Requirements for Standards Based Dynamic Interoperation of Critical Infrastructure Models

Bernard Zeigler

RTSYnc, ACIMS and GMU C4I Center

Zeigler@rtsync.com

Steven B. Hall

Lockheed Martin Space Systems Company

Advanced Technology Center

steve.hall@lmco.com

Abstract

Dynamic Interoperation of Critical Infrastructure Models is a grand challenge in Modeling, Simulation, and Analysis for Homeland Security. HYDRA is an environment developed by NISAC to support automatic coupling of disparate tools that were developed to stand alone in separate locations on different platforms, and implemented in a variety of languages. In this paper, we examine the HYDRA environment in relation to some minimal requirements for dynamic interoperation of critical infrastructure models. We then suggest approaches, based on Discrete Event System Specification to extend it to meet these requirements. Finally, we discuss the potential for a quantum leap in capability to respond to national-scale infrastructure disruption events.

1. Requirements for a Web-Standards Based Dynamic Interoperation Environment for Critical Infrastructure Modeling

The need for a state-of-the art modeling and simulation environment that will provide analysts with the ability to rapidly predict the 1) spatially qualifiable *direct* consequences of complex (multiple) disruptive events on critical infrastructure and resources, as well as 2) their *indirect* effects on interdependent systems including health, financial, power systems, fuel supply chains, transportation systems and emergency response capabilities [1]. Such an environment would support specific questions, e.g., given that a 10 kiloton improvised nuclear device has just been detonated in New York City, where and when would an adversary detonate a second such device to maximally extend the time it takes to recover? To help answer such questions the simulation environment would generate a *model composition plan* and identify which of the disruption, infrastructure and analysis models in its inventory are candidates for participation in the specific scenario instantiations of this search. Among the requirements needed to compose such models to support simulation experimentation are:

- Support for federation of models enabling not only exchange of data using Web standards but also meaningful interoperation via coordinated dynamic simulation interaction.
- Ontology support to frame specific questions in the domain of interest, discover relevant model components and rapidly compose them for federated execution.

In this paper we consider these requirements in more detail and propose an approach based on the HYDRA [2] environment developed by NISAC [3] and the Discrete Event System Specification (DEVS) theory, concepts, and Web-based simulation environments [4].

2. HYDRA Background

The HYDRA architecture [2] provides a web service oriented platform that can vastly increase the types of analyses that can be performed by supporting the interaction and interoperation of infrastructure models as simulation components. However, important capabilities are missing from this architecture that would enable it to serve as a state-of-the art modeling and simulation environment that will provide analysts with the ability to rapidly answer questions concerning infrastructure disruption.

First among the currently missing features, is the capability to compose federations of simulation components that interact in a truly dynamic manner such as facilitated by distributed simulation platforms such as HLA [5]. This requires support for federation of simulation models enabling both exchanges of appropriately specified data but also meaningful interoperation via temporally-coordinated dynamic state-based interaction. The current HYDRA architecture is implicitly designed as a serial work-flow process that exploits the basic functionality of web-service architectures which are message driven, loosely coupled and inherently stateless.

3. Extending HYDRA to Support For Federation of Dynamic Models

Sequential workflow is not a necessary restriction of a SOA foundation. By careful design, it is possible to employ the service-oriented architecture to realize the advantages of message-based, loosely coupled services while supporting highly-interactive, concurrent simulation of truly dynamic systems. Such a result has been accomplished by employing the Discrete Event Systems Specification (DEVS) [4] formalism for defining compositions of models and implementing this specification within the SOA framework, with the result referred to as DEVS/SOA [6].

An enhancement of HYDRA that meets the desired requirements can be obtained by extending the DEVS/SOA environment so that it includes the essential SOA features of the HYDRA environment. The result is a system that exploits the power of DEVS models and simulations which enable truly dynamic interactions among model components[7]. An example of such a need is where critical infrastructure components respond in highly interactive fashion to environmental disruptions. At the same time, the message exchange capabilities of DEVS/SOA can be enhanced with HYDRA message schemata for disruption propagation.

