LA-UR-08-07830

Approved for public release; distribution is unlimited.

Title: Hydra: A Service Oriented Architecture for Scientific Simulation Integration

Author(s): Russell Bent, Tatiana Djidjeva
Birch Hayes, Joe Holland
Hari Khalsa, Steve Linger
Mark Mathis, Sue Mniszewski,
Brian Bush

Intended for: 42nd Annual Simulation Symposium (ANSS'09)

Hydra: A Service Oriented Architecture for Scientific Simulation Integration

Russell Bent, Tatiana Djidjeva, Birch Hayes, Joe Holland, Hari Khalsa, Steve Linger, Mark Mathis, Sue Mniszewski
Los Alamos National Laboratory
{rbent,tatiana,birch,jvh,hari,spl,mmathis,smm}@lanl.gov
and
Brian Bush
National Renewable Energy Laboratory
brian_bush@nrel.gov

Keywords: Service Oriented Architecture, Infrastructure, Hydra, Interoperability

Abstract

One of the current major challenges in scientific modeling and simulation, in particular in the infrastructure-analysis community, is the development of techniques for efficiently and automatically coupling disparate tools that exist in separate locations on different platforms, implemented in a variety of languages and designed to be standalone. Recent advances in web-based platforms for integrating systems such as service-oriented architecture (SOA) provide an opportunity to address these challenges in a systematic fashion. This paper describes Hydra, an integrating architecture for infrastructure modeling and simulation that defines geography-based schemas that, when used to wrap existing tools as web services, allow for seamless interoperability. While conducting new studies in Hydra, existing users of these tools increase their analysis capabilities by assessing how the simulation results of one tool impact the behavior of another tool and can automate existing ad hoc processes and work flows for highly complex scenarios.

1. INTRODUCTION

Over the past fifty years the size and mission of Los Alamos National Laboratory (LANL) has allowed it to make considerable scientific contributions in the area of modeling and simulation, in particular of infrastructure, but this has resulted in the creation of many independent and disparate software projects to address different questions. In recent years, there has been recognition that many of the projects have natural points of interaction and that the integration of projects can produce tools whose scope extends beyond the individual projects. For example, a hurricane-simulation tool and an electric-power-grid-simulation tool have a common interface at the point where hurricane winds and flooding damage electric components. The two simulations, when coupled, allow the creation of a new tool that simulates electric power grid behavior during and after hurricanes. Indeed, recent sponsors of research at LANL have required analysis that necessitate integration [30]. While conceptionally straight-forward, current approaches for scientific integration of simulations tend to be analysis-specific and ad hoc due to a number of technical challenges, including software implementations in different languages and operating system platforms, input/output incompatibilities, non-standard integration interfaces, and requiring considerable manual intervention [1]. This situation is exasperated when considering integration between institutions when even the option of sharing source code is unavailable or undesirable. Thus, there exists a very strong need to create generic software integration architectures for scientific modeling and simulation [11].

This challenge is not unique to modeling and simulation communities. Most notably this challenge arose in business operations where legacy systems for performing specific tasks such as payroll, scheduling, evaluations, etc. over the years required more and more interaction and were not designed to do so. The problem arose during mergers, in particular in the airline (reservation systems) and banking industries (account management) where two (or more) incompatible systems that served the same function needed to be integrated in the new combined company. It also arose as increased collaboration and out-sourcing required business-to-business data sharing. This led to the rise of Service Oriented Architectures (SOA) and web services [21]. The key idea was to develop an architecture for *wrapping* the functionality (*service*) of existing tools and systems in a standard way that was platform and language independent. In order to allow access to the service over a network (i.e. the web) SOA makes use of WSDL (Web Service Description Language) as its standard API. WSDL is an XML-based language for describing how to access a service over the web (typically called a *web service*). WSDL includes information about the communication protocols necessary for connecting with the service, the interfaces for interacting with the service, and the types of data each interface requires as input and output [32].

At the same time the Open Geospatial Consortium (OGC) was developing SOA web service standards for delivering geographic content over the Internet. WFS (Web Feature Service) and WMS (Web Map Service) are two OGC standards that are of interest here. Both use GML (Geographic Markup

Language), an XML-based language, as the message format and provide mechanisms to obtain geography and geography attributes and manipulate that data. The WMS standard is considerably simpler and provides information in picture format that can be displayed in map viewers (i.e. Google Maps). It is useful for applications that wish to display geometric information and are less interested in the attributes associated with the geometry. The WFS standard serves the geography directly as a series of points and allows data (features) to be associated with each geographic object (i.e. population, name, land use, etc.). Often the same underlying data can be served in both formats depending on the needs of the requester [23].

The recent advances in Geographic Information Systems (GIS), both for the desktop and more recently over the web (such as OGC), have allowed for a remarkable rise in infrastructure modeling and simulations based on GIS [11] and its use has spread to a number of governmental organizations such as NOAA, NASA, EPA, and more recently DHS [17] to address these sorts of analysis. The focus of the bulk of such work has been on how to take advantage of GIS to perform common geographic operations and visualize results from multiple models and simulations in a single platform, but has not used geography as a mechanism for tying the functional aspects of modeling and simulation together.

