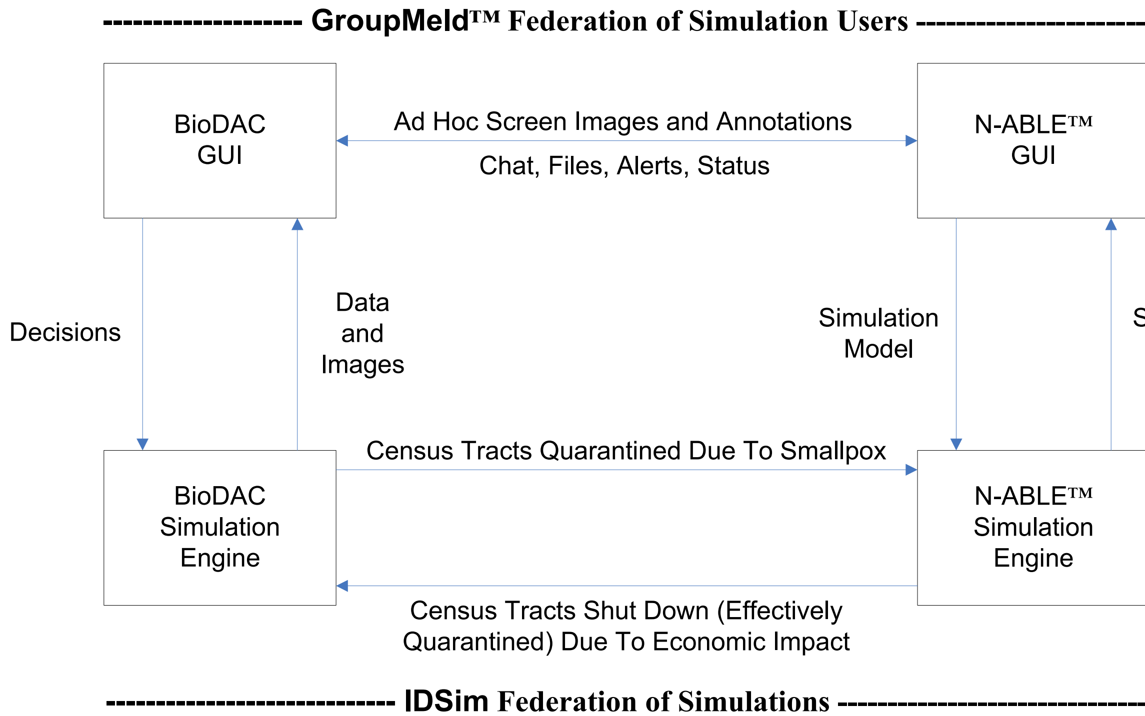

Figure 1. Two Federations under one unifying scenario

2. BioDAC

BioDAC (an abbreviation for the Weapons of Mass Destruction Decision Analysis Center [WMD-DAC] Biological Defense Application) is a component of the WMD-DAC suite of simulation components. BioDAC is used to simulate the release of biological agents and evaluate the efficacy of a large set of response strategies. Three primary roles exist in a BioDAC simulation—Public Health Official (PHO), Navy Official (NO), and Analyst. Figure 2 below portrays the plume associated with the release of a biological warfare agent into the atmosphere.

The graphical user interface of a BioDAC screen is dynamically created depending on the role of the user in the simulation. The simulation consists of multiple components, such as the warfare agent dispersion model (provided by the Defense Threat Reduction Agency Hazard Prediction and Assessment Capability [DTRA HPAC]), the population movement model, the disease model, and the hospital model. Figure 3 below shows the distribution of the disease based on the infection of a population from the plume dispersion in Figure 2.