4. DEVS Background

DEVS provides a computational basis for a formal systems-theoretic framework for modeling and simulation that:

- Exploits the separation between model, experimental frame and simulator
- Supports expression of a wide variety of discrete and continuous model types (see Figure xx)
- Offers a standard for distributed simulation to support interoperability, composability, and reuse via the Systems Entity Structure ontology framework
- Supports automated, integrated complex systems development and testing

DEVS simulations are mediated by the DEVS simulation protocol that specifies an abstract simulation engine that ensures correct distributed simulation of DEVS atomic and coupled models, The DEVS protocol has specific mechanisms for:

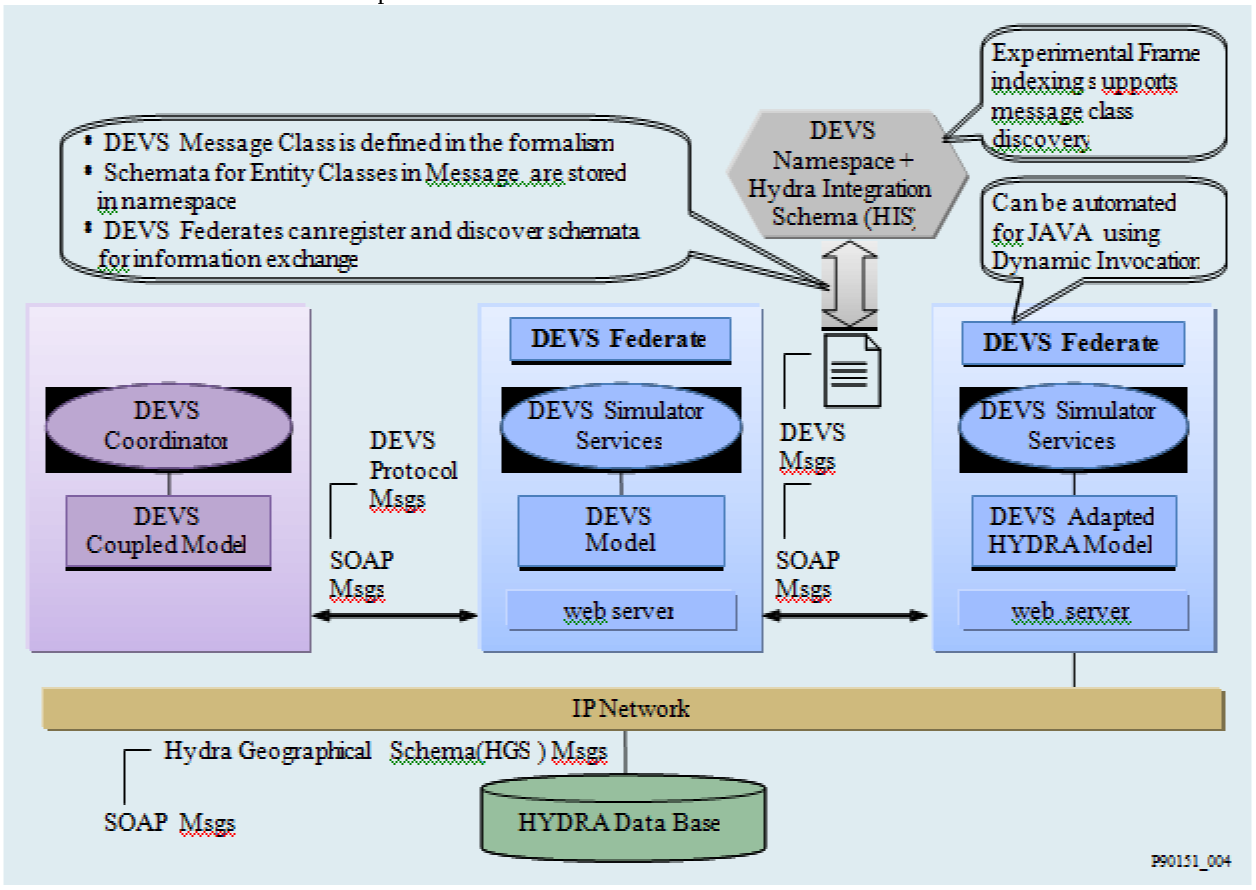
- declaring which components (federates) takes part in the simulation
- declaring how federates exchange information
- executing an iterative cycle that
  - controls how time advances
  - determines when federates exchange messages
  - determines when federates do internal state updating

