

Automated User Interface Generation for Dynamically Integrated Simulations in the SUMMIT Framework

Zach Heath, Ann Hammer, Charles John,
 Karim Mahrous, Steven Mier,
 Stephen Mueller, Todd Plantenga, Andrew Rothfuss,
 Nerayo Teclemariam, Lynn Yang, Ann Yoshimura
 Sandia National Laboratories
 Livermore, CA
 {ahammer, zheath, cjohn, kmmahro, samier,
 snmuell,tplante, arothfu, nptecle,
 liyang, asyosh}@sandia.gov

Jalal Mapar
 Science and Technology Directorate
 Department of Homeland Security
 Washington DC

ABSTRACT

SUMMIT is a Department of Homeland Security Science and Technology Directorate-sponsored software environment for connecting simulation tools that enable the emergency preparedness community to more effectively and economically prepare for and respond to real and potential incidents. SUMMIT allows users to dynamically link, configure and execute a library of models and data sources. All integrated model configuration and result displays cannot be developed *a priori* as the library of data types and available models continues to expand and because SUMMIT's users have different visualization requirements (desktop applications, mobile clients, virtual worlds). We present a data type-centered methodology, and its use in national and regional level exercises, to meet the above challenges. Using a configurable system of data representations, SUMMIT is able to dynamically generate visual representations for data based on the interfaces of the linked models. This approach 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 Principal Member of the Technical Staff at Sandia National Laboratories. He has a MS in computer science and 9 years experience developing software systems that make modeling and simulation more accessible to the emergency preparedness community.

Karim Mahrous is a Manager 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 10 years experience developing software for homeland security applications with an emphasis on modeling and simulation and sensor systems.

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.

Jalal Mapar is the Director of the Resilient Systems Division (RSD) at the DHS Science and Technology (S&T) directorate. His division's mission is to rapidly develop and deliver innovative solutions that enhance the resilience of individuals, communities, and systems by enabling the Whole Community to prevent and protect against threats, mitigate hazards, effectively respond to disasters, and expedite recovery. He manages a portfolio of S&T programs that provide capabilities for the nation's emergency preparedness and response community. Mr. Mapar holds a B.S. and M.S. in Aerospace Engineering from the University of Texas at Austin.

Automated User Interface Generation for Dynamically Integrated Simulations in the SUMMIT Framework

Ann Hammer, Zach Heath, Charles John,

Karim Mahrous, Steven Mier,

Stephen Mueller, Todd Plantenga, Andrew Rothfuss,

Nerayo Teclémariam, Lynn Yang, Ann Yoshimura

Sandia National Laboratories

Livermore, CA

{ahammer, zheath, cjjohn, kmmahro, samier,

snmuell, tplant, arothfu, nptecle,

liyang, asyosh}@sandia.gov

Jalal Mapar

Science and Technology Directorate

Department of Homeland Security

Washington DC

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 Resilient Systems Division (RSD) 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 based 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 two different user interface clients to illustrate user interface generation. First, the Browser Based Client (BBC) is a lightweight HTML/ext JavaScript (Ext JS, 2014) based client that provides a minimal install option for accessing SUMMIT. The BBC makes use of Google Earth (Google Earth, 2014) for displaying Graphical Information System (GIS) data. Second, 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, 2014), 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.

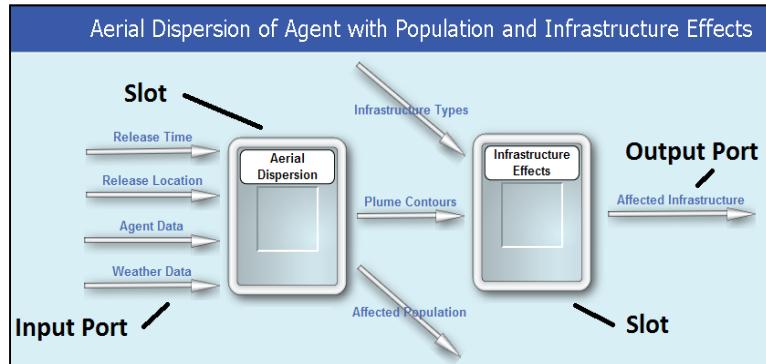


Figure 1 SUMMIT Simulation Template

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 either XML Schema (Fallside, 2004) or the Google Protocol Buffer language (Google Protocol Buffer, 2014). XML Schema is the preferred data type definition language to allow a better integration with the United States government's evolving National Information Exchange Model (NIEM, 2014).