BioDAC is designed for use in simulator-based exercises involving officials from various interested military and governmental agencies. There are plans to extend it to make it capable for use as a decision support system during an actual event (for instance, in estimating the outcomes of various different hypothetical scenarios or decision paths.) BioDAC provides views of the simulated events that correspond to the information the PHO and NO would have access to in the course of an actual attack, including sensor inputs, lab test results and various indicators from the health surveillance system (such as numbers of patients with various symptoms reported by hospitals and emergency rooms), and sales of particular types of remedies by pharmacies. The role of Analyst provides a "God's eye view" that is useful for viewing ground truth.

The decisions that the PHO or the corresponding Navy official can make include the ordering of lab tests, the collection of additional samples for testing, the distribution of prophylaxis, and the closure of selected parts of the city by evacuation or sheltering in place. “Evacuate” means “no one can stay,” while “shelter-in-place” means “no one can leave.” The timing of the various actions the officials may take can have a crucial

21st International Workshop on
Principles of Advanced and Distributed Simulation (PADS'07)
0-7695-2898-8/07 $20.00 © 2007 IEEE

effect on the outcome in terms of the number infected and the number of deaths.

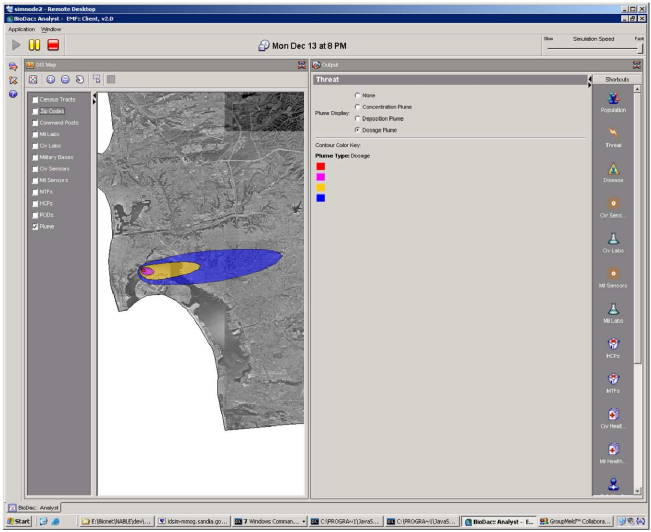

Figure 2. BioDAC plume display

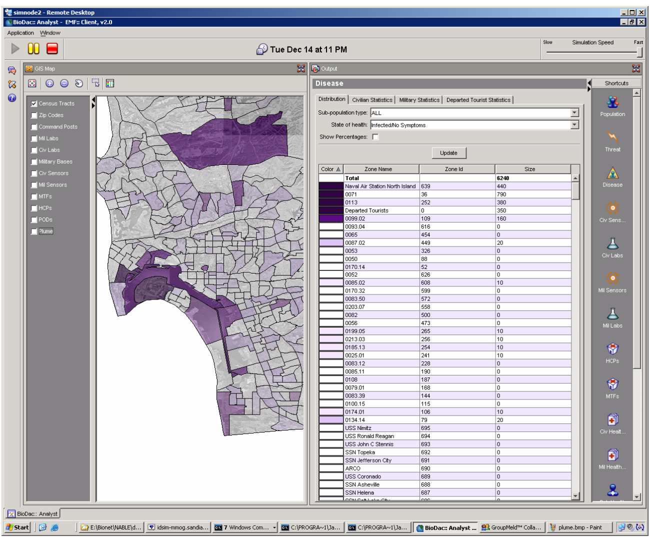

Figure 3. BioDAC disease distribution display

3. N-ABLE™

The NISAC Agent-Based Laboratory for Economics (N-ABLE™) is an agent-based economic modeling and simulation package. It consists of two components—an agent-based simulation engine that can execute either serially or in parallel, and a rich graphical user interface (GUI) that enables collaborative analysis of the simulation results. N-ABLE™ was the first software tool to use the GroupMeld™ collaboration framework to provide collaboration services embedded inside of the application. Figure 4 shows a screenshot of N-ABLE™ with several collaboration services visible—group awareness, public chat, and public screenboard with annotation capability.