Important for the confidence and integrity of model composition on distributed platforms, is the following theoretical result that is based on the underlying DEVS theory:

> *If the federates are all DEVS compliant then the simulation is provably correct in the sense that it guarantees a well-defined resulting structure and behavior as specified by the simulation builder.*

DEVS/SOA implements DEVS modeling and simulation services in a layered fashion. At the top is the application layer that contains models expressed in DEVSML, a way of representing DEVS models in XML, the standard for communicating web-services. This layer provides seamless Interoperation, composition and dynamic scenario construction resulting in portable models in DEVSML that are complete in every respect. These DEVSML models can be ported to any remote location using the

SOAP-based messages and be executed at any remote server. The simulation engine at the next lower layer is transparent to model execution and executes over the next lower SOAP-based middleware layer. The DEVSML model description files in XML contains meta-data information about compliance with simulation 'builds' or versions to provide true interoperability between various simulation engines that



DEVS Message Class is defined in the formalism
Schemata for Entity Classes in Message are stored in namespace
DEVS Federates can register and discover schemata for information exchange
DEVS Namespace + Hydra Integration Schema (HIS
Experimental Frame indexing supports message class discovery
Can be automated for JAVA using Dynamic Invocation
DEVS Coordinator
DEVS Coupled Model
DEVS Protocol Msgs
SOAP Msgs
DEVS Federate
DEVS Simulator Services
DEVS Model
web server
DEVS Msgs
SOAP Msgs
DEVS Federate
DEVS Simulator Services
DEVS Adapted HYDRA Model
web server
IP Network
Hydra Geographical Schema(HGS) Msgs
SOAP Msgs
HYDRA Data Base
P90151_004

Figure 1 DEVS/SOA Distributed Simulation Environment for HYDRA extension

adhere to the underlying DEVS protocol. Such run-time interoperability provides great advantage when models from different repositories are used to compose coupled models using DEVSML Interoperation capabilities [6]. DEVS models communicate via SOAP messages defined in a DEVS message class as specified in the DEVS formalism. Schemata for basic DEVS message classes are stored in the DEVS Namespace. DEVS Federates can register and discover schemata for information exchange. Experimental frame indexing supports message class discovery for newly constructed federates.

5. Model Federation Development and Simulation Support

As illustrated in Figure 1, enabling HYDRA-compliant models to serve as federates in DEVS/SOA simulation, may be accomplished through the following steps:

- Include the HYDRA Integration Schema within the DEVS Namespace – this will allow federates that have updated the HYDRA Data Base to inform other federates about the availability of the newly stored data. By design, HYDRA-compliant federates employ the HYDRA Geographical Schema to store data in the HYDRA Data Base.

- Enable HYDRA-compliant models to operate as discrete event models – this requires that they be adapted to interact via input events and output events with other federates as required by the

DEVS simulation protocol. Such models must be converted from a once-only events of data ingestion and data delivery (at the start and end of their execution respectively), to multiple data ingestions and deliveries that are based on events occurring on a global time line shared with all other components in the federation. Generally, there are two modes of such update reception/production – fixed time step and quantized significant-event updating. Although the former is often used, the latter has a number of advantages in this application. Quantum sizes can be viewed as service level contracts between producers and consumers whereby the smaller the quantum, the smaller the change in data attribute has to be to trigger an update. In basic application traversal of a quantum threshold results in asynchronous events time-managed by the DEVS protocol. Subsequent advanced application may aim at automated control of federate resolutions. For example, consumers may detect the need for more frequent data updates, thereby triggering an automatic upgrade in producer resolution model to meet the more demanding specification [8].