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.

A *template run* represents the instantiation of a simulation template by a user. It organizes the evolving set of selected model wrappers, data sensitivities and assumptions, template level inputs, template level outputs, analysis notes, and generated data representations.

CONFIGURATION DISPLAY GENERATION

The SUMMIT server maintains a repository of simulation templates. This repository contains the structure of the template (connections between input ports, slots, and output ports), an image file (jpg) representing this structure, as well as information on the pixel location and sizes of elements within the image. Both structure and image are created using the SUMMIT software development kit (SUMMIT SDK), which will be described in a future paper. As with templates, SUMMIT provides both metadata and an image representation for each model wrapper. SUMMIT clients can use the pre-generated template and model wrapper images as part of an interface customized for the client type which allows users to associate each abstract slot with a compatible model or data source.

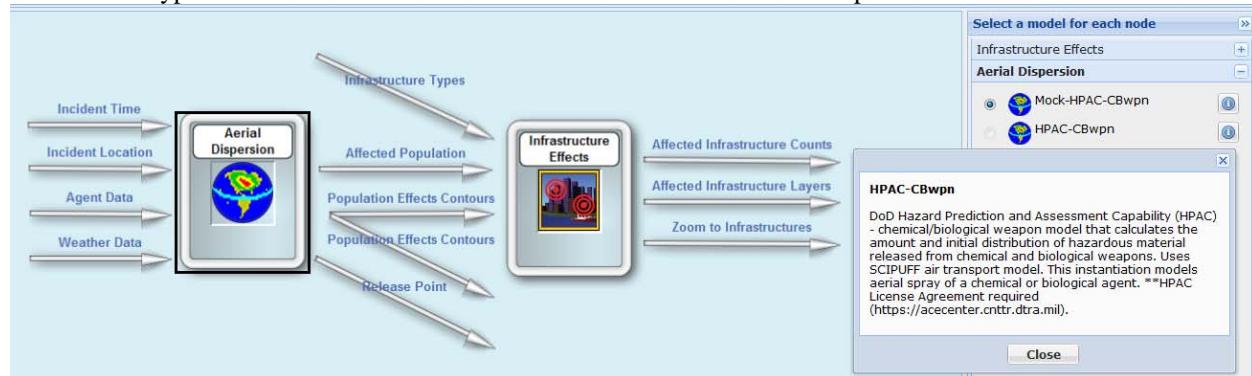


Figure 2 Template with Selected Models

Once a user populates all slots with models, they are presented with an input configuration screen (Figure 3. Generated BBC Configuration Screen). The SUMMIT server provides clients with 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 must provide their users with a mechanism for specifying an input value for each port. The input configuration screen is composed dynamically by the SUMMIT framework based on data type definitions of the template input ports. The SUMMIT SDK allows UI designers to submit input interfaces (called data representation handlers) for each data type. This allows for the reuse of data type user interfaces between templates and clients. When building the input user interface for a template, the client queries the SUMMIT server for input user interfaces for the given input port data types and that clients display capabilities. The client then composes a consolidated display with displays for each port data type. Users can then use each display to set the inputs as necessary. The generated 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.

