Automated User Interface Generation for the Integrated Modeling, Mapping, and Simulation (IMMS) Framework

Zach Heath, Allen Bagwell, Katherine Guzman,
Marilyn Hawley, Karim Mahrous, Stephen Mueller, Todd Plantenga,
Andrew Rothfuss, Tim Sa, Daniel Sinto, Nerayo Teclemariam,
Gary Templet, Christine Yang, Lynn Yang, Ann Yoshimura
Sandia National Laboratories
Livermore, CA
{zheath, afbagwe, kguzman, hawley, kmmahro, snmuell,
tplante, arothfu, tjsa, dsinto, nptecle
gjtempl, clyang, liyang, asyosh}@sandia.gov

Jalal Mapar, Justin Legary
Science and Technology Directorate
Department of Homeland Security
jalal.mapar@dhs.gov,
Justin.Legary@associates.dhs.gov

ABSTRACT

The Integrated Modeling, Mapping, and Simulation (IMMS) effort is developing a prototype simulation environment for connecting and configuring modeling and simulation tools that together enable the emergency preparedness community to more effectively, economically, and rapidly prepare, analyze, train, and respond to real and potential incidents. This effort is funded by the Infrastructure Protection and Disaster Management Division (IDD) of the DHS Science and Technology (S&T) directorate. The principal component of IMMS is the Standard Unified Modeling, Mapping and Integration Toolkit (SUMMIT), which provides a standardized mechanism to allow model and data providers to advertise and make their capabilities accessible to the emergency preparedness community. SUMMIT allows emergency preparedness personnel to easily discover, integrate, configure, execute, and view the results of the nation's available simulation capabilities.

SUMMIT provides an *interactive, modeling environment* in which users dynamically configure and execute a collection of models and data sources, continually adding new models and data and linking them in unanticipated ways. It follows that all model configurations and connectitivities cannot be developed *a priori*. The same is true for visualization of the simulation results as SUMMIT's user base consists of a variety of users with different visualization requirements (desktop applications, mobile clients, virtual worlds). Part of SUMMIT's efforts are focused in addressing these user interface and visualization challenges.

We present a data type centered methodology and software implementation to meet the above challenges. Using a configurable system of user interface handlers (UI handlers) and data representation handlers for data types, SUMMIT is able to dynamically generate visual representations for input and output data based on the interfaces of the linked models and input data. Such a system allows users to make use of disparate modeling capabilities across a variety of tools in unanticipated ways while maintaining a consolidated user interface.

ABOUT THE AUTHORS

Zach Heath is a Senior Member of the Technical Staff at Sandia National Laboratories. He has a MS in computer science and 6 years experience developing software systems that make modeling and simulation more accessible to the emergency preparedness community.

Allen Bagwell is a Senior Member of the Technical Staff at Sandia National Laboratories. He has a MS in computer science and 9 years of experience developing software for next-generation satellite ground stations and simulation analysis tools.

Katherine Guzman is a Senior Member of the Technical Staff at Sandia National Laboratories. She has a PhD in mechanical engineering and 5 years experience conducting analysis that supports decision support tool development and enhanced understanding of homeland security issues.

Karim Mahrous is a Senior Member of the Technical Staff at Sandia National Laboratories. He has a Ph.D. in Computer Science and leverages experience in private industry guiding technology creation focusing on the insertion of modeling and simulation capabilities into non-traditional environments.

Stephen Mueller is a Senior Member of the technical staff at Sandia National Laboratories. He has a MS in computer science and 6 years experience developing software for homeland security applications with an emphasis on modeling and simulation and sensor systems.

Marilyn Hawley is a Senior Member of the Technical Staff at Sandia National Laboratories. She has a BS in computer science and 25 years experience developing simulation software with an emphasis on data visualization.

Todd Plantenga is a Principal Member of the Technical Staff at Sandia National Laboratories. He has a PhD in EE/CS and over 15 years industry experience developing scientific computing models and algorithms. His research interests include mathematical optimization, data mining, and distributed computing.

Andrew Rothfuss is a Member of the Technical Staff at Sandia National Laboratories. He has a BS in Physics and 10 years experience developing software for modeling and simulation applications.

Timothy Sa is a Distinguished Technologist at Sandia National Laboratories. He has over 15 years of experience developing software with an emphasis on user interface design.

Daniel Sinto is a Member of Technical Staff at Sandia National Laboratories. He has a MS in electrical engineering and extensive experience developing web-based applications, agent-based modeling, and data visualization.

Nerayo Teclemariam is a Senior Member of the Technical Staff at Sandia National Laboratories. He has a PhD in chemical engineering and extensive experience developing decision-support tools and modeling and simulation products for the emergency planning and response communities.

Gary Templet is a Senior Member of the technical staff at Sandia National Laboratories where he has 8 years experience developing scientific visualization capabilities. He holds an MS in Mechanical Engineering and is currently pursuing a Ph.D. with a focus in Continuum Mechanics and Applied Math.

Christine Yang is a Principal Member of the Technical Staff at Sandia National Laboratories. She has a MS in mechanical engineering and 15 years experience leading development and deployment of DOE and DHS funded software infrastructures.

Lynn Yang is a Principal Member of the Technical Staff at Sandia National Laboratories. She has ten years of experience leading and supporting system studies on biological and chemical defense for military and civilian programs. She holds a Master's degree in Technology and Public Policy and a Bachelor's degree in Civil Engineering from the Massachusetts Institute of Technology.