- Extend the interfaces of HYDRA-compliant models to

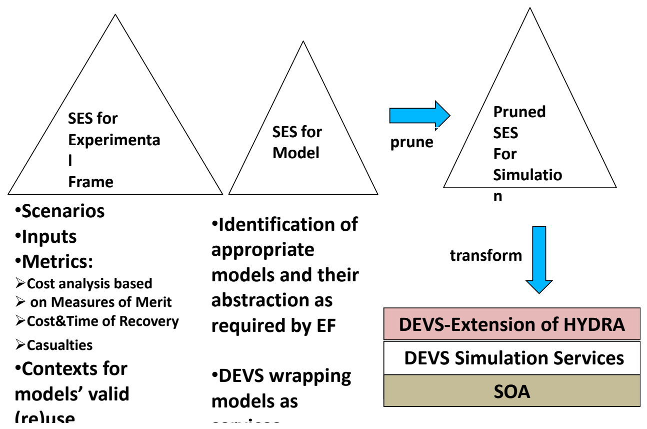


Figure 2 System Entity Structure Support for DEVS/HDRA Model Composition

include those required to interoperate with other DEVS federates as mediated by the DEVS simulation

6. Discovery and Composition of Models

An ontology framework is needed to enable analysts to frame specific questions in the domain of interest, e.g., in regard to system-wide impacts of disruptions and to employ such query specifications to help them discover relevant models in a repository to serve as components in system- wide federations. protocol – these interfaces will expose operations to respond to the protocol's time-management and information-exchange coordination commands. These operations will have been rendered to the models via the refactoring in previous steps. Once discovered, the ontology framework will also support composing a set of such models so that they can be executed in the DEVS/SOA simulation environment. Such an ontology framework can be centered on the System Entity Structure (SES)[9], a framework especially suited for dynamic systems simulation modeling. An SES specifies a family of hierarchically composed models from which a simulation model example can be constructed by applying pruning operations and transformation to DEVS model. In Figure 2, there are two SES components of an overall SES, one for experimental frames and one for models. The SES for frames represents scenarios inputs, metrics for analysis such as cost and time of recovery, and casualties. The SES for models represents the family of all executable hierarchical coupled models that can be specified by pruning operations [9]. Legacy models can be wrapped as DEVS models or accessed as services via a gateway construct [10]. In pruning the SES for frames, the objective is to select an EF that can address the current question or object of analysis. In pruning the SES for models, the objective is to compose a model that a) matches the selected EF in being able to meet the input/output and analytic requirements of the Frame and b) represents the system infrastructure configuration under consideration. Such a model is selected from specialization choices in hierarchical fashion and automatically transformed into a hierarchic composition of models resident in the model repository [3]. The resulting hierarchical, coupled model, representing an infrastructure configuration, can be simulated in distributed fashion by a DEVS simulator as illustrated in Figure 1. Likewise, pruning of the SES for frames results in a hierarchical coupled model that can interacts with the model to stimulate with inputs, observe its outputs and compute the metrics of interest to support evaluation.

7. Conclusions

We have considered support for dynamic interoperation of critical infrastructure models. Two main requirements were discussed: a) ability to federate component models for simulated dynamic interoperation, and b) ontology support to frame specific questions in the domain of interest, discover relevant model components and rapidly compose them for federated execution. The DEVS/SOA and SES modeling and simulation environment has been shown to provide a workable solution to extending the HYDRA environment to provide such support. Beyond such requirements, are additional requirements to support efficient and effective modeling and simulation. Such requirements include 1) tools and methods to select model components to scale analyses to meet resource constraints (time, budget, etc., 2) Ontology framework to support repository of models qualified by their domains of application/validity and execution effectiveness, 3) Support for incorporation of NISAC models and tools into associated repositories with extensibility to newly developed models and tools.