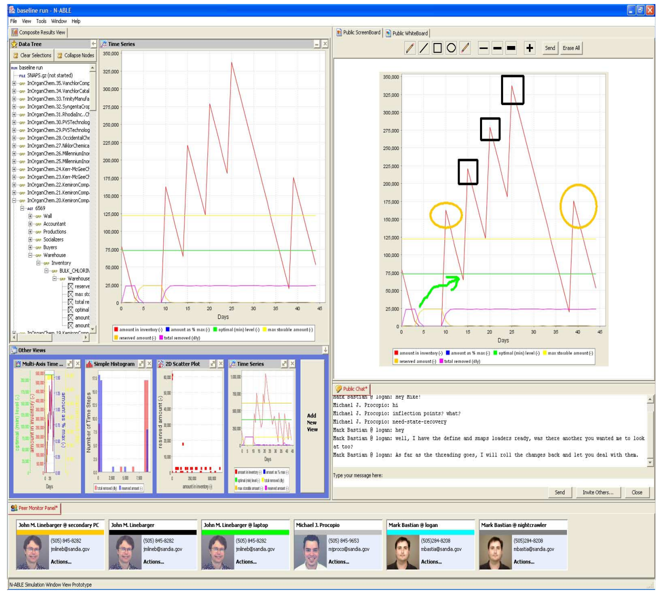

Figure 4. GroupMeld™ collaboration panels embedded in N-ABLE™

A canonical simulation analysis cycle using N-ABLE™ iterates through four stages—modeling, simulation, analysis, and software development. First the simulation model is created using XML (eXtensible Markup Language). Next, the model is submitted to the simulation engine. Then the results of the simulation are validated via a sampling procedure—key outputs of various agent types for representative firms at each level of the supply chain are displayed graphically and shared with others using the screenboard collaboration capability. This validation process is called a "deep dive." Finally, if anomalous results are discovered, a review of the simulation software code is performed, which often results in code changes. This triggers a new iteration of the cycle, in which the simulation model is resubmitted to the simulation engine to run against the updated code, and the results are validated again.

Note that because of the impact of hurricanes Katrina and Rita, resources were diverted at the last minute such that a model of the San Diego economy was not able to be created and validated in time for the final experiment. As a result, an N-ABLE™ emulator was used instead of the full N-ABLE simulation engine (see section 7 below).

4. IDSim

The Interoperable Distributed Simulation (IDSim) framework provides the means to federate two or more autonomous simulators. The motivation for the development of IDSim was the mandate from the

National Infrastructure Simulation and Analysis Center to combine "best of breed" infrastructure models. This translates into the need to integrate disparate simulations, distributed both geographically and organizationally, across a WAN (wide area network). To make this feasible, IDSim was developed with the following design goals in mind: Ease of integration; interoperability, both between platforms and languages; low usage of client resources; secure network communications; and the use of standard, open technologies.

IDSim is built on top of the reference implementation of the Open Grid Services Infrastructure (OGSI), Globus 3.2.1. This implementation uses Web Services technologies as a means of interoperating among different software applications running on heterogeneous platforms over a network, most often a WAN (wide area network). IDSim inherits much of the core functionality of Globus, including its Factory, ServiceGroupRegistration and ServiceGroupEntry services, among others. These services provide the underlying basis for IDSim's FederateFactory, FederationFactory, Federation, Federate, and FederateEntry services. The factory services create instances of Federate, Federation and FederateEntry services. Each Federate service provides the interface for one federation-participating simulator. The Federation services manage a group of federation-member federate services, which support the joining and leaving of the federation. The FederateEntry service provides an interface for each Federate service that has joined a federation. It acts as an entry point into the Federation service.

The "events" or messages that are sent between simulators are specified using XML and XML schema. These technologies provide a language-independent means of defining the data structure to be shared within the federation. Once the data structure is defined, it is compiled into any language that Web Services tools support.

