A SIMULATION-AS-A-SERVICE FRAMEWORK FACILITATING WEBGIS BASED INSTALLATION PLANNING

Z. Zheng [a] , Z.Y. Chang, Y.F. Fei [a]
[a] Dept. of Planning, Beijing Special Civil Engineering Design and Research Institute, Beijing, P.R.C. - sametwoo@sina.com

Commission IV, WG IV/4

KEY WORDS: Web GIS, Multi-Agent Systems, Simulation-as-a-Service, Framework Design, Installation Planning

ABSTRACT:

Installation Planning is constrained by both natural and social conditions, especially for spatially sparse but functionally connected facilities. Simulation is important for proper deploy in space and configuration in function of facilities to make them a cohesive and supportive system to meet users' operation needs. Based on requirement analysis, we propose a framework to combine GIS and Agent simulation to overcome the shortness in temporal analysis and task simulation of traditional GIS. In this framework, Agent based simulation runs as a service on the server, exposes basic simulation functions, such as scenario configuration, simulation control, and simulation data retrieval to installation planners. At the same time, the simulation service is able to utilize various kinds of geoprocessing services in Agents' process logic to make sophisticated spatial inferences and analysis. This simulation-as-a-service framework has many potential benefits, such as easy-to-use, on-demand, shared understanding, and boosted performances. At the end, we present a preliminary implement of this concept using ArcGIS javascript api 4.0 and ArcGIS for server, showing how trip planning and driving can be carried out by agents.

1. INTRODUCTION

In installation planning, it is essential to make sure the location and functional configuration of each facility in it meet the users' needs effectively and efficiently, especially when a number of functionally interrelated facilities should be dispersed into a large area, where the costs of transportations would be high.

As a tool for geographic information management, analysis and decision support, GIS plays an important role in resolve complex installation planning and management problems (Fraile, 2016). Widely used as it is, GIS based planning has limited capability in users' behavior analysis, which is necessary for optimizing configuration of installations.

In recent years, Web GIS moves into maturity, with its easy-to-use, easy-to-share and highly interactivity, it is widely used as a method to communicate with users, and collect users' opinions about the installation plan, such as reservoir siting (Walsh, 2015), risk management (Aye, 2016), environmental impact assessment (Wanderer, 2015), emergency management (Wu, 2013), shield tunnel construction planning (Li, 2013), and landslide disaster information publishing (Yu, 2012). Temporal-spatial pattern researches based on Web GIS are covered frequently nowadays too (Martino, 2016).

On the other hand, shortly after the birth of Web, WBS started to emerge (Byrne,2010). In 1995, first web-front end simulation run as Common Gateway Interface was created. In 1998 first conference dedicated to Web based modelling and simulation was held. From its birth, web based simulation has gained continuously attention and active research. With the development of new technologies, such as Service Oriented Architecture, Web 2.0 and semantic web, web based simulation is getting a broad development opportunity.

In this paper, we try to focus on the spatial configuration and functional configuration problem of installation planning, propose a simulation service framework, and based on this framework, implement some useful functions. We hope to exemplify the usefulness and feasibility of our concept, and accumulate experiences for following works.

2. RELATED TECHNOLOGIES

2.1 Web based simulation

Web based simulation (WBS) can be defined as the use of resources and technologies offered by the World-Wide-Web for interaction with client and server modelling and simulation tools (Byrne, 2010). WBS uses browser to display information and receive user input. It is compatible and well adapted to heterogeneous and distributed environment, making interaction and cooperating between software across different platforms, different operating systems, and different network environments possible. WBS can share situations among many participants, promoting common understanding of problems, it is an important supporting technology for computer supported cooperative work (CSCW).