Ann Yoshimura is a Principal Member of the Technical Staff at Sandia National Laboratories. She has a PhD in chemical engineering and over 20 years experience developing simulations and visual analytic tools.

Jalal Mapar is a Program Manager at the DHS Science and Technology Directorate. He manages a portfolio of S&T programs that provide capabilities for the nation's emergency preparedness and response community. His portfolio primarily focuses on technologies for First Responders: location and health monitoring and Integrated Modeling, Mapping, and Simulation, Training and Exercise, and Incident Management Enterprise systems. Mr. Mapar holds a B.S. and M.S. in Aerospace Engineering from the University of Texas at Austin.

Justin Legary is the S&T Program Manager for Schafer Corporation, directly supporting the DHS Science & Technology Directorate. His current project areas include complex information management architectures, sensor tracking and location systems, modeling & simulation capabilities and cyber security programs related to preparedness and response of the nation's first responders. Mr. Legary is a certified Project Management Professional and holds a bachelor's degree in computer science from the University of Colorado at Boulder.

Automated User Interface Generation for the Integrated Modeling, Mapping, and Simulation (IMMS) Framework

Zach Heath, Allen Bagwell, Katherine Guzman,
Marilyn Hawley, Karim Mahrous, Stephen Mueller, Todd Plantenga,
Andrew Rothfuss, Tim Sa, Daniel Sinto, Nerayo Teclemariam,
Gary Templet, Christine Yang, Lynn Yang, Ann Yoshimura
Sandia National Laboratories
Livermore, CA
{zheath, afbagwe, kguzman, hawley, kmmahro, snmuell,
tplante, arothfu, tjsa, dsinto, nptecle
gjtempl, clyang, liyang, asyosh}@sandia.gov

Jalal Mapar, Justin Legary
Science and Technology Directorate
Department of Homeland Security
jalal.mapar@dhs.gov,
Justin.Legary@associates.dhs.gov

INTRODUCTION

Homeland Security Presidential Directive/HSPD-8 (Bush, 2003) and the subsequent Presidential Policy Directive/PPD-8 (Obama, 2011) identify the need for federal, state, local, and tribal emergency managers and responders to be able to take effective action during a broad set of natural and man-made catastrophic incidents. As these events can have tremendous impact on life, capabilities, and property, the emergency preparedness community relies on pre-event planning and exercises to practice and develop knowledge and understanding of mitigation and response strategies, and potential "what if" trade-offs that might be considered during an event.

Such planning and exercises require the extensive use of scientifically-grounded, reusable quantitative data that can be provided by current and future modeling and simulation tools. Unfortunately, the emergency preparedness community does not have a mechanism for easily identifying and running these tools, nor do they have a way to consistently manage and view the resultant data. Tool developers also lack an adequate mechanism for making their capabilities available and compatible to the emergency preparedness community. To help the emergency preparedness community more effectively, economically, and rapidly prepare, analyze, train, and respond to real or potential incidents, the Integrated Modeling, Mapping, and Simulation (IMMS) effort is designing and prototyping an environment for unifying existing and future data sources and modeling and simulation tools. Using the principal IMMS component—the Standard Unified Modeling, Mapping, and Integration Toolkit (SUMMIT)—emergency preparedness personnel can easily discover, integrate, configure, execute, and view the results of the nation's available simulation capabilities. SUMMIT provides a standardized mechanism that allows model and data providers to advertise and make their capabilities accessible to the emergency preparedness community. This effort is funded by the Infrastructure Protection and Disaster Management Division (IDD) of the DHS Science and Technology (S&T) directorate.

SUMMIT provides a platform-neutral architecture with components that allow interconnected clients to discover available data and modeling tools, configure and execute models, and find, share, and view modeling and simulation results. Further details on the purpose, design, and workings of this architecture can be found in (Friedman-Hill, 2010) and (Plantenga, 2010).

SUMMIT DESCRIPTION

SUMMIT users are able to configure a variety of published model inputs and view their results. Enabling this flexibility presents unique user interface challenges. Since the system makes use of and will continually add new model capabilities created by a variety of external contributors and since users will link these models in unanticipated ways, the number and configuration of possible consolidated input and output displays will grow exponentially and cannot be specified *a priori*. Also adding to the complexity, SUMMIT is being made available to a variety of users with different levels of modeling expertise (novice to advanced) and visualization requirements (desktop applications, mobile clients, immersive virtual worlds).

The primary goal of SUMMIT is to bring science-based modeling and simulation to the emergency preparedness community. Here we describe a data type centered methodology and software implementation to meet the above challenges. Using a configurable system of data type user interface handlers (UI handlers) and data representation handlers, SUMMIT enables interconnected clients to dynamically generate standardized input data types and then output standardized visual representations based on the interfaces of the interconnected models, input data, user requirements, and client capabilities. It is this novel management of data types that allows users to make use of disparate modeling capabilities from a variety of tools in unanticipated ways while maintaining a consistent, consolidated user interface.

This paper presents three different user interface clients to illustrate user interface generation. First, the Browser Based Client (BBC) is a lightweight ext JavaScript (Ext JS, 2011) based client that provides a minimal install option for accessing SUMMIT. The BBC makes use of Google Earth (Google Earth, 2011) for displaying Graphical Information System (GIS) data. Second, the RCP Client is a desktop application built using the Eclipse Rich Client Platform (RCP, 2011) and allows functionality not permitted by a browser based tool. Finally, the 3D Graphical Information System (GIS) client allows for a detailed, immersive, location based view of SUMMIT data.