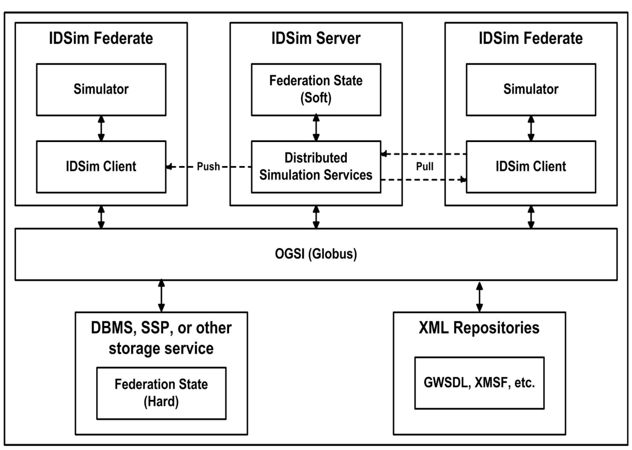

Figure 5. IDSim framework architecture

Figure 5 shows the different architectural components of IDSim and their relationship to each other. The XML repositories provide the definitions to create the IDSim services and the data structures that are used in interactions among federates. The Grid Web Services Description Language (GWSDL) is compiled first to Web Services Description Language (WSDL), and finally into both client and server language stubs. Soft state refers to state that is kept during the federation run, and hard state refers to state that is kept after the run has finished. The soft state associated with each of the services can be obtained synchronously through queries (pull model) or asynchronously through callbacks (push model). Note that if the federation has components that cross firewalls, only the pull model can be used; network callbacks are rarely allowed through firewalls. At Georgia Tech, the IDSim project has recently morphed into a prototype system called Aurora [11], which is focused on running large distributed simulations in a grid computing environment.

5. GroupMeld™

The National Infrastructure Simulation and Analysis Center (NISAC), a program under the United States Department of Homeland Security's Information Analysis and Infrastructure Protection (IAIP) directorate, provides advanced modeling and simulation capabilities for the analysis of critical infrastructures, their interdependencies, vulnerabilities, and complexities. These capabilities help improve the robustness of critical infrastructures of the United States by aiding decision makers in the areas of policy analysis, investment and mitigation planning, education and training, and near real-time assistance to crisis response mobilizations. NISAC and related programs are frequently called upon for Fast Analysis and Simulation Team (FAST) exercises to assess the impact of a potential event on critical infrastructures. The primary metrics for this high-pressure, time-constrained collaboration (which can be characterized as "collaboration in a crisis") are time to solution and quality of solution. A primary time consumer is the information exchange required to establish a common mental model (also called a "common analysis picture") of the problem(s) and solution(s) between all participants in the exercise.

Inspired by observations of several FAST exercises (although it has not been used in an actual FAST exercise to date), the GroupMeld™ software framework for synchronous multimedia collaboration was developed. The goal of this framework is to facilitate real-time collaborative interaction both textually and graphically, in order to allow geographically-distributed analysis teams to integrate multiple perspectives and quickly converge on a shared view of the problem(s) and potential solution(s).

The collaboration functions provided by GroupMeld™ include:

- Pictorial awareness of other members of the virtual team, with visual status change indicators
- Real-time chat and file transfer
- Shared screen images with collaborative annotation capability (a.k.a. "screenboard") as well as a shared whiteboard
- Audible paging capability (to get someone's attention in case they are working on something else).

