Distributed Simulation of Interdependencies in Community Resilience

by

Szu-Yun Lin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering and Scientific Computing)
in the University of Michigan
2020

Doctoral Committee:

Professor Sherif El-Tawil, Chair
Associate Professor Jason P. McCormick
Professor Atul Prakash
Assistant Professor Seymour M.J. Spence

Szu-Yun Lin

sylin@umich.edu

ORCID iD: 0000-0001-5369-2571

DEDICATION

To my beloved family and friends

ACKNOWLEDGEMENTS

During my PhD journey, I have received a great deal of support and assistance. First, I would like to express my sincere gratitude to my advisor, Prof. El-Tawil. With his guidance and support, I have gained the ability to conduct independent and collaborative research, and the other important skills for my further career. He has been very supportive of both my academic research and career plan. I would like to thank my committee members, Prof. McCormick, Prof. Spence, and Prof. Prakash, and the other co-principal investigators in the ICoR group, Prof. Aguirre, Prof. Kamat, and Prof. Menassa, for their valuable and insightful advice to make this work more complete and accurate.

Development of the SRTI software described and used in this dissertation was carried out by the Computational Platform Team under the ICoR (Interdependencies in Community Resilience) project, which was funded by the US National Science Foundation and the University of Michigan. The design, implementation and validation of the SRTI was primarily accomplished by myself, Andrew Hlynka, Harvey Hao Lu and Lichao Xu.

The research cannot be completed without my gorgeous colleagues and friends. My sincere thanks also go to the members in Prof. El-Tawil's research group (Tung-Yu, Yunsu, Yuh-Shiou, Omar, and Saqif) and the ICoR research group (Andrew, Lichao, Harvey, Xi, Ahmed, and Eileen), and friends in CEE. I would like to thank many friends I have made in Ann Arbor and Taiwan for

sharing their knowledge and life experiences. Special thanks to Wei-Chu, Dian-Ru, Teyuh, Jenny, and Serena for being there whenever I needed.

Finally, I would like to thank my parents and brother for their constant love and support. I would not have been able to finish my PhD without them. I am extremely grateful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

When a disruption such as a severe natural event occurs, the interdependencies between the infrastructure systems of society can lead to cascading events that can adversely affect community resilience. Resilience is the ability of a community to withstand, adapt and recover from a disruption, typically measured in terms of loss of life, injuries and economic cost. Studying the interactions between infrastructure systems is complicated by the fact that each system is rooted in a specific field and thus requires crossing disciplinary barriers. To overcome the identified research challenge, this dissertation employs distributed computational simulation to model and investigate the interdependencies that arise during severe disasters and the post-disaster recovery process. It focuses in particular on multi-scale interdependencies and their time-dependent effects on community resilience.

A distributed simulation framework that links each discipline specific simulator using a publish-subscribe data transmission pattern is proposed. The framework's capabilities are demonstrated through a case study of multistory buildings that suffer wind-induced progressive damage. Data transmission is achieved using the Lightweight Communications and Marshalling (LCM) libraries. Building upon the LCM platform, a group of discipline specific computational models with disparate temporal and spatial scales are linked together to investigate the time-dependent interdependencies that arise between water, gas, and electrical power systems during a series of seismic events and the corresponding recovery processes. The results show that ignoring

interdependencies can adversely affect resilience assessments and adopting time-varying recovery strategies can lead to better resilience performance.

An agent-based computational model simulating benefit fraud behavior in the wake of a disaster is used to demonstrate that distributed simulation frameworks can take into account broader socio-technical interactions in resilience research. The study not only considered the effect of micro-level disaster-caused demands but also meso-level social factors on criminal tendencies. The proposed model captures the key characteristics of post-disaster benefit fraud in detail, including the dynamic nature of the criminal process. The results of parametric sensitivity analyses can be used to achieve a meaningful balance between the loss of fraudulent payments and the speed of distributing aid for improving the overall resilience performance of communities.

To provide a scalable, versatile, and user-friendly solution for natural hazards simulations, a new distributed computing tool called Simple Run-Time Infrastructure (SRTI) is employed. The high-level structure, data structure, and fundamental components of SRTI are comprehensively described. The applications of SRTI in natural hazard simulations are presented. The performance of an initial version of the SRTI is compared with the LCM. A cross-language simulation of time-dependent resilience analysis of an electric power system is conducted to show the scalability and flexibility of the improved version of the SRTI, which reduces a user's effort for composing a complex distributed simulation and better handle time management. Lastly, the choice between different versions of SRTI and potential features to develop in the future are discussed.

CHAPTER 1

Introduction

1.1 General

Communities are complex socio-technical systems. They consist of a number of infrastructure systems that are interdependent and continually interacting with one another, including the built environment, elements of social organization, institutional arrangements, and the economy. The built infrastructure contains buildings, bridges, lifeline systems and other physical systems. Social organization is the relationship between and among individuals and social groups, comprised of social power, leadership, community cohesion and crime sub-culture. Institutional arrangements, such as urban plans and disaster mitigation policies, and the economy are other infrastructure systems. These disparate systems operate and interact together quite efficiently most of the time, to the point where community residents rarely notice them on a day-to-day basis.

When a disruption such as a severe natural disaster occurs, the interactions between the infrastructure systems of society can lead to cascading events that can adversely affect community resilience. Resilience is the ability of a community to withstand, adapt and recover from a disruption, typically measured in terms of loss of life, injuries and economic cost. Because interdependencies can be highly influential, key US government documents published by the National Earthquake Resilience (NER, 2011), the Grand Challenges in Earthquake Engineering Research (GCEER, 2011) and the National Institute of Standards and Technology (NIST, 2016)

have called for the development of means by which to better address them in research studies. This, however, is complicated by the fact that each infrastructure system is rooted in a specific discipline or spread across a number of closely related disciplines. Achieving this goal is, therefore, difficult because of the challenges of transcending the disciplinary boundaries.

The vast majority of sub-fields in natural hazards engineering and disaster science areas have evolved to rely on computational models. This common focus on computation provides a powerful means for reaching across the disciplinary boundaries. Yet, doing so is still complicated by the disparate data types as well as temporal and spatial scales that characterize the variety of models used in the different fields. For example, in a seismic scenario, the earthquake event may take place in seconds, while the recovery process may take place over months. Earthquakes may cause damage to a component measured in meters, while repairs to lifelines may be spread across an areas measured in kilometers. As discussed in Chapter 2, to eliminate this multi-scale challenge, many existing resilience studies are independent of time or have focused on only a single spatial scale, for example a single phase of a disaster, a specific structure or geographic area. Lack of consideration of the time- dependent and spatial interactions across different physical and social infrastructure systems may greatly reduce the accuracy and fidelity of simulations.

Motivated by these needs, this dissertation has three main goals. The first one is to develop means by which to model and investigate the interactions between the different infrastructure systems of society during extreme natural disasters and the post-disaster recovery process. The second is to explore the effect of interdependencies on community resilience, with a specific focus on time-dependence. The third goal is to generalize the tools developed in this work into a computational simulation platform that will open the door for researchers from different fields to

connect their models and collaborate in new ways to study hazard mitigation and community resilience.

1.2 Objectives

Along with the primary goals described above, several key research questions that this dissertation aims to tackle include: What types of disciplinary models (simulators) can be linked together to form a meaningful simulation representing a hazard event? What interdependencies arise during severe disasters such as seismic and hurricane events? How can discipline specific computational models 'talk' to one another? How can simulators interact over disparate temporal and spatial scales? These research questions are addressed by achieving a series of specific objectives as follows:

(1) Review existing distributed simulation standards/platforms, and identify their limitations for use in natural hazards simulation. Compare their advantages and disadvantages, and provide design recommendations for new distributed simulation tools.

(2) Model the spatial and time-dependent interactions that occur between critical infrastructure systems using simulators in disparate temporal and spatial scales. Investigate the interdependencies that arise between these systems during a series of seismic events that include short- and long-term recovery processes.

