

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

Jonathan L. Goodall<sup>1</sup>, Member, ASCE

Anthony M. Castronova<sup>2</sup>

Nathan N. Huynh<sup>3</sup>

Juan M. Caicedo<sup>4</sup>, Member, ASCE

Ryan Hall<sup>5</sup>

## 1 ABSTRACT

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

---

<sup>1</sup>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

<sup>2</sup>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

<sup>3</sup>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

<sup>4</sup>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

<sup>5</sup>Engineering Associate, Alliance Consulting Engineers, Inc., P.O. Box 8147, Columbia, S.C. 29202-8147, rhall@allianceCE.com

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

## 17 INTRODUCTION

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

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

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

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

## 56 **BACKGROUND**

57 Past research has shown that civil infrastructure systems may generate complex and  
58 counter-intuitive responses that are undesirable, unpredictable, and compromise the re-  
59 siliency of the system (Rinaldi et al. 2001). This complexity can cause cascading failures  
60 throughout the system in ways that are not immediately intuitive (and therefore predictable)  
61 (Amin 2002; Little 2003). Rinaldi et al. (2001) stressed the need to understand the in-  
62 terdependency of infrastructure systems that are connected as a “system-of-systems” with  
63 bidirectional relationships between system components that produce complex relationships  
64 characterized by feedback and feedforward paths, and intricate branching topologies. These  
65 characteristics of infrastructure systems have motivated researchers to argue that future civil

66 engineers should be master integrators with a view that civil infrastructure is a complex,  
67 holistic system (Bordogna 1998; Folke 2006), able to understand the complex computer and  
68 information technologies including sensing techniques, data models, and data mining ca-  
69 pabilities needed to design and maintain Advanced Infrastructure Systems (AIS) (Garrett  
70 2005).

71 Prior work for integrating civil infrastructure systems has ranged from techniques for sim-  
72 ulating system interactions, to techniques for understanding the sustainability of systems, to  
73 approaches for fostering communication and integration of systems. Simulation techniques  
74 presented in the literature include agent-based modeling (Sanford Bernhardt and McNeil  
75 2008), dynamic programming (Kuhn 2010), and network analysis (Ash and Newth 2007;  
76 Tran et al. 2010). Sustainability of civil infrastructure systems has primarily been addressed  
77 by using life-cycle analysis (Racoviceanu and Karney 2010; Francis et al. 2011). Communica-  
78 tion and integration approaches have included the design of management systems (Halfawy  
79 and Eng 2008), peer-to-peer communication (Zhang et al. 2010), and enterprise-level geo-  
80 graphic information system approaches (Pradhan et al. 2007). Integration of infrastructure  
81 systems has often adopted a network theory approach that relates network properties to  
82 reliability measures (Dueñas-Osorio and Vemuru 2009; Ouyang and Dueñas-Osorio 2011)  
83 or considers infrastructure as multilayer networks interlinked based on physical and socio-  
84 economic factors (Chang et al. 2002; Zhang et al. 2005; Zhang and Peeta 2011). Past work  
85 has also focused on the specific problem of municipal infrastructure systems. For example,  
86 Halfawy and Eng (2008) presented a discussion of the main challenges and proposed specific  
87 solutions for implementing integrated Municipal Infrastructure Management Environments  
88 (MIMEs). The proposed solutions focus on industry standard data integration and software  
89 interoperability standards in the municipal infrastructure domain, and considers existing  
90 standards and their harmonization, refinement, and integration.

91 Prior work on using web services in civil engineering has focused on applications to  
92 both built and natural systems. Vacharasintopchai et al. (2007) presented an architectural

93 framework for the application of the Semantic Web Services technology in computational  
94 mechanics. The work was motivated by the perceived need in the computational mechanics  
95 community to allow various groups of programmers to work collectively to create a sim-  
96 ulation that could be deployed using heterogeneous platforms. Halfawy (2010) presented  
97 municipal integration work that builds from Halfawy and Eng (2008) and proposed web  
98 service components and Geography Markup Language (GML) data standards for MIMEs.  
99 Liu et al. (2003) presented an innovative vision for applying ubiquitous computing where  
100 devices are universally accessible through information services and parties are able to ac-  
101 cess information services to foster collaboration. Liu et al. (2005) expanded on prior work  
102 by presenting an experimental service-composition paradigm for integrating loosely coupled  
103 software components that employs a distributed data-flow approach. Results from this work  
104 suggested that a distributed data flow approach is superior when data volumes are large,  
105 but when data volumes are low, more traditional approaches outperform their proposed ap-  
106 proach. There has also been work in the water resources community to use web services for  
107 integrating heterogeneous databases (Goodall et al. 2008; Horsburgh et al. 2009) and for  
108 model integration (Goodall et al. 2011).

109 This paper expands on past research by focusing specifically on the challenge of inte-  
110 grating multidisciplinary models across civil infrastructure systems using a service-oriented  
111 architecture. The technical approach is most similar to that of Liu et al. (2003) where we  
112 envision civil infrastructure as a distributed system of components that are published as  
113 services to enable interactions based on system interdependencies. Distributed systems are  
114 common across many fields, and for this reason researchers have created tools for managing  
115 distributed systems for business, science, and engineering applications (Foster et al. 2002;  
116 Papazoglou and Georgakopoulos 2003; Foster 2005). Taking the system-of-systems approach  
117 for interdependent critical infrastructures described by Rinaldi et al. (2001) and using the  
118 definition of a system-of-systems as consisting of “multiple, heterogeneous, distributed, oc-  
119 casionally independently operating systems embedded in networks at multiple levels, which

120 evolve over time” introduced by DeLaurentis (2007), Eusgeld et al. (2011) proposed the idea  
121 of hierarchical level architecture for capturing the complexity of system-of-systems. In this  
122 architecture, the lowest level represents system models of a single infrastructure, the middle  
123 level represents interactions between lower level systems, and the high level represents the  
124 global system-of-systems (Eusgeld et al. 2011). In relation to past approaches of interde-  
125 pendent infrastructure systems, Nan and Eusgeld (2011) suggested that the two best known  
126 approaches are complex network theory and object-oriented modeling. Within this context,  
127 our work builds on object-oriented modeling approaches where a system-of-systems is rep-  
128 resented as low level models of single infrastructure systems, which are individually exposed  
129 as web services, and then integrated into a middle level systems through data exchanges  
130 representing system interdependencies.