Inspired by the success of SOA in the business operations community, availability of widely accepted and robust OGC standards, and the utilization of GIS for infrastructure modeling and simulation, this paper presents Hydra, a novel service-oriented architecture for automatic plug-and-play integration of infrastructure models and simulations. This paper focuses on defining standards for scientific simulation integration that take advantage of natural couplings that exist through geography that to the best of our knowledge has yet to be explored in this community. [1]

The architecture is based entirely on SOA and OGC for defining its standards and makes the following key contributions.

- To the best of our knowledge, the first generic architecture for integrating scientific *infrastructure* modeling and simulation via web-based standards. See [7], [22], [26], [29], [10] for examples of workflow modeling in other scientific domains.

- A standard I/O schema for integration of scientific *infrastructure* modeling and simulation that utilizes geographic information.

- Web-enabled access to scientific *infrastructure* modeling and simulation.

- Ability to create arbitrary tools by connecting disparate *infrastructure* simulations in novel ways.

- Ability to consider multiple simulation packages that perform the same function at different levels of fidelity and efficiency in order to find the best match for the task at hand.

Though we believe we are one of the first to propose and implement an SOA standardized architecture for integrating scientific modeling and simulation for infrastructure through geography, it is important to understand that the literature contains years of interoperability research in a number of scientific domains. Some of the recent work has explored web-technologies to achieve interoperability results. We now discuss a small sample of this work to help place Hydra in the greater interoperability research context. Early work on the extensible modeling and simulation framework (XMSF) in [4] and [2] formally defines many of the interoperability research challenges for modeling and simulation, with a focus on web-based technologies to solve them. They develop a framework for addressing these challenges and demonstrate its application on military modeling and simulation. A number of challenges discussed by XMSF also arose in Hydra and Hydra's solutions can be thought of as implementations of parts of XMSF for infrastructure modeling and simulation. Concurrently with XMSF, [1] also discussed many of the same challenges in the context of environmental sciences and suggested web-based solutions to these challenges.

More recently, there has been a push to formally define degrees of composability and interoperability (i.e a Levels of Conceptual Interoperability Model) in order to better understand how to achieve the necessary level of interoperability (see [33] and [29], among others). Reference [29] suggests that data-engineering approaches can be used to address many of the challenges inherent in achieving different levels of interoperability which is similar to the Hydra solution for creating composable and interoperable infrastructure modeling and simulation tools.

Finally a small sampling of literature in other domains yield a number of communities utilizing web services and OGC for interoperability including [25] (crisis management), [26] (earthquake simulations), [22] (bioinformatics), [3] (scientific workflows), [7] (weather), and [24] (electric power command and control) that suggests an SOA approach to infrastructure modeling and simulation is appropriate.

The remainder of this paper is organized as follows. Section 2 describes the standard schemas we have developed for defining geographic interaction points between infrastructure simulation tools. Section 3 provides example implementations of the schema to demonstrate how existing tools were wrapped to conform to the schema. Section 4 discusses three integration examples of the tools described in section 3. Sec-

[1] Hydra is currently available to LANL and its government sponsors. Discussions are ongoing as to how to make Hydra available to the greater scientific community

tion 5 describes two client applications for interacting with the web services in a seamless way. Section 6 concludes our presentation.

2. ARCHITECTURE DESCRIPTION AND STANDARDS

A significant goal of the Hydra architecture is to provide a standard framework for integrating scientific models and simulations with a focus on infrastructure modeling and simulation. This section describes two schemas for integrating infrastructure simulations through geographic information.

2.1. Hydra Geography Schema (HGS)

The Hydra Geography Schema (HGS) aims to describe geographic output of infrastructure simulations (i.e. impact areas computed by hazard simulations (hurricanes, earthquakes, etc.) and infrastructure behavior simulations (outage areas) through a common format using data-engineering practices similar to that of [29]. Because of the disparate types of infrastructure simulations and the eclectic set of analysis tools in use, we have emphasized flexibility and broad applicability in the design: visualization capabilities, compatibility with GML 2.x and 3.x, ESRI shape files, WFS servers, and spatially-extended relational databases. These compatibility requirements led us to a schema described in Table 1.

Field	Description	Allowed Values
IMPACT_UID	A unique identifier for the impact area.	Unique integer.
IMPACT_DOMAINS	The domains(s) [separated by commas] to which the impact applies, or blank if the impact applies to all domains.	Optional character string.
IMPACT_TYPE	The type of impact in the area.	One of *fatality*, *injury*, *infection*, *evacuation*, *contamination*, *destruction*, *damage*, or *outage*.
IMPACT_SUBTYPE	The more specific type of impact: e.g., the type of contamination, infection, etc.	Optional character string
IMPACT_LEVEL	A numeric value quantifying the impact.	Optional real number.
IMPACT_UNITS	The units of measure for the quantification of impact.	Optional character string, unless IMPACT_LEVEL is specified.
IMPACT_BEGINNING	The beginning time of the impact.	Optional time stamp, unless IMPACT_ENDING is specified.
IMPACT_ENDING	The ending time of the impact.	Optional time stamp.
IMPACT_ANNOTATION	A textual description of the impact for the area.	Optional character string.
IMPACT_AREA	A geometric object specifying the area impacted.	Polygon, Point, or Line

Table 1. Hydra Geography Schema.