(3) Address broader socio-technical considerations in resilience research, and conduct computational simulations to study the coupling effects between civil and social systems from both hazards engineering and social science perspective. Explore the combined effect of physical systems (damaged buildings) and social systems (social power, community

cohesion, and crime sub-culture) on benefit fraud behavior following disasters, and assess the impact of post-disaster crimes on community resilience.

(4) Develop a new distributed simulation tool, which is scalable, extensible and easy-to-use for natural hazards simulations. Investigate the data passing and time management issues across computational models in heterogeneous scales.

1.3 Organization of the Dissertation

A brief description of the six chapters comprising this dissertation is provided below.

Chapter 1: Introduction. General information underlying the motivation and goals of this research is presented. The key research problems that need to be addressed are outlined. The objectives and organization of this dissertation are also highlighted.

Chapter 2: Distributed Simulation Frameworks. This chapter systematically reviews the state-of-the-art in distributed simulation standards and tools. A distributed simulation framework that treats each discipline specific simulator as a black box subscribing to data from other simulators and/or publishing its results for others to use is proposed. The data transmission is enabled using Lightweight Communications and Marshalling (LCM) libraries. A case study of wind-induced progressive damage due to both the structural response of a building and external wind pressures is demonstrated.

Chapter 3: Time-Dependent Resilience Assessment of Interdependent Lifeline Systems. This chapter describes the resilience assessment of three interdependent lifeline systems, i.e. water, gas, and electrical power systems, subjected to a series of seismic events. The simulation is comprised of a group of independent simulators that interact together through a publish-subscribe pattern for data management. The framework addresses the spatial and time-dependent interactions

that arise between lifeline systems as the hazard and subsequent restoration processes unfold. Multiscale interdependency within the described hazard event and time-dependent effects of resource allocation strategies on community resilience are discussed. The ability of the framework to address the linkages that exist between civil and social systems is also presented.

Chapter 4: Effect of Benefit Fraud in the Wake of Disaster on Community Resilience. In this chapter, an agent-based computational model, which includes a simulation environment of a community facing a natural disaster, fraudsters, and application inspectors, is created to simulate benefit fraud behavior following disasters. The proposed model considers the effect of both micro-level disaster demands caused by building damages and meso-level social variables on benefit fraud, and estimates the cost to communities associated with these post-disaster crimes. The statistical data from the government reports in the aftermath of Hurricane Katrina and Rita is adopted for the calibration and validation.

Chapter 5: Simple Run-Time Infrastructure (SRTI) for Distributed Simulation of Community Resilience. This chapter introduces a new, scalable, versatile, and user-friendly distributed computing software solution, Simple Run-Time Infrastructure (SRTI). It is described comprehensively from the high-level structure to its fundamental components. The motivations, target users and development history of the SRTI are provided. The same case study of wind-excited buildings suffering progressive damage in Chapter 2 is conducted using the SRTI v1.00.00 to demonstrate the capabilities and compare the performance. Then, a cross-platform resilience analysis of a lifeline system subjected to an earthquake is built using the SRTI v2 GUI and executed through the SRTI v2.00.00 to show its scalability and flexibility. Lastly, a summary of work in progress and the potential features of the SRTI to develop is discussed.

Chapter 6: Summary, Conclusions, and Future Research. This chapter summarizes the key contributions and findings that can be drawn from this dissertation. Recommendations for future research in the area of interdependencies in community resilience and distributed disaster simulation are presented.

1.4 Publications from the Dissertation

Chapter 2 and Chapter 3 have been published as journal papers. Chapter 4 and the first half of Chapter 5 have been submitted to the American Society of Civil Engineers (ASCE) journals and under review. The second half of Chapter 5 is in preparation and will soon be submitted for publication. Details are listed below:

Lin, S.-Y., Chuang, W.C., Xu, L., El-Tawil, S., Spence, S.M.J., Kamat, V.R., Menassa, C.C., and McCormick, J. (2019). "Framework for Modeling Interdependent Effects in Natural Disasters: Application to Wind Engineering." *Journal of Structural Engineering*, 145(5): 04019025, DOI: 10.1061/(ASCE)ST.1943-541X.0002310. **(Chapter 2)**

Lin, S.-Y. and El-Tawil, S. (2020). "Time-Dependent Resilience Assessment of Seismic Damage and Restoration of Interdependent Lifeline Systems." *Journal of Infrastructure Systems*, DOI: 10.1061/(ASCE)IS.1943-555X.0000522. **(Chapter 3)**

Lin, S.-Y. and El-Tawil, S., Aguirre, B.E. "Computational Simulation of Benefit Fraud and Community Resilience in the Wake of Disaster." *Natural Hazards Review*, under review. **(Chapter 4)**

Lin, S.-Y., Hlynka, A. W., Xu, L., Lu, H., El-Tawil, S., Kamat, V.R., Prakash A., Menassa, C.C., Spence, S. M. J., McCormick, J., and Aguirre, B. "Simple Run-Time Infrastructure (SRTI):

A Distributed Computing Software Solution for Simulating." *Natural Hazards Review*, under review. **(Chapter 5)**

Lin, S.-Y., Hlynka, A. W., Xu, L., Lu, H., Sediek, O.A., El-Tawil, S., Kamat, V.R., Prakash A., Menassa, C.C., Spence, S. M. J., McCormick, J., and Aguirre, B. "Simple Run-Time Infrastructure (SRTI): A Distributed Computing Platform with Extremely Low Barrier to Entry." In preparation. **(Chapter 5)**

CHAPTER 2

Distributed Simulation Frameworks

2.1 General

This chapter reviews the existing modeling techniques used to study the interdependencies that arise during severe disasters (Section 2.2.1), and the state-of-the-art in distributed simulation standards and tools (Section 2.2.2). A distributed simulation framework that treats each discipline specific simulator as a black box subscribing to data from other simulators and/or publishing its results for others to use is proposed (Section 2.3). The data transmission is practiced using the Lightweight Communications and Marshalling (LCM) libraries. A case study of wind-induced progressive damage due to both the structural response of a building and external wind pressures is demonstrated (Section 2.4). Lastly, the advantages and potential of the proposed framework for modeling community resilience are discussed (Section 2.5).

2.2 Background

2.2.1 Modeling of interdependences in hazard engineering

As discussed in Chapter 1, the interactions that occur between the various components of a natural disaster scenario are important to model in order to create a realistic assessment of the situation. Rinaldi et al. (2001) categorized interdependencies into four main classes: physical, cyber, geographic, and logical. Lee et al. (2007) identified five types of interrelationship between infrastructure systems, i.e., input, mutual, shared, exclusive, and co-located dependence. Zhang

and Peeta (2011) categorized the interdependencies among infrastructure systems into the following types: functional, physical, budgetary, market and economic interdependency. Eusgeld et al. (2008) and Ouyang (2014) suggested that existing methods used to study interdependent critical infrastructure systems can be broadly grouped into several types: empirical, agent-based, system dynamics, economic theory and network-based approaches. As noted in Ouyang (2014), each type of modeling approach has its strengths and weaknesses, but it is clear from the literature that the two main classes of tools used in natural hazard simulations are agent-based models and network-based models.

Agent-based modeling is a computational method that allows embedded agents, which are computational entities that represent the behavior of autonomous actors, to interact together with a view towards assessing their effects on the overall system. Examples of agent-based studies in the natural hazards field can be found in Reilly et al. (2017), Barton et al. (2000) and Schoenwald et al. (2004). The flexibility and capability of capturing behavior at the component (agent) level allow agent-based models to address many types of interdependencies and integrate well with other modeling techniques. This view is widely held as discussed in Ouyang (2014), GCEER (2011) and NER (2011). However, the key challenge in implementing agent-based models is managing the information that must be transmitted between agents and other simulation models. In many cases, the information format is hardwired into the simulation platform itself, which severely limits the ability of the platform to be extended to handle new situations and accommodate new models. For example, in Chang and Miles (2004) and Miles and Chang (2003, 2007), the agents were implemented in a Matlab/Simulink software environment which restricts future growth to only that particular platform. There is a handful of instances in the literature where more powerful data management techniques, such as high level architecture (discussed later on in this chapter), have

been used in natural disaster simulations, e.g. Fiedrich (2006), Jain and McLean (2006), Casalicchio et al. (2007), Eusgeld and Nan (2009), and Eusgeld et al. (2011).