## 131 **METHODOLOGY**

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

### 141 **Design of Models as Web Services**

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

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

159 REST is a general standard for providing interoperability between clients and servers that  
160 can benefit from an additional layer of software to provide domain specificity (Foster 2005).  
161 Examples of domain specific web service standards applicable to civil infrastructure systems  
162 are those from the Open Geospatial Consortium (OGC). OGC is a non-profit organization  
163 that has created widely used standards for publishing geospatial data. These standards  
164 are used in conjunction with the lower level service-oriented architecture standards, such  
165 as REST or SOAP, and provide additional specificity for the interactions between client  
166 applications and web services when dealing with geospatial data (Kiehle 2006). The Web  
167 Processing Service (WPS) is one of the OGC standards and is designed for performing  
168 server-side data processing operations (Schaeffer 2008). The WPS standard consists of three  
169 methods: GetCapabilities, DescribeProcess, and Execute. The GetCapabilities method is  
170 used by the client to query metadata that describes the processing services provided by the  
171 server (Schut and Whiteside 2007). This functionality is implemented at the server level and  
172 retrieves information about all available web processes in a single call. The DescribePro-  
173 cess and Execute methods are implemented at the process level, meaning they are unique

174 to each web process on the server. The DescribeProcess operation provides the client with  
175 metadata describing a specific process (Schut and Whiteside 2007). This operation is useful  
176 for determining the required inputs, as well as the outputs calculated by a process. Fi-  
177 nally, the Execute operation enables a client to specify inputs and run a web process (Schut  
178 and Whiteside 2007). The Execute operation can return structured data such as XML or  
179 JavaScript Object Notation (JSON), as well as various file formats (e.g. Network Common  
180 Data Form (NetCDF), JPG, TIFF, etc.). Furthermore, the data can consist of three pos-  
181 sible output types: literal, complex, or boundingbox. Since WPS supports XML, it is also  
182 possible to encode data using a specific markup schema, for example geographic data using  
183 the Geography Markup Language (GML).

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



201 the DELETE command using the same URI.

## 202 **Implementing Models as Web Services**

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

### 213 *Water Service*

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

$$Q = \frac{1.49}{n} AR^{\frac{2}{3}} S_f^{\frac{1}{2}} \quad (1)$$

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

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

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

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

### 243 *Transportation Service*

244 The transportation web service utilizes DTALite, an open-source dynamic traffic as-  
 245 signment model (<https://sites.google.com/site/dtalite/home>). The system architecture of  
 246 DTALite is shown in Fig. 4. As shown, DTALite, like other Dynamic Traffic Assignment  
 247 (DTA) models, takes as input a transportation network and an Origin-Destination (OD)

248 matrix that specifies trips between traffic analysis zones (in this study, the OD reflects the  
249 morning rush hours in downtown Columbia, SC), models the traffic flow evolution in a net-  
250 work using advanced network algorithms and trip-maker behavior models, and provides as  
251 output link and path travel times. DTALite can also capture the effect of road closures,  
252 real-time traffic information provision via Advanced Traveler Information System (ATIS),  
253 and routing of traffic via Dynamic Message Signs (DMS).

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

267 DTALite was deployed as a web service by creating a Python wrapper designed to interact  
268 with the DTALite application when invoked by client applications. It was assumed that the  
269 necessary input files are pre-loaded onto the server. During model simulation, the client  
270 application invokes the DTALite web resource (i.e. the Python wrapper) using the POST  
271 method. This enables the client to send road closures data, encoded in eXtensive Markup  
272 Language (XML) to the DTALite web resource. At this time, the client can also designate  
273 if DMS or an Advanced Traffic Information System (ATIS) will be used. These data are  
274 received by the web resource and saved within specific DTALite input files. Next, the web

275 resource executes the DTALite application, extracts the results from an output file, and sends  
276 them back to the client. This approach enables the client to change simulation parameters  
277 such as road closures, assuming that the model was created first on a desktop computer and  
278 then upload to the web server.

### 279 *Structural Reliability Service*

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

289 The scour process is very complex (Yanmaz et al. 1991; Richardson and Panchang  
290 1998), and the modeling and simulation of scour in bridge foundations is an active area of  
291 research. A complex scour model could be included as a separate component of the proposed  
292 simulation framework. However, we assumed a simplified model where the stiffness of the  
293 foundation is a function of the river's stage. The model is inspired by results reported in  
294 the Federal Highway Administration HEC-18 report (Richardson et al. 1993). The report  
295 summarizes the scour depth of different models as a function of river stage and the models  
296 indicate an exponential relationship between scour depth and stage. Given that stiffness  
297 of the foundation and scour are inversely proportional, the stiffness of the foundation is  
298 modeled using the equation

$$k(h) = R * k_s * exp(-a * h) \quad (3)$$

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

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

316 The probability of failure of the bridge to a particular river stage is calculated before  
317 the simulation starts using an in-house Matlab finite element toolbox. The results of this  
318 simulation are stored in a database and the structural reliability service provides client  
319 applications access to these data. We used this data staging approach because the structural  
320 model is time consuming and does not easily lend itself to on demand processing. However,  
321 the service could be implemented such that the reliability of the structure is calculated  
322 as requested, if needed, or so that the Matlab program is running “behind the scenes” to  
323 update the SQL database if any changes are made in the assumptions of the analysis. The  
324 structural reliability service could also be complemented with structural health monitoring  
325 services using global or local structural health monitoring techniques (Brownjohn 2007)

326 through future work. The simulation would then be configured to let the client consume  
327 the structural health monitoring service directly, or to let the structural reliability service  
328 consume structural health monitoring service to enhance the calculations of the probability  
329 of damage.

## 330 **EXAMPLE APPLICATION**

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

### 336 **Study Area**

337 The Rocky Branch Watershed (Fig. 5) drains 11 km<sup>2</sup> of Columbia, SC. The watershed  
338 is urbanized consisting of commercial districts, university property, and residential neigh-  
339 borhoods. Due in part to urbanization of the watershed, high intensity storms often cause  
340 flooding within the watershed. During these flooding events, roads at low lying areas must  
341 be closed by the city as a safety precaution. We have selected this watershed because the  
342 frequent flooding that the City of Columbia is attempting to alleviate. However, the ap-  
343 proach and tools developed through this study are applicable to other challenges in urban  
344 infrastructure integration beyond this specific example application.

345 Existing data for the watershed includes stream gaging stations maintained by the USGS  
346 and the University of South Carolina, as well as a weather station maintained by the Univer-  
347 sity of South Carolina (Fig. 5). Data collected at the USGS stations are available through  
348 the National Water Information System and include river stage on a 15-minute time interval.  
349 One river stage and one rainfall gauge capable of recording observations on a 15-second inter-  
350 val were installed as part of this study. This high frequency data is important because of the  
351 quick response of the watershed to rainfall events (i.e., the time to peak streamflow can be  
352 30 or 45 minutes from the beginning of the rain storm). The river stage at the USC gage is

353 measured using a KPSI Submersible Hydrostatic Level Transducer and rainfall is measured  
354 using a stainless steel tipping bucket rain gage manufactured by Sutron Corporation. The  
355 rain gage has an orifice diameter of 7.87 inches (20 cm) and measures rainfall in 0.01 inch  
356 increments, to an accuracy of 2%.