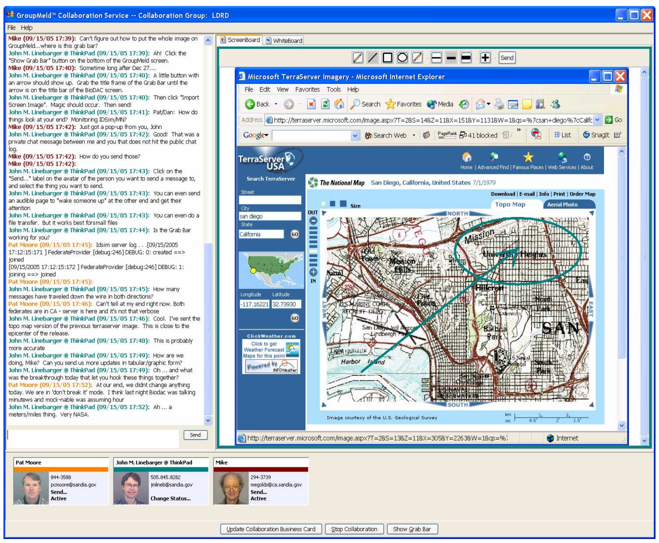

Figure 6. Standalone GroupMeld™ collaboration application

The collaboration scope of each capability is chosen from three levels, which can co-exist simultaneously: Full group ("public" collaboration); subgroup ("restricted" collaboration); and person-to-person ("private" collaboration). Three usage models for GroupMeld™ have been observed:

- Programmatically embedded inside of a simulation application (*e.g.*, as used by N-ABLE™ in Figure 4 above)
- Standalone GroupMeld™ collaboration application (see Figure 6 above)
- Hybrid (invoked as a separate window inside the Java Virtual Machine of another simulation application). This approach allows panels from the simulation application to be dragged and dropped directly onto the screenboard of the GroupMeld™ application.

GroupMeld™ was developed using the Java programming language, and has been deployed as a programmable collaboration library with an application programming interface (API). The library enables collaboration *through* a particular software application, thus forming an application-centered collaboration community. RMI over IIOP (Remote Method Invocation over the Internet Inter-ORB Protocol) is used as the distributed communication mechanism. The use of Java provides cross-platform portability—GroupMeld™ currently runs on Windows, Macintosh, and Linux computers. The framework is deployed as a set of Java packages in a single JAR (Java ARchive) file. The Java drag-and-drop API is used to drag a simulation graph or OpenGL (Open Graphics Language) image onto the screenboard panel. Each collaborator is both a client of and a server to all the other collaborators in the session, so the network topology is truly peer-to-peer. The communication functions are multithreaded, so reader-writer locks are used to protect shared data structures. An instance of the CORBA (Common Object Request Broker Architecture) Naming Service is used to keep track of all the participants in the collaborative session as well as their current subgroup structure. Subgroups can be nested to an arbitrary depth. A limitation of this architecture is that all computers must be on the same network or security domain; collaboration transactions cannot currently traverse a firewall.

6. Integration Details

We wished to synchronize the BioDAC and IDSim federations so that the two would function as if part of one federation, with causality relationships maintained among their events. We could not carry out the synchronization at the Run Time Infrastructure (RTI) level, because the RTI used with BioDAC is a commercial product (pRTI™ from Pitch AB) and we do not have access to its source code. Therefore we carried out the synchronization at the federate level.

We introduced an extra federate, a so-called *gateway* federate, into each federation and allowed the two gateways to communicate with each other via a direct TCP link. It would probably have been possible to use a single gateway process that was a federate in both federations, but it appeared less straightforward in implementation, so for the sake of our schedule we took the more direct approach.

There are known problems with connecting federations at the federate level [12]. Our connection scheme sidesteps the problems by connecting the two gateways with a TCP/IP link, as explained below.

It was necessary to propagate interactions of certain message classes from the originating federation to the other federation. In fact, there were only two such message classes, one going each direction, and we handled them as special cases. If we had wished to implement a more general scheme, we might have marked such message interaction classes by placing them in a separate subclass, say InteractionRoot.Interfederation.