The network-based approach is mainly used to model interdependencies across lifelines or other network systems. Using this method, Hernandez-Fajardo and Dueñas-Osorio (2011) assessed the propagation of fragility across electric power and water distribution systems after an external perturbation. Guidotti et al. (2016) and Ellingwood et al. (2016) modeled the effect of water and electric power systems interdependencies on community resilience. The network-based approach employs interdependence matrices, which indicate the strength of dependency of one network on another at the component level (Ellingwood et al., 2016). While interdependence matrices are conceptually simple, their application to multi-network systems can be cumbersome to formulate because of their two-dimensional nature. In such cases, multiple interdependence matrices must be used to model the relationship between different pairs of networks, resulting in an exponential increase in the number of interdependence matrices with increasing number of networks. In addition, interdependence relationships must necessarily be simple ones, e.g. link (one), no link (zero) or partial link (between zero and one), because they are represented by a single entry in a matrix. Another key limitation of this approach is that it only applies to systems that can be simplified into nodes and links. In spite of their drawbacks and primarily because the basic idea is simple, numerous other studies have used interdependence matrices including Cimellaro et al. (2014) and Cimellaro et al. (2016).

The above discussion indicates that common implementations of the two main classes of techniques used to model interactions in natural hazard situations have a number of drawbacks. With this as motivation, a new simulation framework that addresses these limitations is developed in this chapter. The platform employs a publish-subscribe pattern for data management and is

designed to enable the different components of a hazard simulation, each represented by a simulator, to interact together. Each simulator publishes its results (in a "message") to a corresponding "channel." Other simulators, which need the information, subscribe to the channels and receive published messages from them. There are no restrictions on how each simulator is developed or operates; for example, it could be agent-based, network-based, a Matlab/Simulink model or developed using some other methodology. Each simulator is viewed as a black box that subscribes to data from other simulators and/or publishes its results for other simulators to use. Modifiability and scalability are the key advantages of this methodology. In particular, it allows simulators to be replaced based on different theories or algorithms and permits new simulators to be added to existing simulation frameworks, allowing for increasing levels of complexity.

2.2.2 Existing distributed simulation platforms

Distributed simulation is a computational technique that facilitates interactions between simulation models regardless of the type of models employed or computing platforms on which they run (Fujimoto, 2015). The interactions in question pertain primarily to data communication and action synchronization. Several frameworks have been developed to enable distributed simulations in a number of fields such as robotics, manufacturing, war-gaming, and the Internet-of-Things (IoT).

Distributed simulation platforms fall into two general categories, standards-based and ad-hoc. The former is typically geared towards general purpose applications and governed by a set of formal specifications intended to ensure successful execution of extensive, complex simulations. However, the broad and comprehensive nature of the standards complicates adoption by new users. This situation has prompted the development of commercial software solutions to facilitate

adoption, but these proprietary applications prevent the development of truly open source platforms. By reducing the rules governing the development of new simulations, ad-hoc platforms are generally more straightforward to adopt and use and are therefore attractive for new users wary of the steep learning curve associated with standards-based tools. However, ad-hoc platforms shift the responsibility for handling various essential features of distributed simulations to the users.

One of the earliest standards-based tools is the Distributed Interactive Simulation (DIS) platform (IEEE, 1993) developed by the US Department of Defense. DIS is currently maintained and developed by the Simulation Interoperability Standards Group (SISO,) and documented under IEEE Standard 1278 series since 1993 (IEEE, 2012, 2015).

High-Level Architecture (HLA) is another set of standards (IEEE, 2010b, 2010a, 2010c) for distributed simulation. It was also developed by the US Department of Defense. It is a set of standards that foster interoperability and therefore facilitate future simulation efforts. There are three primary components to the HLA specification: the HLA rules [IEEE 1516 (IEEE, 2010a)], the HLA interface specification (IFSpec) [IEEE 1516.1 (IEEE, 2010b)], and the Object Model Template (OMT) [IEEE 1516.2 (IEEE, 2010c)]. The HLA interface specification defines the functional modes of interaction between multiple models and the framework's Run-Time Infrastructure (RTI). The RTI is a middleware that must conform to the HLA specifications and provide simulation facilitation services such as time and data synchronization, bi-directional communication, and data exchange among various participating models to enable an HLA-compliant simulation environment. All simulation entities (or objects) and interactions (or transactions) that are part of a composed simulation are defined in a common, pre-agreed format that will support multiple model interaction, data exchange, and reusability. HLA was originally developed for military war-gaming applications (Hollenbach, 2009), although it is now being

applied for other purposes (Cayirci, 2013). A prominent alternative that also originates from the US DoD is the Test and Training Enabling Architecture (TENA) (Powell and Noseworthy, 2012). Similar to HLA, TENA uses a middleware to organize message communication.

Distributed Data Services (DDS) is a middleware protocol and standard managed by the Object Management Group (OMG, 2015) that provides low-latency data communications through a data-centric publish–subscribe model. Its objective is to allow scalable, real-time, reliable and interoperable data exchanges. Like the RTI in HLA, the Global Data Space in DDS is the place where applications can share information by simply reading and writing data objects. DDS features a rich set of Quality of Service (QoS) properties that assure data reliability, ownership, durability and delivery deadlines (OMG, 2015). However, it does not have specific time management properties like HLA. DDS standards are open-source and have been applied in many fields including naval combat systems (Khaefi et al., 2015), remote systems control (Garcia-Valls and Basanta-Val, 2013), and Internet-of-Things (IoT) (Beckmann and Dedi, 2015).

An example of an ad-hoc distributed simulation platform is the Lightweight Communication and Marshalling (LCM) (LCM, 2018). LCM is an open-source, seamless connection, language and platform independent, inter-process communication tool. It was developed and employed for modeling the information flow that takes place between robot swarms and is especially efficient for its intended application because it features low-latency and does not require a centralized hub (Huang et al., 2010). As such, data communication in LCM is not controlled by middleware, e.g. the RTI in HLA that imposes restrictive rules on data formats, thereby permitting rapid and straightforward modification of the platform for the intended purpose. In this chapter, the LCM are employed to build the distributed simulation framework.

2.3 Distributed Simulation Framework Using LCM

The key idea in the developed distributed simulation framework is that each component of a disaster simulation, e.g. a wind loading model or structural response model, is represented by a simulator that models its behavior. Since there are well-established simulation models in various fields, the biggest challenge of realizing a distributed simulation platform is finding a way to integrate these computational models together and managing the data transmitted between them. The Lightweight Communications and Marshalling (LCM) libraries (LCM, 2018) are ideal for this purpose.

In LCM, the type-defined and marshalled messages pass through a channel, to which simulators can subscribe to required data from others and publish results for other simulators to use. The data transmitting method in LCM complies with the User Datagram Protocol (UDP) multicast (Postel, 1980) which is a low-latency data transport protocol. It allows LCM to perform real-time communication between various workstations with no need for a centralized hub or middleware as in other distributed simulation systems.

Another favorable feature of LCM is that it is platform and language independent. Hence, the developed system can be extended through addition of new simulators provided by other users no matter what programming language they use provided they have a means of sending and receiving external data. Besides the simple message-passing feature, LCM also provides several tools to help users debug, record and inspect message traffic through the channels, e.g., *lcm-spy* and *lcm-logplayer*. *lcm-spy* is a Java implementation that displays live message traffic and allows developers to identify common run-time errors, e.g., invalid data and format mismatches. *lcm-logplayer* is a playback tool to replay the actual transmitted messages. These tools enable users to observe and analyze live message traffic without affecting system performance (Huang et al., 2010).