Each simulation tool may output one or more records of the type described in Table 1: Each record is identified uniquely and is associated with a single geographic object. The domain of the application and the type of impact is described along with a quantification of the impact and its beginning and ending time. A textual annotation provides an end-user-readable description of the result. Since many simulations do not provide the full set of fields, a number of the fields are optional. A simulation may also augment HGS with additional attributes where appropriate. HGS provides an input/output standard in which web service implementations (wrappers) of existing tools conform to and promote ease of integration.

2.2. Hydra Integration Schema (HIS)

The HGS provides a common data format for communicating geographic information between infrastructure models and simulations. It remains to describe a structure for transmitting the data that is scalable and easy to use. As the data itself can be arbitrarily large and recipients of the data may not require the data in its entirety, we did not define a schema to directly send the data between simulation tools. Instead, messages about the data's location are transmitted and it is up to the recipient web service to use that message to obtain the pieces of the data it needs and translate that data into an appropriate format. HGS lends itself to two message (integration) architectures both based on standard protocols. The first schema focuses on retrieving the data via WFS and includes three string fields: a URL of the WFS server, a namespace of the data, and a Contextual Query Language (CQL) [6] filter for obtaining specific data out of the namespace. The second schema focuses on retrieving the data via a georeferenced database such as PostgreSQL. It consists of four string fields including: a URL for the database server, the name of the database, the table in the database, and a CQL filter for obtaining specific data out of the table. We refer to individual instantiations of the schema as tuples and HIS provides a definition for transmitting sets of tuples between services.

HIS provides a common language for creating web services in the Hydra architecture and allows us to address the data transfer challenge described in [10]. Every web service that obeys the Hydra architecture expects HIS as part of its input, retrieves the data that HIS references, and translates the retrieved HGS into input for the underlying modeling and simulation tool (such as damage to components). In addition, every web service is expected to provide as part of its output HIS tuples. By obeying each rule, the output of any other Hydra web service can be passed as input to any Hydra web service in a seamless, plug-and-play fashion. Of course, the value of such interactions is only as good as the overlap between the two services (for example, the output of a Florida model may have limited effect as input to a California model), yet the behavior remains defined.

The utility of HGS and HIS confirms many of the results and observations found in [2] and [29]. In the context of the Levels of Conceptual Interoperability Models (LCIM) of [29], HIS provides an example of a syntactic interoperability level as it specifies a common structure to exchange information. HGS provides a hybrid of syntactic and semantic levels as it specifies a structure for storing information (syntactic) and using model-based data engineering practices to impose semantics on the information that can be stored. Thus, HGS serves the same Common Reference Model role for infrastructure simulations as C2IEDM does for [29] and [2] for military applications. We next describe specific implementations of the Hydra architecture and integrations.

3. ARCHITECTURE IMPLEMENTATION

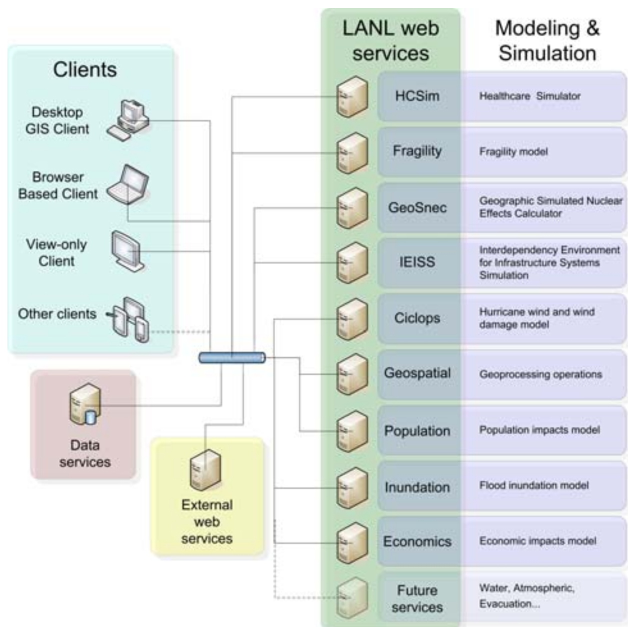

Figure 1. Hydra-implementations.

The importance and utility of the Hydra architecture is best illustrated via demonstrations of how it is used and implemented. Figure 1 provides an overview of our implementations of the Hydra architecture to date. On the right hand side is an enumeration of the scientific models and simulations that have been wrapped as web services to conform to HGS and HIS. The left hand side shows how users may interact with the services (via clients), indicates the existence of any data sources for running the services, and defines points of entry for interacting with external web services.