DEVS/SOA and SES environment can be placed as the lower layers within a layered architecture for M&S. [11] Consequently, they can supply services that can be augmented at the higher layers with capabilities to address the additional requirements just mentioned.

Extreme challenges are posed by real time decision making in response to critical infrastructure disruptions [12,13]. Such challenges include long standing problems in M&S theory and methodology such as the simplification and composition of disparate, complicated, multi-scale models with the attendant trade-offs between execution speed and the accuracy that models can exhibit. These challenges are amplified in the case of real-time decision making in the face of infrastructure attacks for a number of reasons: 1) As a new area of concern, there are few abstractions that are known to be useful in developing concise, valid, and rapidly executable models applicable to objectives-driven simplification of infrastructure models developed for purposes other than disruption analysis. 2) We have very little experience with how to automate simplification processes as is necessary to meet the objectives of disruption analysis in the severely time-constrained reaction to crisis events. 3) Likewise, we have limited experience with automation of model simplification processes that propagate explanations and justifications of the steps taken to create desired simulation models – such outputs being needed to support confidence of human crisis managers in the resulting simulations and the decisions make with them.

The Layered Architecture for M&S would set the stage for application of modeling and simulation methodology [5,9] to these extreme challenges with the potential for a quantum leap in capability to respond to national-scale infrastructure disruption events.

8. References

1. William R. Graham. *Report of the Commission to Assess the Threat to the United States from EMP Attack: Critical National Infrastructures*. McLean, VA: Congressional EMP Commission, 2008. Available at: http://works.bepress.com/george_h_baker/17
2. Bent, Russell, et.al. Hydra: A Service Oriented Architecture for Scientific Simulation Integration. 42nd Annual Simulation Symposium (ANSS'09)
3. NISAC:http://www.sandia.gov/mission/homeland/programs/critical/nisac.html
4. B. P Zeigler, H. Praehofer, T. G. Kim, *Theory of Modeling and Simulation*, Academic Press, 2000
5. Frederick Kuhl, Richard Weatherly, Judith Dahmann, "Creating Computer Simulation Systems: An Introduction to the High Level Architecture", Prentice Hall PTR, 1999
6. Saurabh Mittal, José L. Risco-Martín & Bernard P. Zeigler: DEVS/SOA: A Cross-Platform Framework for Net-Centric Modeling and Simulation in DEVS Unified Process. SIMULATION: Transactions of SCS 85(7): 419-450 (2009)
7. James Nutaro. Discrete Event Simulation of Continuous Systems. In Dynamic Models by P.A. Fishwick, 2006
8. James J. Nutaro, Bernard P. Zeigler, Rajanikanth Jammalamadaka, and Salil R. Akerkar. Discrete event solution of gas dynamics within the DEVS framework. In International Conference on Computational Science, volume 2660 of Lecture Notes in Computer Science, pages 319-328. Springer, 2003.
9. B.P. Zeigler, and P. Hammonds, "Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange", New York, NY: Academic Press, 2007

10. Bernard Zeigler, Saurabh Mittal, Xiaolin Hu "Towards a Formal Standard for Interoperability in M&S/System of Systems Integration", AFCEA-George Mason University Symposium, May, 2008

11. H. Sarjoughian, B. Zeigler, and S. Hall, ―A Layered Modeling and Simulation Architecture for Agent-Based System Development‖, Proceedings of the IEEE 89 (2); 201-213, 2001
12. Hall, Steven B, "The Changing Role of M&S in Developing Tomorrow's Military Complex Adaptive Systems", Invited address to the National Defense Industrial Association, 2007
13. Hall, Steven B, "Learning in a Complex Adaptive System for ISR Resource Management". Spring Simulation