The practice of LCM can be divided into three dimensions: type definition, marshalling, and communication, as its name suggests. First, all transmitted messages must be well defined and conform to the type specification of LCM, which is language-natural and similar to C. Then, the language-specific data structures, called classes, for all messages can be automatically generated and marshalled into a binding by the lcm-gen tool in LCM. In order to conduct valid communications, all distributed simulators must have the same type definition binding, i.e., the same message interpretations. Types of messages are checked by a prepending fingerprint function of LCM before they are passed across computers. Once all type checks are completed, the adequately defined messages are ready to be transmitted to the different channels where receivers can subscribe to the channels pertinent to them. By allowing uninteresting data to be skipped, a simple and efficient inter-process communication is then established.

The concept behind the distributed computing platform is portrayed in Figure 2-1, which depicts an LCM setup for a wind-excited building event. Simulators pertinent to such an event (shown in Figure 2-1) are discussed in detail later. A disaster simulation may execute on a single computer or a set of machines connected to one another through a local area network (LAN) or wide area network (WAN) via the Internet. Simulators are decentralized and send or receive messages based on their needs. In this way, each discrete event simulation can be developed individually and all computational models are reusable and extensible in a plug-and-play sense.

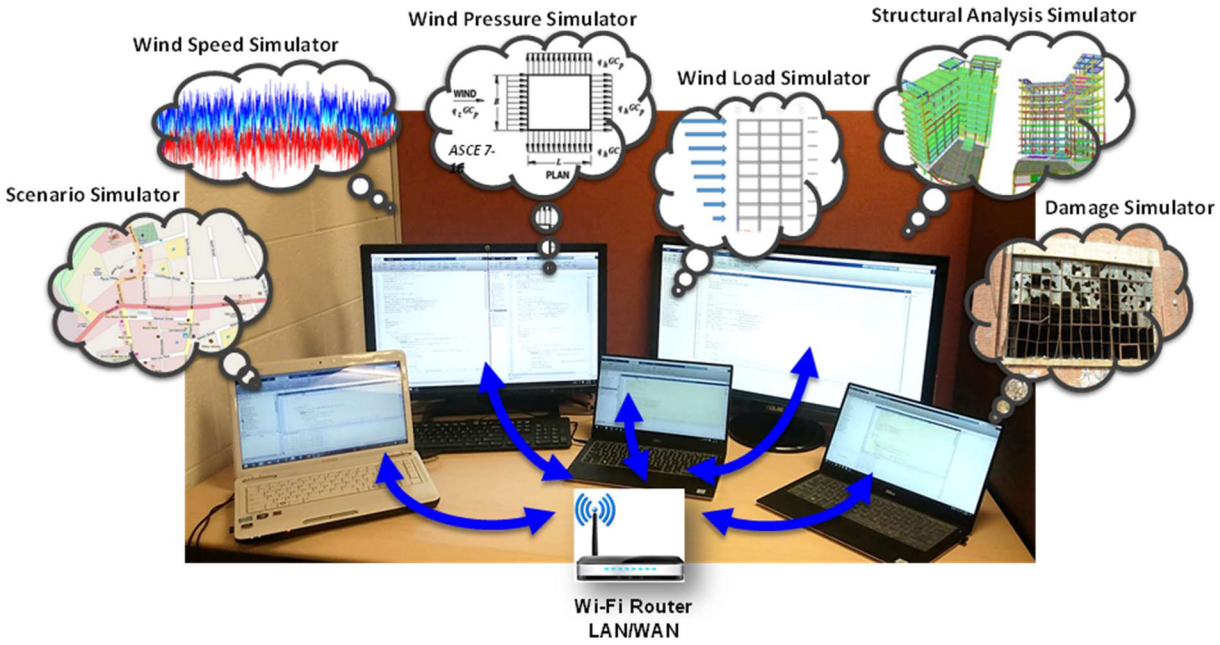


Figure 2-1. Distributed simulation platform for a wind-loading event. Each simulator runs on a separate computer in the shown example

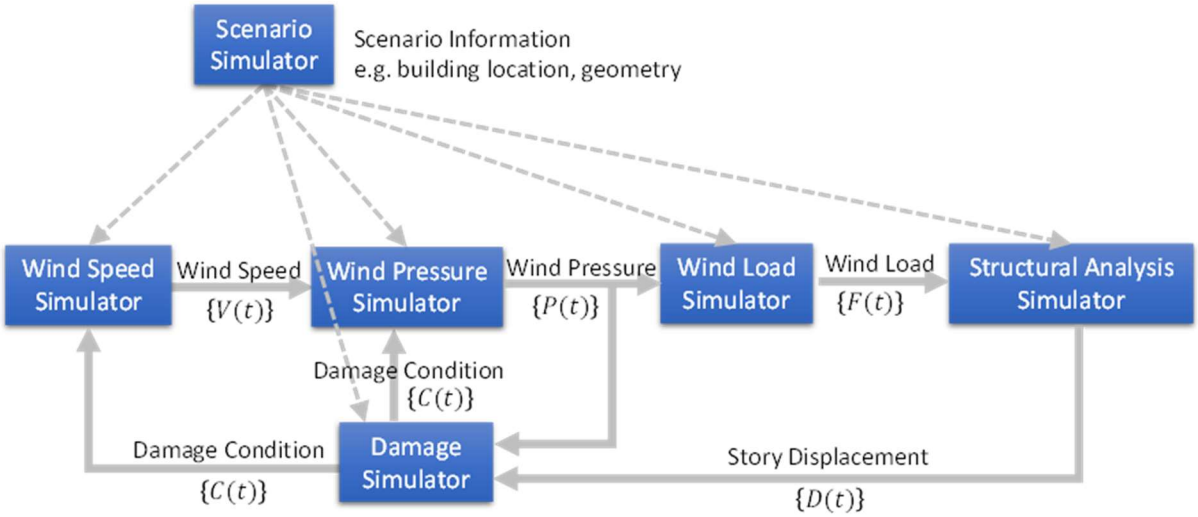


Figure 2-2. Schematic diagram of LCM framework

2.4 Case Study: Application to Wind Engineering

Analysis of a collection of multistory buildings that suffer progressive damage during a wind event is presented to demonstrate the capabilities of the developed framework and show that it can lead to meaningful results. Although this example comes from the field of structural engineering, the concept can be applied to address other more complex situations, as discussed later on in this chapter. As shown in Figure 2-1, the problem is represented by six separate simulators: *Scenario Simulator*, *Wind Speed Simulator*, *Wind Pressure Simulator*, *Wind Load Simulator*, *Structural Analysis Simulator*, and *Damage Simulator*. The framework, analysis procedure and data flow are shown in Figure 2-2.

In Figure 2-2, the *Scenario Simulator* broadcasts information about the situation being simulated, specifically static information such as the location and geometry of the affected buildings. The *Wind Speed Simulator* generates time dependent wind speeds at each story of each building as a function of floor height. The *Damage Simulator* provides the damage condition of each building and assesses if windows are broken due to excessive wind pressures or story drift ratios. The *Wind Pressure Simulator* determines local wind pressures based on associated wind speeds and the damage condition of the windows. The *Wind Load Simulator* computes the equivalent wind forces applied at each story as a function of the applied wind pressures. The *Structural Analysis Simulator* calculates the dynamic response of the building and outputs the displacements of each story. These are then used to check for structural response-induced damage by the *Damage Simulator*.

As shown in Figure 2-2, each simulator represents one part of the wind performance workflow, is executed on a separate computer, and connects to other simulators through the LCM framework. The types of messages produced are shown in Table 2-1 while the messages published

and subscribed to by each simulator are listed in Table 2-2. The LCM channels and message flows are shown in Figure 2-3. Example Matlab code of the Structural Analysis Simulator using LCM is shown Figure 2-4, which includes the codes for initializing the LCM, subscribing, receiving, marshalling and publishing messages.

