Dependability of Active Emergency Response Systems

Jane W. S. Liu
Institute of Information Science
Academia Sinica
Taipei, Taiwan
e-mail: janeliu@iis.sinica.edu.tw

Edward T. H. Chu
Computer Science and Information Engineering
National Yunlin Science and Tech. University
Yunlin, Taiwan
e-mail: Edwardchu@yuntech.edu.tw

***Abstract*— Recent technological and infrastructure advances along several fronts have enabled smart embedded devices, systems and applications that can enhance preparedness of our living environments against common natural and man-made disasters. They can also help us to be safer when disasters strike. This paper first discusses issues in configurability, maintainability and safety specific to this type of smart things and systems. It then describes models and tools for assessing their effectiveness and ensuring their safety.**

Keywords - disaster preparedness and response; system safety; cyber-physical elements; simulation environment; testbed

I. INTRODUCTION

The term *Active Emergency Response Systems* (AERS) [1] refers to systems of smart embedded devices and mobile applications that can process standard-compliant disaster alert messages from authorized senders and respond by taking appropriate actions to prevent loss of lives, reduce chance of injuries and minimize property damages and economical losses when the forewarned disaster strikes. We call such devices and applications *iGaDs* (*intelligent Guards against Disasters*) collectively [2]-[4]. Examples of iGaDs include smart devices that shut natural gas intake valves and turn off electricity to prevent fire, open doors to ease evacuation, bring elevators to the ground floor, turn on hazard flashers and warn the drivers of trucks and cars on highways, and deliver location-, environment- and situation-specific alerts and instructions to people via their mobile devices upon receiving an alert of a strong earthquake.

iGaDs and AERS have been made feasible in developed regions by recent advances along four directions: First, advances in sensor and analysis technologies have enabled the predication and detection of common types of natural disasters and issuance of accurate early warnings about them. For example, in developed countries frequented by earthquakes, systems of strong motion sensors networked via RF links with computers running analysis tools can generate early warnings of strong earthquakes within second(s) of their occurrences, providing receivers in affected areas with warnings, often second(s) before ground motion starts.

The second enabler is Common Alert Protocol (CAP) for encoding alert messages [5]. The OASIS standard has been adopted in US, Canada, Australia and parts of Asian Pacific region, including Taiwan and Japan. Being XML-based, CAP alert messages can be processed automatically by smart devices and applications. Hereafter, we assume that all alert messages are in CAP format and sometimes call iGaDs CAP-aware devices, systems or applications.

Third, iGaDs and AERS are enabled by platforms for receiving and authenticating CAP-compliant alerts from alerting authorities and then broadcasting them. An example is Integrated Public Alert and Warning System (IPAWS) - OPEN [6], which has been operational in USA and Canada since 2011 [6]. IPAWS-OPEN and similarly platforms in other parts of the world enable CAP alerts to be disseminated via multiple communication pathways, including broadcast channels, cellular broadcast and Internet.

The fourth enabler is Building Information Models (BIM) [7] and associated digital data exchange standards. BIM has been adopted increasingly more widely. The integration of BIM with facility management and building automation systems (e.g., [8] [9]) has enabled the systems to provide 3D-4D data on buildings and their facilities, interior layouts, and so on that are vital to support decisions of individual iGaDs in their choices of protective actions.

To illustrate this, Figure 1 shows an earthquake scenario: A strong earthquake alert in CAP format is issued by Central Weather Bureau, the agency authorized to issue such alerts in Taiwan. Today, earthquake alerts are sent directly to safety equipment of power plants, trains and fabrication lines. Alerts are also sent to Emergency Alert Services (EAS) and mobile alert services, including Google Public Alerts. These services in turn warn the general public. Limitation in human's ability to react in time and the lack of specific instructions limit the effectiveness of the warnings.