WBS has a lot of advantages, such as easy to use, convenient for cooperating, flexible deployment and authorization, strong compatibility, and cross-platform. It has induced interests from domains like education & training (Cant, 2014; Granlund, 2000; Beam, 2006), design and production (Li, 2013; Wang, 2012), and military modelling and simulations (Page, 2000). There are also some works trying to incorporating GIS with WBS in ecology (Adeva, 2012) and agriculture planning (Rao, 2007). WBS is especially attractive in its capability for resource integration. A lot of works has been done in this area, including: implementing computation models as web services (Johansson, 2003), packaging DEVS resources into web services to promoting interactions, publishing general simulation models as services to support discrete event simulations (Gyimesi, 2008),

https://doi.org/10.5194/isprs-archives-XLII-2-W7-193-2017

and organizing computation services for task analysis (Wang, 2010; Senin, 2003).

The enabling technologies (Byrne, 2010) of WBS can be classified as middleware, database management system, web language & tools, and simulation tools and libraries. As the result of the openness of WBS and rapid development of technologies, a great many of architectures and technologies, such as Web Service, HLA, Semantic Web, EJB, CORBA， has been developed and used in WBS. WBS tools such as Insight Maker (Fortmann-Roe, 2014), which supports system dynamics and agent based simulation, and JSIM (Miller, 2000), which is based on component technology has been proposed by researchers.

2.2 Web GIS

Web GIS was first introduced to realise online accesses to geographic information databases. One representative project is Alexandria project. Users can access, browse and download maps, orthophotos, SPOT images and Landsat Images by remote web clients. After the Alexandria project, Terraserver from Microsoft, EOS data gateway from NASA, and MapConnect from Australia has put into use. Accessing geographic data through web has become usual practices. By the late 1990s, distributed and interoperable processing architectures based on Internet helped to improve geographic information integration. Web GIS got the capability to support teams in different places with information sharing and cooperating. In the 2000s, geographic information exchange and storage was promoted with several geographic information specific markup languages such as GML and KML. In recent years, developers and researchers from outside of the traditional geographic industry made great contributions to the development of Web GIS, new concepts such as service-oriented architecture, collaborative applications, and loosely coupled structure, has promoted many new types of Web GIS applications. (Raper, 2009)

Interaction and cooperation are the most prominent features of Web GIS. A lot of researches have adopted Web GIS in the solutions of their problems. For example, in the risk management area, Aye et al. proposed a collaborative Web GIS framework to improve the intra-department collaboration (Aye, 2016); in the construction planning area, Walsh et al. designed a reservoir siting tool to interact with publics for screening and evaluation.

Now, developing of Web GIS is backed by a great many specifications. Open Geospatial Consortium released specifications for network transformation of geographic information (such as WMS, WFS and WCS), specifications for data format (such as SFS, GML and KML), and specifications for data search (such as OGC-CSW and OGC-WPS). These specifications set common standards for Web GIS development, promised the interoperability of Web GIS applications. There are many open source software and libraries for Web GIS application, which roughly divided into 3 categories: spatial data storage (such as PostGIS, MySQL and Spaital), spatial data publishing and sharing (such as MapServer, GeoServer, GoogleMaps and GoogleEarth), and geoprocessing and spatial analysis (such as 52°North WPS、PyWPS、GeoServer and degreeWPS). All these technologies provided a steady base for Web GIS application developing. (Swain, 2015)

3. REQUIREMENT ANALYSIS

Requirement analysis is the first step of framework designing. The goal of this framework is to support the analysis of spatial configuration and functional configuration of spatially dispersed and functionally related facilities. This framework will incorporate the power of simulation and GIS to help users answer what-if questions about their plans, support them making better decisions.

Requirements can be divided into functional requirements and system requirements. Functional requirements are the characters and services the system should provide. System requirements state the technology constraints that the system should respect. We consider these requirements together, and listed them as follows:

1. The system should have sufficient capability to represent facility users' activity. This means our system should be capable of setting a virtual environment to host the facilities of the installation in planning and their users, so we can simulate typical activities of the users of these facilities, to see what aspects of the activities will be affected and how will these aspects be affected by the installation configuration.
The system should have sufficient capability to model the facility users' tasks, the physical constraints and logical constraints of these tasks. It should also have sufficient capability to model the interactions between facility user and the geographic environment, including spatial related deductions and decisions, such as route planning and moving along the route.