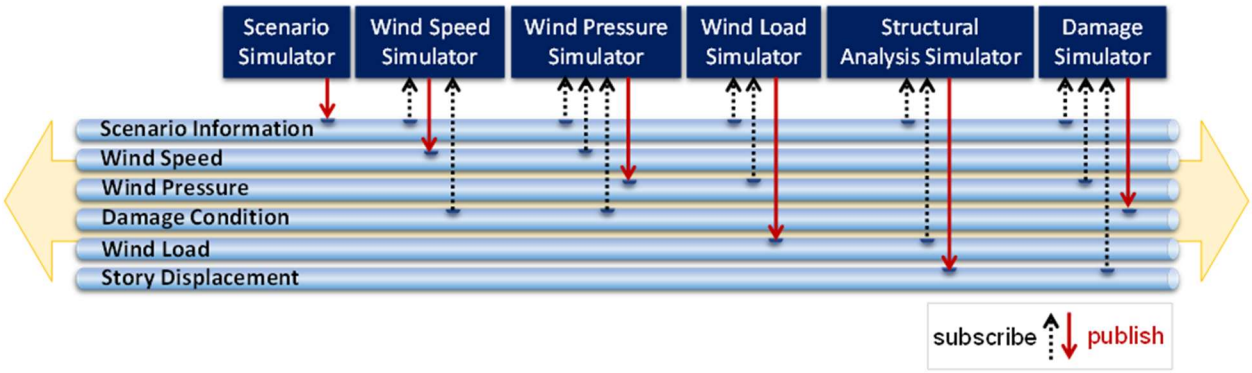


Figure 2-3. LCM channels and message flows

Table 2-1. Types of messages produced

Code	Message Description
I	A matrix containing scenario information (location and geometry of buildings) that does not change with time
V(*t*)	A vector of wind speed for each story at time *t*
P(*t*)	A vector of wind pressure for each story at time *t*
F(*t*)	A vector of wind load for each story at time *t*
D(*t*)	A vector of story displacements at time *t*
C(*t*)	A vector of damage condition for windows at time *t*

Table 2-2. Messages published or subscribed to by the simulators

Simulator	Message Published	Message Subscribed
Scenario Simulator	*I*	-
Wind Speed Simulator	*V*(*t*)	*I*, *C*(*t*)
Wind Pressure Simulator	*P*(*t*)	*I*, *V*(*t*), *C*(*t*)
Wind Load Simulator	*F*(*t*)	*I*, *P*(*t*)
Structural Analysis Simulator	*D*(*t*)	*I*, *F*(*t*)
Damage Simulator	*C*(*t*)	*I*, *P*(*t*), *D*(*t*)

```
%======== Add LCM and the message bindings to the MATLAB classpath =======
javaaddpath lcm.jar
javaaddpath message_definition.jar  % compiled message definition binding

%======== Initialize LCM ================================================
lc = lcm.lcm.LCM.getSingleton();

%======== Subscribe messages ============================================
aggre_scenario = lcm.lcm.MessageAggregator();
lc.subscribe('Scenario', aggre_scenario);
aggre_windload = lcm.lcm.MessageAggregator();
lc.subscribe('WindLoad', aggre_windload);

%======== Receive subscribed messages ===================================
millis_to_wait = 1000;
msg_scenario = aggre_scenario.getNextMessage(millis_to_wait);
m_sce = whlcm.scenario_t(msg_scenario.data);
msg_windload = aggre_windload.getNextMessage(millis_to_wait);
m_windload = whlcm.windload_t(msg_windload.data);
% Based on the definition in 'message_definition.jar'
% 'm_sce' contains variables:'m_sce.nbuildings', 'm_sce.buildinginfo'
% 'm_windload' contains variables:'m_windload.step', 'm_windload.loads'

%======= Calculate displacement =========================================
displacements = StruAnaModel...
    (m_sce.nbuildings,m_sce.buildinginfo,m_windload.step,m_windload.loads);

%======= Marshal messages ================================================
msg = whlcm.storydisp_t();
msg.step = m_windload.gstep;
msg.step_time = m_windload.gstep_time;
msg.buildings = m_sce.nbuildings;
msg.disps = displacements;

%======= Publish messages ================================================
lc.publish('StoryDisp', msg);
```

Figure 2-4. Example Matlab code of Structural Analysis Simulator using LCM

The computational problem is solved by incrementally stepping through time. Increments are typically applied in an Euler sense, where initial conditions are based on the previous time step and assumed to prevail during the increment. They are then updated at the beginning of the next increment. For added accuracy and to resolve situations that change substantially during an increment, iterations can be performed within each increment. For example, the effective wind pressure is a function of the condition of windows (broken or not) and the area of the opening, but

windows may be broken by excessive wind pressure or story drift. As shown in Figure 2-2, The *Damage Simulator, Wind Pressure Simulator, Structural Analysis Simulator* are therefore interdependent and iterate in a local loop in order to reach a solution in which window damage and associated pressures are consistent at each time step. Iteration stops when the damage condition is consistent with the wind pressure, typically within two cycles. The converged window status is used as the initial damage state for the first iteration of the next time step. As demonstrated later on, for small time steps, it is feasible to skip iterations and conduct the analysis in a purely incremental manner.

2.4.1 Description of simulators

Scenario Simulator

The *Scenario Simulator* represents an urban community consisting of four buildings, as rendered in Figure 2-5 and shown in plan in Figure 2-6(a). Each building has a 42.67 x 30.48 m rectangular plan [see Figure 2-6(b)], flat roof, and an inter-story height h of 3.96 m. The buildings have different heights and orientations. In particular, buildings B1 and B4 are 8- and 4-stories, respectively, and are orientated so that the incident wind impacts the long side of the building. Buildings B2 and B3 are 20-story buildings that are orientated so that the wind impacts the short side of B2 and the long side of B3. The geometry of all four buildings is summarized in Table 2-3 and Figure 2-7, where *B* and *L* are the orthogonal horizontal dimensions of the buildings.

Special moment resisting steel frames are used for lateral resistance of the buildings. The frames are located at the perimeter of each building, as indicated in Figure 2 6(b). Member sizes are summarized in Table 2-4 and additional structural details can be found in NIST (2010). For the wind direction illustrated in Figure 2-6(a), the two moment frames parallel to the wind direction

will resist the alongwind forces. With the exception of the outer three meters of each face, windows of width 1.2 m are assumed to cover the entire face of each building. As such, there are 20 windows along the short side and 30 windows along the long side of each building at each floor, as illustrated in Figure 2-8.