Figure 1. A earthquake scenario illustrating active use of alert [2]

Our white paper [2] advocates an alternative: Broadcast the alerts in the original CAP format directly to iGaDs pervasively deployed throughout our living environment. CAP-aware embedded devices can respond with humanly impossible speed to make the environment safer in ways illustrated by the examples mentioned earlier and shown in the lower right corner of Figure 1. CAP-aware mobile applications can instruct people how to stay safe based the seismic codes of buildings, interior layouts, and furnishings around them. Indeed, if such applications were available at the time of 2011 5.8 Virginia Earthquake [10], most people from New York City to Washington DC would be instructed to stay where they were: That is, do not evacuate. The chaos and economic loss occurred on the day could be avoided.

From this and other scenarios [2], one can see that iGaDs are mission critical. Ubiquitous iGaDs are Internet of Things (IoTs), and AERS containing iGaDs and remote and local sensors are cyber-physical systems. So, the title "No dependability, no internet of things" of the article [11] published by Newsroom Editor of European Commission is applicable to iGaDs/AERS. Challenges in making them adaptable and dependable, unless satisfactorily overcome, are roadblocks to their becoming pervasive elements of future disaster prepared smart living environment.

Following this introduction, Section II presents related work on dependability of IoTs and cyber-physical system in general and discusses dependability issues specific to iGaDs and AERS. To date, the results of our work include iGaDs and AERS prototypes built for proof of concept purposes and as solutions of configurability and adaptability problems. They are described in Section III. Safety is an important dependability requirement of iGaDs and AERS. Section IV describes our current and future work on models and tools for assessing the safety of AERS containing a large number of diverse iGaDs. Section V summarizes the paper.

II. Related Work

The above-mentioned statement on dependability of IoTs [11] and similar observation by researchers and developers worldwide have motivated vast efforts on IoT dependability. Examples of recent results include mechanisms and protocols for enhanced availability and reliability of IoTs and networks and middleware in applications/services built from them [12]-[14]. Other efforts (e.g., [15]-[18]) aim at providing frameworks, tools, benchmarks to support the design, implementation and assessment of dependable IoT applications and cyber-physical systems. These applications and systems, including AERS, have long lifetime. Support infrastructures, including tools for maintenance and upgrade, need to be put in place (e.g., in [19]) to ensure non-disruptive operations of existing devices and systems as they adapt to inevitable changes in message delivery platforms, message format standards, security mechanisms, and technological advances during their lifetime.

Our work on the dependability of iGaDs and AERS has the same general goal as these related efforts. We leverage existing solutions as much as possible. Section III will present examples. By doing so, we can better focus on dependability issues specific to iGaDs and AERS.

A focal point of our current effort is safety of AERS that contain vast numbers of diverse iGaDs and local sensors (e.g., intelligent emergency evacuation systems for large and complex buildings). To explain the challenges, we note that an iGaD may need to process at the same time multiple types of alerts (e.g., a strong earthquake alert for the region and a local fire or flash flood alarm) that call for conflicting responses (e.g., open all doors and close some doors, respectively). Alerts may be cancelled and reissued as conditions changes. Even most advanced disaster prediction and detection systems may issue false alarms and have missed detections. Protocols for handling such events need to be put in place, however rarely they may happen. Even when all alert messages arrive correctly and in time and all devices function correctly, the combinations of their actions may lead to catastrophic consequences.

Section IV will further elaborate issues related to safety of AERS and present our current work on building an extensible simulation framework, called AERS Simulation Framework (AERS-SF). The framework is agent-based. It resembles many existing toolkits (e.g., [20]-[22]) for the development of agent-based applications in their use of agents as model elements. Existing safety studies and emergency and disaster simulators (e.g., [23]-[25]) typically consider specific kind of emergency (e.g., fire) in a specific environment (e.g., in high rises or planes). In contrast, AERS-SF aims to provide models, tools and benchmarks needed to support simulation of diverse AERS in diverse operating environments and disaster scenarios for sake of assessing safety of AERS throughout their development.

III. Configurable and Adaptable Prototypes

Thus far, our work aims to demonstrate the concept of configurable and adaptable AERS [1]-[4] for homes, office buildings, and large public places. They contain diverse iGaDs capable of responding to alerts of natural disasters affecting the region in general, as well as alerts of emergency conditions within the building.