BACKGROUND AND RELATED WORK

Much prior work has gone into developing modeling and simulation tools and integration frameworks. The Department of Defense's initiated High-Level Architecture (HLA) provides a widely used architecture for integrating discrete event simulation models (Kuhl, 1999). The Dynamic Information Architecture System (DIAS) is a simulation framework developed by Argonne National Laboratory (Simunich, 2005). The Fast Analysis Infrastructure Tool (FAIT) integrates a set of infrastructure models for scenario analysis (FAIT, 2010).

The concept of integrating data from disparate sources is not a new one. There are multiple Geospatial Information System (GIS) visualization tools that integrate GIS data from multiple sources, such as Google Earth (Google Earth, 2011), the DHS iCAV (iCAV, 2010) and DoD *Palenterra* (Beaulieu, 2004).

In addition, previous work describes dynamic user interface generation. For example, there are multiple user-interface definition languages, such as XForms (XForm, 2011) and XUL (XUL, 2011). These languages allow UIs to be specified independently of the platforms and programming languages they will be used with. There are also tools that automatically generate UIs. For example, OpenXava (OpenXava, 2011) is a Java-based tool that automatically generates web interfaces from Java classes and Metawidget (Metawidget, 2011) which generates native UI widgets automatically from a machine service description.

A goal of SUMMIT is to provide an architecture that allows for the reuse of existing simulation frameworks such as HLA, in part by providing access to key utility services that make integrating with these simulation frameworks much easier. By separating the data type definition from model integration and visualization SUMMIT brings the MVC (model-view-controller) design pattern (Burbeck, 1987) to modeling and simulation. SUMMIT allows one type of user to define data types in an abstract but usable manner while model developers can pull these definitions from the server to write model interfaces, and UI designers can develop components for customized visualization for each data type. To that end, a key contribution of SUMMIT is its novel architecture, which allows for dynamically generating standardized UIs. This is noteworthy as the simulations are composed of disparate models and data sources. SUMMIT's architecture is extensible to new UIs as new models, data types, and display features are introduced.

SIMULATION TEMPLATES AND DATA TYPES

The *simulation template* is the primary interface for SUMMIT users to access its modeling and simulation capabilities. A *template* is an abstract representation of an incident to be simulated. It provides a pattern or *workflow* for *federating* (linking together and rendering interoperable) models and data to address that incident.

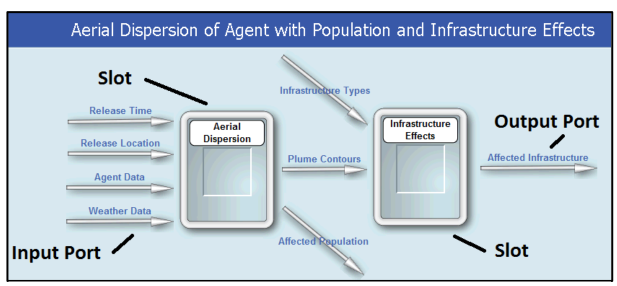


Figure 1. SUMMIT Simulation Template

Each template contains one or more *slots*, which are abstract placeholders for models or data sources needed to simulate the incident. A slot is composed of a set of input and output ports, with each port assigned a data type definition. A template connects a set of slots through their input and output ports in a *directed acyclical graph*.

A data type specifies the structure and type of content in messages passed between SUMMIT models and tools. SUMMIT allows for an evolving and community-driven definition of these data types and uses existing standards where possible. Data types are currently defined using the Google Protocol Buffer language (Google Protocol Buffer, 2010). However, SUMMIT's data type definition language is transitioning to XML Schema (Fallside, 2004) to allow a better integration with the National Information Exchange Model (NIEM, 2010).

A *model wrapper* connects the model or data source with a slot interface that allows the SUMMIT server to communicate with the model. This communication includes delivery of model inputs, execution and halting of the model by the server, logging and progress reporting by the model, and delivering model results to the server.

CONFIGURATION DISPLAY GENERATION

The SUMMIT server maintains a repository of simulation templates. This repository contains the structure of the template, including data slots, as well as an image representing this structure. Both structure and image are created using the SUMMIT software development kit (SUMMIT SDK), which will be described in a future paper. SUMMIT clients can use the pre-generated template image as part of an interface which allows users to associate each abstract slot with a compatible model or data source. As with templates, SUMMIT provides both metadata and an image representation for each model wrapper. These may be used in visualization or user interfaces.

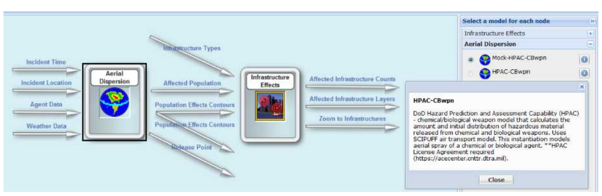


Figure 2. Template with Selected Models

Once a user populates all slots with models, they are presented with an input configuration screen. The SUMMIT clients provide a list of template level input ports associated with a template. The port information contains the name, description, and data type for that port. SUMMIT clients provide their users with a mechanism for specifying input for each port. Next, the populated and configured template is executed by the system.

The input configuration screen is composed dynamically by the SUMMIT system based on data type definitions. The SUMMIT SDK allows users to submit input interfaces (called UI Handlers) for each data type. This allows reuse of data type user interfaces within a template. When building the input user interface for a template, the client queries the SUMMIT server for user interfaces for the given data types. The client then composes a consolidated display with displays for each data type. Users can then use each display to set the inputs as necessary. User interface code running on the client can validate user inputs before sending them to the server along with a request to execute the template.