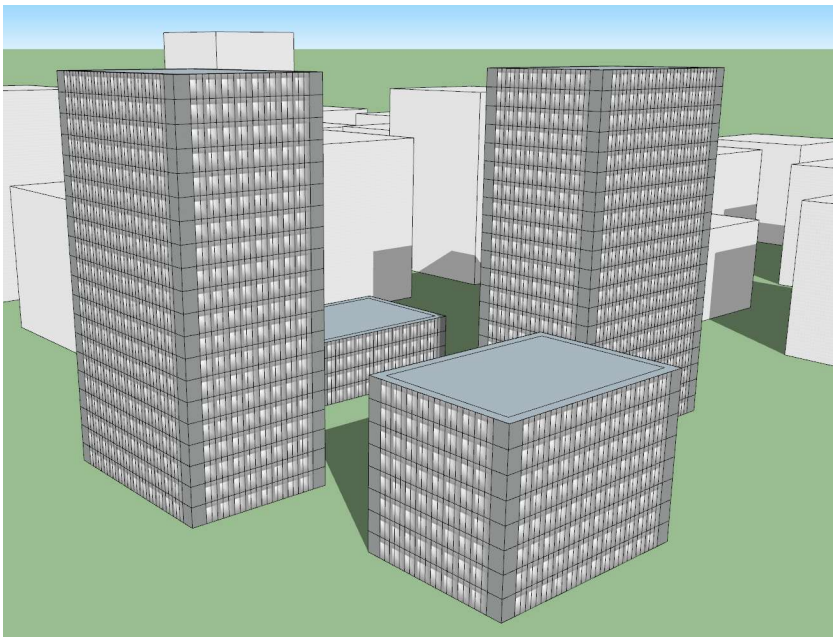


Figure 2-5. 3D rendering of the four buildings considered in the case study

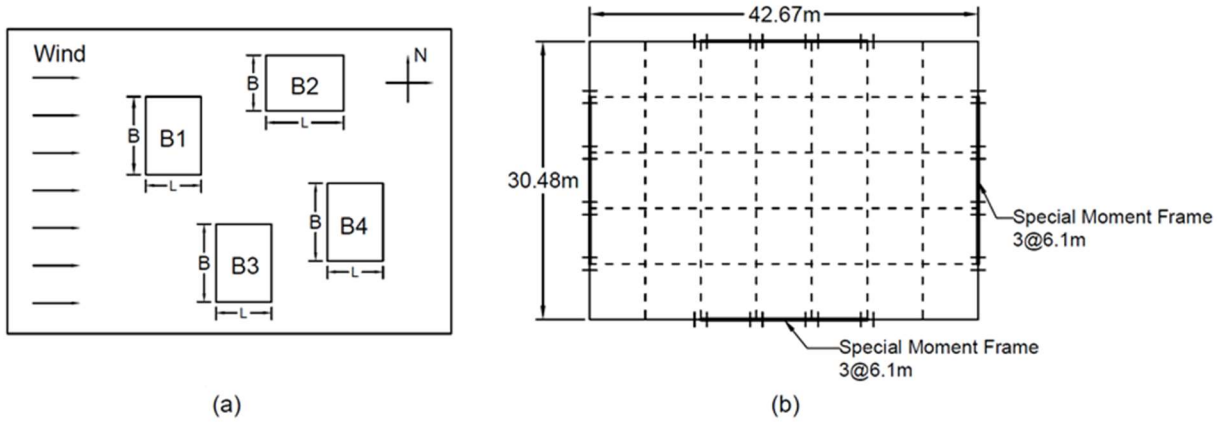


Figure 2-6. Plan view of: (a) the four building community; and (b) structural layout of each Building

Table 2-3. Geometric characterization for the buildings of the case study

Building	Story	*B* (m)	*L* (m)
B1	8	42.67	30.48
B2	20	30.48	42.67
B3	20	42.67	30.48
B4	4	42.67	30.48

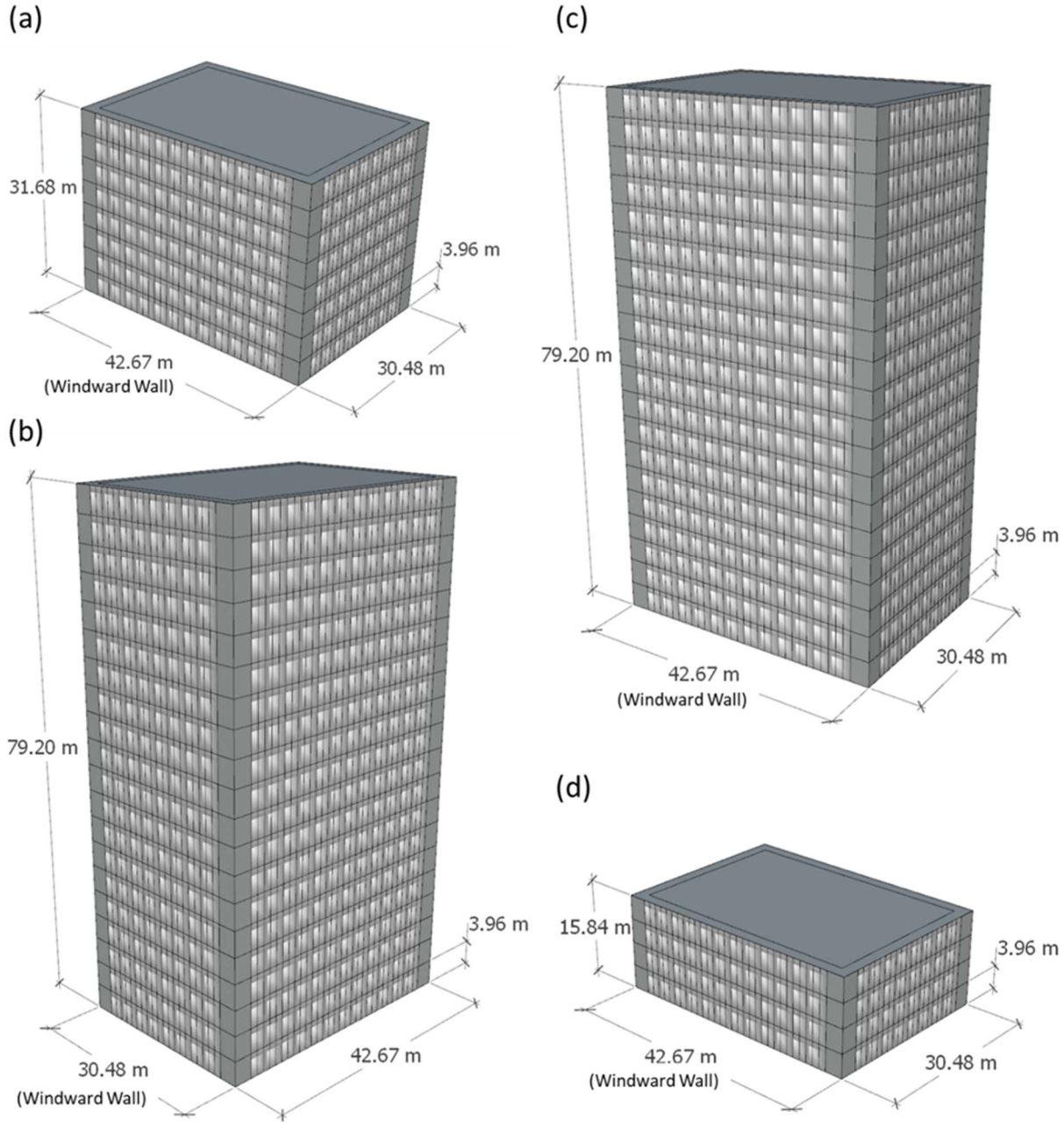


Figure 2-7. Geometry characterization of the simulated buildings: (a) B1, 8 stories; (b) B2, 20 stories; (c) B3, 20 stories; (d) B4, 4 stories

Table 2-4. Member sizes for special moment resisting steel frames

Story	Beam size	Exterior column size	Interior column size	Doubler plate size (mm)	
				Exterior column	Interior column
B1 (8-stories)					
1	W30×108	W24×131	W24×162	1.59	14.29
2	W30×116	W24×131	W24×162	1.59	9.53
3	W30×116	W24×131	W24×162	1.59	17.46
4	W27×94	W24×131	W24×162	0	9.53
5	W27×94	W24×131	W24×131	0	14.29
6	W24×84	W24×131	W24×131	0	11.11
7	W24×84	W24×94	W24×94	0	14.29
8	W21×68	W24×94	W24×94	0	7.94
B2 and B3 (20-stories)					
1	W33×169	W14×426	W24×335	0	6.35
2	W33×169	W14×426	W24×335	0	6.35
3	W33×169	W14×426	W24×335	0	6.35
4	W33×169	W14×426	W24×335	0	6.35
5	W33×169	W14×398	W24×335	0	6.35
6	W33×169	W14×398	W24×335	0	6.35
7	W33×169	W14×370	W24×335	0	6.35
8	W33×169	W14×370	W24×335	0	6.35
9	W33×141	W14×311	W24×279	0	7.94
10	W33×141	W14×311	W24×279	0	7.94
11	W33×141	W14×283	W24×250	0	12.7
12	W33×141	W14×283	W24×250	0	12.7
13	W33×141	W14×233	W24×250	1.59	12.7
14	W33×141	W14×233	W24×250	1.59	12.7
15	W30×108	W14×159	W24×162	6.35	14.29
16	W30×108	W14×159	W24×162	6.35	14.29
17	W30×108	W14×132	W24×162	9.53	14.29
18	W30×108	W14×132	W24×162	9.53	14.29
19	W24×62	W14×132	W24×103	0	4.76
20	W24×62	W14×132	W24×103	0	4.76
B4 (4-stories)					
1	W21×73	W24×103	W24×103	0	7.94
2	W21×73	W24×103	W24×103	0	7.94
3	W21×57	W24×62	W24×62	0	7.94
4	W21×57	W24×62	W24×62	0	7.94

Source: Adapted from NIST (2010).