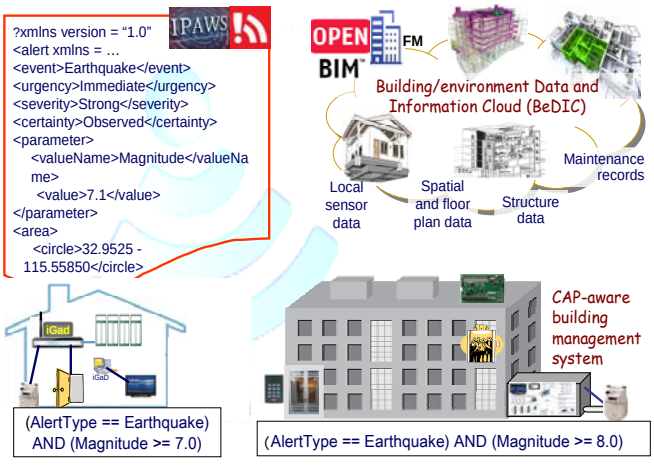

Figure 2. Underlying assuptions

Figure 2 highlights three of the underlying assumptions: First, all messages are compliant to the XML-based CAP standard. They are sent by trustworthy entities (e.g., in US,

responsible authorities via IPAWS-OPEN) and the building management system. So, their contents can be secured and authenticated by the existing XML security mechanism [26].

Second, the decisions of individual iGaDs on whether and how to respond to an alert are based in part on the alert type and severity specified by the alert. In an AERS for indoor spaces, their decisions are also based on data on the building, including its seismic code and maintenance records. For example, suppose that the home and office building in Figure 2 are designed to withstand earthquakes of magnitude 7.0 and 8.0, respectively. Then, CAP-aware door and gas value controllers in the home should respond to the magnitude 7.8 earthquake alert in Figure 1, but the devices of the same types in the office building should ignore the alert. Building data are provided by an information system, called Building and environment Data and Information Cloud (BeDIC) in Figure 2. It contains datasets selected from BIM and facility management system of the building.

Third, the response decision of an iGaDs also depends on how the device(s) is used and data (e.g., sensor data) from local sources. For example, upon receiving a Enhanced Fujita (EF) [27] scale 5 tornado alert, an iGaD controlling a public shelter door should open the door unconditionally. An iGaD controlling the front door of a house may wait until the tornado is about to strike the house, indicated by drastic decrease of outside air pressure, and then opens the door.

From these examples, we can see that iGaDs must be configurable and customizable, not only at installation times but also at maintenance and runtimes. Figure 3 shows an architectural framework for iGaDs for building configurable and customizable iGaDs for diverse purposes from the same set of components [2][3]. Specifically, every iGaDs has a CAP message processor/parser for validating CAP-compliances of the message and extracting from each CAP message the type and severity of the disaster, areas targeted by the message and so on. Every iGaD has a location filter that determines whether the device is located in an affected area and hence is targeted by the alert. An embedded iGaD has a device controller that interfaces with one or more physical devices. Customization of the kinds mentioned above is enabled by using a rule engine to process action activation rules such as the ones shown in Figure 2. The rules are selected and their parameters set at installation and maintenance time of each iGaD.

Some iGaDs are reachable only via the Internet. Examples include CAP-aware elevator, smart gas valve and door controllers. These devices receive alerts relayed by the building (home) management system that is connected to the Internet and serves as an aggregation server. Clearly, iGaDs and people can take protective actions in preparation of an imminent calamity only when they receive warnings about the calamity in time. This means that the end-to-end delay of earthquake warning messages should be a second or less, and delay for tornado and flash flood warnings a minute to a few minutes, and so on. Performance data of Asynchronous Message Delivery Service (AMeDS) [3] [4] for delivering CAP messages asynchronously over the Internet show that end-to-end delay requirements of this order are feasible and AMeDS offers a way to do so.