2. The system should provide installation planners with functions to control the simulation. Installation planners should be able to setup the simulation scenario, start, stop and monitor the simulation.

3. The system should have good availability. The negative effects of system maintenance, such as model updating, UI upgrading, and functional extension, should be reduced to the minimum extent.

4. The system should simulate a large area. The extent of the simulation area should be large enough to match the planning area, typically, several hundred kilometres long and several hundred kilometres width.

5. The system should be extensible. Simulation model should reflect installation planner's needs, and can be added, modified and deleted easily.

4. FRAMEWORK DESIGN

4.1 Framework Architecture

In order to realise the requirements listed in the previous section, we adopt a Web GIS based architecture. Facility users' activity simulation is designed as a service running separately on the backend server, this service can be seen as a resource which can be used later in other places. The benefit of this is that all resources, (including geographic information and simulation) can be integrated and managed with mature web technologies, providing good interactivity and availability.

The architecture of the framework is displayed in Figure. 1. The system is composed of two loosely coupled modules, which is, the geographic service and the simulation service. Both services can be consumed by a remote installation planner through a web browser. The geographic service provides installation planner with maps and spatial analysis functions; The simulation service provides installation planner with simulation scenario setup, simulation control, and monitor functions. The simulation models can also use geoprocessing services provided by geographic service in their business logic to make space related deductions and decisions.

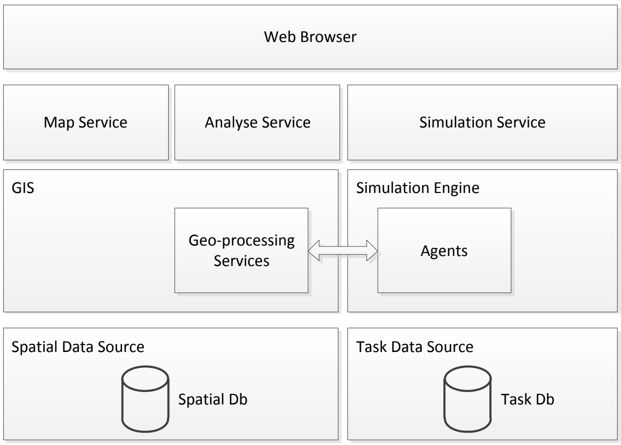


Figure 1. Framework Architecture

4.2 Implementing Technologies

The implementation of the framework is divided into two parts: the server side and the client side.

4.2.1 Server side

There are essentially 3 kinds of services running in the server side, which is, map service, geo-analysis service, and simulation service. For map service, there are many open source projects provides good enough functions for map rendering, such as Mapserver and Geoserver, nearly all of them is compatible with OGC's WMS specification and WMTS specification. This framework adopts WMS and WMTS as map service specification, so it's free to choose map server as long as it supports these specifications. Geo-analysis service is based on geoprocessing services. Geoprocessing service can be built on OGC's Web Processing Service, which is now supported by most open source geoprocessing software/libraries, such as degreeWPS, pyWPS, and 52°North WPS. Complex geo-analysis can be orchestrated by simple geo-process through BPEL specification and other ways (Weiser, 2007). Spatial data storage can be provided by several open source spatial database, among them, PostgreSQL with PostGIS extension is most widely used.

While open source software/libraries are capable of providing a solid foundation for the architecture depicted in Figure 1. Most modern commercial GIS are capable of providing both map service and geoprocessing service, and it is usually easier to use than open source tools. As a pilot study, we choose ArcGIS for server as the supplier for both map service and geoprocessing service to focus on the simulation service concept. The map service of ArcGIS for server is used to display base map. Map can be prepared and published from ArcGIS Desktop. Once published, the map can be consumed over the web. Geospatial query and analysis can be supported by Feature Services, Geometry Services, Network Analysis Service and Geoprocessing Services. Models can issue a task to these services and get responses in specified formats from them.