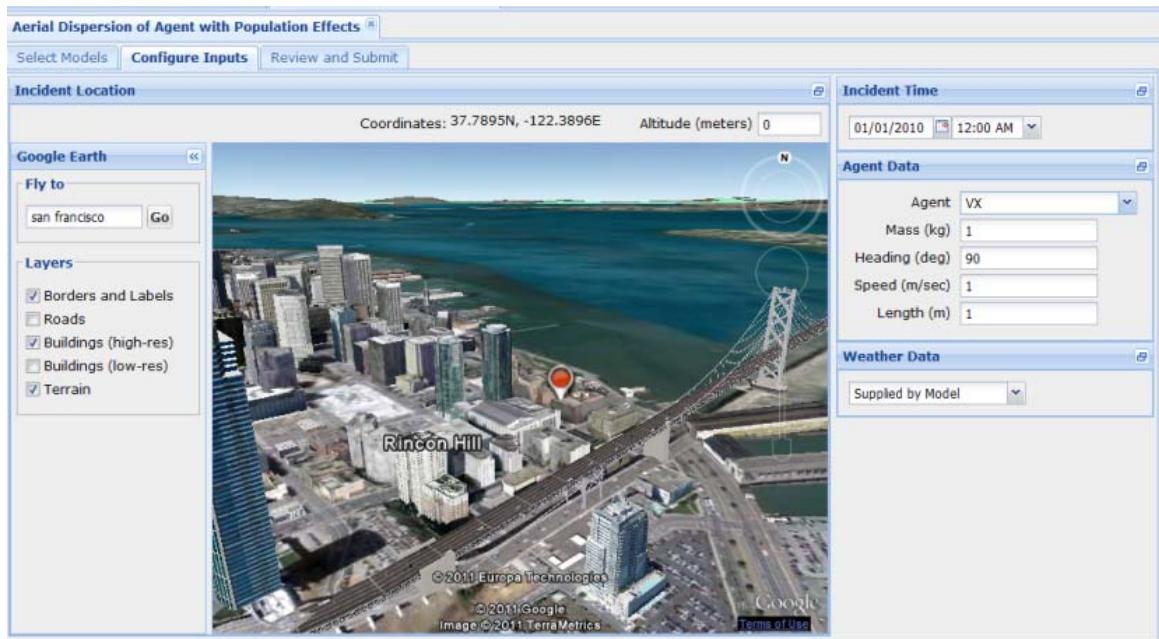


Figure 3. Generated BBC Configuration Screen

For the BBC client, UI designers submit HTML Form based data representation handlers to the SUMMIT server using the SDK. This is done by submitting Java code that is able to generate HTML/JavaScript for the display. A HTML iframe element will be created for each port display and will be filled by this generated HTML. The generated HTML can either allow the user to submit complete data type instances directly to the server, or interpret user inputs and make use of utilities on the server for building up complete data type instances. The SUMMIT framework provides a JavaScript API to make creating these displays easier.

The data representation handler interface has an event notification capability that allows a data representation handler to listen for updates published by other data representation 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.

RESULT DISPLAY GENERATION

When a template run is configured and submitted for execution by the client, the server executes the new template run with the configured models and inputs. Further details on the coordination and execution of models in the SUMMIT framework are discussed in (Friedman-Hill, 2010). After the template run is complete, the results are

displayed to the user. Similar to how configuration data representation handlers are used for generating configuration displays, the BBC employs output data representation handlers for results data types to dynamically create the results display. When building the results display, the BBC client queries the SUMMIT server for result data representation handlers. The BBC runs a data representation handler to convert raw output data to a format amenable for a particular display type (Figure 4). A data type may have multiple data 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:

- Keyhole Markup Language (KML 2014) for Geographic Information System (GIS) based data
- Chart/Table for displaying tables, line charts, and bar charts of data
- HTML5 which allow complete customization of display.
- Comma Separated Value (CSV) so the data can be imported into other analysis tools such as Microsoft Excel
- PDF snippets for generating aggregate PDF reports for a template run that can be distributed to people that do not have access to SUMMIT

Much like configuration data representation handlers, users are able to submit custom data representation handlers through the SUMMIT SDK.

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 (Figure 5). 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. For each returned representation, the BBC will provide a labeled checkbox that when clicked will display the KML representation on the Google Earth display. To make the GIS representation creation process easier, the SUMMIT server 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 data representation handler that needs to display county boundaries within a state can query the GIS database for this information.

