*This is an Accepted Manuscript of an article published in the Journal of Computing in Civil Engineering in August 2013, available online: DOI: 10.1061/(ASCE)CP.1943-5487.0000364*

# SIMULATION OF INTEGRATED URBAN INFRASTRUCTURE SYSTEMS: A SERVICE-ORIENTED APPROACH

Jonathan L. Goodall [1], Member, ASCE
Anthony M. Castronova [2]
Nathan N. Huynh [3]
Juan M. Caicedo [4], Member, ASCE
Ryan Hall[5]

## ABSTRACT

This paper explores service-oriented architectures as an approach for simulating integrated urban infrastructure as a system-of-systems. Models representing three individual infrastructure systems (water, transportation, and structures) are written as web services so that they can be linked through the exchange of data into an integrated system. An example application is presented where the service-oriented approach is used to simulate an integrated urban infrastructure system in Columbia, South Carolina during a flooding event that causes road closures. Findings of this work are that (i) service-oriented architectures are well suited for urban infrastructure system integration, primarily because of the benefit in handling model heterogeneities including differences in conceptual design and technical implementation, and (ii) it is possible to extend the Open Geospatial Consortium (OGC) Web

---

[1] Assistant Professor, Dept. of Civil and Environmental Engineering, College of Engineering and Computing, Univ. of South Carolina, 300 Main St., Columbia, SC 29208. E-mail: goodall@cec.sc.edu

[2] Postdoctoral Researcher, Dept. of Civil and Environmental Engineering, College of Engineering and Computing, Univ. of South Carolina, 300 Main St., Columbia, SC 29208. E-mail: castrona@cec.sc.edu

[3] Assistant Professor, Dept. of Civil and Environmental Engineering, College of Engineering and Computing, Univ. of South Carolina, 300 Main St., Columbia, SC 29208. E-mail: huynhn@cec.sc.edu

[4] Associate Professor, Dept. of Civil and Environmental Engineering, College of Engineering and Computing, Univ. of South Carolina, 300 Main St., Columbia, SC 29208. E-mail: caicedo@cec.sc.edu

[5] Engineering Associate, Alliance Consulting Engineers, Inc., P.O. Box 8147, Columbia, S.C. 29202-8147, rhall@allianceCE.com

Processing Service (WPS) standard to expose models as web services to achieve a service-oriented modeling system. Results of the example application demonstrate the improvement in alleviating traffic congestion due to flooding by automating the exchange of data between the urban water modeling systems with other components of the civil infrastructure system.

**Keywords:** urban infrastructure; system integration; service-oriented architectures

# INTRODUCTION

Urban infrastructure is often designed as a set of stand-alone systems with little consideration for interactions between physical infrastructure systems and their environments. While system interactions are considered in the design and retrofitting of civil infrastructure, e.g. considering the likelihood of river flow rates when designing a bridge or culvert, they are largely ignored for the majority of the infrastructure life-cycle. While tools like sensing networks and models are being used for real-time and adaptive management within many parts of the overall urban infrastructure system, there has been less work across civil engineering disciplines to connect the various pieces of the urban infrastructure into a holistic system. Such work is needed, however, in order to consider inter-system interactions and dynamics throughout the entire life cycle of civil infrastructure. This cross-disciplinary civil infrastructure integration is the focus of our work.

Flooding in an urban environment is one scenario that illustrates the need for cross-disciplinary civil infrastructure system integration. In water resources, hydrologic and hydraulic models are capable of running in real-time using physical or statistical approaches that offer sufficient accuracy and performance to forecast river levels, velocities, and flow rates throughout a river network system. This information is valuable to other parts of the civil infrastructure system. Transportation models, for example, could use these forecasted flows from the river system to determine how traffic should be rerouted to avoid hazardous roads prior to or during a storm event. Likewise, bridge monitoring systems can use information about forecasted river levels in combination with its own structural monitoring data to improve estimates of the probability of bridge failure during a flooding event due to scour.

There are many other examples, both for real-time and long term management applications, where if civil infrastructure systems were able to seamlessly transfer data and information in an automated way, without the need for human intervention, then the integrated system would be more reliable and serviceable.

While there are clear advantages to having an integrated civil infrastructure system, achieving such a system is challenging. Garrett (2005) summarizes some of the challenges grouped along social, economic, and technological dimensions in the context of Advanced Infrastructure Systems (AIS). Within this context, our focus is primarily on the technical challenges and, even more specifically, overcoming the heterogeneity amongst approaches used to design, model, and manage parts of the civil infrastructure system. We propose the use of a service-oriented architecture to overcome these heterogeneity challenges. The objective of this paper is therefore to apply service-oriented computing concepts for the specific problem of integrating civil infrastructure systems using a system-of-systems approach. In doing so, a key contribution of our work is to address the challenges of integrating modeling methodologies that have been adopted within single infrastructure systems (water, transportation, structures) without imposing a single modeling methodology across all infrastructure subsystems.

## BACKGROUND

Past research has shown that civil infrastructure systems may generate complex and counter-intuitive responses that are undesirable, unpredictable, and compromise the resiliency of the system (Rinaldi et al. 2001). This complexity can cause cascading failures throughout the system in ways that are not immediately intuitive (and therefore predictable) (Amin 2002; Little 2003). Rinaldi et al. (2001) stressed the need to understand the interdependency of infrastructure systems that are connected as a "system-of-systems" with bidirectional relationships between system components that produce complex relationships characterized by feedback and feedforward paths, and intricate branching topologies. These characteristics of infrastructure systems have motivated researchers to argue that future civil engineers should be master integrators with a view that civil infrastructure is a complex, holistic system (Bordogna 1998; Folke 2006), able to understand the complex computer and information technologies including sensing techniques, data models, and data mining capabilities needed to design and maintain Advanced Infrastructure Systems (AIS) (Garrett 2005).

Prior work for integrating civil infrastructure systems has ranged from techniques for simulating system interactions, to techniques for understanding the sustainability of systems, to approaches for fostering communication and integration of systems. Simulation techniques presented in the literature include agent-based modeling (Sanford Bernhardt and McNeil 2008), dynamic programming (Kuhn 2010), and network analysis (Ash and Newth 2007; Tran et al. 2010). Sustainability of civil infrastructure systems has primarily been addressed by using life-cyle analysis (Racoviceanu and Karney 2010; Francis et al. 2011). Communication and integration approaches have included the design of management systems (Halfawy and Eng 2008), peer-to-peer communication (Zhang et al. 2010), and enterprise-level geographic information system approaches (Pradhan et al. 2007). Integration of infrastructure systems has often adopted a network theory approach that relates network properties to reliability measures (Dueñas-Osorio and Vemuru 2009; Ouyang and Dueñas-Osorio 2011) or considers infrastructure as multilayer networks interlinked based on physical and socio-economic factors (Chang et al. 2002; Zhang et al. 2005; Zhang and Peeta 2011). Past work has also focused on the specific problem of municipal infrastructure systems. For example, Halfawy and Eng (2008) presented a discussion of the main challenges and proposed specific solutions for implementing integrated Municipal Infrastructure Management Environments (MIMEs). The proposed solutions focus on industry standard data integration and software interoperability standards in the municipal infrastructure domain, and considers existing standards and their harmonization, refinement, and integration.

Prior work on using web services in civil engineering has focused on applications to both built and natural systems. Vacharasintopchai et al. (2007) presented an architectural framework for the application of the Semantic Web Services technology in computational mechanics. The work was motivated by the perceived need in the computational mechanics community to allow various groups of programmers to work collectively to create a simulation that could be deployed using heterogeneous platforms. Halfawy (2010) presented municipal integration work that builds from Halfawy and Eng (2008) and proposed web service components and Geography Markup Language (GML) data standards for MIMEs. Liu et al. (2003) presented an innovative vision for applying ubiquitous computing where devices are universally accessible through information services and parties are able to access information services to foster collaboration. Liu et al. (2005) expanded on prior work by presenting an experimental service-composition paradigm for integrating loosely coupled software components that employs a distributed data-flow approach. Results from this work suggested that a distributed data flow approach is superior when data volumes are large, but when data volumes are low, more traditional approaches outperform their proposed approach. There has also been work in the water resources community to use web services for integrating heterogeneous databases (Goodall et al. 2008; Horsburgh et al. 2009) and for model integration (Goodall et al. 2011).

This paper expands on past research by focusing specifically on the challenge of integrating multidisciplinary models across civil infrastructure systems using a service-oriented architecture. The technical approach is most similar to that of Liu et al. (2003) where we envision civil infrastructure as a distributed system of components that are published as services to enable interactions based on system interdependencies. Distributed systems are common across many fields, and for this reason researchers have created tools for managing distributed systems for business, science, and engineering applications (Foster et al. 2002; Papazoglou and Georgakopoulos 2003; Foster 2005). Taking the system-of-systems approach for interdependent critical infrastructures described by Rinaldi et al. (2001) and using the definition of a system-of-systems as consisting of "multiple, heterogeneous, distributed, occasionally independently operating systems embedded in networks at multiple levels, which evolve over time" introduced by DeLaurentis (2007), Eusgeld et al. (2011) proposed the idea of hierarchical level architecture for capturing the complexity of system-of-systems. In this architecture, the lowest level represents system models of a single infrastructure, the middle level represents interactions between lower level systems, and the high level represents the global system-of-systems (Eusgeld et al. 2011). In relation to past approaches of interdependent infrastructure systems, Nan and Eusgeld (2011) suggested that the two best known approaches are complex network theory and object-oriented modeling. Within this context, our work builds on object-oriented modeling approaches where a system-of-systems is represented as low level models of single infrastructure systems, which are individually exposed as web services, and then integrated into a middle level systems through data exchanges representing system interdependencies.