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

366 A rail bridge crossing the Rocky Branch Watershed and Sumter Street is the focus of  
367 the structural reliability service. A finite element model of the bridge (Fig. 6) was used to  
368 calculate the bridge displacements. The model has 664 nodes, 480 beam elements, 312 rigid  
369 links, 228 lumped masses and 32 linear springs to model the interaction with the soil using  
370 a Winkler model.  $R$  (Equation 3) was defined with a mean of one and a standard deviation  
371 of 0.1. This is within the limits reported on the literature for the variation of the stiffness  
372 of soils (Jones et al. 2002). The bridge model was loaded with Hopper cars that have a  
373 load capacity of 224,500 lb traveling at low speed because the bridge is located in an urban  
374 area. Uncertainty in the loads was considered by assuming that the load of each car follows  
375 a normal distribution of mean 224,500 lb and a standard deviation of 11,225 lb. The state of  
376 the structural component was also considered uncertain and a normally distributed Young  
377 modulus with mean of 29,000 ksi and a standard deviation of 300 ksi. The yielding stress  
378 was considered as a normal random variable with a mean of 50 ksi and a standard deviation  
379 of 5 ksi. Simulation results indicate that the structure will not reach yielding at any stage

380 level. Therefore, the limit state corresponding to the yield stress does not control. The limit  
381 stage for maximum displacement was considered as 3 inches and determined the failure of  
382 the structure. Fig. 7 shows a representative example of the histograms for these random  
383 variables (Young's modulus), a Probability Density Function (PDF) of the displacement of  
384 the bridge at the point of interest, and the probability of failure (displacement greater than  
385 3 inches) as a function of the river stage.

## 386 **System Integration**

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

397 The data flow during the integrated infrastructure simulation (Fig. 9) consists of multiple  
398 web service calls, all mediated by a client-side controller. In this study, simulation begins by  
399 issuing the WPS Execute operation using the HTTP GET method on the water service and  
400 providing a bounding box as input. This triggers the water service to produce output for all  
401 known locations within the requested rectangle, and returns these values to the client. The  
402 calculated flood stage is then used to determine which roads are affected by the flood water.  
403 This calculation requires GIS processing, and is currently performed on the client machine.  
404 Next, the Execute operation is issued once again on the water service, only this time specific  
405 locations are provided as input. The locations represent areas of interest, namely locations  
406 in which the river water impacts bridge supports. The water service returns river stage and



407 streamflow calculations for the gages nearest to these requested bridge locations. Since the  
408 river stage measured at the streamflow gages likely differs from the stage at the bridges,  
409 additional calculations are performed by the model coordinator to translate streamflow at  
410 the gage location into river stage at the bridge location. This is done by first assuming  
411 that the flow rate at both locations is the same. Next, using simplified trapezoidal channel  
412 geometries, shown in Table 1, Manning’s equation was used to back calculate river depth  
413 at the bridge from the known streamflow at the nearest gage. This new river stage is then  
414 supplied as input for the bridge service by invoking the WPS Execute operation using the  
415 HTTP GET method. Using this data, in addition to known bridge properties, the bridge  
416 service calculates the probability that a specific structure will fail. These calculations are  
417 sent back to the client-side controller which determines if it is necessary to close the bridge  
418 (and neighboring roads) due to safety concerns. All road closures are aggregated and sent to  
419 the transportation service by invoking the WPS Execute operation using the POST method,  
420 which enables input data to be supplied within an eXtensible Markup Language (XML)  
421 encoding. The transportation service uses this input to calculate the average travel time  
422 through the road network. This sequence continues over the entire simulation time horizon.

## 423 **Simulation Results**

424 The example application simulation captures the dynamic interaction and interdepend-  
425 encies of the subsystems (i.e. water resource, transportation, and bridge). The results  
426 calculated by each subsystem at every designated time step over the course of the simulation  
427 horizon (Fig. 10) show that at three separate gauging locations, the river stage continues  
428 to increase until overtopping and subsequent road closures occur, indicated by the dashed  
429 lines. Once the river stage decreases and flooding subsides, roadways are reopened to the  
430 public. Similarly, one bridge in the test network is severely affected by the forces imposed by  
431 rising water levels, to the extent that the probability of failure approaches 50%. When the  
432 probability of failure reached 40%, the bridge and neighboring roadways were closed due to  
433 safety concerns. This threshold is dependent on the bridge owner’s preferences and can be

434 adjusted accordingly. Unlike the previous scenario, the bridge and neighboring roads were  
435 not reopened to the public because the bridge would have to undergo an official evaluation  
436 before deemed safe.

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

## 456 **DISCUSSION**

457 This paper aims to better understand the benefits and challenges of applying a service-  
458 oriented architecture for representing civil infrastructure as a “system-of-systems.” The key  
459 benefit is that service-oriented architectures allow modelers to combine heterogeneous com-  
460 puting resources into an integrated system. Due to the loose-coupling nature of web services,

461 each model maintains its independence, which is a benefit because the core development team  
462 can maintain control over the model and therefore it can evolve over time within the larger  
463 modeling system. Models in this system are meant to be generic and can be applied across  
464 a variety of use cases (the example application in this paper presented one potential use  
465 case). Therefore, while the service logic described in this paper may be overly complex for  
466 the particular example application demonstration, each service might be used for multiple  
467 applications in a production environment, thus requiring a sophisticated implementation not  
468 tailored to this specific use case. That said, if the architecture was deployed for a single use  
469 case, services could be tailored to match the desired level of complication required by the  
470 use case.

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

487 A key contribution of this work is the translation needed to integrate low-level infras-

488 structure systems into a higher-level system-of-systems. We achieved this goal by creating  
489 workflows that transformed information provided by a model service into the specific input  
490 needed by a second service. In our example application, the water service provides only  
491 water heights and flows at specific locations, yet the transportation and structure services  
492 required information at other locations within the study area. To transform this information,  
493 we developed client-side code that uses the water height information provided by the water  
494 service within a Geographic Information System (GIS) framework to determine road clos-  
495 ings. Therefore, geospatial referencing of information provided the context for translating  
496 information between the models. This road closing information can then be fed to the trans-  
497 portation service for traffic re-routing. In a more general sense, this technique demonstrates  
498 the benefit of general services that can be used by client-side code that acts as a communica-  
499 tion mediator between different models of civil infrastructure subsystems. However, it also  
500 demonstrates the cost of having to maintain more sophisticated client-side code to perform  
501 the system integration.