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

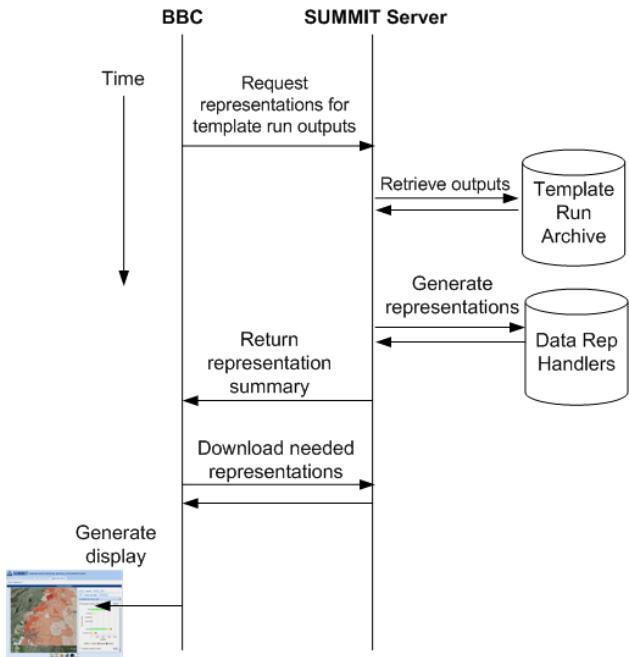


Figure 4. Data representation handler mechanism

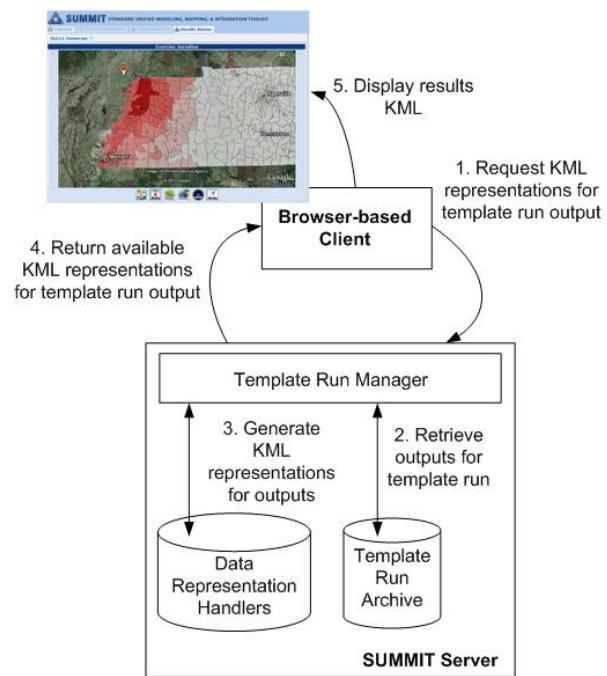


Figure 5. Data representation handler sequence diagram

data 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 data representation handlers provide 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 client also displays 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. Owners of the template run also have the option to increase the sensitivity level of a template run if needed due to the combination or nature of the data.