Currently SUMMIT supports configuration UI Handlers for two clients. For the RCP client, users submit Standard Widget Toolkit (SWT) code (SWT, 2011). This code must implement the configuration handler interface to be compatible with the RCP client. The interface allows the RCP client to verify input values for the given data type. Also, the RCP UI handler interface has an event notification capability that allows a UI handler to listen for updates published by other UI handlers. This is useful in correlating the display of related data inputs. An example is location and time; changing the location may require a change in the displayed time zone.

```
interface DataTypeConfigurationHandler {

//Add a gui for the given port/datatype
//to the configuration ui
void addDatatypeConfigurationUI(Port port,
TemplateConfigurationManager manager);

//Get the configuration data object
//produced by this ui
Object getConfiguredData();

//Display the input object on the ui
void loadConfiguredData(Object obj);

//Is the ui filled in by the user and
//valid
isConfigured();

//Set hanlder's ui elements so they cannot
//be changed by the user
void setReadOnly(boolean readOnly);

//clean up after no longer needed
void cleanUp();
}
```

Figure 3. RCP Configuration Handler Interface

For the BBC client, users submit ext JS/html-based UI Handlers (Ext JS, 2011). This is done by submitting to the server via the SDK one or more files containing custom JavaScript/HTML allowing users to create instances of the data type.

Currently, the BBC does not have an event notification mechanism to allow input ports to communicate with each other.

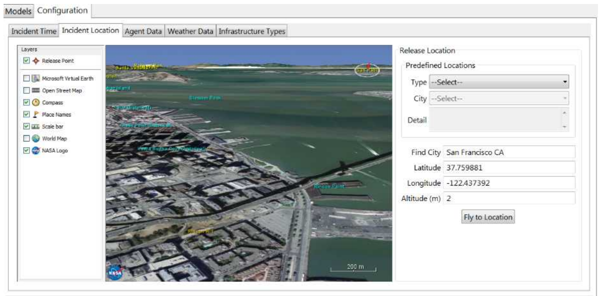


Figure 4. Generated RCP Configuration Screen

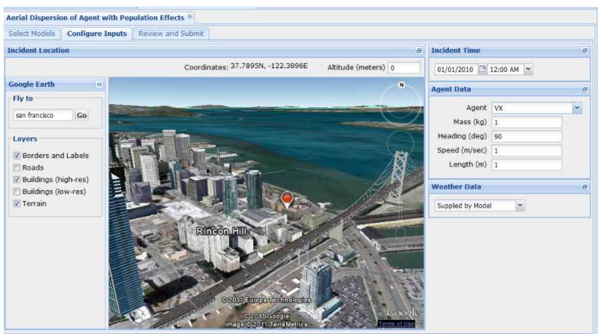


Figure 5. Generated BBC Configuration Screen

Simple UI Handlers for a data type can also be automatically generated and displayed in the RCP and BBC clients by examining the Google Protocol Buffer (GPB) definition. An algorithm is employed to incrementally build a composite UI element by adding a labeled UI element for each field in the data type. A many–to-one mapping exists between simple field types and known UI elements. For example, a text box can be used to configure a field representing a string or number, check boxes can be used for Boolean types, and combo boxes can be used for enumerations. The algorithm is applied recursively when a field of complex type such as a GPB Message or a repeated (list) field is reached. This method of producing a UI Handler is only adequate for simple data types as it does not allow for much user help or control complexity. As SUMMIT transitions to the use of XML Schema for data type definition, additional features will be added to streamline the process for automatic generation of more complex data types. This will allow data type designers to suggest the type of controls for various fields as well as include useful tips and information that should be displayed to the user as they configure the fields.

RESULT DISPLAY GENERATION

When a template is configured and submitted for execution, the server creates and executes a new *template run* with the configured models and inputs. After the template run is complete, the results are displayed to the user. Similar to how configuration UI handlers are used for generating configuration displays, the RCP employs UI handlers for results data types to create the results display. When building the results display, the RCP client queries the SUMMIT server for result UI handlers. Just like configuration UI handlers, users submit result UI handlers for data types through SWT packages for the RCP. The result handlers can be reused anytime a data type appears within another template.

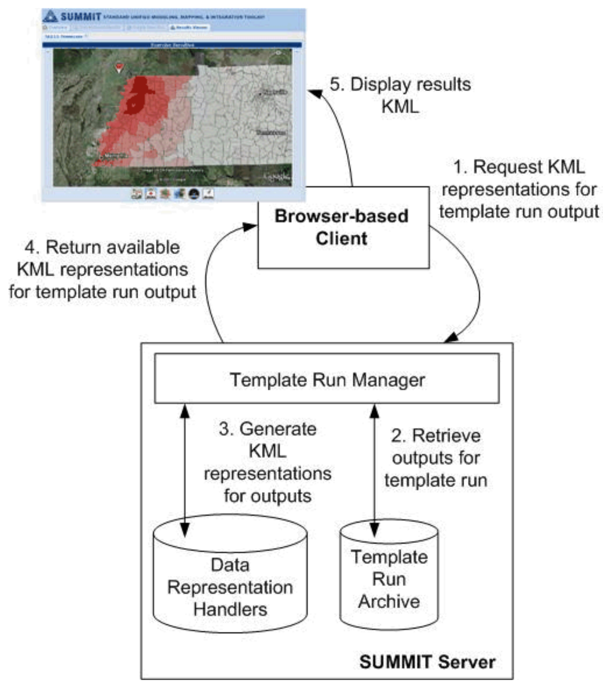


Figure 6. Data representation handler mechanism

For results display, the BBC uses *data representation handlers* which provide a flexible and extensible way for displaying results. The BBC runs a data representation handler to convert raw output data to a format amenable for a particular display. A data type may have multiple representation handlers, which allows results to be viewed in multiple formats and on multiple displays. Current generic output formats that are supported in SUMMIT include KML and chart/table; planned output types include HTML5, SVG, image, and PDF. Much like configuration and result UI handlers, users will be able to submit custom data representation handlers and choose the representation handler most appropriate to their context.