The two federations, BioDAC and IDSIM, are both conservatively synchronized [13]. When a federate in one federation sends an interaction at time t1 with arrival time t2, t1 < t2, we must ensure that the simulation time of a federate that receives it in the other federation does not exceed t2. Thus each federation must be able to constrain the simulation time advance of the other. In addition, for the sake of performance, we wish to allow as great a lookahead value for each federate as possible, in order to provide maximum concurrency.

What lookahead guarantees can the application make for interactions that originate in one federation and reach a destination in the other federation? Suppose that the originating federation can make the guarantee that the difference between the arrival time and the send time (t2 and t1 above) will be no less than L, even if the receiving federate is in the other federation. We call L the *interfederation lookahead*.

In order for the interaction to reach the other federation, it must first be sent, using the usual RTI mechanisms, in the originating federation, where it will be received by that federation's gateway federate. Then it must pass through the TCP link to the gateway federate in the destination federation. Finally, that gateway must send it (via the usual RTI mechanisms) into the destination federation, where the destination federate receives it. Thus there are three contributions to the interfederation lookahead: the lookahead L1 of the originating federate, the lookahead L2 of the gateway federate in the destination federation, and the amount by which the simulation time in the destination gateway can be ahead of the simulation time in the source gateway.

We follow a simple scheme to bound the difference in the simulation time between the two federations. Each gateway federate is time-regulating, so it can constrain the advance of simulation time in its own federation. A gateway requests an advance in simulation time only when it believes its own simulation time is not greater than that of the other gateway. The advance that it requests is its estimate of the other gateway's simulation time plus S, where S is a parameter.

In order to make this scheme work, the gateway must have information about the simulation time of the other gateway. To provide this information, every time a gateway federate receives a time advance grant from its federation's RTI, it sends a notification of the advance across the TCP link to the other gateway. Thus two kinds of messages go across the TCP link, interactions traveling from one federation to the other and time advance grant notifications. The simulation time in a time advance grant notification provides to the receiving gateway a lower bound on the simulation time of the other gateway. Thus the simulation times of the two gateways can differ by at most S.

Now we are able to state that the interfederation lookahead L must be at least as great as L1 + S + L2. For simplicity, let L1 = S = L2. Then the lookahead of a gateway federate may be no greater than L/3.

The event loops in the two gateway federates are simple and similar. The only events are the arrival of an interaction originating in the other federation (to be sent into the local federation), the arrival of an interaction originating in the local federation (to be sent across the TCP link to the other gateway), the arrival of a time advance grant notification from the other federation (used to update the gateway's estimate of the other gateway's simulation time), and a time advance grant from the local RTI. Note that the scheme outlined above will work to couple any two simulators that have the concepts of lookahead, time advance requests, and time advance grants.

BioDAC and IDSim operate in this application at similar resolution and time scales, so only minor data format and units conversions are necessary when an interaction crosses federation boundaries. All such conversions were handled in the gateway federate on the BioDAC side.

There are two sources of inefficiency in the synchronization scheme. The constraint on the gateway lookahead value serves to limit concurrency within each federation, and the fact that each gateway can delay the time advance requests of the other gateway limits concurrency between the federations. The effect of these factors on performance depends entirely on the characteristics of the application (the actual lookahead values and the frequency of synchronization relative to computation). For this particular application, the effects are minimal. The run times of BioDAC with and without IDSim differ by less than three percent on a typical scenario.