FEMA NLE 2011 and Regional Exercise Use Cases

SUMMIT has been used extensively to support federal, state, local, tribal and territorial exercises including the recent Blue Surge 2013 and National Level Exercise 2011 (NLE 11), organized by the Federal Emergency Management Agency (FEMA) (NLE, 2011). The NLE 11 scenario exercised was a catastrophic earthquake in the New Madrid Seismic Zone, and Blue Surge was a earthquake/tsunami scenario located in the Caribbean. SUMMIT supported the exercises 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, 2014). Hospital supply usage was modeled using the Agency for Healthcare Research and Quality (AHRQ) hospital surge model (AHRQ, 2011). The template used contained outputs for casualty counts by census tract and casualty distributions by hospital, among several other outputs. The data type format for these outputs was XML Schema. **Error! Reference source not found.** 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 exercise 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.

```

- <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 4. Sample casualty counts XML output from Hazus earthquake effects model

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. **Error! Reference source not found.** (a) shows the casualty counts KML and chart displayed in the BBC. **Error! Reference source not found.** (b) shows the casualty counts table. **Error! Reference source not found.** (c) shows the graph depicting patient arrivals by hospital.

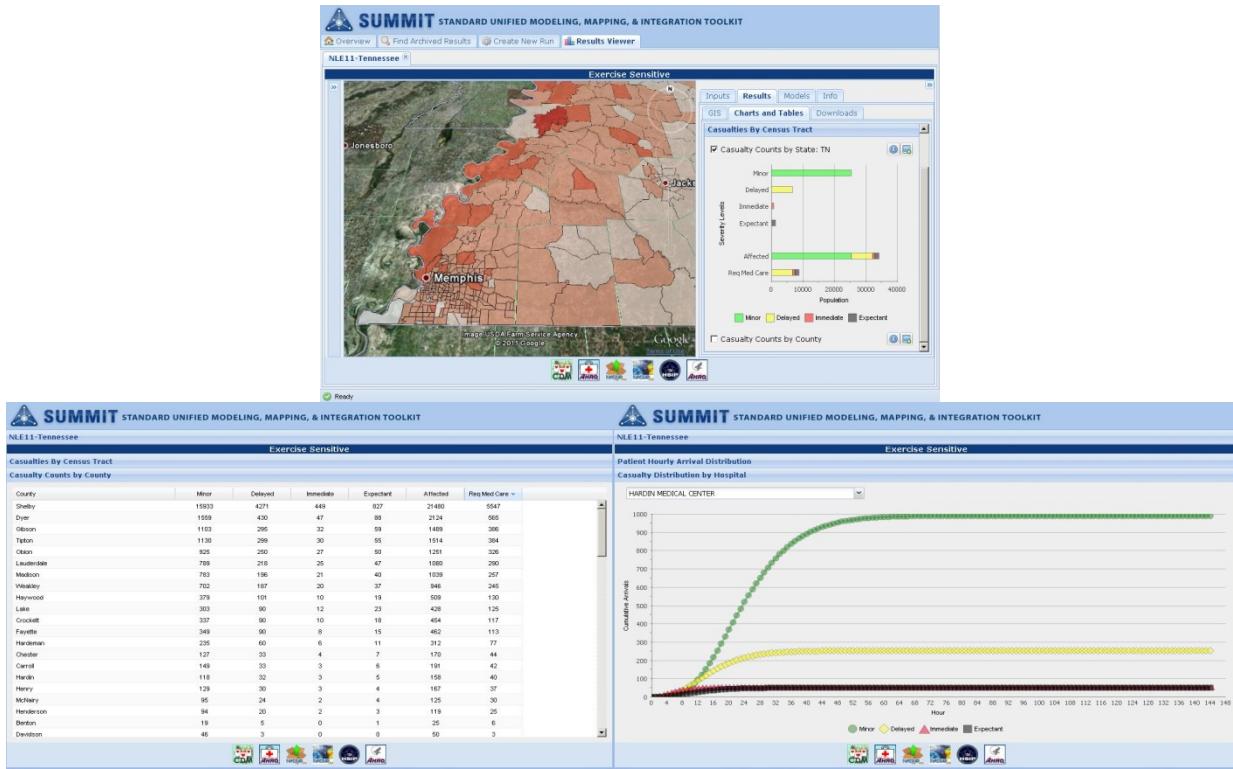


Figure 5. BBC NLE 11 Results Screenshots (a) TOP: KML showing census tracts colored by total casualties with chart showing total casualties; (b) LEFT: Table showing total casualties per county; (c) RIGHT: Graph showing cumulative patient arrivals per hour

Outputs from the BBC were viewed and utilized by exercise controllers throughout Blue Surge and 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.

3D DATA REPRESENTATION HANDLERS

A SUMMIT Client for the representation of output data was developed as a plug-in for Geoweb3d (Geoweb3d, 2014), 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.

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.