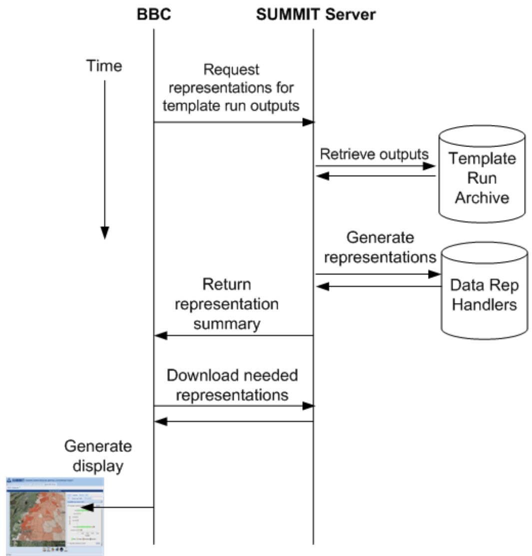


Figure 7. Data representation handler sequence diagram

A SUMMIT client can query the server for a summary of data representations available for a given template run. The server returns a list of metadata of the available representations, including the name, description, MIME-type, and a link to the actual content. Clients can request all representations or representations of a specific type. This gives clients the ability to offer the user a choice of representations for viewing or download. For example, the BBC supports the display of KML files through the Google Earth plug-in. Therefore, when the BBC displays the results for a template run, for each output it makes a server request for all data representations with MIME-type of KML. The SUMMIT server also has a simple GIS feature database that contains geography information for regions of interest for use with representation handlers that need geography data. For instance, a KML representation handler that needs to display county boundaries within a state can query the GIS database for this information.

```
DataRepresentationSummary {
      String handlerId;
      String name;
      String description;
      String mimeType;
      long size;
      String iconURL;
      String dataURL;
      String dataContent;
}
```

Figure 8. Data representation metadata returned by server

Since data conversion can potentially be an expensive and time consuming operation for large datasets, data representations are generated on-demand by the server. When a client requests a specific data representation for a template run, the server first checks if the representation has already been generated. If not, it runs the appropriate representation handler to convert the output data. The generated output is cached for reuse so that the handler code is not called for subsequent requests for the same output representation.

The BBC representation handler provides flexibility in displaying results through custom view types. Different view types may hide certain outputs from the user, or use different representation handlers. For instance, there may be an "expert" view, which shows all the data outputs, whereas an "exercise x" view would hide certain outputs not applicable for the exercise.

As part of results display, the BBC and RCP clients also display model attribution information, sensitivity level, and additional caveats associated with a template run. The model attribution information includes the names, descriptions, and icons of all models and data sources used in the template run. The sensitivity level for a template run is equal to the highest sensitivity level among the models used in the run. For instance, if a template run uses three models, two of which have a sensitivity level of "None", but one model is "Official Use Only", then the template run sensitivity would be "Official Use Only". Caveats for a template run may include guidance on how results should or should not be utilized. When calculating the caveats for a template run, the caveats for all the models used in the template run are aggregated together.

NLE 2011 Use Case

SUMMIT was used to support the National Level Exercise 2011 (NLE 11), organized by the Federal Emergency Management Agency (FEMA) and held in May 2011 (NLE, 2011). The scenario exercised was a catastrophic earthquake in the New Madrid Seismic Zone. SUMMIT supported the exercise by providing simulated ground truth data, including casualty counts, infrastructure damage, and hospital supply usage. Earthquake effects were modeled using FEMA's Hazus model (Hazus, 2011). Hospital supply usage was modeled using the Agency for Healthcare Research and Quality (AHRQ) hospital surge model (AHRQ, 2011). The template used in NLE 11 contained outputs for casualty counts by census tract and casualty distributions by hospital, among several other outputs.

The data type format for these outputs were XML. Figure 9 shows a sample of the casualty counts output XML file. In order to display the casualty counts and casualty distribution data to the user, the BBC used the data representation handler mechanism described in the previous section. There were four representation handlers used for the NLE 11 data:

- *KML representation handler* – Converts casualty counts XML file to multiple KML files, color-coded by number of casualties for each census tract. Each KML file represents a different casualty severity level. Uses the GIS database to obtain the geometry for each census tract.
- *Chart representation handler* – Reads-in casualty counts XML file and produces a chart with total casualties for each severity level.
- *Table representation handler* – Reads-in casualty counts XML file and produces a table showing total casualty counts for each county.
- *Graph representation handler* – Reads-in casualty counts distribution XML file and produces graphs containing cumulative patient arrivals by hour for each hospital.