In order to evaluate the effects of installation plan on real users, it is important to reflect the autonomous behaviors of facility users. Agent-based simulation paradigm is adopted to meet this need. As stated by (Brown, 2005), integrating GIS with agent-based simulation, need to address 4 basic relationships, which are identity, causal, temporal and topological relationship. In our architecture, a service based approach is adopted. Agent platform is responsible for time management and task schedule, spatial query and analysis is realized through geoprocessing service.

Popular agent platforms in urban and architecture research include Swarm, Mason, Repast, StarLogo, Netlogo and so on (Chen, 2012). Among them, Mason has a separate design for UI and model scheduling system, supports intrinsic data types for continuous and discrete spatial structures and network data structure for relations, is more suitable for our architecture.So we choose Mason as the simulation engine for the simulation service.

4.2.2 Client side

At the client end, browser is used to consume geographic service and simulation service, and display maps and simulation situations. The inputs for simulation scenario setup, simulation control are also read from installation planners.
Javascript is used as the develop language of the browser-end functions. Supporting API can be Openlayers or ArcGIS Javascript API. As we chose ArcGIS for server to provide map service, we adopt ArcGIS Javascript API as the client side developing language just for convenience.

4.3 COMPOSING SIMULATION SERVICE

4.3.1 Simulation service in detail

Simulation service provides users with scenario editing, simulation control, and simulation monitoring functions in the form of EJB running in Glassfish application server. Users interact with simulation service through RESTful APIs, and receive real-time simulation status through WebSocket protocol. The simulation service EJB encapsulates Mason engine to actually carry out simulations.

It is vital for agents to make spatial related deliberations when they are used to simulate human behavior to assess installation plans. Geoprocessing services are used to meet this need. A geoprocessing service is a stateless calculation service that can provide operations for agent to make spatial query and analysis. These operations will help agents make better decisions in a complex topographic environment. The decision is based on the comprehensive perception involving spatial, social, and task conditions, these conditions are kept in the Environment Description data structure managed by the mason engine. The Environment Description is composed of a general relation network, which stores all kinds of relationships except task assignment relation (see section 4.3.2), a task assignment network, which describes the task assignment for every agent, and a location grid, which maintains the spatial relationship between agents. The Environment Description is updated after each action carried out by agents, to reflect the common knowledge shared by all agents. Agents do their tasks according to the Environment Description and data stored in the task database. This design will facilitate modeling and simulation for collaborative complex tasks.

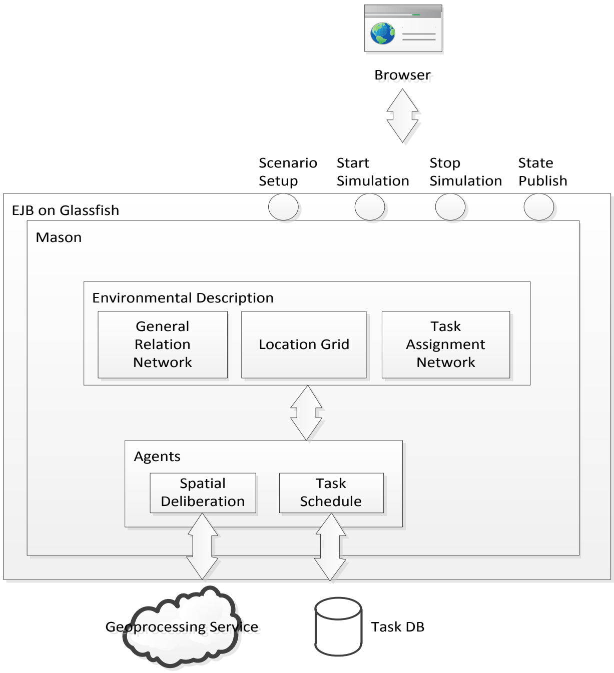


Figure 2. Simulation Service in Detail

4.3.2 Modelling in the simulation service

The model system of the simulation service focuses on the behavior of facility users.All models can be categorized into 3 types: Entity, Behavior, and Relation. Entity is the abstract class of every perceivable objects, including facilities, roads, hills, and users of facilities. Relation is the abstract class of all interactions and links between entities. Behavior is an independent model class. Behavior models can be “attached” to some kind of entity, giving it the capability to change the states of entities both of itself and others.