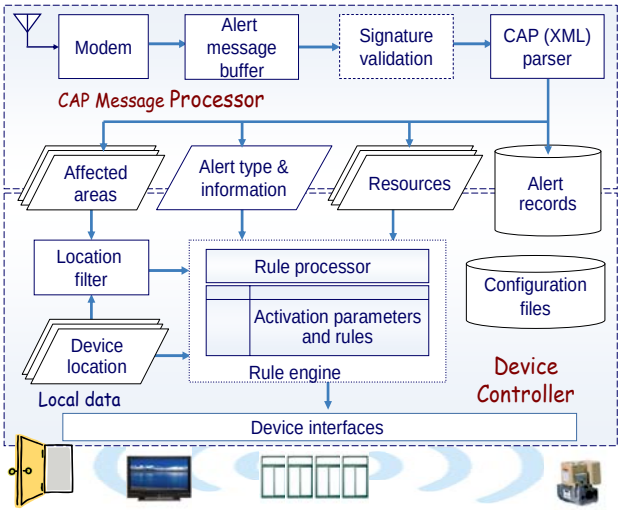

Figure 3. iGaD architecture and key components

IV. AERS SIMULAITON FRAMEWORK

Again, a major thrust of our current work is on safety of AERS, in particular, systems containing a large number of diverse iGaDs and local sensors and serving large complex buildings and facilities, such as transport hubs, major hospitals, sports centers, and shopping malls. A common definition of safety is the absence of *dangerous conditions* that can cause death, injury, damage to property and economical loss [28]. This definition is not appropriate for AERS since such systems work in the presence of dangerous conditions. As an alternative definition of safety, we may say that an AERS is *safe* if its actions never create new dangerous conditions and never increase the probability of occurrence of dangerous conditions known to exist when the system is not in use.

We work with a definition that is more practical from the point of view of validation: We say that a system is *safe as specified* when it always removes the dangerous conditions identified by disaster and emergency response experts and defined in its safety requirement specification. We need to be able assess to what degree a given AERS is safe (i.e., safe as specified) under all likely operating conditions/demands, including occurrences of nearly simultaneous multiple alerts that require conflicting responses; arbitrary sequences of alerts, cancellations, and re-issuances; and false alarms and missed detections of specified rates. The combined actions of a large number of iGaDs may lead to unexpected dangerous conditions, even when all alerts are correct and delivered in time and every device and application works correctly. The problem of making AERS serving large public buildings safe is further complicated by two factors. First, iGaDs may need to collaborate and coordinate their actions for error/failure handling and conflict resolution purposes. The complexity thus introduced may actually make the system less safe. The second complicating factor is the presence of people and crowds, who are also smart entities and may respond to alerts on their own in unsafe ways unless constrained from doing so. The problem is to identify the constraints.

 ISBN: 978-1-61208-429-9

Motivated by the fact that highly available, secure and configurable and maintainable AERS may nevertheless be unsafe, we are developing the simulation framework AERS-SF capable of supporting simulation experiments on diverse AERS for purposes of finding safety flaws and assessing their safety throughout their design, development and deployment. We also want to evaluate via simulation constraints on operations of the system and its components, which when adhered to, can make the system safer.

Figure 4 shows the major components of AERS-SF. The framework will offer libraries of models, tools, and test scenarios generators, together with a simulation environment, using which a user (i.e., a designer or a developer) can construct customized simulator(s) of his/her AERS in building(s) targeted by the system and conduct experiments with design choices (e.g., action activation rules and conflict resolution protocols) of individual iGaDs and alternative Standard Operating Procedures (SOPs) governing alert cancellations and false alarms, for the system as a whole. Specifically, AERS-SF model libraries have (1) agent-based models of active entities in AERS and operating environment, including executable models of iGaDs; (2) behavior models of people as individuals and as members of crowds; (3) BIM-based models of representative buildings and facilities controlled by iGaDs; and (4) conflict resolution and collaboration protocols for iGaDs and representative SOPs. Similar to model libraries of the Agent-Based Disaster Simulation Environment ABDiSE [22], AERS-SF model libraries are extensible: Model elements in the underlying model of each simulation experiment are dynamically loaded during set up and initialization time. The user can add new types of models by providing dynamic linked library functions defining the behavior of new types.