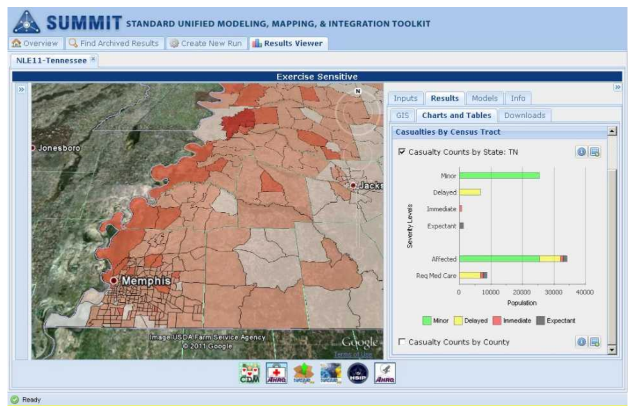
The charts, tables, and graphs produced by the representation handlers are defined using an XML description language that the BBC interprets for display.

When displaying one of the NLE 11 template runs, the BBC requests all output representations for the template run from the server. Upon receiving this request, the SUMMIT server executes the KML, chart, table and graph representation handlers. Once the representation handlers are finished executing, the SUMMIT server returns the data representation summary to the BBC, which includes links to download the resultant KML, chart, and table description files. Figure 10 (a) shows the casualty counts KML and chart displayed in the BBC. Figure 10 (b) shows the casualty counts table. Figure 10 (c) shows the graph depicting patient arrivals by hospital.

Outputs from the BBC were viewed and utilized by exercise controllers throughout NLE 11. While most data requests from the exercise controllers were satisfied using the existing representation handlers, there were requests for output formats not yet supported in SUMMIT. This demonstrates the need for an extensible architecture that allows flexibility to add new output and display capabilities, for which the data representation mechanism is designed.


```
- <CasualtyCounts>
  - <CasualtyCount>
      <CensusTract>47003950300</CensusTract>
    - <Location>
        <Latitude>35.455402</Latitude>
        <Longitude>-86.585105</Longitude>
      </Location>
      <Population>3195</Population>
    - <SeverityCount>
        <Type>Sev1</Type>
        <Probability>8.87202186E-05</Probability>
        <Count>0.283461094</Count>
      </SeverityCount>
  ⋮
```


Figure 9. Sample casualty counts XML output from Hazus earthquake effects model



SUMMIT STANDARD UNIFIED MODELING, MAPPING, & INTEGRATION TOOLKIT

NLE11-Tennessee

Exercise Sensitive

Casualties By Census Tract

Casualty Counts by County

County	Minor	Delayed	Immediate	Expectant	Affected	Req Med Care
Shelby	15933	4271	449	827	21480	5547
Dyer	1559	430	47	88	2124	565
Gibson	1103	295	32	59	1489	386
Tipton	1130	299	30	55	1514	384
Obion	925	250	27	50	1251	326
Lauderdale	789	218	25	47	1080	290
Madison	783	196	21	40	1039	257
Weakley	702	187	20	37	946	245
Haywood	379	101	10	19	509	130
Lake	303	90	12	23	428	125
Crockett	337	90	10	18	454	117
Fayette	349	90	8	15	462	113
Hardeman	235	60	6	11	312	77
Chester	127	33	4	7	170	44
Carroll	149	33	3	6	191	42
Hardin	118	32	3	5	158	40
Henry	129	30	3	4	167	37
McNairy	95	24	2	4	125	30
Henderson	94	20	2	3	119	25
Benton	19	5	0	1	25	6
Davidson	46	3	0	0	50	3

3D DATA REPRESENATION HANDLERS

A SUMMIT Client for the representation of output data was developed as a plug-in for Geoweb3d (Geoweb3d, 2011), a visualization platform combining GIS standards with a game-quality 3D engine. Geoweb3d allows procedural scene generation using standardized GIS formats, as well as dynamic loading and modification of data through a JavaScript API.

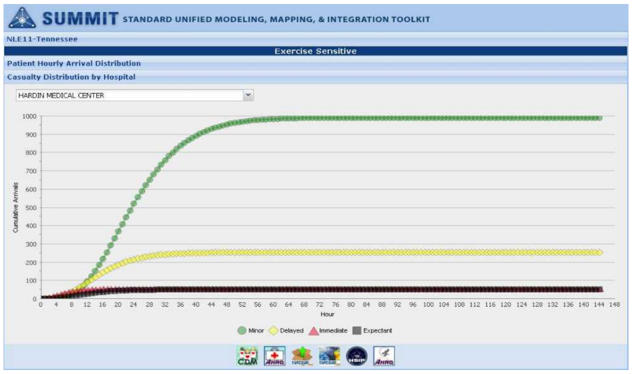


Figure 10. BBC NLE 11 Results Screenshots (a) TOP: KML showing census tracts colored by total casualties with chart showing total casualties; (b) MIDDLE: Table showing total casualties per county; (c) BOTTOM: Graph showing cumulative patient arrivals per hour

A *scene* is defined as a geo-specific location (such as a city, county, or region) that has both static and dynamic components. Static components are those that are not affected by SUMMIT-generated data and include satellite imagery, terrain elevation, and other aesthetic layers meant to make the scene visually appealing (e.g. lampposts, trees, benches, etc.). Dynamic components are data elements that are either generated or modified by SUMMIT data.

Figure 11 is an overview of the architecture used to procedurally generate a scene. A *representation layer* contains indexable geometry that can be modified by a *SUMMIT data set*. The final data representation is created by putting the representation layer and the Data Set through a *representation operation*. A *view* is comprised of several representations (which are all active at the same time). Similar views are grouped together in a *view grouping*. A scene can contain multiple view groupings, with each grouping having at most one view active.