# METHODOLOGY

A simplistic view of the desired system representation is shown in Fig. 1. In this representation, the services for each system allow for the integration into a system-of-systems. The service-oriented architecture designed to create this system-of-systems was developed in three phases: (i) identify appropriate web service standards for use in urban infrastructure systems, (ii) implement our existing models and sensor systems as web services to facilitate integration, and (iii) design client software system to coordinate service integration for an urban flooding example application. In this section we discuss the first two design-oriented tasks, and in the Example Application section we discuss the third implementation-oriented task.

## Design of Models as Web Services

Service-oriented architectures follow a client/server paradigm where the server publishes a web service and the client uses that web service to perform a specific task. Different standards have been developed to facilitate communication between clients and servers. For example, the Simple Object Access Protocol (SOAP) can be used to encode data exchanges between clients and servers in a service-oriented architecture (Christensen et al. 2001). The

Web Service Definition Language (WSDL) is typically used to define a web service when using SOAP. WSDL is an XML-based file that specifies the methods that a service can perform including the inputs and outputs for each method (Pautasso et al. 2008). A client reads the WSDL file for a web service in order to interact with it through method calls. Another approach for web service communication is defined by the REpresentational State Transfer (REST) specification. REST is a self descriptive specification that utilizes Uniform Resource Identifiers (URI's) to direct the client to a specific service on the server, and as a result, there is no need for WSDL's (Fielding 2000). A service is manipulated by the client using HTTP methods: GET, PUT, POST, and DELETE. In general, REST can be thought of as a simpler web service implementation than SOAP (Ray and Kulchenko 2003), and it has gained popularity in part for this reason. We adopted REST in this work because of its simplicity and wide adoption.

REST is a general standard for providing interoperability between clients and servers that can benefit from an additional layer of software to provide domain specificity (Foster 2005). Examples of domain specific web service standards applicable to civil infrastructure systems are those from the Open Geospatial Consortium (OGC). OGC is a non-profit organization that has created widely used standards for publishing geospatial data. These standards are used in conjunction with the lower level service-oriented architecture standards, such as REST or SOAP, and provide additional specificity for the interactions between client applications and web services when dealing with geospatial data (Kiehle 2006). The Web Processing Service (WPS) is one of the OGC standards and is designed for performing server-side data processing operations (Schaeffer 2008). The WPS standard consists of three methods: GetCapabilities, DescribeProcess, and Execute. The GetCapabilities method is used by the client to query metadata that describes the processing services provided by the server (Schut and Whiteside 2007). This functionality is implemented at the server level and retrieves information about all available web processes in a single call. The DescribeProcess and Execute methods are implemented at the process level, meaning they are unique to each web process on the server. The DescribeProcess operation provides the client with metadata describing a specific process (Schut and Whiteside 2007). This operation is useful for determining the required inputs, as well as the outputs calculated by a process. Finally, the Execute operation enables a client to specify inputs and run a web process (Schut and Whiteside 2007). The Execute operation can return structured data such as XML or JavaScript Object Notation (JSON), as well as various file formats (e.g. Network Common Data Form (NetCDF), JPG, TIFF, etc.). Furthermore, the data can consist of three possible output types: literal, complex, or boundingbox. Since WPS supports XML, it is also possible to encode data using a specific markup schema, for example geographic data using the Geography Markup Language (GML).

In this paper, we use the OGC WPS and REST to expose models of the civil infrastructure system to a client application. Our approach consists of a server with a set of models that implement the WPS interface; therefore, they have defined operations when a client calls the GetCapabilities, DescribeProcess, and Execute methods (Fig. 2). We leveraged the open source Python PyWPS library (http://pywps.wald.intevation.org) to implement this solution and through prior work (Castronova et al. 2012) extended the software in order to maintain state on the server. This extension was necessary to allow the models in our system to maintain a session with specific clients that are running a model interactively through service calls. The existing PyWPS library is designed to use REST and implements GET and POST methods. Our extension adds to this by implementing the DELETE method in order to remove session data from the server, as shown in Fig. 2. The implementation begins by the client first constructing the URI that specifies the WPS method, resource, and input parameters. The URI is invoked on the server using either a DELETE, POST, or GET command, and depending on the command used, a specific action is invoked on the server and output data is returned to the client. For example, a model can be run by calling the WPS Execute method with the input data for the model using a POST or GET command. Once the client has finished using the model, the session can be ended by calling the DELETE command using the same URI.

## Implementing Models as Web Services

Three model services were constructed to represent the water, transportation, and structural systems in the integrated model illustrated in Fig. 1. It is important to note that our emphasis in this work is on the framework required for system integration including mechanisms for establishing interoperability across system components. Thus, it was sufficient to start with simple models for building and testing the integration framework. By establishing service interfaces and communication standards, it will be possible to evolve the models while still maintaining system interoperability. It is also important to note that the models are general and not specific to a given use case. Therefore, while an example application is provided following this section as one potential use case for the framework, the services were not designed to be used only for this specific use case.

### *Water Service*

The water web service estimates streamflow based on observations of river stage measured by a sensor network. Both river stage and streamflow are made available by the water service to the transportation and structures services during model simulation. Streamflow is modeled at a given location along a stream network where stage is known using the Manning equation (Eq. 1)

$$Q = \frac{1.49}{n} A R^{\frac{2}{3}} S_f^{\frac{1}{2}} \tag{1}$$

where $n$ is Manning's roughness coefficient, $R$ is the hydraulic radius of the channel, and $S_f$ is the friction slope (Chow et al. 1988). For simplicity, we assumed a simple trapezoidal channel geometry so hydraulic radius can be approximated by Eq. 2 where $B$ is the bottom width of the channel, $y$ is the depth of water, and $z$ is the horizontal slope of the river banks (Mays 2005).

$$R = \frac{(B + zy)y}{B + 2y\sqrt{1 + z^2}} \tag{2}$$

If detailed cross-sectional data is available, this information could be easily incorporated into the service to relax the assumption of a trapezoidal channel geometry.

To determine flood inundation, the observed river stage is used along with channel and floodplain geometries. The calculation makes use of geoprocessing routines available in a Geographic Information System (GIS) and involves several steps as outlined in Fig. 3. First, a point is placed at the location of river stage gaging station. Next, the elevation at this point is extracted from a given Digital Elevation Model (DEM) and is used to determine the water depth (i.e., the elevation of the river stage relative to the land surface). The water surface elevation is then subtracted from the land surface elevation so that locations where flooding has occurred can be identified by having values less than zero. The areas where flooding may occur are converted into polygon features. Next, the location of the river overtopping the channel is buffered by a distance proportional to the river height as a simple means for estimating a margin of safety factor. The buffered region is then intersected with the potential flood region to determine the locations that are at high risk of flooding. Finally, this region is used to determine the roads that will be affected by the flood. While this approach ignores many of the complicated hydraulic and hydrologic conditions that occur during floods that would be needed for a realistic flood model, it does provide a means for estimating the likelihood that the river will inundate roads – valuable information for traffic operations.

### *Transportation Service*

The transportation web service utilizes DTALite, an open-source dynamic traffic assignment model (https://sites.google.com/site/dtalite/home). The system architecture of DTALite is shown in Fig. 4. As shown, DTALite, like other Dynamic Traffic Assignment (DTA) models, takes as input a transportation network and an Origin-Destination (OD) matrix that specifies trips between traffic analysis zones (in this study, the OD reflects the morning rush hours in downtown Columbia, SC), models the traffic flow evolution in a network using advanced network algorithms and trip-maker behavior models, and provides as output link and path travel times. DTALite can also capture the effect of road closures, real-time traffic information provision via Advanced Traveler Information System (ATIS), and routing of traffic via Dynamic Message Signs (DMS).

In the event of flooding, DTALite receives information about road closures from the water web service and it in turn modifies the transportation network to reflect the change in capacity. Specifically, the capacity of the links corresponding to flooded roads are set to zero. Under flooding conditions, tripmakers who enter a road that is closed won't be able to move and will have to wait until the road's capacity is restored. Tripmakers with access to real-time information will avoid the closed roads when selecting the shortest path to take at the time of departure. To assess the benefit of trip-makers having access to real-time traffic information and thus prevailing road closures, the pre-trip information feature in DTALite is used. Pre-trip information means that trip-makers know the prevailing travel times on all links in the network (including road closures) at the time of their departure and will choose the shortest paths to their destinations and thereby choose paths that avoid the closed roads. The key performance measure used in this study is the average network travel time of all trip-makers who completed their trips.

DTALite was deployed as a web service by creating a Python wrapper designed to interact with the DTALite application when invoked by client applications. It was assumed that the necessary input files are pre-loaded onto the server. During model simulation, the client application invokes the DTALite web resource (i.e. the Python wrapper) using the POST method. This enables the client to send road closures data, encoded in eXtensive Markup Language (XML) to the DTALite web resource. At this time, the client can also designate if DMS or an Advanced Traffic Information System (ATIS) will be used. These data are received by the web resource and saved within specific DTALite input files. Next, the web resource executes the DTALite application, extracts the results from an output file, and sends them back to the client. This approach enables the client to change simulation parameters such as road closures, assuming that the model was created first on a desktop computer and then upload to the web server.