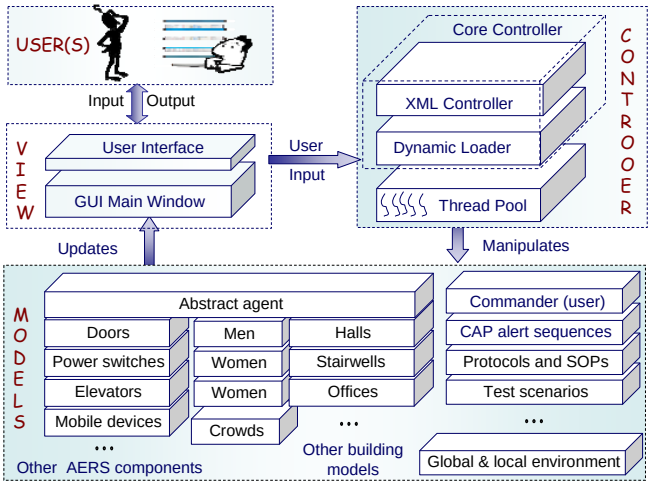

Figure 4. Structure and major components of AERS-SF

To support what-if experiments, the framework will also have extensible libraries of test scenarios. In particular, it will provide traces of disaster and emergency alerts, both actual traces from CAP alert message records that have been released as open data in many countries and synthetic traces that can be used as benchmark input to the system being evaluated. Some of the scenarios detailing the development of emergencies within the targeted building are generated from historical records of common types of disasters and local emergencies. For example, scenario generation scripts can use as input information extracted from historical records on impacts of past typhoons and debris flows on similar buildings. We also plan to link AERS-SF with ABDiSE and through it, to import external disaster simulation programs.

AERS-SF will adopt two other features of ABDiSE. One is to build model elements on common-sense concepts. For example, every simulation experiment has one and only one simulation world, i.e., the geographical area specified by the user for the experiment at set up time. The world may have many regions with specified boundaries. The simulation world has a global environment, and some regions may have local environments that differ from the global environment. Each environment is defined by a set of environment parameters. The behaviors of all agents around any point in space and time within a region depend on the values of local environment parameters at that point in space and time. Thus, we eliminate the need to model sensors explicitly.

Also, similar to ABDiSE, AERS-SF makes tools for building the underlying model for each series of simulation experiments and for controlling simulation runs accessible to the user from the GUI of the framework. Figure 5 uses a marked up screen dump of ABDiSE to illustrate this point. The most prominently displayed tool is the Map Explorer in area B, which displays a 2-D map of a region (e.g., an office area shown here). The tool provides the user with an easy way to specify locations of agents (e.g., two CAP-aware doors). Area A provides access to tools using which the user can select and retrieve model elements from libraries and use them to construct and customize simulation models of the target AERS and its operating environment. When new agent types need to be created, a click of “Create New Agent” button in area A is the first step. Area C displays the list of all model elements that have been selected. Area D lets the user set up and control simulation experiments (e.g., lengths of time steps and the current simulation run). Area E lets the user to specific environment parameters of the region displayed in area B. The user can also visualize via the GUI the development of the scenario within the part displayed in area B during the simulation run.