SUMMIT can output a variety of data types and, necessarily, representations can take on many forms and are a versatile construct. The same data set can be used to affect different types of layers: data containing building damage can be used to color building footprints based on the damage amount, or used to deform a 3D model of the building (Figure 12). In the first representation the representation layer would be a set of building footprint polygons and the operation would map the damage output to a color and then apply it to the correct footprint, in the second example the representation layer is a set of 3D building models and the operation would modify the 3D geometry to display the required amount of damage. The same data sets and representation layers can be used in different representations.

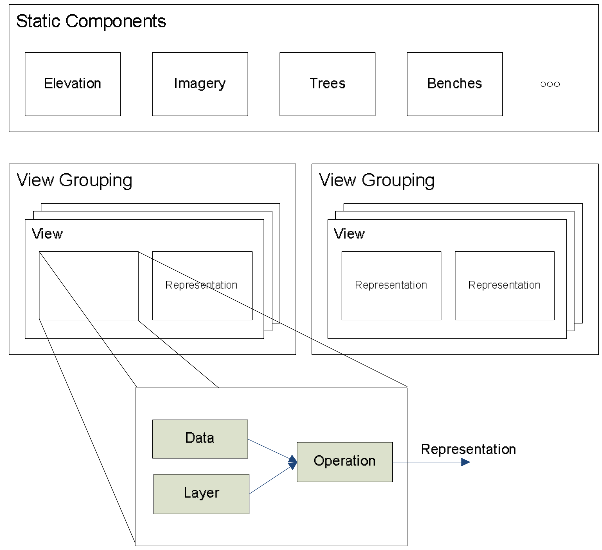


Figure 11. Scene Generation Overview

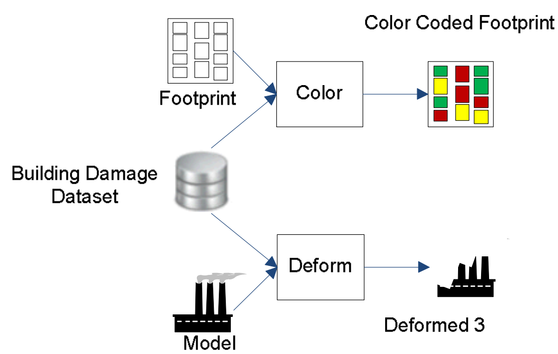


Figure 12. Creating Data Representations

To implement the above concepts in Geoweb3d, a scene manifest was created as a JSON data structure. The manifest contains the static and dynamic assets necessary for generating a scene. Some representations are pre-processed and then imported directly (due to the computationally-intensive nature of the operations), and others are created at run-time. As an illustrative example, SUMMIT simulated an earthquake in the Midwest and was used to create a scene for Jonesboro, AR. The scene had two view groupings: Census Tract and Building Damage (Figure 13). The Census Tract grouping was used to visualize the average and most likely damage in a particular census tract, while the Building Damage view allowed for different visualizations of the building data.

Exercise planners and participants can have varied roles that drive different visualization requirements. A planner or an analyst may find viewing colored polygons more helpful, while a "boots-on-the-ground" player may prefer a photo-realistic view of the surroundings. Three different views from the view grouping above can be seen in Figure 14 for a residential neighborhood.

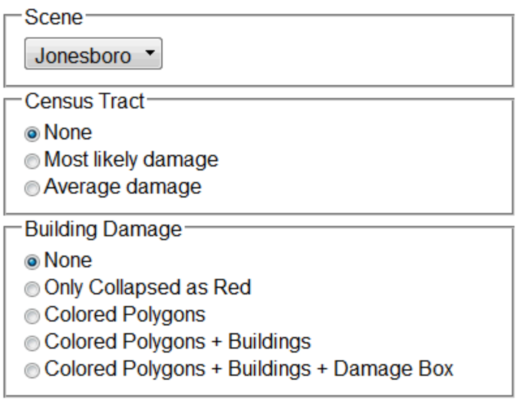
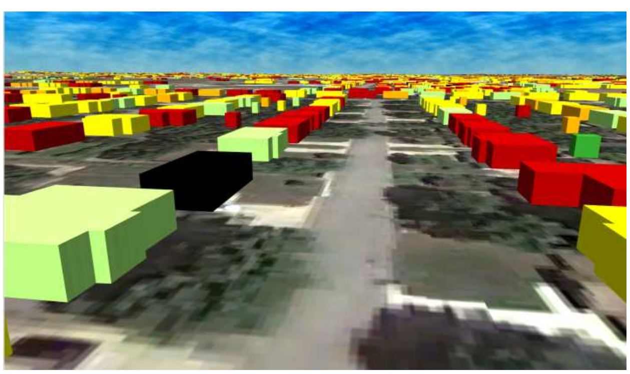


Figure 13. View Groupings for a scene

In Figure 14 (b) and (c), the realistic building damage was created using two representations: 1) deforming a KML model of the building in question; 2) dropping rubble piles at the vertices of the building footprint. The level of damage indicated the amount of formation and the size of the rubble pile needed, respectively. In (c) a third representation was used where the building footprint layer was extruded, rendered opaque, and colored.



SUMMARY and FUTURE WORK

In this paper we provided an overview of SUMMIT's data type centered methodology and software implementation for the automatic generation of user interface displays. Such a methodology allows users to configure and display the results of disparate modeling capabilities across a variety of tools in unanticipated ways while maintaining a consolidated user interface and maximizing software reuse. This methodology also provides flexibility allowing the system to evolve as new requirements and display capabilities arise. Providing this consolidation and adaptability is vital in reaching the goal of bringing science based modeling and simulation to the emergency planning and management community.