*Structural Reliability Service*

The structural reliability web service estimates the probability of failure of bridges based on finite element models. Structural reliability is usually calculated as a function of time, showing the probability of failure of the structure as the strength of the structural system decreases due to aging and other processes (Okasha et al. 2011). In this particular paper we are interested in investigating the probability of failure due to scour of its foundation. This is calculated by developing a finite element model of the structure where the load, material characteristics, and soil characteristics are considered uncertain. The soil is modeled using a Winkler model (Makris and Gazetas 1992; Zarafshan et al. 2011) where the soil is represented with a linear spring.

The scour process is very complex (Yanmaz et al. 1991; Richardson and Panchang 1998), and the modeling and simulation of scour in bridge foundations is an active area of research. A complex scour model could be included as a separate component of the proposed simulation framework. However, we assumed a simplified model where the stiffness of the foundation is a function of the river's stage. The model is inspired by results reported in the Federal Highway Administration HEC-18 report (Richardson et al. 1993). The report summarizes the scour depth of different models as a function of river stage and the models indicate an exponential relationship between scour depth and stage. Given that stiffness of the foundation and scour are inversely proportional, the stiffness of the foundation is modeled using the equation

$$k(h) = R * k_s * exp(-a * h) \tag{3}$$

where $k(h)$ is the spring constant as a function of the water stage $h$, $k_s$ is the original stiffness and $a$ is a constant that describes the reduction of stiffness. The constant $a$ is calculated by finding the river stage that would create an expected loss of the foundation of the bridge. $R$ is a normally distributed random number that indicates the uncertainty in the foundation characteristics. Uncertainty in the bridge's materials is considered by multiplying the Young's modulus by a normally distributed random number. Similarly, uncertainty in the live load is simulated by multiplying the vehicle's weight by a normally distributed random number.

The location of the vehicle on the bridge influences the response of the structure. An influence diagram was generated to determine the vehicle location that creates the highest displacement at the point of interest. All subsequent calculations were performed with the vehicle located at the critical location. The probability of failure of the structure is calculated as the probability of the structure exceeding a limit state. In this study two limit stages were considered: the structural elements exceeding yielding stresses, and excessive displacements of the supports creating structural instabilities (i.e. the beams losing their support due to excessive displacements). Here, we considered failure when the top of a bridge's pier has displaced more than a predetermined value.

The probability of failure of the bridge to a particular river stage is calculated before the simulation starts using an in-house Matlab finite element toolbox. The results of this simulation are stored in a database and the structural reliability service provides client applications access to these data. We used this data staging approach because the structural model is time consuming and does not easily lend itself to on demand processing. However, the service could be implemented such that the reliability of the structure is calculated as requested, if needed, or so that the Matlab program is running "behind the scenes" to update the SQL database if any changes are made in the assumptions of the analysis. The structural reliability service could also be complemented with structural health monitoring services using global or local structural health monitoring techniques (Brownjohn 2007) through future work. The simulation would then be configured to let the client consume the structural health monitoring service directly, or to let the structural reliability service consume structural health monitoring service to enhance the calculations of the probability of damage.

## EXAMPLE APPLICATION

As a demonstration of the service-oriented approach, we modeled a historical flooding event in the Rocky Branch Watershed in downtown Columbia, South Carolina. We investigated a scenario that emphasizes real-time communication between hydrologic, transportation, and bridge components during a flooding event in order to understand how system integration can improve traffic management.

### Study Area

The Rocky Branch Watershed (Fig. 5) drains 11 $km^2$ of Columbia, SC. The watershed is urbanized consisting of commercial districts, university property, and residential neighborhoods. Due in part to urbanization of the watershed, high intensity storms often cause flooding within the watershed. During these flooding events, roads at low lying areas must be closed by the city as a safety precaution. We have selected this watershed because the frequent flooding that the City of Columbia is attempting to alleviate. However, the approach and tools developed through this study are applicable to other challenges in urban infrastructure integration beyond this specific example application.

Existing data for the watershed includes stream gaging stations maintained by the USGS and the University of South Carolina, as well as a weather station maintained by the University of South Carolina (Fig. 5). Data collected at the USGS stations are available through the National Water Information System and include river stage on a 15-minute time interval. One river stage and one rainfall gauge capable of recording observations on a 15-second interval were installed as part of this study. This high frequency data is important because of the quick response of the watershed to rainfall events (i.e., the time to peak streamflow can be 30 or 45 minutes from the beginning of the rain storm). The river stage at the USC gage is measured using a KPSI Submersible Hydrostatic Level Transducer and rainfall is measured using a stainless steel tipping bucket rain gage manufactured by Sutron Corporation. The rain gage has an orifice diameter of 7.87 inches (20 cm) and measures rainfall in 0.01 inch increments, to an accuracy of 2%.

The transportation network was extracted from a larger regional network made available by the Central Midlands Council of Governments (CMCOG). The network extraction was performed using TransCAD, a GIS based transportation software. Additionally, TransCAD was used to output data in the format required by DTALite. The network covers an area of 7.3 km$^2$, which is smaller than the watershed area because it focuses specifically on downtown Columbia. There are a total of 898 links and 313 nodes within the network. In our simulations, approximately 850 vehicles within the transportation network reached their destination over a period of 60 minutes; this demand is derived from the overall regional network demand data provided by CMCOG.

A rail bridge crossing the Rocky Branch Watershed and Sumter Street is the focus of the structural reliability service. A finite element model of the bridge (Fig. 6) was used to calculate the bridge displacements. The model has 664 nodes, 480 beam elements, 312 rigid links, 228 lumped masses and 32 linear springs to model the interaction with the soil using a Winkler model. R (Equation 3) was defined with a mean of one and a standard deviation of 0.1. This is within the limits reported on the literature for the variation of the stiffness of soils (Jones et al. 2002). The bridge model was loaded with Hopper cars that have a load capacity of 224,500 lb traveling at low speed because the bridge is located in an urban area. Uncertainty in the loads was considered by assuming that the load of each car follows a normal distribution of mean 224,500 lb and a standard deviation of 11,225 lb. The state of the structural component was also considered uncertain and a normally distributed Young modulus with mean of 29,000 ksi and a standard deviation of 300 ksi. The yielding stress was considered as a normal random variable with a mean of 50 ksi and a standard deviation of 5 ksi. Simulation results indicate that the structure will not reach yielding at any stage level. Therefore, the limit state corresponding to the yield stress does not control. The limit stage for maximum displacement was considered as 3 inches and determined the failure of the structure. Fig. 7 shows a representative example of the histograms for these random variables (Young's modulus), a Probability Density Function (PDF) of the displacement of the bridge at the point of interest, and the probability of failure (displacement greater than 3 inches) as a function of the river stage.

## System Integration

Our approach for system integration follows a centralized paradigm where the services within the system communicate through a client-side *model coordinator* (Fig. 8). Using this approach each model remains independent of other models, and any required translation functionality is implemented in the model coordinator. This implementation has advantages in that it provides the client much greater control over its own execution and facilitates the exchange of information via web services. This allows the web services to be generic and reusable across a wide range of applications because they are not client-specific. While we built our prototype system using a centralized service integration paradigm, we acknowledge the possibility of a more decentralized approach where services are directly chained into a workflow with minimal translation of data exchanges between services.

The data flow during the integrated infrastructure simulation (Fig. 9) consists of multiple web service calls, all mediated by a client-side controller. In this study, simulation begins by issuing the WPS Execute operation using the HTTP GET method on the water service and providing a bounding box as input. This triggers the water service to produce output for all known locations within the requested rectangle, and returns these values to the client. The calculated flood stage is then used to determine which roads are affected by the flood water. This calculation requires GIS processing, and is currently performed on the client machine. Next, the Execute operation is issued once again on the water service, only this time specific locations are provided as input. The locations represent areas of interest, namely locations in which the river water impacts bridge supports. The water service returns river stage and streamflow calculations for the gages nearest to these requested bridge locations. Since the river stage measured at the streamflow gages likely differs from the stage at the bridges, additional calculations are performed by the model coordinator to translate streamflow at the gage location into river stage at the bridge location. This is done by first assuming that the flow rate at both locations is the same. Next, using simplified trapezoidal channel geometries, shown in Table 1, Manning's equation was used to back calculate river depth at the bridge from the known streamflow at the nearest gage. This new river stage is then supplied as input for the bridge service by invoking the WPS Execute operation using the HTTP GET method. Using this data, in addition to known bridge properties, the bridge service calculates the probability that a specific structure will fail. These calculations are sent back to the client-side controller which determines if it is necessary to close the bridge (and neighboring roads) due to safety concerns. All road closures are aggregated and sent to the transportation service by invoking the WPS Execute operation using the POST method, which enables input data to be supplied within an eXtensible Markup Language (XML) encoding. The transportation service uses this input to calculate the average travel time through the road network. This sequence continues over the entire simulation time horizon.