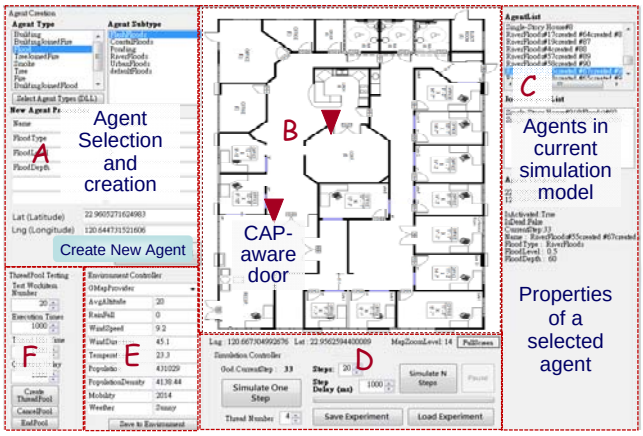

Figure 5. GUI, tools and use scenario

V. CONCLUSION

The previous sections first presented the need for AERS and ways to make them configurable, maintainable and secure. Among all attributes of dependability, safety is the most challenging one for AERS for reasons stated earlier. We are developing the simulation framework AERS-SF designed to support the use of simulation as a tool for assessing the safety of AERS of diverse AERS in diverse operating environments throughout their development and deployment process. Thus far, we have been focusing on its design; especially we want to make sure that the framework will support the underlying models, simulation methods, data capture and analysis methods required to meet its design goals. We have adopted some of the approaches of ABDiSE. Compared with that framework, AERS-SF is far more complex in almost all aspects. Nevertheless, we believe that the software architecture of ABDiSE, as well as some of its software components, can be adopted and enhanced to give the implementation of AERS-SF a head start.

ACKNOWLEDGMENT

This work is supported by the Academia Sinica, Sustainability Science Research Project OpenISDM.

REFERENCES

[1] C. Y. Lin, E. T.-H. Chu, L.-W. Ku, and J. W. S. Liu, "Active Disaster Response System for a Smart Building," Sensors, 14, 2014, pp.17451-17470, doi:10.3390/s140917451.

[2] J. W. S. Liu, E. T. H. Chu and C. S. Shih, "Cyber-Physical Element of Disaster Prepared Smart Environment," IEEE Computer, Vol. 46, No. 2, Feb. 2013, pp. 69 – 75, doi:10.1109/MC.2012.149.

[3] W. P. Laio, Y. Z. Ou, E. T. H. Chu, C. S. Shih, and J. W. S. Liu, "Ubiquitous Smart Devices and Applications for Disaster Preparedness," Proc. of the 2012 Int. Symp. Ubiquitous Intelligence & Computing, Frontiers Workshop, IEEE Press, Sep. 2012, pp. 764 – 770, doi:10.1109/UIC-ATC.2012.98.

[4] Y. Z. Ou, C. M. Huang, C. T. Hu, E. T. H. Chu, C. S. Shih, and J. W. S. Liu, "Responsive Alert Delivery over IP Network," Proc. of IEEE Int. Conf. on Cyber Physical Systems, Networks and Applications, IEEE Press, Aug. 2013, pp. 19 – 25, doi:10.1109/CPSNA.2013.6614241.

[5] Common Alert Protocol Version 1.2, OASIS Standard, http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html (Retrieved: June 2015).

[6] Integrated Public Alert & Warning System (IPAWS)-OPEN, https://www.fema.gov/integrated-public-alert-warning-system-open-platform-emergency-networks, (Retrieved: June 2015).

[7] Building Information Models/Modeling (BIM) - Wikipedia, http://en.wikipedia.org/wiki/Building_information_modeling, (Retrieved: June 2015).

[8] P. Teicholz (editor). BIM for Facility Managers, Mar 2013, IFMA Foundation.

[9] G. Percivall, "Smart Cities Spatial Information Framework," Jan 2015, Open Geospatial Consortium Document, https://portal.opengeospatial.org/files/?artifact_id=61188, (Retrieved: June 2015).

[10] 2011 Magnitude 5.8 Virginia Earthquake, August 23, 2011, http://en.wikipedia.org/wiki/2011_Virginia_earthquake, (Retrieved: June 2015).

[11] "No dependability, no Internet of Things," Mar. 2004, https://ec.europa.eu/digital-agenda/en/news/no-dependability-no-internet-things, (Retrieved : June 2015)

[12] P. H. Su, C. S. Shih, J. Y-J. Hsu, J.Y.-J., K. J. Lin and Y. C. Wang, "Decentralized fault tolerance mechanism for intelligent IoT/M2M middleware," Proc. of IEEE World Forum on IoT, IEEE Press, Mar. 2014, pp. 45 – 50, doi:10.1109/WF-IoT.2014.6803115.