Entity model can be subclassified into Agent, Task, and Passive Object (Figure. 3). Agent can attach behavior models to simulate real behavior of facility users, Passive Object (PO) cannot attach behavior. Task is a special kind of Entity, it can be seen as the script of the Agent, the behavior model can read this script, and carry out actions accordingly.

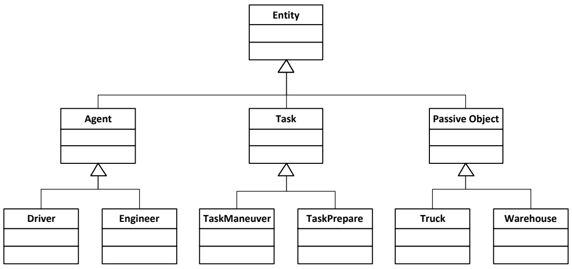


Figure 3. A Part of Entity Meta Model

Figure. 4 depicts a part of relation meta model. Relation can be classified into 3 types: interaction, execution, and link. Interaction is the effect Agents exert on other Agents or POs. It will change the state of the target entity; TaskAssign is the relation between an Agent and its task, representing the task assignment. Link is the basic fact among Agents and POs, It can be seen as a predicate, in both physical domain and cognitive domain, representing concepts such as connected, dependency, acquaintance, and so on.

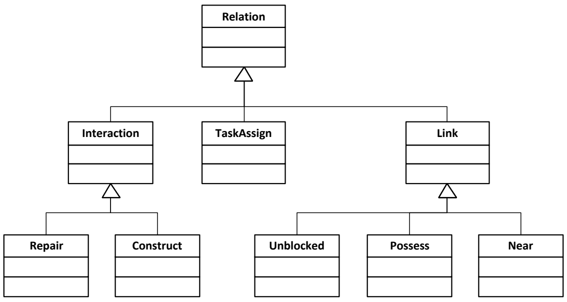


Figure 4. Meta Model of Relation

Behavior model typically carries out the perception-deliberating-action loop. Each agent class has their corresponding behavior model. Behavior model is flexible, it can be added, replaced, and removed as needed.

5. PRELIMINARY IMPLIMENTATION

Based on the framework proposed above, we made a preliminary implementation. In this implementation, we realised a simple demo system to study the driving behavior of users between widely dispersed facilities. Users of the facilities are tasked to drive from one facility to another, each user have one or more such tasks to complete. Users are free to choose the driving route and schedule their tasks according to the possibility and emergency. Facility planners want to find out the optimise location of the facilities based on the drive patterns of the users.

In this demo system, users are modelled as autonomous agents, they pick up tasks from their task list and schedule these tasks based on their beliefs. The driving behavior is divided into two parts: route planning and position calculation. Route planning is carried out by route service of ArcGIS for server. Position calculation is done in agent’s models.