## Simulation Results

The example application simulation captures the dynamic interaction and interdependencies of the subsystems (i.e. water resource, transportation, and bridge). The results calculated by each subsystem at every designated time step over the course of the simulation horizon (Fig. 10) show that at three separate gauging locations, the river stage continues to increase until overtopping and subsequent road closures occur, indicated by the dashed lines. Once the river stage decreases and flooding subsides, roadways are reopened to the public. Similarly, one bridge in the test network is severely affected by the forces imposed by rising water levels, to the extent that the probability of failure approaches 50%. When the probability of failure reached 40%, the bridge and neighboring roadways were closed due to safety concerns. This threshold is dependent on the bridge owner's preferences and can be adjusted accordingly. Unlike the previous scenario, the bridge and neighboring roads were not reopened to the public because the bridge would have to undergo an official evaluation before deemed safe.

The final plot illustrates the average travel time through the road network under four different scenarios. The base case, represented by the horizontal line, offers a perspective of the typical travel time without a flooding event and therefore no roads are closed. In this scenario, the average network travel time for all travelers is about 4 minutes. The other lines show the average network travel time when roads are closed as a result of the storm event assuming (i) no communication to travelers about road closures, (ii) 50% of users have access to pre-trip information, and (iii) 100% of users have access to pre-trip information. As expected, when travelers have access to real-time travel information via ATIS, the impact of the road closure is less (about 10 minutes less at the peak of the flooding) because travelers chose paths that avoid the closed roads. This time savings reduces to approximately 4 minutes when only 50% of the tripmakers have access to real-time information. Without the integration of the separate subsystems used in this model, the closure of roads would be done in a reactive instead of proactive manner. In the reactive scenario, after some time (around 30 to 60 minutes) the roads have been flooded the city police would arrive on the scene and erect barricades to close down the roads. During this time, some vehicles may attempt to pass flooded roads and in the process of doing so endanger their lives and others around them. In the proactive scenario, the police would know in advance when the roads will be flooded. Therefore, they can close roads forecasted to be flooded before they are actually overtopped to prevent any vehicle crossing.

## DISCUSSION

This paper aims to better understand the benefits and challenges of applying a service-oriented architecture for representing civil infrastructure as a "system-of-systems." The key benefit is that service-oriented architectures allow modelers to combine heterogeneous computing resources into an integrated system. Due to the loose-coupling nature of web services, each model maintains its independence, which is a benefit because the core development team can maintain control over the model and therefore it can evolve over time within the larger modeling system. Models in this system are meant to be generic and can be applied across a variety of use cases (the example application in this paper presented one potential use case). Therefore, while the service logic described in this paper may be overly complex for the particular example application demonstration, each service might be used for multiple applications in a production environment, thus requiring a sophisticated implementation not tailored to this specific use case. That said, if the architecture was deployed for a single use case, services could be tailored to match the desired level of complication required by the use case.

We have already experienced the benefit of this loosely-coupled system design in a number of ways. In building the example application, we were able to work independently on our individual models of the single infrastructure systems. Throughout our work, these models evolved in complication while still functioning within the large system context. We were able to represent our individual infrastructure systems using the modeling methodology and technical implementation that best suited the problem. For example, the water service used GIS operations and was implemented in a Linux OS, while the transportation service used a network-based model and was implemented on a Windows OS. Instead of recompiling each of these models to run under the same operating system or adopting a single means for abstracting the environment across all three sub-systems, we were able to achieve an integrated system by exposing each model as a web service. Therefore, while it is well known that the service-oriented approach will introduce computational overhead during simulation runtime due to network data communication latencies, a broader view of benchmarking should also consider the time required to build and maintain a state-of-the-art simulation model, and at this level our approach offers considerable time savings by leveraging existing models that were rapidly integrated using services to create a detailed system representation.

A key contribution of this work is the translation needed to integrate low-level infrastructure systems into a higher-level system-of-systems. We achieved this goal by creating workflows that transformed information provided by a model service into the specific input needed by a second service. In our example application, the water service provides only water heights and flows at specific locations, yet the transportation and structure services required information at other locations within the study area. To transform this information, we developed client-side code that uses the water height information provided by the water service within a Geographic Information System (GIS) framework to determine road closings. Therefore, geospatial referencing of information provided the context for translating information between the models. This road closing information can then be fed to the transportation service for traffic re-routing. In a more general sense, this technique demonstrates the benefit of general services that can be used by client-side code that acts as a communication mediator between different models of civil infrastructure subsystems. However, it also demonstrates the cost of having to maintain more sophisticated client-side code to perform the system integration.

The key challenge in making a service-oriented approach work for civil infrastructure systems is the need to provide exact specification for interface standards and data exchanges. In this work, we used the OGC WPS as the interface standard. As we discussed before, the web processing service was not specifically designed as a way of exposing models, but rather as a means for exposing similar data processing routines as web services (Goodall et al. 2011). We believe that a new modeling web service standard would be ideal; however in this work we conformed to existing standards in order to be compliant with existing approaches and to leverage existing tools such as PyWPS. In addition to interface standards, there is the need to clearly define data exchanges that go beyond describing objects, but also describe semantics in a consistent way so that mismatches between systems can be overcome. To accomplish this would require identifying existing or construct new ontologies for each discipline (water, structures, transportation) to facilitate both syntactic and semantic mediation within the integrated system simulation. Each service can, and most likely will, be implemented on its own spatial and/or temporal grid or use different internal geospatial data representations (lines, polygons, and volumes) and their own internal vocabularies and semantics. The key to simulating a system where services have different spatial or temporal scales is to fully document data exchanges according to a well defined ontology, and to include data transformations to rescale the output of one component to satisfy the input needs of another component.

Our example application does not fully address many of the challenges that would be encountered when implementing a service-oriented approach at a production level. Specific challenges that would need to be addressed more fully include (i) security, (ii) performance, and (iii) fault tolerance. If models are exposed on the web as services, there is the potential for misuse of the services. One could easily imagine sensitive civil infrastructure services that must be tightly guarded against malicious use. This challenge is not unique to civil infrastructure systems and cyber-security is an active area of research and development. Therefore, while we have not implemented security measures in this example application, we are confident that this challenge could be overcome with proper information technology implementations.

Performance may also be an issue for service-oriented architecture approaches because of the need to transmit data over a network. A service-oriented architecture approach is not ideal for a system with numerous interactions and transmission of data on individual time steps, as there could be a significant performance cost associated with transmitting large volumes of data over the Internet. The concept of granularity of system decomposition is important. Therefore, care should be taken not to decompose the system into too fine of granular components. Performance tests could be easily developed through system simulation to understand costs and bottlenecks within the system and to reconfigure components to achieve acceptable performance metrics for the intended use of the system.

Finally, fault tolerance is particularly important in a service-oriented architecture because of the variety of ways in which the system could be susceptible to faults including network outages, failed sensors, temporarily unavailable servers or services, either intentional or unintentional misuse of the system. These issues must be anticipated and properly handled. Many of these challenges can be easily handled with proper software engineering to ensure graceful handling of system faults. However, there is no way to guarantee that all components of the system will be functional at all times, and creating a highly reliable system could become cost prohibitive if it requires multiple servers for backup and load balancing.

## CONCLUSIONS

The overarching objective of this work was to apply service-oriented architectures for integrating models for individual infrastructure systems into a system-of-systems. By doing so, one of our goals was to better understand if this technical solution to creating distributed systems could be used to overcome specific challenges associated with integrating diverse methodologies used to model individual systems present within urban infrastructure systems. Based on our study, we conclude that the approach has merit and is well suited for this particular application. A primary reason for reaching this conclusion is that the heterogeneity present in our models of water, transportation, and structural systems was significant, making a tight integration (meaning the code is recompiled into a single application) challenging or even impossible. While loose integration using web services does introduce a computational overhead associated with exchanging data across the Internet, in our example application and for expected uses of such an integrated model (e.g., real-time day-to-day traffic management by traffic management centers), the size of data exchanges between the models will be generally small, making the overhead acceptable for the intended use of the model.

A second conclusion from this work is that a domain-specific software layer is beneficial for abstracting the modeler from the lower-level web service protocols. Our implementation approach made use of REST as the lower-level web service protocol and the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard as the domain-specific software layer. We considered alternative approaches including using REST alone or creating our own domain-specific software layer. However, we found the OGC WPS useful in providing an additional layer of abstraction between the modeler and the lower-level web service protocols. Furthermore, while we needed to modify the OGC WPS for our purposes, this modification is a more effective solution than building a custom solution for civil infrastructure systems.

The outcome of the integrated urban infrastructure system example application demonstrated the benefit of system integration as measured by the improvement in traffic operations if information from a water resource model and bridge structure model were made available in real-time to improve traffic management and decision making. In the example application, each model is written in a different programming language, some models operate in Windows OS and others in Linux OS, and each model had a unique way of abstracting the real-world system into objects for use within the model. By standardizing interfaces and communications between the models, it allows for each model to remain independent within the system. This means that each model can evolve independently as long as interfaces between models are maintained. Our future work will focus on advancing the individual models within the prototype system to more accurately simulate integrated infrastructure systems.

Finally, it is important to note that, while we have achieved interoperability across three distinct components of the civil infrastructure system, our approach can be improved through future work and future research. Future work should include improving the current approaches for security, performance, and fault tolerance. We classify these extensions as future work rather than future research because there are existing approaches for overcoming these challenges within the context of cyber-systems. Future research should focus on creating ontologies to more elegantly handle the syntactic and semantic heterogeneities between models. In this paper, we used an ad-hoc approach for overcoming these integration challenges that included spatiotemporal and semantic mismatches between models. This is not the ideal long term solution and further research is needed to establish or merge standards across civil infrastructure system domains to better facilitate integration through service-oriented or other approaches.

## ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1100227.

## REFERENCES

Amin, M. (2002). "Modeling and control of complex interactive networks [guest editorial]." *Control Systems Magazine, IEEE*, 22(1), 22 –27.

Ash, J. and Newth, D. (2007). "Optimizing complex networks for resilience against cascading failure." *Physica A: Statistical Mechanics and its Applications*, 380, 673 – 683.

Bordogna, J. (1998). "Tomorrow's civil systems engineer—the master integrator." *Journal of Professional Issues in Engineering Education and Practice*, 124(2), 48–50.

Brownjohn, J. (2007). "Structural health monitoring of civil infrastructure." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), 589–622.

Castronova, A., Goodall, J., and Elag, M. M. (2012). "Models as web services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard." *Environmental Modelling & Software*, In Review.

Chang, S., Svekla, W., and Shinozuka, M. (2002). "Linking infrastructure and urban economy: simulation of water-disruption impacts in earthquakes." *Environment and Planning B*, 29(2), 281–302.

Chow, V. T., Maidment, D. R., and Mays, L. W. (1988). *Applied Hydrology.* McGraw-Hill Inc, Boston, Massachusetts.

Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). "Web services description language (WSDL) 1.1." *Report no.*, World Wide Web Consortium.

DeLaurentis, D. (2007). "Role of humans in complexity of a system-of-systems." *Proceedings of the 1st International Conference on Digital Human Modeling*, Springer-Verlag. 363–371.

Dueñas-Osorio, L. and Vemuru, S. (2009). "Cascading failures in complex infrastructure systems." *Structural safety*, 31(2), 157–167.

Eusgeld, I., Nan, C., and Dietz, S. (2011). "“system-of-systems” approach for interdependent critical infrastructures." *Reliability Engineering and System Safety*.

Fielding, R. (2000). "Architectural styles and the design of network-based software architectures," PhD thesis, University of California.

Folke, C. (2006). "Resilience: The emergence of a perspective for social-ecological systems analyses." *Global Environmental Change*, 16(3), 253 – 267.

Foster, I. (2005). "Service-Oriented science." *Science*, 308(5723), 814–817.

Foster, I., Kesselman, C., Nick, J. M., and Tuecke, S. (2002). "Grid services for distributed system integration." *Computer*, 35, 37–46.

Francis, R., Falconi, S., Nateghi, R., and Guikema, S. (2011). "Probabilistic life cycle analysis model for evaluating electric power infrastructure risk mitigation investments." *Climatic change*, 106(1), 31–55.

Garrett, J. H. (2005). "Advanced infrastructure systems: Definitions, vision, research, and responsibilities." *Computing in Civil Engineering 2005*, Cancun, Mexico. 4–4.

Goodall, J., Horsburgh, J., Whiteaker, T., Maidment, D., and Zaslavsky, I. (2008). "A first approach to web services for the national water information system." *Environmental Modelling & Software*, 23(4), 404–411.

Goodall, J., Robinson, B., and Castronova, A. (2011). "Modeling water resource systems using a service-oriented computing paradigm." *Environmental Modelling & Software*.

Halfawy, M. (2010). "Municipal information models and federated software architecture for implementing integrated infrastructure management environments." *Automation in Construction*, 19(4), 433–446.

Halfawy, M. and Eng, P. (2008). "Integration of municipal infrastructure asset management processes: challenges and solutions." *Journal of Computing in Civil Engineering*, 22, 216.

Horsburgh, J., Tarboton, D., Piasecki, M., Maidment, D., Zaslavsky, I., Valentine, D., and Whitenack, T. (2009). "An integrated system for publishing environmental observations data." *Environmental Modelling & Software*, 24(8), 879–888.

Jones, A., Kramer, S., and Arduino, P. (2002). "Estimation of uncertainty in geotechnical properties for performance-based earthquake engineering." *Report No. 2002/16*, Pacific Earthquake Engineering Research Center, College of Engineering, University of California, Berkeley.

Kiehle, C. (2006). "Business logic for geoprocessing of distributed geodata." *Computers & Geosciences*, 32(10), 1746–1757.

Kuhn, K. D. (2010). "Network-level infrastructure management using approximate dynamic programming." *Journal of Infrastructure Systems*, 16(2), 103–111.

Little, R. G. (2003). "Toward more robust infrastructure: Observations on improving the resilience and reliability of critical systems." *HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 2*, Washington, DC, USA. IEEE Computer Society, 58.1.

Liu, D., Cheng, J., Law, K., Wiederhold, G., and Sriram, R. (2003). "Engineering information service infrastructure for ubiquitous computing." *Journal of Computing in Civil Engineering*, 17, 219.

Liu, D., Peng, J., Law, K., Wiederhold, G., and Sriram, R. (2005). "Composition of engineering web services with distributed data-flows and computations." *Advanced Engineering Informatics*, 19(1), 25–42.

Makris, N. and Gazetas, G. (1992). "Dynamic pile-soil-pile interaction. Part II: Lateral and seismic response." *Earthquake Engineering & Structural Dynamics*, 21(2), 145–162.

Mays, L. W. (2005). *Water Resources Engineering*. John Wiley and Sons, Inc, Hoboken.

Nan, C. and Eusgeld, I. (2011). "Adopting hla standard for interdependency study." *Reliability Engineering & System Safety*, 96(1), 149–159.

Okasha, N., Frangopol, D., and Orcesi, A. (2011). "Automated finite element updating using strain data for the lifetime reliability assessment of bridges." *Reliability Engineering & System Safety*.

Ouyang, M. and Dueñas-Osorio, L. (2011). "An approach to design interface topologies across interdependent urban infrastructure systems." *Reliability Engineering & System Safety.*

Papazoglou, M. P. and Georgakopoulos, D. (2003). "Service-oriented computing: Introduction." *Communications of the ACM*, 46(10), 24.

Pautasso, C., Zimmermann, O., and Leymann, F. (2008). "Restful web services vs. "big" web services." *Proceeding of the 17th international conference on World Wide Web*, Beijing, China. 805.

Pradhan, A., Laefer, D., Rasdorf, W., et al. (2007). "Infrastructure management information system framework requirements for disasters." *Journal of Computing in Civil Engineering*, 21, 90.

Racoviceanu, A. I. and Karney, B. W. (2010). "Life-cycle perspective on residential water conservation strategies." *Journal of Infrastructure Systems*, 16(1), 40–49.

Ray, R. and Kulchenko, P. (2003). *Programming Web Services with Perl.* O'Reilly Media.

Richardson, E., Harrison, L., Richardson, J., and Davis, S. (1993). "Evaluating scour at bridges: Federal highway administration hydraulic engineering circular no. 18 (hec-18), publication no." *Report no.*, FHWA-IP-90-017.

Richardson, J. and Panchang, V. (1998). "Three-dimensional simulation of scour-inducing flow at bridge piers." *Journal of Hydraulic Engineering*, 124(5), 530–541.

Rinaldi, S., Peerenboom, J., and Kelly, T. (2001). "Identifying, understanding, and analyzing critical infrastructure interdependencies." *Control Systems Magazine, IEEE*, 21(6), 11–25.

Sanford Bernhardt, K. L. and McNeil, S. (2008). "Agent-based modeling: Approach for improving infrastructure management." *Journal of Infrastructure Systems*, 14(3), 253–261.

Schaeffer, B. (2008). "Towards a transactional web processing service." *Proceedings of the GI-Days, Münster.*

Schut, P. and Whiteside, A. (2007). "OpenGIS Web Processing Service." *Open Geospatial Consortium, Wayland, MA, USA.*

Tran, H. D., Perera, B. J. C., and Ng, A. W. M. (2010). "Markov and neural network models for prediction of structural deterioration of storm-water pipe assets." *Journal of Infrastructure Systems*, 16(2), 167–171.

U.S. Census Bureau (2012). TIGER/Line Roads Shapefile product courtesy of the U.S. Census Bureau.

U.S. Geological Survey (2014). National Elevatoin Dataset courtesy of the U.S. Geological Survey.

Vacharasintopchai, T., Barry, W., Wuwongse, V., and Kanok-Nukulchai, W. (2007). "Semantic web services framework for computational mechanics." *Journal of Computing in Civil Engineering*, 21, 65.

Yanmaz, A. et al. (1991). "Study of time-depenbent local scour around bridge piers." *Journal of Hydraulic Engineering*, 117, 1247.

Zarafshan, A., Iranmanesh, A., and Ansari, F. (2011). "A vibration based method and sensor for monitoring of bridge scour." *Journal of Bridge Engineering*, 1, 253.

Zhang, G., Smith, B., and Guo, J. (2010). "Peer-to-peer-based publish/subscribe architecture for advanced infrastructure systems." *Journal of Computing in Civil Engineering*, 24, 65.

Zhang, P. and Peeta, S. (2011). "A generalized modeling framework to analyze interdependencies among infrastructure systems." *Transportation Research Part B: Methodological*, 45(3), 553–579.