7. Results of Small Pilot Experiment

A small pilot experiment was performed in order to investigate the use of these integration frameworks in exploring crisis management strategies during a bioterrorism attack. Three such strategies were explored—the use of BioDAC alone; the integration of BioDAC and N-ABLE™ in order to determine which census tracts have been economically disrupted by quarantine actions taken by a PHO, which are subsequently left alone; and the integration of BioDAC and N-ABLE™ in which a PHO immediately shelters in place census tracts that are determined by N-ABLE™ to be economically disrupted. The operating hypothesis for the experiment was that the economy itself dampens the spread of a pathogen, primarily by impacting commuter population movements, and informs the choice of crisis management responses. The BioDAC simulation and community of simulation users were located in Livermore, California, while the N-ABLE™ emulator and community of N-ABLE™ users were in Albuquerque, New Mexico. Since the incubation period for smallpox is 7 to 17 days, the scenario of a smallpox release over San Diego ran for 33 simulation days, which took approximately 2.5 hours of wall clock time. BioDAC displays the list and a graphical representation of economically disrupted census tracts as soon as they are received from N-ABLE™. Census tracts that were sheltered in place by the PHO are outlined in magenta, and census tracts that were determined by N-ABLE™ to be economically disrupted, because they were dependent on labor in the quarantined census tracts, are outlined in blue.

Because of the impact of hurricanes Katrina and Rita on staff availability, a model of the San Diego economy could not be created in time for the experiment, so the N-ABLE™ simulation engine proper was not used. Instead, a limited-functionality N-ABLE™ emulator was developed that took a list of census tracts that were quarantined by a PHO and randomly selected other census tracts that were considered likely to be economically disrupted by the quarantine due to labor dependencies.

Considerable expertise is required to play the role of a PHO in a BioDAC simulation. As a result, it was challenging to perform simulation runs that were consistent enough to allow valid comparison of the three factor levels of the experiment. Only one full experiment cycle (three runs), by an experienced BioDAC user, met that criteria. The results are displayed in Table 1. Though we were gratified that the outcome of one of the crisis management strategies was a lower number of deaths, it must be stressed that the results are not representative for at least two reasons: The extremely small sample size, and the use of an emulator of the N-ABLE™ simulation engine instead of the real N-ABLE™ simulation running against a full agent-based model of the San Diego economy. However, the significance of the pilot experiment is that the integration frameworks allow a far greater range of crisis management options to be explored and evaluated than would be possible without the use of the frameworks.

Several benefits of linking both simulations and simulation users together under a common scenario were observed not only during the pilot experiments but also in the several milestone demonstrations leading up to the pilot experiments. As the scenario progressed, chat messages and screen images were exchanged between simulation user communities to keep each community apprised of the status of the other simulation. The screen images were annotated to highlight significant details, and options to improve the crisis management response and the configuration of the simulations were discussed and evaluated. The collaboration environment seemed to prove as useful in training the simulation users and tuning the simulation scenario as it did for monitoring and controlling the scenario. In short, it provided an additional source of input for the computational steering and human-in-the-loop mechanisms of the simulation codes.

Response Strategy	**Population Category**	**Percent change from running BioDAC alone**
Determine which census tracts are economically disrupted	Avg. Infected/No Symptoms	0
	Max. Infected/No Symptoms	0
	Avg. Mildly Infected	0
	Max. Mildly Infected	0
	Avg. Severely Infected	0
	Max. Severely Infected	0
	Recovered	0
	Dead	0
	Immune	0
Determine which census tracts are economically disrupted and immediately shelter them in place	Avg. Infected/No Symptoms	-3
	Max. Infected/No Symptoms	-4
	Avg. Mildly Infected	0
	Max. Mildly Infected	2
	Avg. Severely Infected	2
	Max. Severely Infected	1
	Recovered	-13
	Dead	-20
	Immune	0

Table 1. Results from small pilot experiment

Numerous benefits were also observed from the use of programmable collaboration capabilities instead of a standalone collaboration application. Because a context switch to another application was not required, better task focus was promoted, and the observed content of the collaboration was almost entirely "on track." Tight coupling with the application also enabled simulation results to be easily shared (*e.g.*, drag and drop of simulation results graphs to the collaboration screenboard for easy annotation), and allowed application-specific data structures to be communicated and decoded through the collaboration framework.