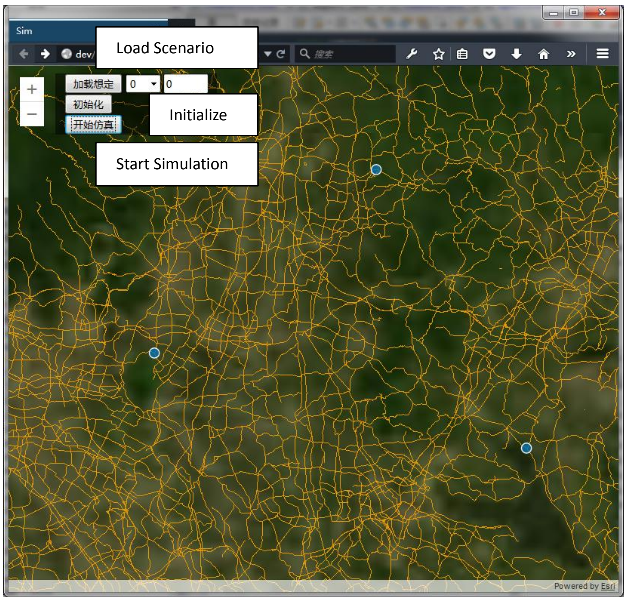
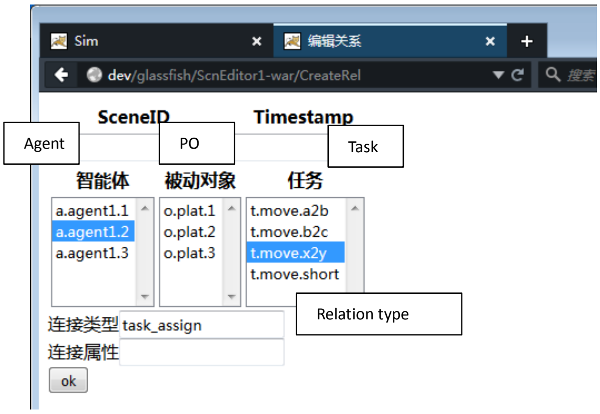


Figure 5. Screenshot of the Driving Simulation

The agent based simulation module is hosted at the Glassfish Application Server as an Enterprise Java Bean (EJB), publishing the positions of agents by websocket. The simulation scenario manager is also realised as an EJB, providing facility planners with basic scenario management functions. At the browser side, ArcGIS Javascript API 4.0 is used to display maps and monitor the driving positions, and provide basic control of loading scenario, initializing scenario and start simulation.



连接类型 task_assign

Figure 6. Scenario Editor UI of Task Assignment

6. CONCLUTION

Through the preliminary implementation, we verified the correctness and effectiveness of the simulation framework. Integrating the function of GIS and agent based simulation as services, will incorporate the capabilities of geographic information management, mapping, and spatial analysis from GIS with the capabilities of task modelling, complex system simulation from Agent based simulations, helping the planning of complex installations in the context of everyday usage.

This simulation-as-a-service framework has many potential benefits, such as easy-to-use, on-demand, shared understanding, and boosted performances. We believe it is a promising direction for following works.

REFERENCES

Adeva, J.J. Garcia, et.al., 2012. Web-based simulation of fruit fly to support biosecurity decision-making. *Ecological Informatics*, 9(3), pp. 19-36.

Aye, Z.C., et. al., 2016. A collaborative(web-GIS) framework based on empirical data collected from three case studies in Europe for risk management of hydro-meteorological hazards. *International Journal of Disaster Risk Reduction,* 15, pp. 10-23.

Beam, E.L., et.al., 2006. Design and Development of Bioterrorism Response Education Programs. *Clinical Simulation in Nursing Education,* 2(2), pp. 65-68.

Brown, D.G., et al. 2005. Spatial process and data models: Toward integration of agent-based models and GIS. *Journal of Geographical Systems*, 7(1), pp. 25-47.

Byrne, J., et.al., 2010. A review of Web-based simulation and supporting tools. *Simulation Modelling Practice and Theory*, 18(3), pp. 253-276.

Cant, R.P., and S.J. Cooper, 2014. Simulation in the Internet age: The place of Web-based simulation in nursing education. An integrative review. *Nurse Education Today*, 34(12), pp. 1435-1442.

Liang Chen, 2012. Agent-based modeling in urban and architectural research: A brief literature review, *Frontiers of Architectural Research*, 1(2), pp. 166-177.