502 The key challenge in making a service-oriented approach work for civil infrastructure  
503 systems is the need to provide exact specification for interface standards and data exchanges.  
504 In this work, we used the OGC WPS as the interface standard. As we discussed before, the  
505 web processing service was not specifically designed as a way of exposing models, but rather  
506 as a means for exposing similar data processing routines as web services (Goodall et al. 2011).  
507 We believe that a new modeling web service standard would be ideal; however in this work  
508 we conformed to existing standards in order to be compliant with existing approaches and to  
509 leverage existing tools such as PyWPS. In addition to interface standards, there is the need to  
510 clearly define data exchanges that go beyond describing objects, but also describe semantics  
511 in a consistent way so that mismatches between systems can be overcome. To accomplish  
512 this would require identifying existing or construct new ontologies for each discipline (water,  
513 structures, transportation) to facilitate both syntactic and semantic mediation within the  
514 integrated system simulation. Each service can, and most likely will, be implemented on its

515 own spatial and/or temporal grid or use different internal geospatial data representations  
516 (lines, polygons, and volumes) and their own internal vocabularies and semantics. The  
517 key to simulating a system where services have different spatial or temporal scales is to  
518 fully document data exchanges according to a well defined ontology, and to include data  
519 transformations to rescale the output of one component to satisfy the input needs of another  
520 component.

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

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

540 Finally, fault tolerance is particularly important in a service-oriented architecture because  
541 of the variety of ways in which the system could be susceptible to faults including network

542 outages, failed sensors, temporarily unavailable servers or services, either intentional or un-  
543 intentional misuse of the system. These issues must be anticipated and properly handled.  
544 Many of these challenges can be easily handled with proper software engineering to ensure  
545 graceful handling of system faults. However, there is no way to guarantee that all compo-  
546 nents of the system will be functional at all times, and creating a highly reliable system  
547 could become cost prohibitive if it requires multiple servers for backup and load balancing.

## 548 **CONCLUSIONS**

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

564 A second conclusion from this work is that a domain-specific software layer is beneficial  
565 for abstracting the modeler from the lower-level web service protocols. Our implementation  
566 approach made use of REST as the lower-level web service protocol and the Open Geospatial  
567 Consortium (OGC) Web Processing Service (WPS) standard as the domain-specific software  
568 layer. We considered alternative approaches including using REST alone or creating our own

569 domain-specific software layer. However, we found the OGC WPS useful in providing an  
570 additional layer of abstraction between the modeler and the lower-level web service protocols.  
571 Furthermore, while we needed to modify the OGC WPS for our purposes, this modification  
572 is a more effective solution than building a custom solution for civil infrastructure systems.

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

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

## 596 ACKNOWLEDGMENTS

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

## 599 REFERENCES

- 600 Amin, M. (2002). “Modeling and control of complex interactive networks [guest editorial].”  
601 *Control Systems Magazine, IEEE*, 22(1), 22–27.
- 602 Ash, J. and Newth, D. (2007). “Optimizing complex networks for resilience against cascading  
603 failure.” *Physica A: Statistical Mechanics and its Applications*, 380, 673–683.
- 604 Bordogna, J. (1998). “Tomorrow’s civil systems engineer—the master integrator.” *Journal*  
605 *of Professional Issues in Engineering Education and Practice*, 124(2), 48–50.
- 606 Brownjohn, J. (2007). “Structural health monitoring of civil infrastructure.” *Philosophical*  
607 *Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*,  
608 365(1851), 589–622.
- 609 Castronova, A., Goodall, J., and Elag, M. M. (2012). “Models as web services using the Open  
610 Geospatial Consortium (OGC) Web Processing Service (WPS) standard.” *Environmental*  
611 *Modelling & Software*, In Review.
- 612 Chang, S., Svekla, W., and Shinozuka, M. (2002). “Linking infrastructure and urban econ-  
613 omy: simulation of water-disruption impacts in earthquakes.” *Environment and Planning*  
614 *B*, 29(2), 281–302.
- 615 Chow, V. T., Maidment, D. R., and Mays, L. W. (1988). *Applied Hydrology*. McGraw-Hill  
616 Inc, Boston, Massachusetts.
- 617 Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2001). “Web services  
618 description language (WSDL) 1.1.” *Report no.*, World Wide Web Consortium.
- 619 DeLaurentis, D. (2007). “Role of humans in complexity of a system-of-systems.” *Proceedings*  
620 *of the 1st International Conference on Digital Human Modeling*, Springer-Verlag, 363–371.
- 621 Dueñas-Osorio, L. and Vemuru, S. (2009). “Cascading failures in complex infrastructure  
622 systems.” *Structural safety*, 31(2), 157–167.



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

625 Fielding, R. (2000). “Architectural styles and the design of network-based software architec-  
626 tures,” PhD thesis, University of California.

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

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

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

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

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

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

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

642 Halfawy, M. (2010). “Municipal information models and federated software architecture for  
643 implementing integrated infrastructure management environments.” *Automation in Con-  
644 struction*, 19(4), 433–446.

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

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

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

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

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

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

662 Liu, D., Cheng, J., Law, K., Wiederhold, G., and Sriram, R. (2003). “Engineering infor-  
663 mation service infrastructure for ubiquitous computing.” *Journal of Computing in Civil  
664 Engineering*, 17, 219.

665 Liu, D., Peng, J., Law, K., Wiederhold, G., and Sriram, R. (2005). “Composition of engi-  
666 neering web services with distributed data-flows and computations.” *Advanced Engineering  
667 Informatics*, 19(1), 25–42.

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

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

671 Nan, C. and Eusgeld, I. (2011). “Adopting hla standard for interdependency study.” *Relia-  
672 bility Engineering & System Safety*, 96(1), 149–159.

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

676 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

704 models for prediction of structural deterioration of storm-water pipe assets.” *Journal of*  
705 *Infrastructure Systems*, 16(2), 167–171.

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

708 U.S. Geological Survey (2014). National Elevation Dataset courtesy of the U.S. Geological  
709 Survey.

710 Vacharasintopchai, T., Barry, W., Wuwongse, V., and Kanok-Nukulchai, W. (2007). “Se-  
711 mantic web services framework for computational mechanics.” *Journal of Computing in*  
712 *Civil Engineering*, 21, 65.

713 Yanmaz, A. et al. (1991). “Study of time-dependent local scour around bridge piers.” *Journal*  
714 *of Hydraulic Engineering*, 117, 1247.