Error! Reference source not found. is an overview of the architecture used to procedurally generate a scene. A *representation layer* contains indexed 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 (**Error! Reference source not found.**). 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.

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. 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 for a residential neighborhood.

In Figure (b), 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.

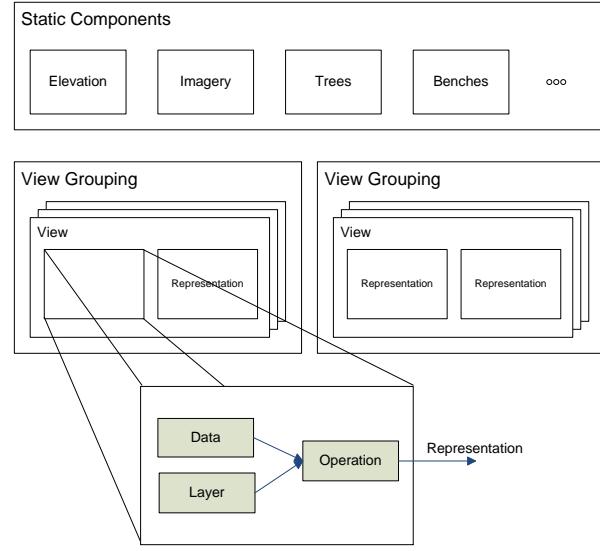


Figure 8. Scene Generation Overview

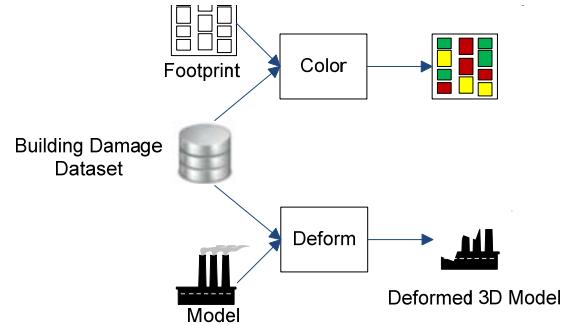


Figure 9.

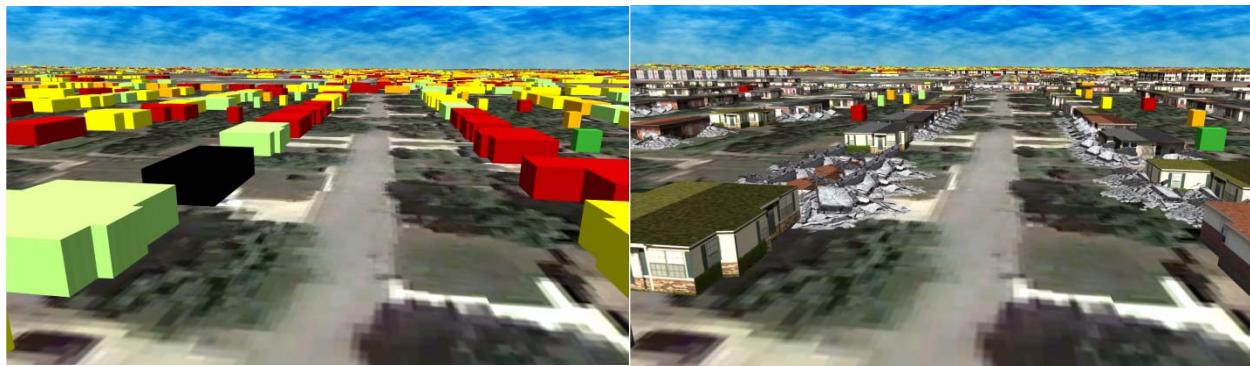


Figure 10. Two different views from a grouping: (a) Left: extruded polygons; (b) Right: realistic building damage

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.

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. 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 and will ease the development of data representations. 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.

Through our participation in training exercises we have identified Geographic Information System (GIS) data as being the most important type of data to be represented. We are evaluating alternate mechanisms for making this data available to clients such as having SUMMIT publish the data through the Web Map Service Interface (WMS)

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

ACKNOWLEDGEMENTS

This work was funded by the Resilient Systems Division (RSD) 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 (2014). Retrieved February 2014 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 (2014) Retrieved February 2014 from <http://www.geoweb3d.com>.

Google Earth (2014). Retrieved February 2014 from <http://www.google.com/earth/index.html>

Google Protocol Buffers (2014). Retrieved February 2014 from <https://developers.google.com/protocol-buffers>.

Hazus (2014). Retrieved February 2014 from <http://www.fema.gov/hazus>.

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

KML (2014) Retrieved February 2014 from <https://developers.google.com/kml/documentation>

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 (2014). Retrieved February 2013 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.

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>.