Zhang, P., Peeta, S., and Friesz, T. (2005). "Dynamic game theoretic model of multi-layer infrastructure networks." *Networks and Spatial Economics*, 5(2), 147–178.

**TABLE 1. Simplified channel geometries of Rocky Branch at the Catawaba gage and bridge crossing locations.**

| Location | Bottom Width, $B$ (m) | Bank Slope, $z$ | Bed Slope, $S_0$ | Roughness Coefficient, $n$ |
|---|---|---|---|---|
| Gage at Catawaba St. | 1.8 | 2.0 | 0.02 | 0.03 |
| Norfolk Southern crossing at Sumter St. | 3.0 | 1.0 | 0.02 | 0.02 |

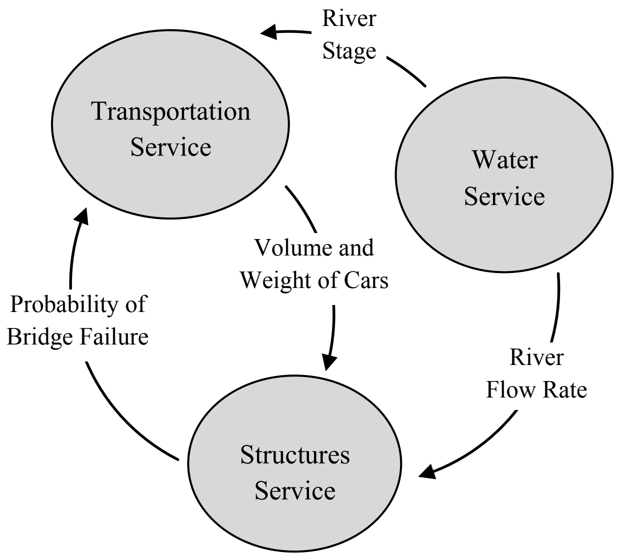

**FIG. 1. Urban infrastructure system as a set of services with data communications across standardized interfaces.**

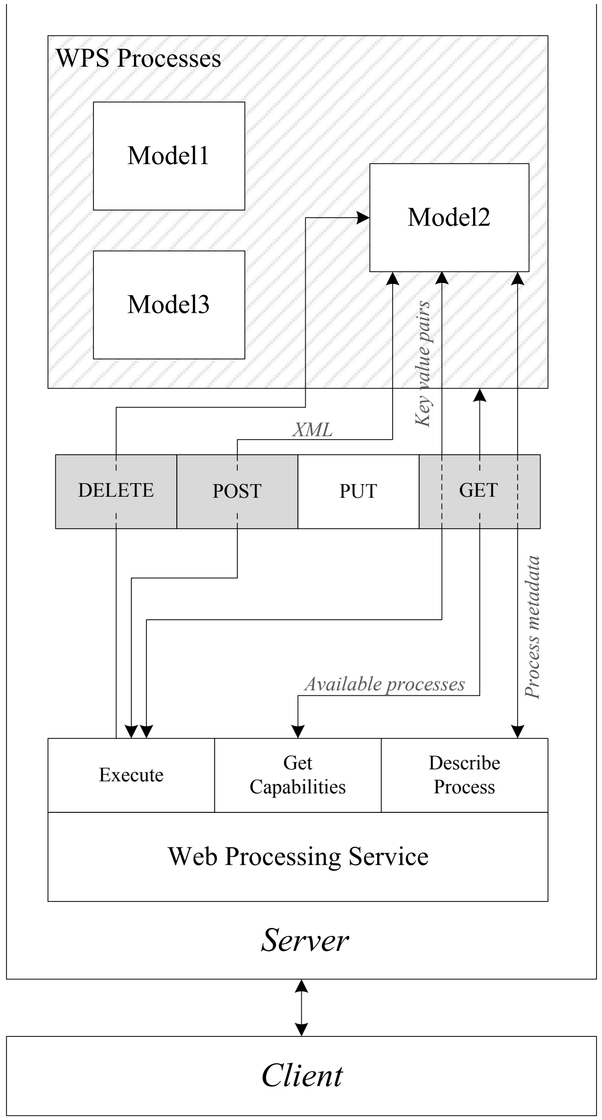

**FIG. 2. Server-side implementation of models using OGC Web Processing Service (WPS) and REST.**

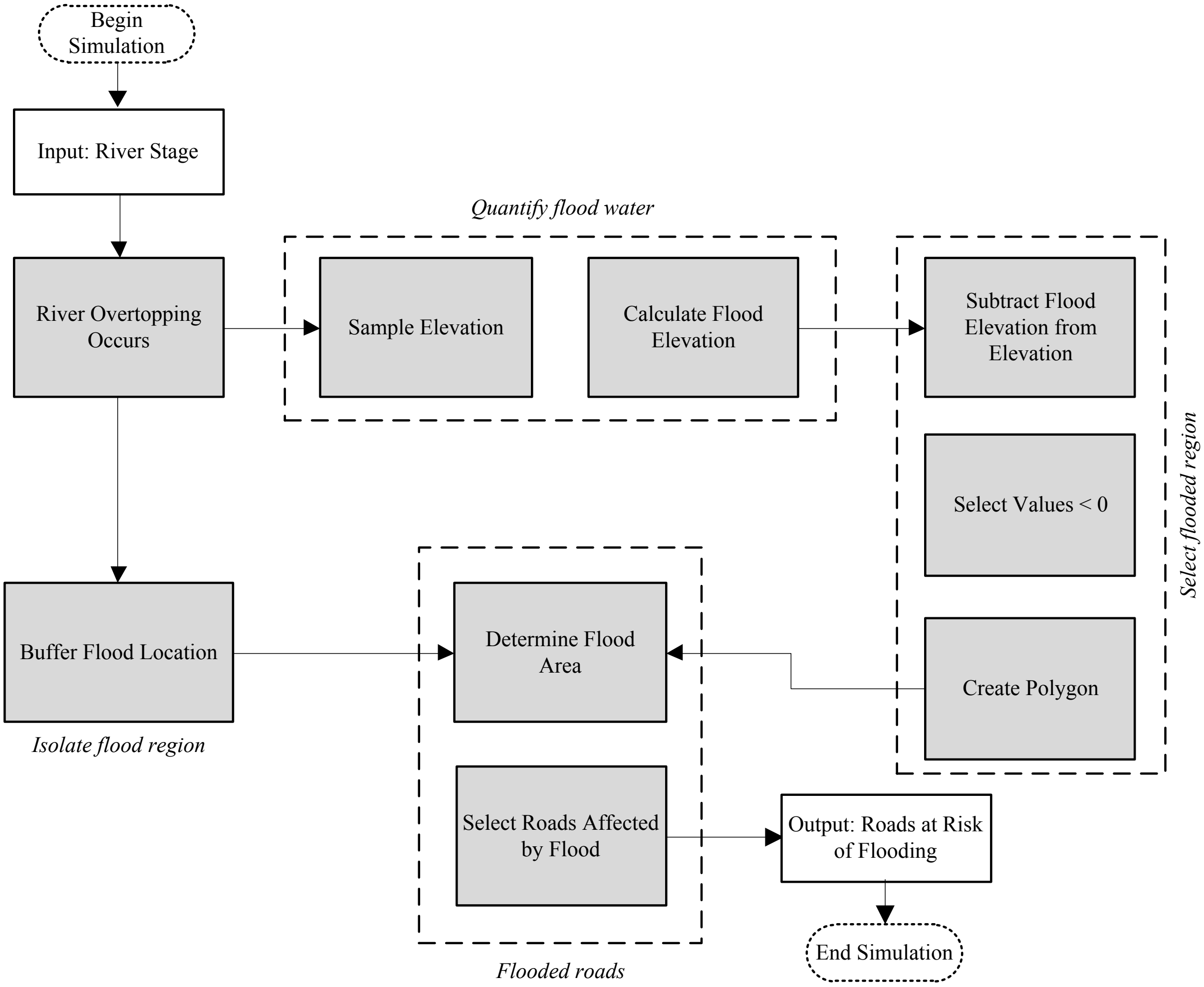