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

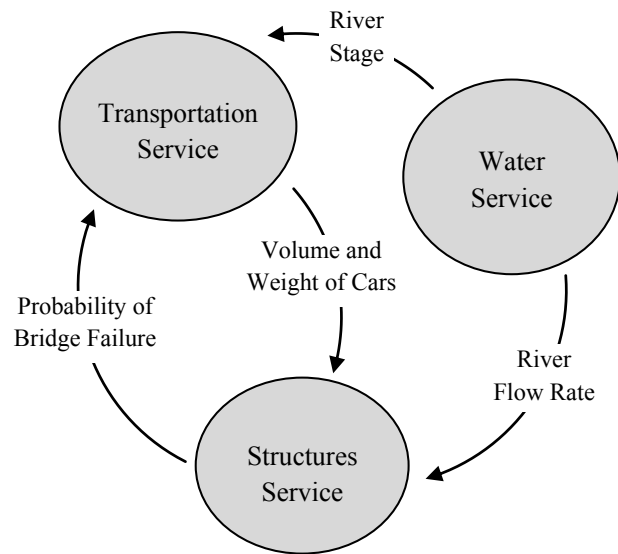
717 Zhang, G., Smith, B., and Guo, J. (2010). “Peer-to-peer-based publish/subscribe architecture  
718 for advanced infrastructure systems.” *Journal of Computing in Civil Engineering*, 24, 65.

719 Zhang, P. and Peeta, S. (2011). “A generalized modeling framework to analyze interdepen-  
720 dencies among infrastructure systems.” *Transportation Research Part B: Methodological*,  
721 45(3), 553–579.

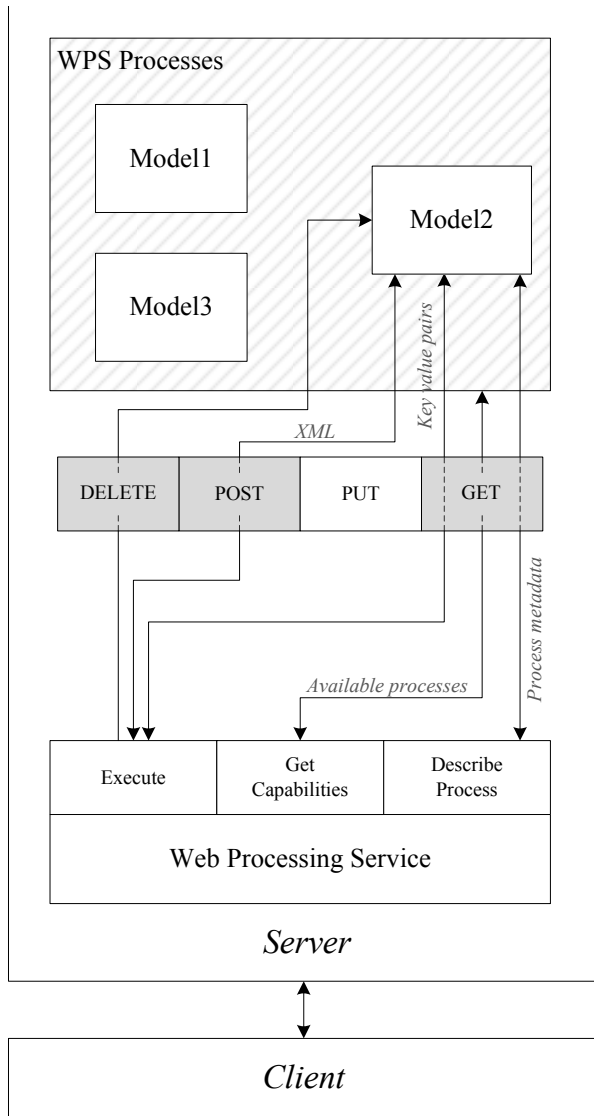
722 Zhang, P., Peeta, S., and Friesz, T. (2005). “Dynamic game theoretic model of multi-layer  
723 infrastructure networks.” *Networks and Spatial Economics*, 5(2), 147–178.

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

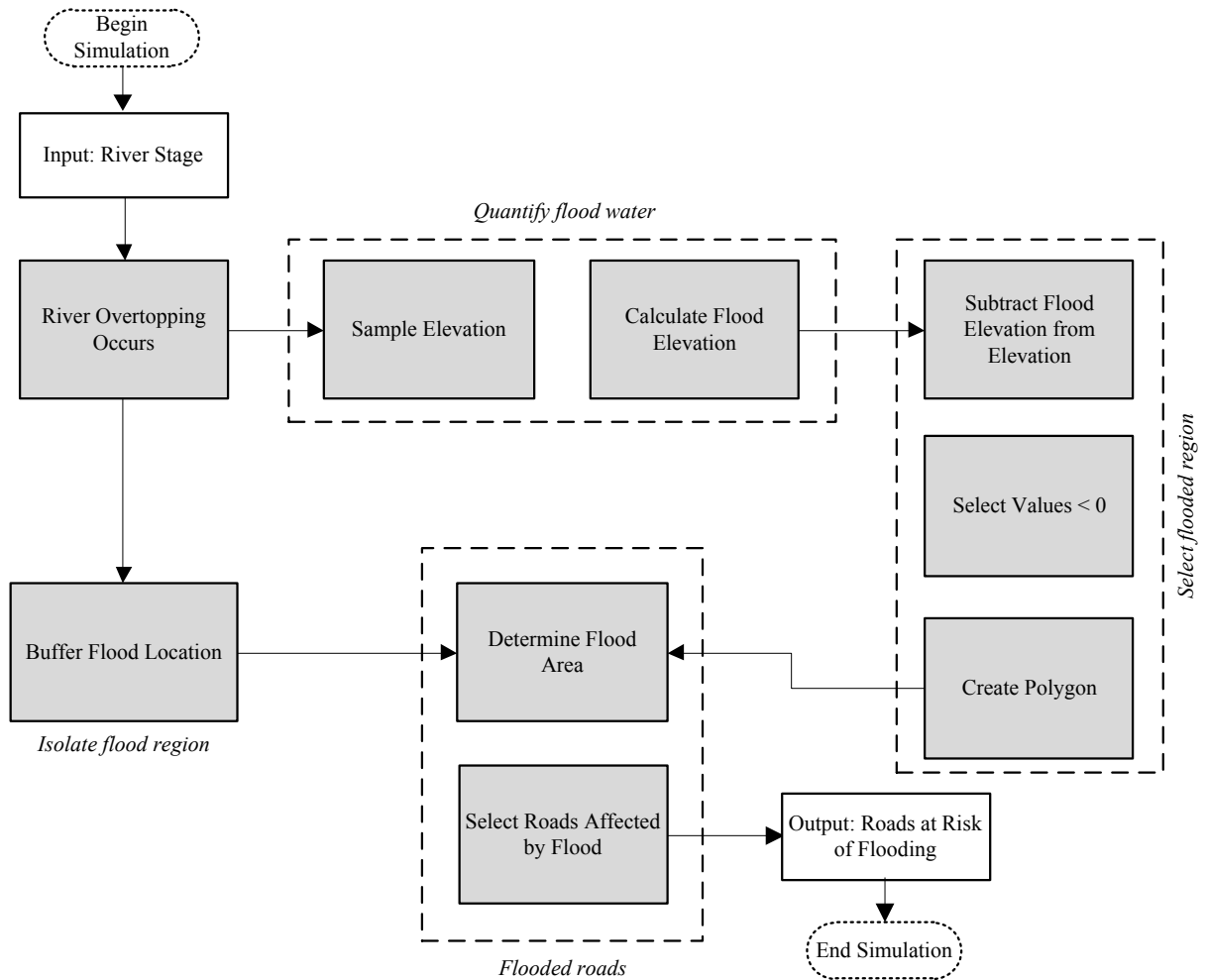
Location	Bottom Width, $B$ (m)	Bank Slope, $z$	Bed Slope, $S_0$	Roughness Coefficient, $n$
Gage at Catawaba St.	1.8	2.0	0.02	0.03
Norfolk Southern crossing at Sumter St.	3.0	1.0	0.02	0.02