Geographic Simple Nuclear Effects Calculator (SNEC) SNEC is a fast simulation of nuclear detonations that is used provide highly approximate results in fast turnaround situations [28]. The output of this tool consists of polygons describing blast, overpressure, and radiation.

Fragility Fragility is a tool for calculating the probability of damage to a building given the type (and magnitude) of the event it is subjected to and how the building was constructed [8]. The input of this tool consists of geo-referenced polygons or grids of a hazard (such as flood, earthquake, etc.). The output of this tool is a list of infrastructure components (from the data service) and the probability each component is damaged.

Population This is a software tool for estimating population at arbitrary resolutions based on aggregate data stored in the data service [18]. The output of this tool is the number of people in the HGS input shapes.

Infrastructure Simulation (IEISS) IEISS is a set of software tools for simulating the behavior of physical infrastructure systems such as electric power, natural gas, telecommunications, etc. [5],[31]. IEISS allows users to deactivate components in the infrastructure models, which the Hydra-compliant web service interface uses to conform to HGS by deactivating all assets in polygons defined by HGS (or creating scenarios if the HGS originates from Fragility). The output of IEISS is geographic representations of outage areas for each infrastructure and this is exported directly to an HGS-compliant PostgreSQL database.

Health Care Simulation (HCSim) HCSim is an agent-based health care facility simulation system for predicting health-care surges from natural and man made disasters. HCSim allows users to deactivate components in the infrastructure models and provide patient profiles, which the Hydra-compliant web service interface uses to conform to HGS by deactivating all assets in polygons defined by the HGS (or creating scenarios if HGS originates from Fragility) and uses output from the population service to generate patients. The output of HCSim is a geographic representation of aggregated patient outcomes.

Economic This is a software tool for calculating direct, indirect, and induced economic costs of infrastructure and business damage as well as loss of service. [19].

Ciclops This a tool for simulating hurricanes based on the work of [14], [16], [27]. It exports maximum wind velocities in HGS.

Inundation This a tool for simulating floods in two dimensions [13]. It exports maximum flood depths in HGS.

Geospatial This is a set of geographic utilities built as a convenience web service for Hydra services to conform to Hydra standards similar and serves a similar purpose as the tool described in [12].

4. INTEGRATION IMPLEMENTATION

Each of the tools described in the prior section were designed to produce results independent of one another, each answering very specific questions. However when questions arise such as *How would a nuclear event affect the economy?* or *How will the effects of a hurricane cascade through the nation's infrastructure systems?* it is clear that none of these tools alone are sufficient to answer these questions. Instead it is a combination of these tools that can answer such questions. The challenge, as described earlier, is to design an integration architecture so that it is not locked into a specific pattern. This section describes three integrations of the tools

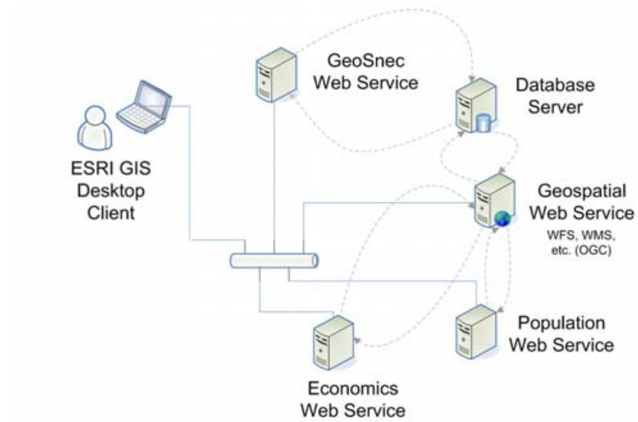

Figure 2. GeoSnec Integration Example

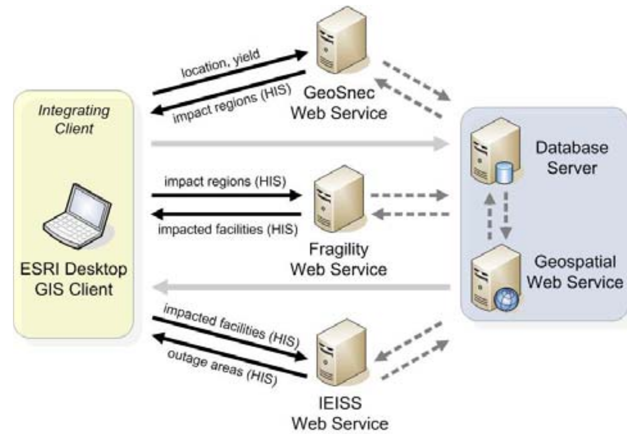

Figure 3. Fragility Integration Example

described in Section 3 using the standards of Section 2 to demonstrate how the architecture answers this challenge by supporting three very different integrations.