**FIG. 3. A flowchart of the GIS-based calculation for flood inundation.**

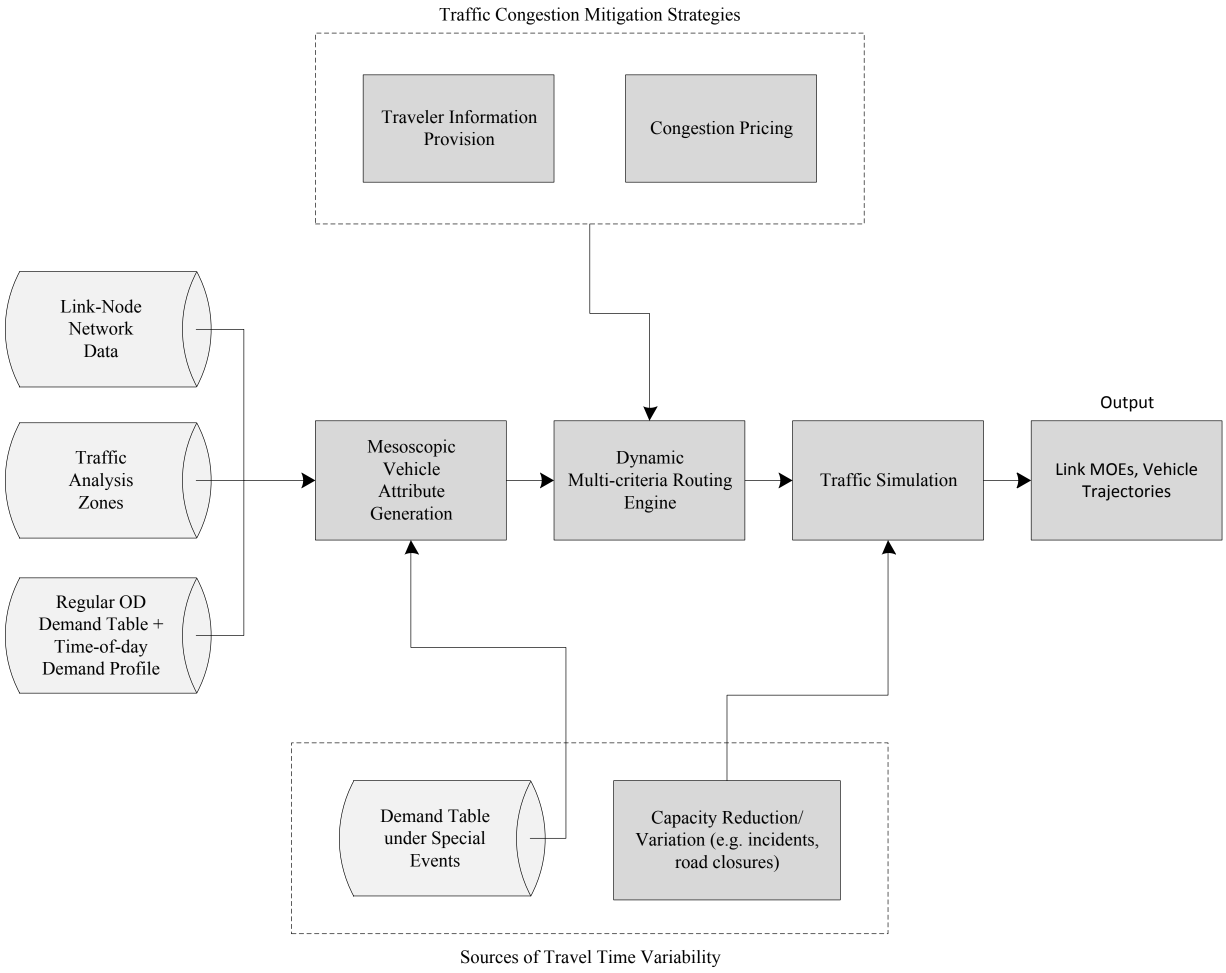

**FIG. 4. System architecture of the open-source Dynamic Traffic Assignment Simulation Engine.**

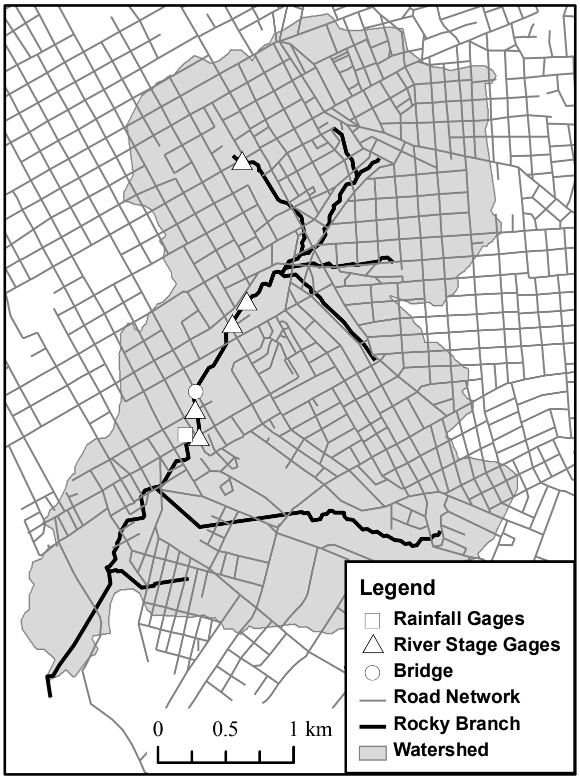

**FIG. 5. The study area: Rocky Branch Watershed in Columbia, South Carolina. Data from U.S. Census Bureau (2012) and derived from U.S. Geological Survey (2014).**

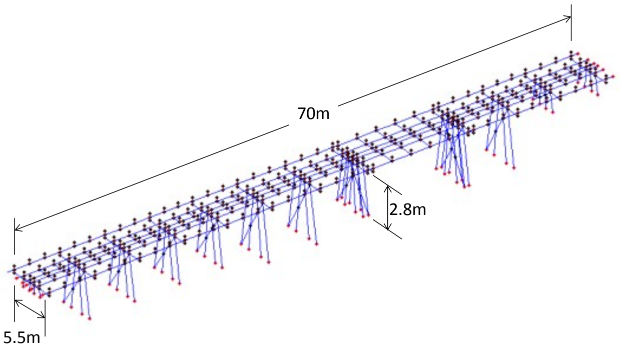

FIG. 6. Finite element model of the railway bridge.

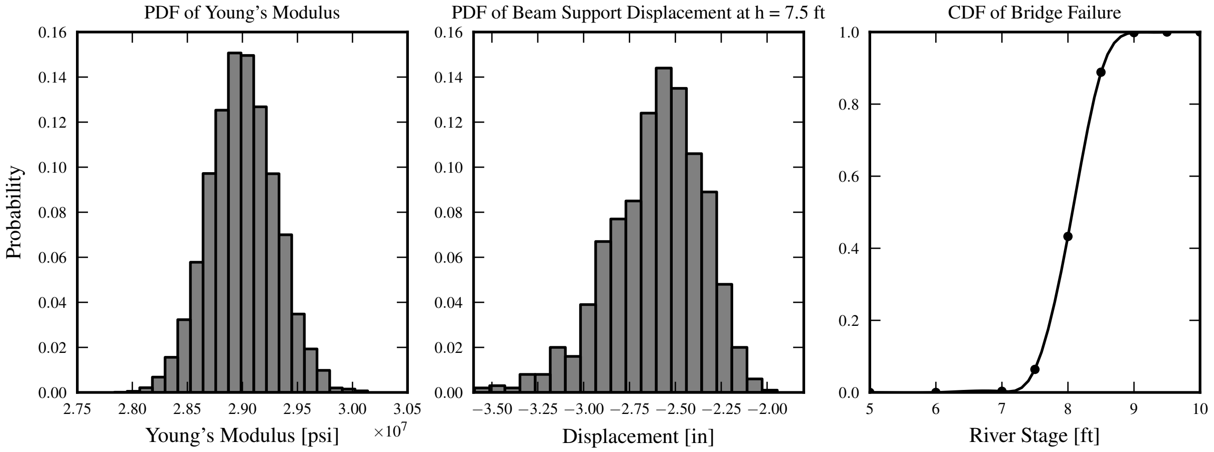

**FIG. 7. Probability Density Functions (PDFs) of Young's Modulus, the displacement of the beam at h = 7.5 ft, and the Cumulative Distribution Function (CDF) of the probability that the bridge will fail as a function of the river stage.**