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

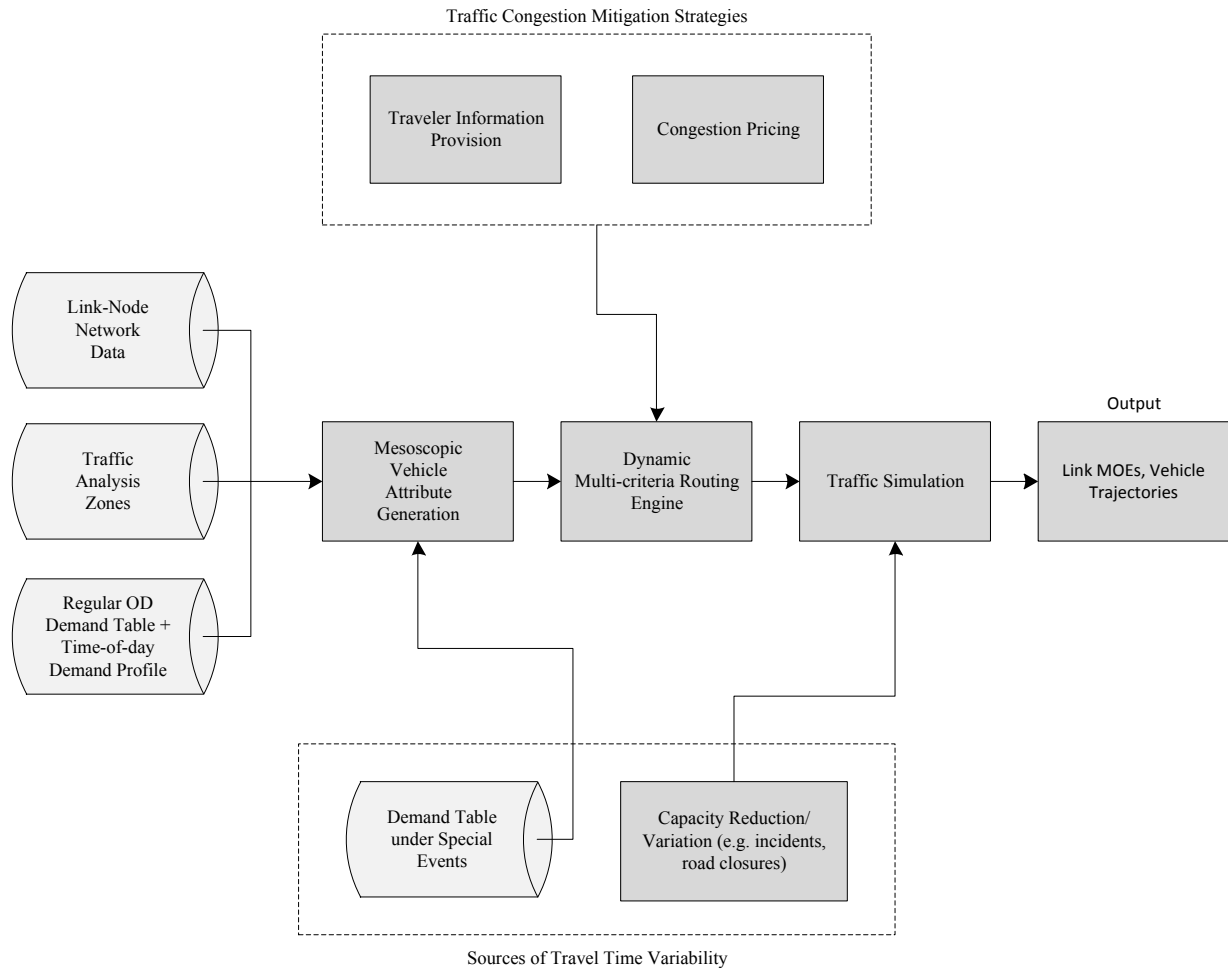


**FIG. 2. Server-side implementation of models using OGC Web Processing Service (WPS) and REST.**

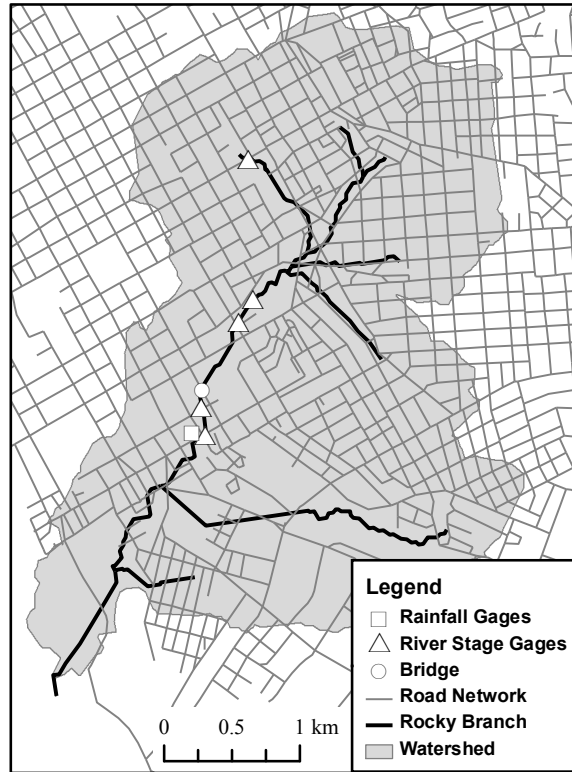


**FIG. 3. A flowchart of the GIS-based calculation for flood inundation.**

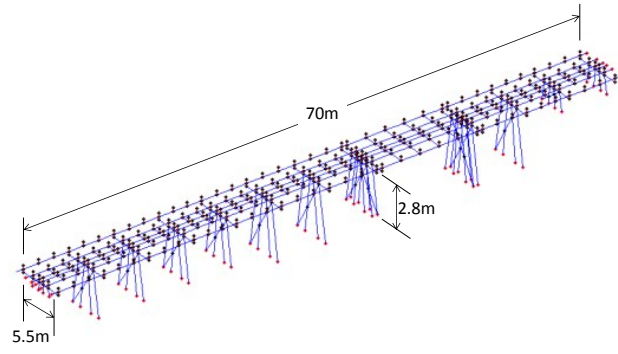