Fortmann-Roe, S., 2014. Insight Maker: A general-purpose tool for web-based modeling & simulation, *Simulation Modelling Practice and Theory*, 47, pp. 28-45.

Fraile, A., et.al. 2016. Decision model for siting transport and logistic facilities in urban environments: A methodological approach. *Journal of Computational and Applied Mathematics*, 291(1), pp. 478-487.

Granlund, R., et.al. 2000. Designing web-based simulation for learning, *Future Generation Computer Systems*, 17(2), pp. 171-185.

Gyimesi, M., 2008. Web Service with generic simulation models for discrete event simulation, *Mathematics and Computers in Simulation*, 79(4), pp. 964-971.

Johansson, B., and P. Krus, 2003. A web service approach for model integration in computational design. In: *23rd Computers and Information in Engineering Conference,* Chicago, Illinois, USA, Vol. 1, Parts A and B, pp. 247-254.

Li, X.J., and H. Zhu, 2013. Development of a web-based information system for shield tunnel construction projects. *Tunnelling and Underground Space Technology*, 37(6), pp. 146-156.

Martino, F.D., et.al., 2016. WebGIS based on spatio-temporal hot spots: an application to oto-laryngo-pharyngeal diseases. *Soft Computing,* 6, pp. 1-13.

Miller, J.A., et.al., 2000. The JSIM web-based simulation environment. *Future Generation Computer Systems,* 17(2), pp. 119-133.

Page, E.H., et.al., 2000. Investigating the application of web-based simulation principles within the architecture for a next-generation computer generated forces model, *Future Generation Computer Systems*, 17(2), pp. 159-169.

Rao, M.F., et.al. 2007. A Web-based GIS decision support system for managing and planning USDA’s Conservation Reserve Program (CRP), *Environmental Modelling and Software*, 22(9), pp. 1270-1280.

Raper, J., et.al. 2009. Geographic information Science. *Annual Review of Information Science and Technology*, 43(1), pp. 1-117.

Senin, N., et.al. 2003. Distributed object-based modeling in design simulation marketplace. *Journal of Mechanical Design*, 125(1), pp. 2–13.

Swain, N.R., 2015. A review of open source software solutions for developing water resources web applications. *Environmental Modeling & Software*, 67(C), pp. 108-117.

Wainer, G.A., et.al., 2008. Distributed simulation of DEVS and cell-DEVS models in CD++ using web-service. *Simulation Modelling Practice and Theory*, 16(9), pp. 1266-1292.

Walsh, S.J., et.al., 2015. A reservoir siting tool for North

Carolina: System design & operations for screening and evaluation. *Applied Geography*, 60, pp. 139-149.

Wanderer, T., and S. Herle, 2015. Creating a spatial multi-criteria decision support system for energy related integrated environmental impact assessment. *Environmental Impact Assessment Review*, 52, pp. 2-8.

Wang, H., et.al., 2010. Towards a collaborative modeling and simulation platform on the Internet. *Advanced Engineering Informatics*, 24(2), pp. 208-218.

Wang H.W., and H.M. Zhang, 2012. Using collaborative computing technologies to enable the sharing and integration of simulation services for product design. *Simulation Modelling Practice and Theory*, 27, pp. 47-64.

Stollberg, B., and A. Zipf. 2007. OGC Web Processing Service Interface for Web Service Orchestration Aggregating Geoprocessing Services in a Bomb Threat Scenario. In: *Web and Wireless Geographical Information Systems, International Symposium, W2gis 2007*, Cardiff, Uk, Proceedings DBLP, pp. 239-251.

Wu, A., et.al., 2013. Supporting collaborative sense-making in emergency management through geo-visualization. *International Journal of Human-Computer Studies*, 71(1), pp. 4-23.

Yu, Z.S., et.al., 2012. The Research on Landslide Disaster Information Publishing System Based on WebGIS. *Energy Procedia*, 16, pp. 1199-1205.