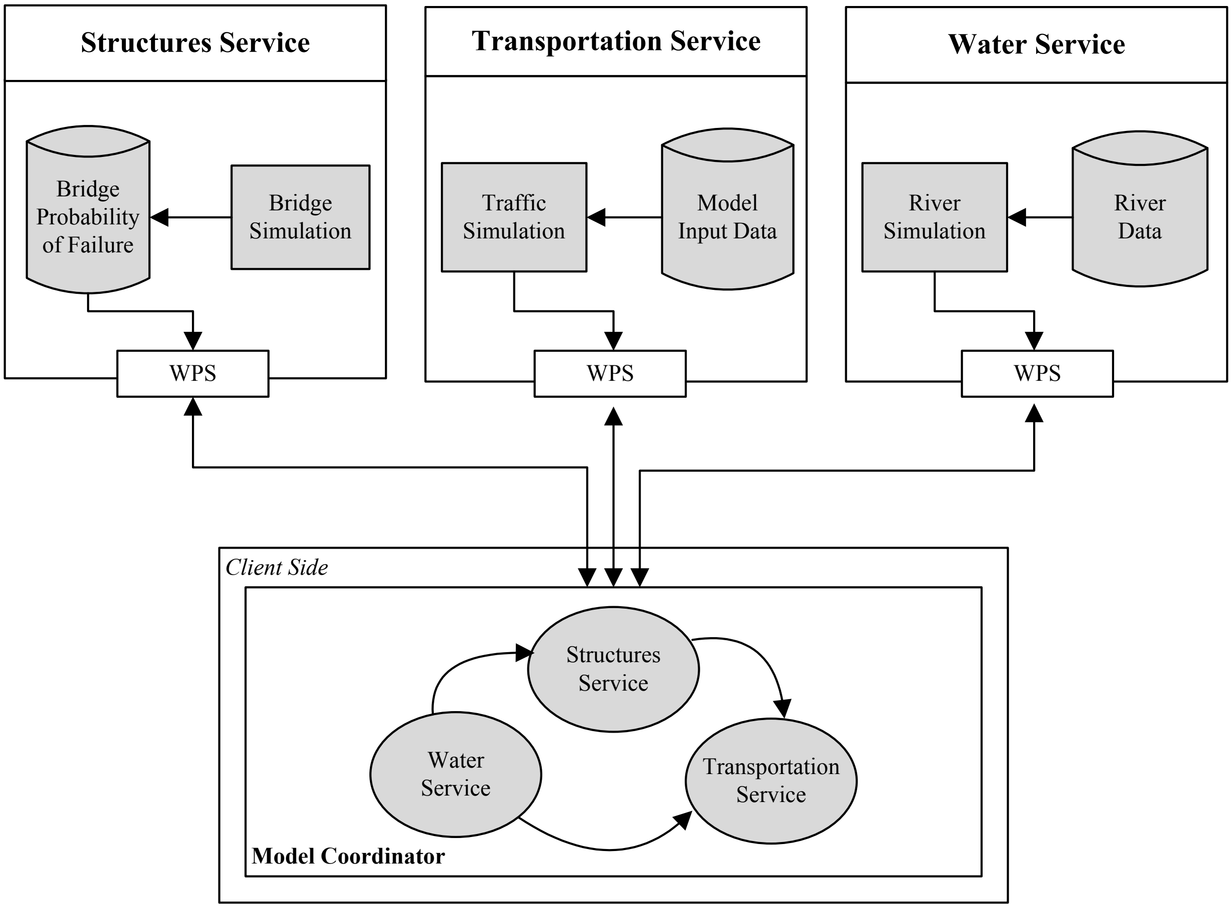

FIG. 8. Overall system architecture where structures, transportation, and water services are consumed and data exchanges are orchestrated using a client-side model coordinator.

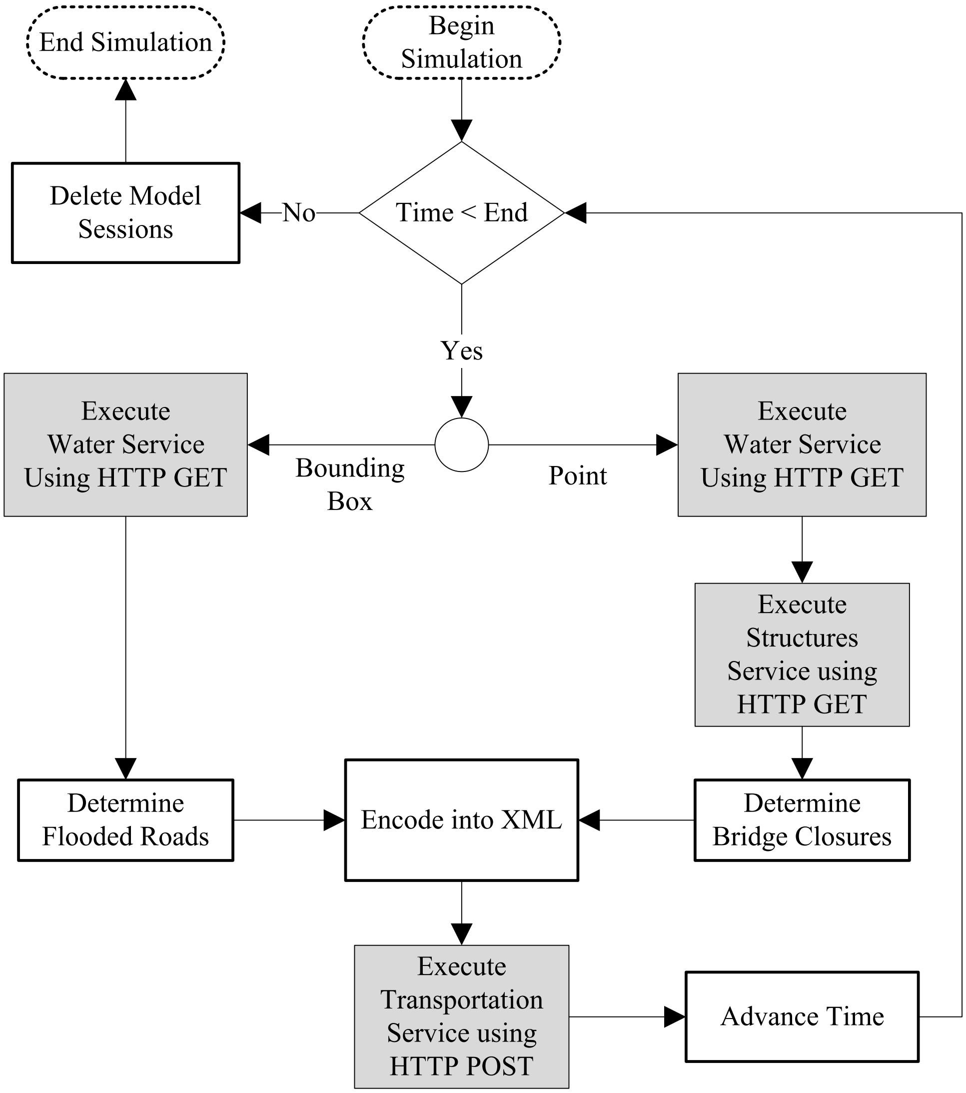

**FIG. 9. Flowchart describing the data communication between the water, structures, and transportation web services including logic required to translate information between services.**

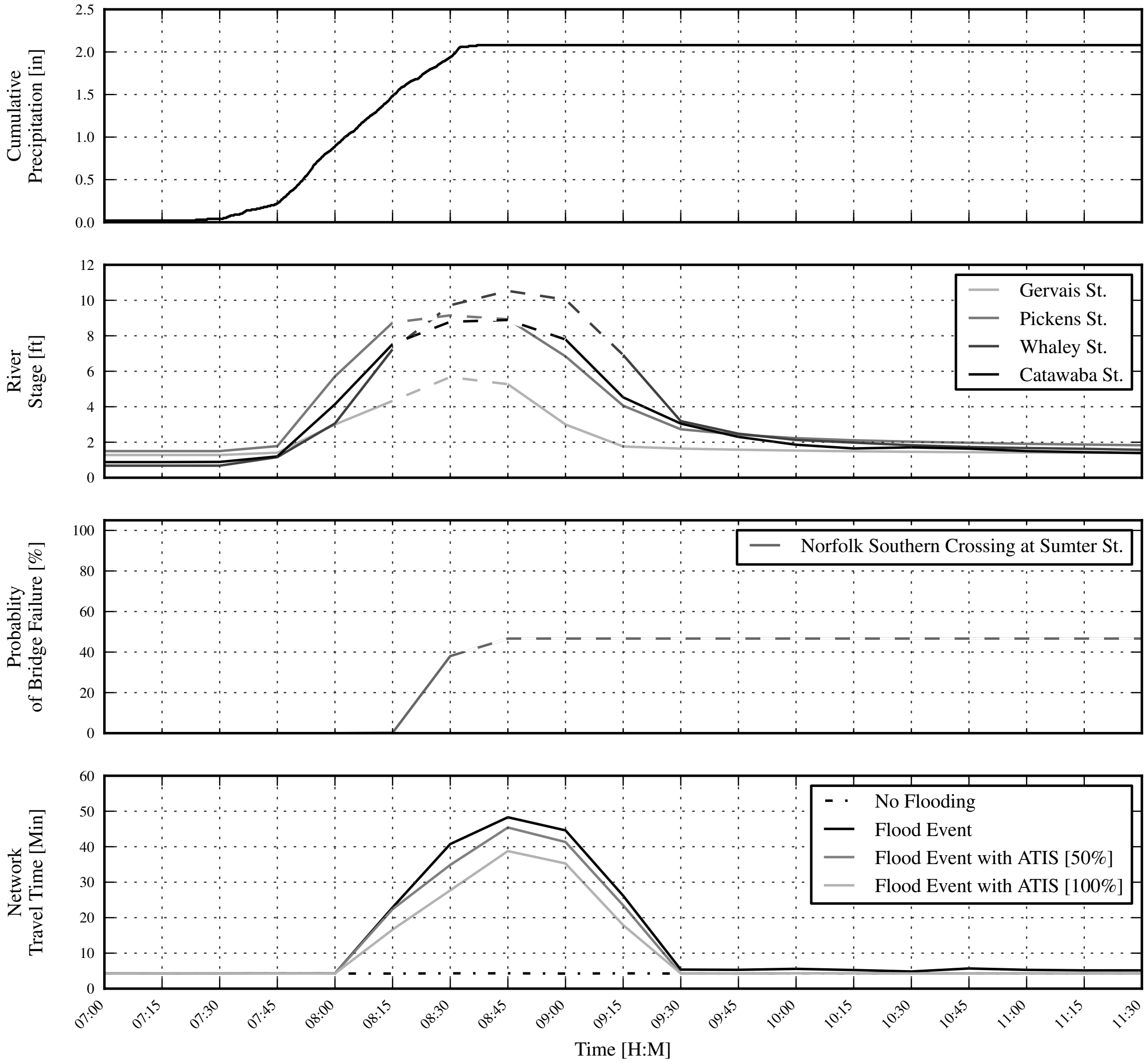

**FIG. 10. Results for each web service process within the infrastructure model system, for storm event on September 23, 2011. Also included is the measured precipitation that drives the physical system. Dashed lines in the river stage figure indicate out-of-bank conditions and dashed lines in the probability of bridge failure figure indicate bridge closure conditions.**