Economics and Population Integration Example Figure 2 suggests a simple and illustrative connectivity view of an integrating application for computing the economic and population impacts of a nuclear event. On the left-hand side is the integrating client that the user uses to invoke the application (Section 5 describes integrating clients in more detail). The SNEC simulation tool is launched by the integrating application (the horizontal tube in the middle of the picture) using parameters provided by the client (such as location and yield) to begin the simulation which stores its results in an HGS-compliant database. The location of the results (in HIS format) are returned to the integrating application which are provided as inputs to the economics and populations services. These two services provide answers on the number of people impacted by the nuclear event and the economic cost of the event which are returned by the integrating application to the client (in HIS) for display.

Infrastructure Simulation Integration Example Figure 3 provides a slightly more complicated probabilistic application for integrating SNEC with IEISS using the Hydra architecture and the tools of Section 3. This figure shows the integration application in terms of the flow of information the application facilitates. Once again, the SNEC application is used to generate a geographic representation of a nuclear effect. This information is provided to the fragility tool via HIS. The fragility tool produces a probabilistic representation of how likely infrastructure assets are to be damaged by the nuclear event. This information is provided to IEISS via HIS which is used to create an ensemble of possible infrastructure damage scenarios. IEISS simulates the scenarios to create a probabilistic model of outaged service areas for each commodity in the infrastructure model (electric power, telecommunications, natural gas, etc.). This result is provided by the application via HIS to the client for display.

Hurricane Fast Response Process Example Finally, Figure 4 describes the LANL process for performing hurricane fast response studies for the U.S. Department of Homeland Security. This process includes up to thirty individuals organized into eight teams (each color of a box represents a team) at two institutions (LANL and Sandia National Laboratories) with nineteen subprocesses. These studies are provided when a hurricane landfall is imminent in the United States and is intended to provide the federal government with an estimate on the destruction and cost to restore an area once the hurricane passes. Due to the nature of the event, results are required quickly, typically within four to eight hours (otherwise passing events render the study useless) of a pre-landfall request. Achieving this goal is often difficult due to the ad hoc, inconsistent, and manual interactions to achieve integration between each of the subprocesses. This example illustrates the motivation for Hydra and demonstrates one of the main integration applications that is the goal of this project. The figure shows the progress in achieving this goal by highlighting the implemented Hydra compatible simulations.

Interestingly, the Hydra philosophy allows integration applications to choose which simulation package to use for a specific piece of the integration as long as each piece complies with HIS and HGS. For example, if a user desires to use a higher fidelity but more computationally intensive simulation to model nuclear effects, that user can replace SNEC without changing any portion of the integrating client other than the parameter for which nuclear effect simulation tool to use. These examples serve to illustrate the flexibility of Hydra and how it has been used to automate a variety of services. The next step for future work is to quantify the benefits of using Hydra (beyond the time-savings and consistency benefits) by comparing the results of the Hydra-enabled processes with the processes they are replacing.

5. INTEGRATION CLIENTS

This section now describes how *integrating clients* are constructed to allow seamless composability of the tools de-

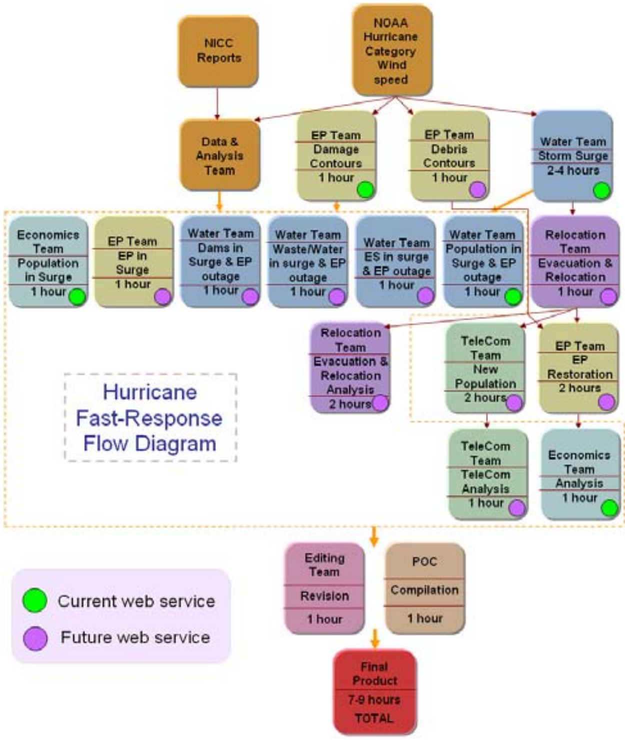

Figure 4. Hurricane Integration Example

scribed in the Section 3 to create the types of applications described in Section 4.

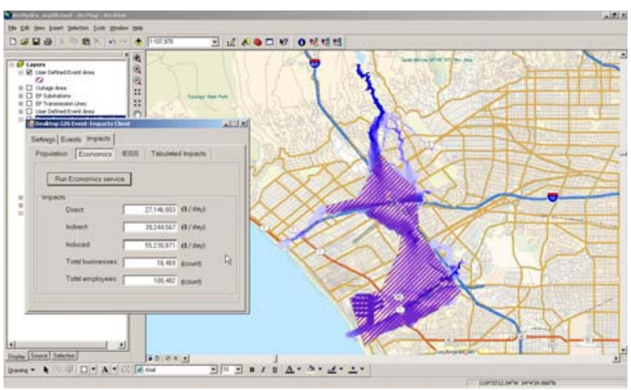

Figure 5. ArcHydra is a GIS-based application that integrates the LANL web services.