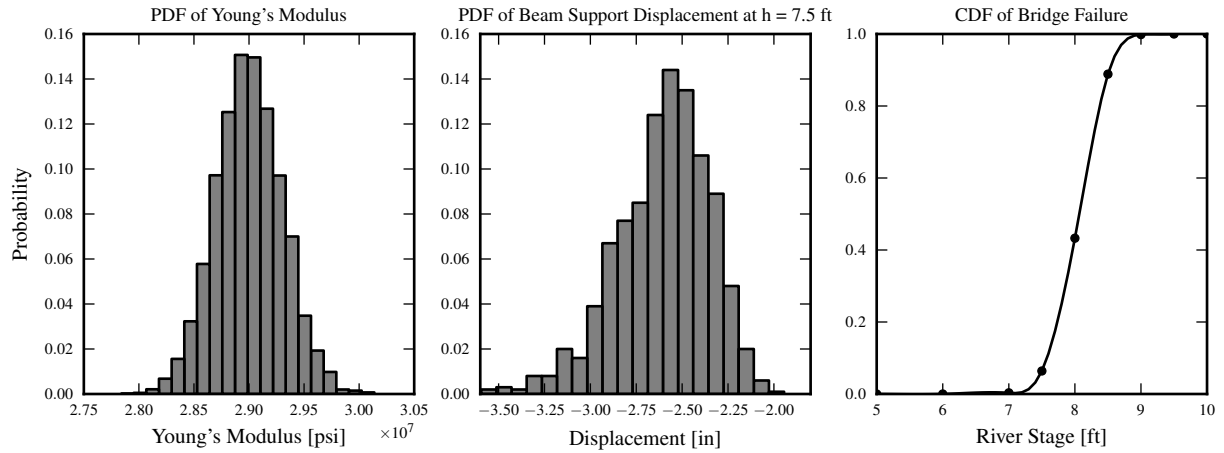
**FIG. 4. System architecture of the open-source Dynamic Traffic Assignment Simulation Engine.**



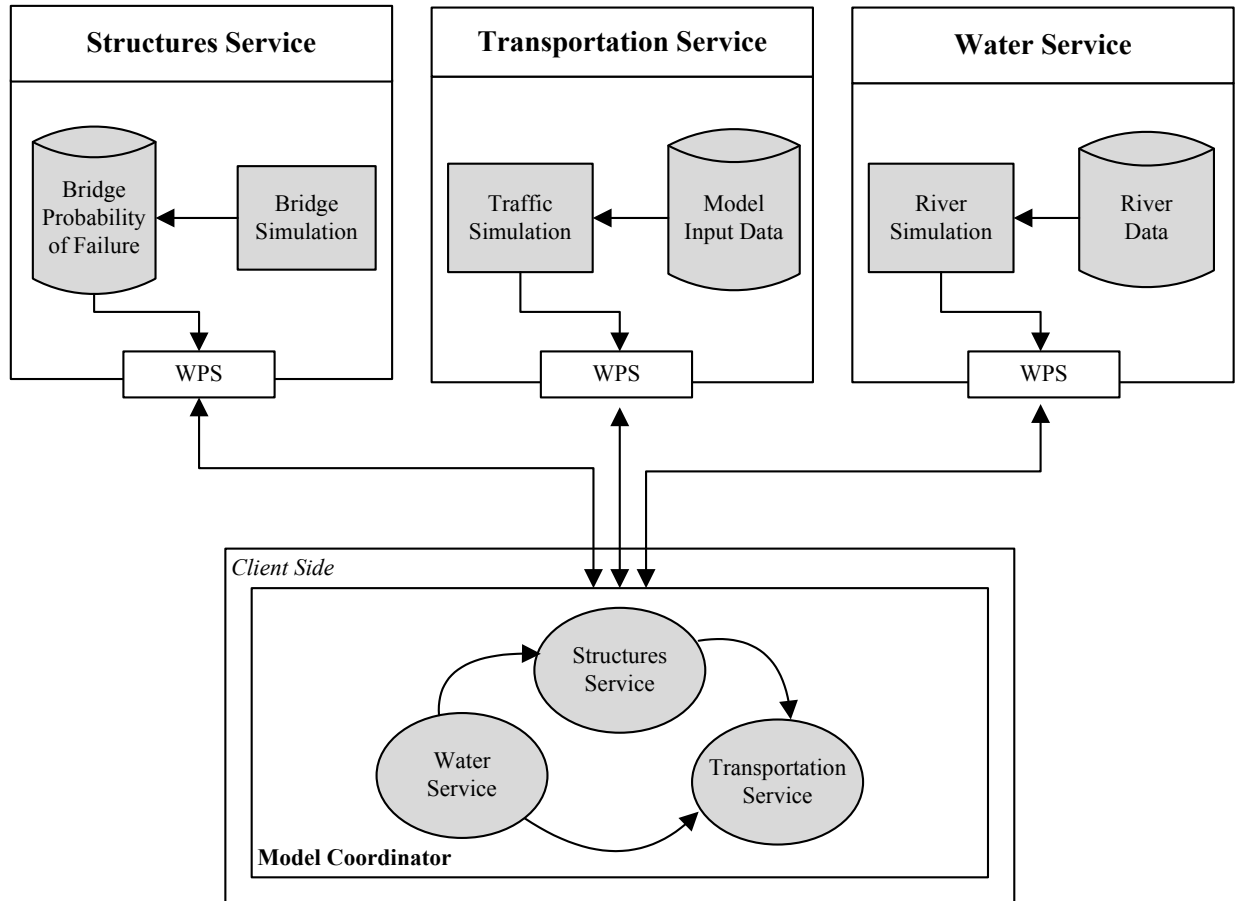
**FIG. 5. The study area: Rocky Branch Watershed in Columbia, South Carolina. Data from U.S. Census Bureau (2012) and derived from U.S. Geological Survey (2014).**