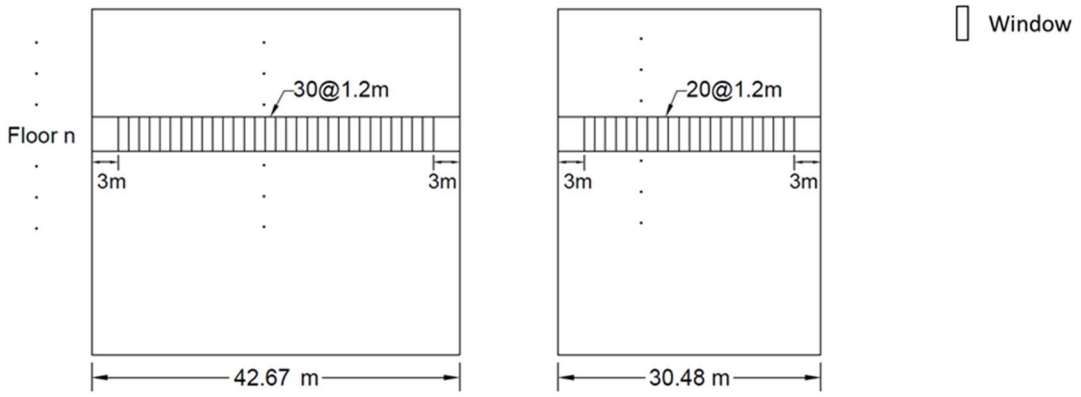


Figure 2-8. Window location of the representative buildings

Wind Speed Simulator[1]

The *Wind Speed Simulator* generates the fluctuating wind speeds at specific heights through the following quasi-steady relation:

$$v_{z_j}^2(t) = \left(\bar{v}_{z_j} + v_j(t)\right)^2 \approx \bar{v}_{z_j}^2 + 2\bar{v}_{z_j} v_j(t), \quad j = 1,2,\dots,n \tag{2-1}$$

where n is the total number of heights of interest, $\bar{v}_{z_j}$ is the maximum mean hourly wind velocity at height z_j, and $v_j(t)$ is the corresponding fluctuating component of the wind speed simulated through the model outlined in Deodatis (1996). The wind speed $\bar{v}_{z_j}$, is extrapolated from meteorological data recorded at height H_{met} through the probabilistic transformation outlined in Chuang and Spence (2017). This formulation allows for the explicit modeling of the uncertainties affecting parameters such as site roughness and observational errors. For this case study, the largest yearly meteorological wind speed is taken as 40 m/s, averaged over $\tau = 1$ hour with a roughness

[1] Prof. Seymour M.J. Spence and Dr. Wei-Chu Chuang contributed to the design and implementation of the *Wind Speed Simulator* and *Wind Pressure Simulator*.

length of $z_{01} = 0.05$ and a meteorological height of $H_{met} = 10$ m. The roughness length at the site of the four buildings, z_0, is taken as 2 m. In calibrating the model, the total duration of the wind storm is taken as $T = 4{,}021$ s with a sampling frequency of 1.27 Hz.

Wind Pressure Simulator[1]

In this simulator, the wind pressures acting on the main wind force resisting system (MWFRS) and components and cladding (C&C) at each time step are determined based on the wind speeds derived from the *Wind Speed Simulator* and Enclosure Classification derived from the *Damage Simulator*. To estimate the wind pressure acting on the building, the velocity pressure at height z above ground is first identified from the following equation:

$$q_z(t) = 0.5\rho {v_z}^2(t) \tag{2-2}$$

where $\rho = 1.25 kg/m^3$ is the air density. The wind pressures for the MWFRS at height z are then determined as

$$p_z = qGC_p - q_i(GC_{pi}) \tag{2-3}$$

where: $q = q_z$ for windward walls; $q = q_h$ for the leeward walls and sidewalls with h the eave height; $G = 0.85$ and is the gust-effect factor; C_p is the external pressure coefficient defined in Table 2-5; $q_i = q_h$; and C_{pi} is the internal pressure coefficient defined in Table 2-6. In particular, C_{pi} depends on the Enclosure Classification, which is determined by the *Damage Simulator* for each building at each time step. To estimate the wind pressures acting on the C&C, the procedures of the ASCE 7-16 (ASCE, 2017) provisions are followed. In this respect, the local wind pressures are given by Equation (2-3) while considering the following external pressure coefficient GC_p:

$$\begin{aligned} &\text{if } A_{eff} < A_{eff1} \Rightarrow GC_p = \left(GC_p\right)_1 \\ &\text{if } A_{eff} \geq A_{eff2} \Rightarrow GC_p = \left(GC_p\right)_2 \\ &\text{if } A_{eff1} \leq A_{eff} < A_{eff2} \Rightarrow GC_p = \left(GC_p\right)_1 + g\log\left(A_{eff}/A_{eff1}\right) \end{aligned} \tag{2-4}$$

where $\left(GC_p\right)_1$ and $\left(GC_p\right)_2$ are the external pressure coefficients, A_{eff1} and A_{eff2} are the effective areas, and g is a coefficient depending on the wall and pressure distribution under consideration (ASCE, 2017). Because only the windows are considered damageable in this work, only zone 4, as specified in ASCE 7-16, of each wall is of interest, as summarized in Table 2-7.

Table 2-5. External pressure coefficient C_p

Surface	*L/B*	C_p
Windward wall	All values	0.8
Leeward wall	$0-1$	-0.5
	2	-0.3
	≥ 4	-0.2
Sidewall	All values	-0.7

Source: Data from ASCE (2017).

Table 2-6. Internal pressure coefficient C_{pi}

Enclosure Classification	Internal Pressure	C_{pi}
Enclosed buildings	Moderate	$\pm$0.18*
Partially enclosed buildings	High	$\pm$0.55*
Partially open buildings	Negligible	0

Source: Data from ASCE (2017).
* The plus and minus signs indicate pressures acting toward and away from the surfaces

Table 2-7. Effective areas A_{eff1}, A_{eff2}, corresponding external pressure coefficients $(GC_p)_1$, $(GC_p)_2$ and coefficient g for Zone 4 of a wall

Zone	Designed Pressure	$(GC_p)_1$	$(GC_p)_2$	A_{eff1} (m^2)	A_{eff2} (m^2)	g
4 and 5	Positive	0.9	0.6	1.9	46.5	-0.215
4	Negative	-0.9	-0.7	1.9	46.5	0.143

Wind Load Simulator

The equivalent wind force F is calculated by multiplying the influence area A and the combination of wind pressure p on windward and leeward walls which gives:

$$F = A\left[qGC_p - q_i\left(GC_{pi}\right)\right]_w - A\left[qGC_p - q_i\left(GC_{pi}\right)\right]_l \tag{2-5}$$

where the subscripts w and l indicate the windward and leeward walls while $A = B \cdot h$ is the influence area.