ArcHydra Our first example of an integrating client of the LANL web services is *ArcHydra*. This Geographic Information System (GIS)-based application, shown in Figure 5, provides an analyst with a flexible and intuitive interface for composing integration scenarios. The ArcHydra menu is chiefly divided into events and impacts. The events menu tab enables users to simulate events, such as hurricanes, nuclear blasts, and flood inundation, which may result in impact regions. Through the impacts menu tab, any potential direct impacts or cascading impacts affecting the operation of na-

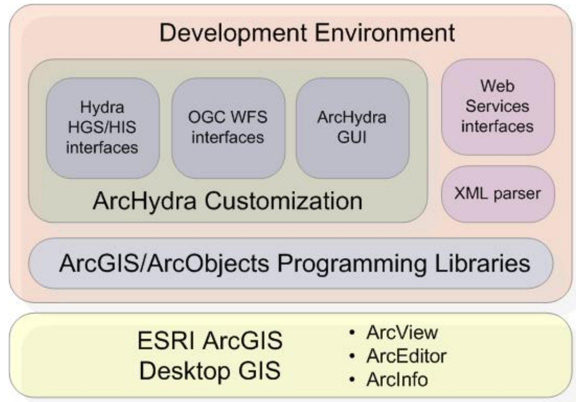

Figure 6. ArcHydra is a custom application developed with the ArcGIS libraries and web service standards that extends the functionality of the ESRI ArcGIS Desktop GIS products.

tional infrastructure can be estimated. Intermediate and final geospatial results, calculated by web services (i.e., models) and provided in HGS, are rendered on the integrating client's GIS map. The generality of the HIS standard enables ArcHydra users to compose scenarios from available events and impacts through point-and-click menus and execute simulations from start to finish entirely within the same integrating-client application.

Built within the ESRI ArcGIS Desktop GIS environment as a custom extension, ArcHydra extends the mature and full-featured capabilities of the popular commercial off-the-shelf (COTS) ArcMap software package. Figure 6 shows the main development components and environment of the ArcHydra application. ArcHydra leverages the rich geographical interface of the GIS environment both for capturing user gestures as model inputs and for displaying geospatial model results. The intrinsic functionality of GIS enables the use of a multitude of standard base-map layers or problem-specific map layers to provide geographic context and meaning to model results. Some examples of these supporting map layers include elevation or topographic data, road networks, 3 D building sets, emergency services, and census demographics.

The web-services-aware development environment used to construct ArcHydra provides built-in functionality to find and employ the methods of web services exposed through the WSDL API for use within the ArcHydra client. The underlying software developed for ArcHydra includes libraries to work with the HIS standard protocols. By developing these reusable libraries and by imposing the HIS standard on all web services, integrating additional new LANL web services has been greatly simplified within ArcHydra. The flexibility and extensibility of the HIS standard enables the future inclusion of other events (i.e., models via web services) that are not initially based on area impacts, such as point and line releases of chemical, biological, and radiological agents (though these examples do ultimately result in area impacts because of the

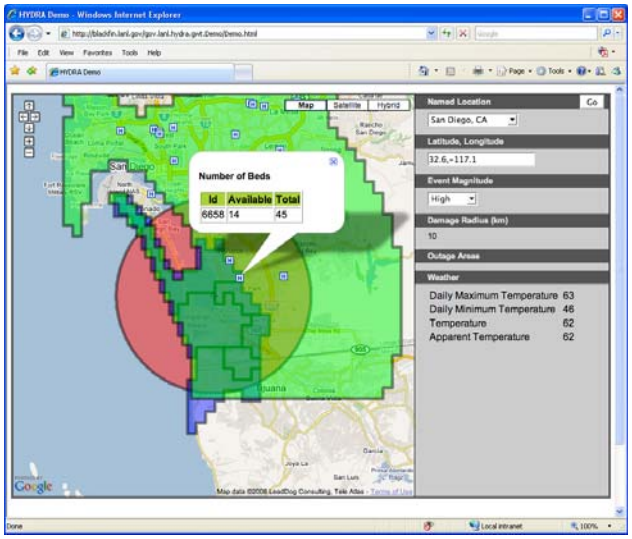

Figure 7. The Hydra web-based application integrates LANL and external web services within a browser.

dispersive effects of atmospheric conditions). These specific atmospheric types of initiating events currently exist as standalone applications, and are slated to be included as future LANL scientific web services within the Hydra framework, and within the ArcHydra integration client.