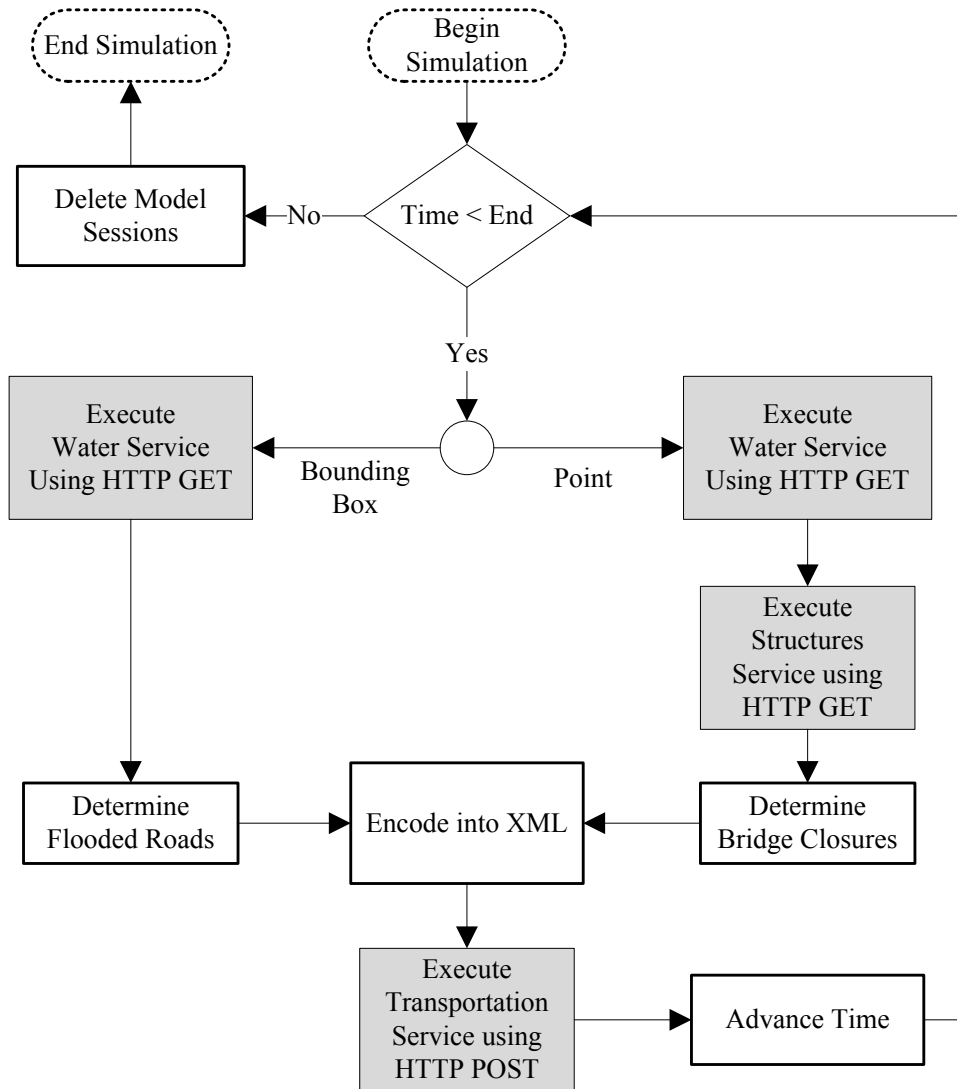
**FIG. 6. Finite element model of the railway bridge.**



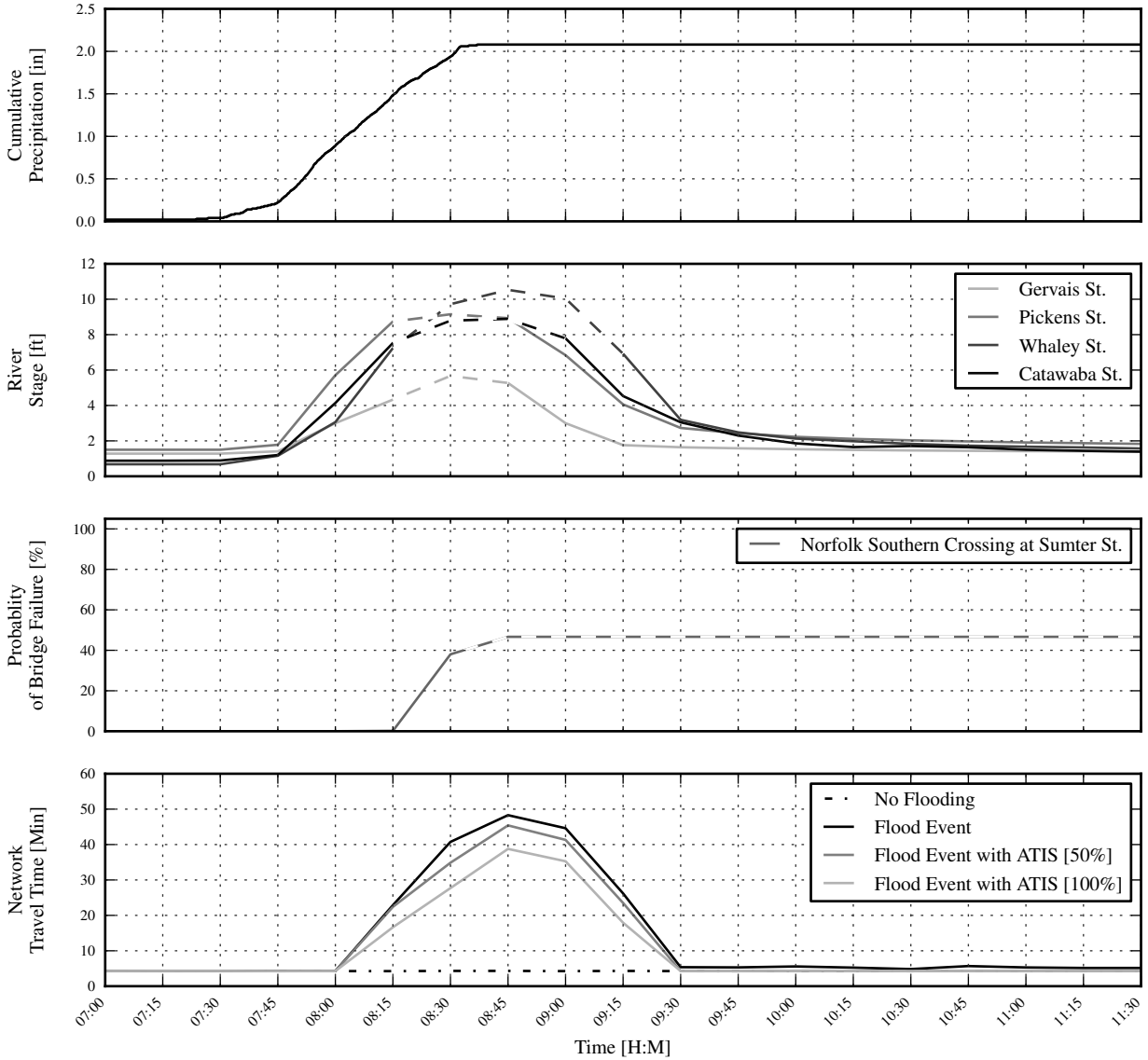
**FIG. 7. Probability Density Functions (PDFs) of Young's Modulus, the displacement of the beam at  $h = 7.5$  ft, and the Cumulative Distribution Function (CDF) of the probability that the bridge will fail as a function of the river stage.**



**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.**