Figure 14. Three different views from a grouping: (a) TOP: extruded polygons; (b) MIDDLE: realistic building damage; (c) BOTTOM: realistic damage with damage boxes

The SUMMIT reference implementation currently under development by Sandia National Laboratories and a consortium of other parties follows an iterative, spiral development process. As development progresses the product is used at various national and regional emergency management training exercise and made available to model and data source providers. This tests the tool and generates additional system requirements.

Current planned architectural extensions to SUMMIT's user interface include ability to configure and display time varying data encountered in distributed discrete event simulations as well as human-in-the-loop simulations. Another extension is providing data type definitions based on XML Schema. This will allow

SUMMIT to better integrate with the National Information Exchange Model (NIEM, 2010) and will affect how data type designers configure the automatic generation of complex, well documented, user interfaces for specifying and displaying data. To this end, we are investigating a generic UI description language interpretable by a variety of clients. The UI description could be provided in addition to or as part of the data type schema. Additions are also being made to the data representation handler interface. New representation types are in development as well as the ability to allow clients to customize template results to allow, for example, custom coloring, binning, and filtering based on user-specified parameters. A more ambitious addition being looked into is to add the ability to display native user interfaces provided by the modeling and data sources in the SUMMIT federation.

The SUMMIT framework and project information is hosted at https://dhs-summit.us.

ACKNOWLEDGEMENTS

This work was funded by the Infrastructure Protection and Disaster Management Division (IDD) of the Department of Homeland Security (DHS) Science and Technology (S&T) directorate.

REFERENCES

Agency for Healthcare Research and Quality (AHRQ) Hospital Surge Model (2011). Retrieved June 2011 from http://hospitalsurgemodel.ahrq.gov.

Beaulieu, B.R. (2004). "Security through The Palanterrs", *GeoIntelligence*, Aug 2004.

Burbeck, S. (1987), Applications Programming in Smalltalk-80: How to use Model-View-Controller, http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html

Bush, G.W. (2003), "Homeland Security Presidential Directive / HSPD-8". Retrieved June 2011 from http://www.fas.org/irp/offdocs/nspd/hspd-8.html.

Ext JS (2011). Retrieved June 2011 from http://www.sencha.com/products/extjs.

FAIT: Fast Analysis Infrastructure Tool (2011). Retrieved June 2011 from http://sandia.gov/nisac/fait.html.

D. C. Fallside (Eds) (2004). ``XML Schema Part 0: Primer'', W3C Recommendation, http://www.w3.org/TR/xmlschema-0.

Friedman-Hill, E. Plantenga, T., & Ammerlahn, H. (2010). "Simulation Templates in the SUMMIT System", in *2010 SISO Spring Interoperability Workshop*, Orlando, FL, Apr 2010.

Geoweb3d (2011) Retrieved June 2011 from http://www.geoweb3d.com.

Google Earth (2011). Retrieved June 2011 from http://www.google.com/earth/index.html

Google Protocol Buffers (2010). Retrieved May 2010 from http://code.google.com/apis/protocolbuffers.

Hazus (2011). Retrieved June 2011 from http://www.fema.gov/plan/prevent/hazus.

iCAV: Integrated Common Analytical Viewer (2011). Retrieved June 2011 from https://icav.dhs.gov.

Kuhl, F., Weatherly, R., & Dahman, J. (1999). Creating Computer Simulation Systems: An Introduction to the High Level Architecture, Upper Saddle River: Prentice-Hall.

Mapar, J., T. Hoette, K. Mahrous, C. Pancerella, T. Plantenga, C. Yang, L. Yang, M. Hopmeier (2010), "Integrated Modeling, Mapping, and Simulation (IMMS) Framework for Exercise and Response Planning", Conference Paper, MODSIM World Conference, Hampton, VA, October 2010.

Metawidget (2011). Retrieved June 2011 from http://metawidget.org.

NIEM: National Information Exchange Model (2010). Retrieved May 2010 from http://www.niem.gov.

NLE 2011: National Level Exercise (2011). Retrieved June 2011 from http://www.fema.gov/media/fact_sheets/nle2011_fs.shtm.

Obama, B. (2011), "Presidential Policy Directive/PPD-8: National Preparedness". Retrieved June 2011 from http://www.dhs.gov/xabout/laws/gc_1215444247124.shtm.

OpenXava (2011). Retrieved June 2011 from http://www.openxava.org.

Plantenga, T., Friedman-Hill, E. (2010). "Integrated Modeling, Mapping, and Simulation (IMMS) Framework for Planning Exercises", in 2010 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), Orlando, FL, Nov 2010.

RCP: Rich Client Platform for Eclipse (2011). Retrieved June 2011 from http://wiki.eclipse.org/index.php/Rich_Client_Platform.

SWT: Standard Widget Toolkit (2011). Retrieved June 2011 from http://www.eclipse.org/swt

Simunich, K.L. (2005). "Dynamic Information Architecture System (DIAS): Developer's Guide", Argonne National Laboratory, ANL/DIS-0501, February 2005.

UICDS: Unified Incident Command & Decision Support (2011). Retrieved June 2011 from http://www.uicds.us.

XForms (2011). Retrieved June 2011 from http://www.w3.org/TR/xforms.

XUL: XML User Interface Language (2011). Retrieved June 2011 from https://developer.mozilla.org/en/xul.