Browser-Based Integrating Client A second example of an integrating client of LANL web services is a web browser-based application. Shown in Figure 7, this lightweight client executes within readily-available web browsers and requires minimal system resources as compared with a heavy client, such as ArcHydra. Accordingly, the browser-based integrating client has recognized limits on capabilities and sophistication. Even with limitations, the browser framework offers a contemporary interactive look-and-feel with a reasonable level of performance appropriate for many users.

The browser-based client utilizes both internal LANL web services and external web services. The application was developed using the Google Web Toolkit (GWT), which is freely-available from Google. The GWT product allows developers to write applications using Java; the Java code is then translated to JavaScript using the GWT Java-to-JavaScript Compiler, and then interpreted within the HTML-web browser environment. Figure 8 shows the components of the integration environment for the web-based client. The mapping component is provided by the Google Maps JavaScript API and enables the Google Maps geospatial data and map gestures to be embedded in the browser-based client. Additional technologies are employed to manage the interactions between the LANL web services, external web services and the various components to construct the integrating client. PHP is used to communicate with web services; XSL (Extensible Stylesheet Language) is used to translate geospatial data from the GML format to the Google (OGC) KML

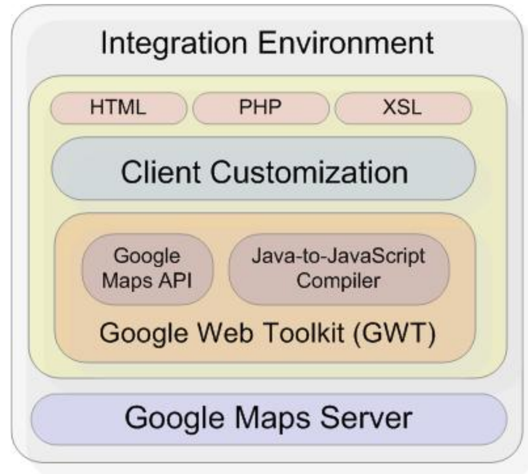

Figure 8. The Hydra web-based application composes infrastructure simulations by integrating LANL web services and external web services within a web browser environment.

format for rendering on the client map. Interestingly, many of the features of this client address the open GIS challenges of [20].

6. CONCLUSION

The Hydra architecture described in this paper allows infrastructure modelers to vastly increase the types of analysis that they can perform by allowing them to integrate their existing tools efficiently and effectively. Existing (manual) processes and work flows for integrating tools are now considerably faster through the use of the Hydra architecture and there are plans to replace many of the manual processes currently in use with Hydra applications. The success of the Hydra architecture to date presents a number of interesting future directions. First, we plan to create new schemas for interactions between simulations other than geography when that is appropriate. Second, SOA-based architectures are inherently focused on functionality, state-less, message driven, and loosely coupled. Thus, the described Hydra architecture is implicitly designed as a serial process. It will be important to explore architectures for supporting highly-interactive, parallel integrations that may take advantage of recently SOA-enabled architectures such as HLA [15] and achieve the dynamic interoperability level of [29]. Third, in order to expose many of the Hydra-compliant services outside of LANL and its sponsors, we will need to build security into the architecture. It will be interesting to explore the newer versions of the Globus SOA architecture and the extensive XMSF approach to security [2] to determine if their approaches solve many of these open issues for infrastructure modeling and simulation [9].

Finally, there are number of purely scientific questions that can be explored via arbitrary model integration. For example, models and simulations may run at differing levels of resolution and SOA integration allows researchers to investigate methods for appropriately integrating multiple resolutions of models together. This scenario was also one of the motivations for XMSF [2]. Second, now that the Hydra architecture has automated processes such as those described in Section 4, it will be interesting to compare the results of the automated processes with the manual, human-intensive processes to quantify how much time is saved and the error avoidance rate. It is also important to compare the quality of the automated results with the manual results to determine if automation results in any degradation in analysis quality.

Acknowledgment This work was supported in part by the U.S. Department of Homeland Security's National Infrastructure Analysis and Simulation Center (NISAC) Program. We also thank the three anonymous referees whose comments vastly improved the quality of this presentation.

REFERENCES

[1] R. Argent. An overview of model integration for environmental applications-components, frameworks and semantics. *Environmental Modeling and Software*, 19:219–234, 2003.

[2] C. Blais, D. Brutzman, D. Drake, D. Moen, K. Morse, M. Pullen, and A. Tolk. Extensible modeling and simulation framework (xmsf) 2004 project summary report. *Naval Postgraduate School, NPS-MV-05-002*, pages 1–101, 2005.

[3] S. Bowers and B. Ludascher. Actor-oriented design of scientific workflows. In *24th International Conference on Conceptual Modeling*, pages 369–384, 2005.

[4] D. Brutzman, M. Zyda, J. Pullen, and K. Morse. Extensible modeling and simulation framework (xmsf) challenges for web-based modeling. *Findings and Recommendations Report: Technical Challenges Workshop, Strategic Oppurtunities Symposium*, pages 1–52, 2002.

[5] Brian Bush and et al. Interdependent energy infrastructure simulation system - ieiss technical reference manual. *LANL Technical Report LANL-D4-05-0027*, pages 1–254, 2005.