8. Future Work

The most obvious path forward for future work is to use the real N-ABLE™ simulation engine instead of an N-ABLE™ emulator. Since the operating hypothesis of the experiment scenario was that the economy itself can contribute to the dampening of a pathogen, using the real N-ABLE™ may lead to significant results that would be of interest not only to the crisis management community, but also to the economic simulation and distributed simulation communities. A secondary path forward would be to involve other simulations from other National Laboratories in another, larger, integrated scenario. And since three collaboration groups were involved, one subgroup each for the BioDAC and N-ABLE™ simulation users and a full group that linked the two user communities, a third area of research would be to analyze the timing and content of intragroup versus intergroup communication to see what patterns emerge.

9. Acknowledgements

Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

10. References

[1] Judith S. Dahmann, Richard M. Fujimoto, & Richard M. Weatherly. (1998). "The DoD High Level Architecture: An Update." *Proceedings of the 1998 Winter Simulation Conference*, 13-16 December, Washington, D.C., USA, 797-804.

[2] J. Venkateswaran, Young-Jun Son, & A. Jones. (2004). "Hierarchical Production Planning Using a Hybrid System Dynamic-Discrete Event Simulation Architecture." *Proceedings of the 2004 Winter Simulation Conference*, 5-8 Dec., Washington, D.C., 1094-1102.

[3] H. Ammerlahn, M. Hawley, & D. Manley. (2004). "Weapons of Mass Destruction Decision Analysis Center (WMD-DAC): Simulation-based Exercises and Analyses for WMD Detection and Response." *Proceedings of the 2004 Conference on Science and Technology for Chem-Bio Information Systems (CBIS)*, 18-21 October, Williamsburg, VA, USA.

[4] Eric D. Eidson & Mark A. Ehlen. (2005). "NISAC Agent-Based Laboratory for Economics (N-ABLE™): Overview of Agent and Simulation Architectures." *Sandia National Laboratories Technical Report SAND2005-0263*, Albuquerque, New Mexico.

[5] J. Brad Fitzgibbons, Richard M. Fujimoto, Daniel Fellig, Stephen D. Kleban, & Andrew J. Scholand. (2004). "IDSim: An Extensible Framework for Interoperable Distributed Simulation." *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*, 6-9 June, San Diego, California, USA, 532-539.

[6] John M. Linebarger, Andrew J. Scholand, Mark A. Ehlen, & Michael J. Procopio. (2005). "Benefits of Synchronous Collaboration Support for an Application-Centered Analysis Team Working on Complex Problems: A Case Study." *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work (Group'05)*, 6-9 November, Sanibel Island, FL, USA, 51-60.

[7] John M. Linebarger, Andrew J. Scholand, & Mark A. Ehlen. (2006). "Representations and Metaphors for the Structure of Synchronous Multimedia Collaboration within Task-Oriented, Time-Constrained Distributed Teams." *Proceedings of the 39th Hawaii International Conference on System Sciences (HICSS 39)*, January 4-7, Kauai, HI, USA, p. 58a.

[8] USJFCOM Millennium Challenge 02 (2002). http://www.jfcom.mil/about/experiments/mc02.htm.

[9] USJFCOM Multinational Experiment 4 (2006). http://www.jfcom.mil/about/experiments/mne4.htm.

[10] Eznia InfoWorkSpace 3.0 (2006.) https://www.ezenia.com/infoworkspace.asp.

[11] Alfred Park & Richard M. Fujimoto. (2006). "Aurora" An Approach to High Throughput Parallel Simulation." *Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation (PADS'06)*, 24-26 May, Singapore, 3-10.

[12] Juergen Dingel, David Garlan, & Craig Damon. (2002). "Bridging the HLA: Problems and Solutions". *Proceedings of the Sixth IEEE International Workshop on Distributed Simulation and Real Time Applications (DS-RT '02)*, 11-13 October, Fort Worth, TX, USA, 33-42.

[13] Richard M. Fujimoto. (2000). *Parallel and Distributed Simulation Systems.* New York: John Wiley & Sons.