[13] S, Cherrier, Y. M. Ghamri-Doudane, S. Lohier, and G. Roussel, "Fault-recovery and coherence in Internet of Things choreographies," Proc. of IEEE World Forum on IoT, IEEE press, Mar. 2014, pp. 532 – 537, doi:10.1109/WF-IoT.2014.6803224

[14] N. Maalel, N., E. Natalizio, A. Bouabdallah, P. Roux and M. Kellil, "Reliability for emergency applications in Internet of Things," Proc. of Int. Conf. on Distributed Computing in Sensor Systems (DCOSS), IEEE Press, Mar. 2015, pp. 361 – 366, doi:10.1109/DCOSS.2013.40.

[15] RELYonIT Project, http://www.relyonit.eu/, (Retrieved: June 2015).

[16] RERUM: REliable, Resilient and secUre IoT for sMart city applications, https://ict-rerum.eu/, (Retrieved: Jun 2015).

[17] L. Wu, and G. Kaiser, "FARE: A Framework for Benchmarking Reliability of Cyber-Physical Systems," Proc. of IEEE 2013 Long Island Systems, Applications and Technology Conference, IEEE press, May 2013, pp. 1-6, doi:10.1109/LISAT.2013.6578226.

[18] L. Silva, R. Leandro, D. Macedo , and L. A. Guedes, "A Dependability Evaluation Tool for the Internet of Things," ACM Jr of Comp. and Elect. Eng, Vol. 39, No.7, Oct. 2013, pp. 2005-2018, doi: 10.1016/j.compeleceng.2013.04.021.

[19] NIST Cyber-Physical Systems Testbed Workshop, http://www.nist.gov/cps/cyber-physical-systems-testbed-workshop.cfm, (Retrieved: Jun 2015)

[20] C. M. Macal and M.l J. North, "Introductory Tutorial: Agent-Based Modeling and Simulation", Journal of Simulation, Vol. 4, 2010, pp. 151–162. doi:10.1057/jos.2010.3.

[21] Rob Allan. "Survey of Agent Based Modeling and Simulation Tools," http://www.grids.ac.uk/Complex/ABMS/, (Retrieved: June 2015).

[22] Hsu, T. L. and J. W.S. Liu, "An Agent-Based Disaster Simulation Environment," Academia Sinica Technical Report No. TR -IIS-15-005, March 2015.

[23] Case Study: Sophisticated fire safety systems enable rapid isolation of incidents and evacuation of occupants, http://w3.siemens.com/topics/global/en/sustainable-cities/resilience/pages/sophisticated-fire-safety-systems.aspx, (Retrieved: June 2015)

[24] A. Sagun, C. J. Anumba, and D. Bouchlaghem, "Safety Issues in Building Design to Cope with Extreme Events: Case Study of an Evacuation Process," J. Archit, Eng, Vol. 20, No.3, Sept. 2014, doi: 10.1061/(ASCE)AE.1943-5568.0000147.

[25] B. Wang, H. Li, Y. Rezgui, A. Bradley, and H. N. Ong, "BIM Based Virtual Envionrment for Fire Emergency Evacauation," The Sci. World Jr., 2014, Article ID 589016, 22 pages.

[26] W3C XML Security 2.0, http://www.w3.org/TR/xmlsec-reqs2/, April 2013, (Retrieved: June 2015).

[27] Enhanced Fujita (EF) scale, from Wikipedia, http://en.wikipedia.org/wiki/Enhanced_Fujita_scale, (Retrieved: June 2015).

[28] NASA System Safety Handbook, Vol. 1, NASA/SP-2010-580, Nov. 2011, http://www.w3.org/TR/xmlsec-reqs2/, (Retrieved: June 2015).

 ISBN: 978-1-61208-429-9