[6] Contextual Query Language. *http://www.loc.gov/standards/sru/specs/cql.html.* Accessed November 2008.

[7] K. Droegemier and et al. Service oriented environments for dynamically interacting with mesoscale weather. *Computing in Science Engineering*, 7 (6):12–29, 2005.

[8] Federal Emergency Management Agency. Multi-hazard loss estimation methodology. *HAZUS-MH MR3 Technical Manual*, 2003.

[9] I. Foster. Globus toolkit version 4: Software for service-oriented systems. *IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779*, pages 2–13, 2006.

[10] G. Fox and D. Gannon. Workflow in grid systems. *Concurrency and Computation: Practice and Experience*, 18 (10):1009–1019, 2006.

[11] H. Gimblett. *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Simulating Social and Ecological Processes*. Oxford University Press, 2002.

[12] E. Jager, I. Altintas, J. Zhang, B. Ludascher, D. Pennington, and W. Michener. A scientific workflow approach to distributed geospatial data processing using web services. In *Proceedings of the 17th international conference on scientific and statistical database management*, pages 87–90, Santa Barbara, CA, 2005.

[13] D. Judi, S. Burian, B. Okhuysen, and T. McPherson. Emergency flood management using computational fluid dynamics. In *Proceedings of the Annual Water Resources Conference (AWRA)*, 2008.

[14] J. Kapalan and M. DeMaria. Simple empirical model for predicting the decay of tropical cyclone winds after landfall. *Journal of Applied Meteorology*, 34:2499, 1995.

[15] F. Kuhl, R. Weatherly, and J. Dahmann. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Prentice Hall, 1999.

[16] C. Martino, K. Cheung, A. Phadke, and S. Houston. Simulation of hurricane waves with parametric wind fields. *MTS/IEEE Conference (OCEAN)*, 2:1323–1330, 2001.

[17] K. McDonald and L. Di. Geoss architecture implementation pilot phase 2 kickoff workshop. *NOAA-NASA GOES-R and GMU CSISS Joint Efforts for Persistent GOES Data Services, Weather Scenarios, Web Geoporcessing Services, and BPEL-based Workflows*, 2008.

[18] T. McPherson, J. Rush, H. Khalsa, A. Ivey, and M. Brown. A day-night population exchange model for better exposure and consequence management assessments. *86th AMS (American Meteorological Society) Annual Meeting, Atlanta, GA.*, pages 1–6, 2006.

[19] Minnesota Implan Group. *http://www.implan.com*. Accessed November 2008.

[20] E. Nash, P. Korduan, S. Abele, and G. Hobona. Design requirements for an ajax and web-service based generic internet gis client. In *Eleventh AGILE International Conference on Geographic Information Science*, pages 1–6, Girona, Spain, 2008.

[21] E. Newcomer and G. Lomow. *Understanding SOA with Web Services*. Addison-Wesley Professional, New York, 2004.

[22] T. Oinn and et al. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20 (17):3045–3054, 2004.

[23] Open Geospatial Consortium. *http://www.opengeospatial.org.* Accessed November 2008.

[24] J. Pathak, Y. Li, V. Honavar, and J. McCalley. A service-oriented architecture for electric power transmission system asset management. In *Service-Oriented Computing (ICSOC)*, pages 26–37, 2006.

[25] S. Pezanowski, B. Tomaszewski, and A. MacEachren. An open geospatial standards-enabled google earth application to support crisis management. *Geomatics Solutions for Disaster Management*, pages 225–238, 2007.

[26] M. Pierce, G. Fox, M. Aktas, G. Aydin, H. Gadgil, Z. Qi, and A. Sayer. A the quakesim project: Web services for managing geophysical data and applications. *Pure and Applied Geophysics*, 165 (3-4):635–651, 2008.

[27] R. Schwerdt, F. Ho, and R. Watkins. Meteorological criteria for standard project hurricane and probable maximum hurricane windfields, gulf and east coasts of the united states. *U.S Dept. of Commerce NOAA Technical Report NWS23*, pages 1–343, 1979.

[28] J. St. Ledger. User guide for the los alamos national laboratory simple nuclear effects calculator. *Los Alamos National Laboratory, LA-CP-02-0580, Official Use Only, Export Controlled*, 2002.

[29] A. Tolk, C. Turnitsa, S. Diallo, and L. Winters. Composable m&s web services for net-centric applications. *Journal for Defense Modeling and Simulation*, 3 (1):27–44, 2006.

[30] U.S. Department of Homeland Security. National infrastructure protection plan. 2006.

[31] R. Waupotitsch, S. Eidenbenz, L. Kroc, and J. Smith. Multi-scale integrated information and telecommunications system (miits): First results from a large-scale end-to-end network simulator. *Proceedings of Winter Simulation Conference*, pages 2132–2139, 2006.

[32] Web Services Description Language. *http://www.w3.org/TR/wsdl.* Accessed November 2008.

[33] B. Zeigler and P. Hammonds. *Modeling and Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange*. Academic Press, Burlington, MA, 2007.