Structural Analysis Simulator

The *Structural Analysis Simulator* estimates the dynamic responses of a building using a central difference scheme. In particular, a floor-wise lumped mass model is employed considering only the degrees-of-freedom (DOFs) of interest (e.g. the horizontal displacement at each floor). As such, the equations of motion can be expressed through static condensation as

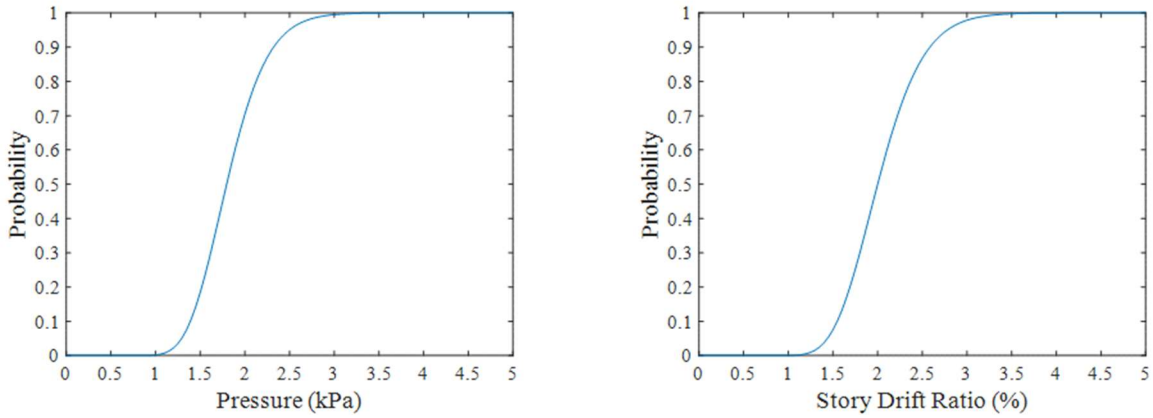
$$\begin{bmatrix} \mathbf{M_{tt}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}_\mathbf{t}(t) \\ \ddot{\mathbf{u}}_\mathbf{r}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{C_{tt}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}}_\mathbf{t}(t) \\ \dot{\mathbf{u}}_\mathbf{r}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{K_{tt}} & \mathbf{K_{tr}} \\ \mathbf{K_{rt}} & \mathbf{K_{rr}} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_\mathbf{t}(t) \\ \mathbf{u}_\mathbf{r}(t) \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}(t) \\ \mathbf{0} \end{Bmatrix} \tag{2-6}$$

where $\mathbf{u_t}(t)$ and $\mathbf{u_r}(t)$ are the vectors of the dynamically significant and secondary DOFs respectively; $\mathbf{C_{tt}}$ is the damping matrix of the condensed system; $\mathbf{K_{tt}}$, $\mathbf{K_{tr}}$, $\mathbf{K_{rt}}$ and $\mathbf{K_{rr}}$ are the stiffness submatrices obtained from the partitioning of the complete stiffness matrix $\mathbf{K}$; $\mathbf{M_{tt}}$ is the lumped mass matrix; and $\mathbf{F}(t)$ is the vector of stochastic forcing functions. Damping is calibrated by considering a modal damping ratio of 0.02 for each mode.

Damage Simulator

In this simulator, window damage and Enclosure Classification of a building are determined at each time step of the simulation. In particular, a window is considered damaged –

resulting in an opening in the wall – if the wind pressure or drift ratio exceed predefined pressure and drift capacities defined through fragility functions. For this case study, the fragility functions are assumed to be lognormally distributed with dispersion $\beta = 0.2$. The mean value associated with the pressure fragility functions is taken as 1.8 kPa for windows on the windward walls and 3.6 kPa for windows on all other surfaces (FEMA, 2006). A mean value of 0.02 is assumed for all fragility functions associated with drift induced damage (Sivanerupan et al., 2014). The fragility functions associated with pressure induced damage to the windward wall and drift induced damage are shown in Figure 2-9. It should be observed that in calculating the wind-induced dynamic response, only the alongwind direction is considered. Therefore, damage due to inter-story drift could occur only on the sidewalls.



(a) (b)

Figure 2-9. Fragility functions for (a) wind pressure on windward wall and (b) inter-story drift

The Enclosure Classification is essential for estimating the internal pressure of a wind excited building. In this case study, three categories are assumed, namely, Enclosed, Partially Enclosed and Partially Open. The criteria necessary for satisfying each category are summarized in Table 2-8. In particular, a building is classified as "Enclosed" if there are insufficient openings in the building envelope for the air to flow through. An internal pressure change can be observed

in a “Partially Enclosed” building where there are sufficient openings in one wall to allow for the flow of air through the envelope while all other walls remain practically unbreached. The classification of “Partially Open” is assumed when internal pressure is negligible.

Table 2-8. Criteria for enclosure classification

Enclosure Classification	Criteria for Enclosure Classification
Enclosed	$A_o < \min(0.01A_g, 0.37m^2)$ $A_o > 1.10A_{oi}$
Partially Enclosed	$A_o > \min(0.01A_g, 0.37m^2)$ $A_{oi}/A_{gi} \leq 0.2$
Partially Open	Others

Note: A_o= total area of openings in a wall receiving positive external pressure; A_g= gross area of that wall where A_o is identified; A_{oi}= sum of area of openings in the building envelope not including A_o; A_{gi}= sum of gross surface areas of the building envelope not including A_g.

2.4.2 Results and discussion

The simulators outlined above are used to analyze the wind-induced response of the four buildings defining the community, and both incremental-iterative and incremental (where iterations are not performed) simulations are conducted. Unlike the incremental scheme, which directly exercises the simulators sequentially within the time step, the incremental-iterative scheme additionally checks to see if the Enclosure Classification changes within the step. If it does, then the updated Enclosure Classification is used to re-calculate wind pressure. Iterations are continued until all Enclosure Classification values are consistent for each building. In general, two iterations suffice to reach a converged damage state. Also, since Enclosure Classification changes are associated with irreversible damage, iterations are generally applied early on in the analysis history, meaning that later steps do not require iterations because the damage state has already been determined. Within the 5,000 steps needed to complete a simulation, the increase in

computational time needed to address iterations is about 10% of the total simulation time. The increase is relatively small because iterations are only needed in the first few steps until the damage state settles down.

The results of incremental-iterative and incremental analysis are compared to demonstrate the effect of local iteration loops on the accuracy of the solution. The roof displacements of building B3 computed through the two analysis methods are plotted in Figure 2-10(a) and its enclosure classification over time is shown in Figure 2-10(b). It is clear that there are substantial differences between the incremental-iterative and incremental schemes in the early part of the analysis associated with the rapid changes in enclosure classification. However, the differences decrease quickly and eventually become negligible. As such, the results of incremental analysis, which are faster to conduct, as discussed above, are used in the following discussions.

The dynamic roof displacement responses of the buildings are shown in Figure 2-11 for the duration of the event. With respect to envelope damage, the windows of building B4 remain intact while those of building B1, B2 and B3 suffer substantial damage due to excessive wind pressures and story drifts. In particular, this envelope damage causes a change in Enclosure Classification for buildings B1, B2 and B3 from “Enclosed” to “Partially Open”.

Figure 2-12 shows the 3D rendering of window damage for building B1. The accumulation of window damage over the height of buildings B1, B2, and B3 at some representative time steps, are shown in Figure 2-13 to Figure 2-15 (damaged windows are marked as light-colored blocks while undamaged windows are shown as dark-colored blocks). The result of B4 is omitted because the window damage rarely occurs in such a low-rise building (4-stories) for the studied wind event. It can be observed that as the wind event progresses, the windows towards the top of the windward

walls sustain the majority of the damage due to the larger wind pressures associated with these areas.

To demonstrate the feasibility of the developed distributed platform, Figure 2-16 compares the results computed using the distributed and integrated schemes, i.e. all simulators are integrated together into a single software package. As can be seen, the two responses are identical. While the distributed scheme may seem more complex to conduct than an integrated one with no change in accuracy, reformulating the problem into the distributed format with a publish-subscribe data pattern opens up new possibilities to modify the simulation in a decentralized way as discussed next.

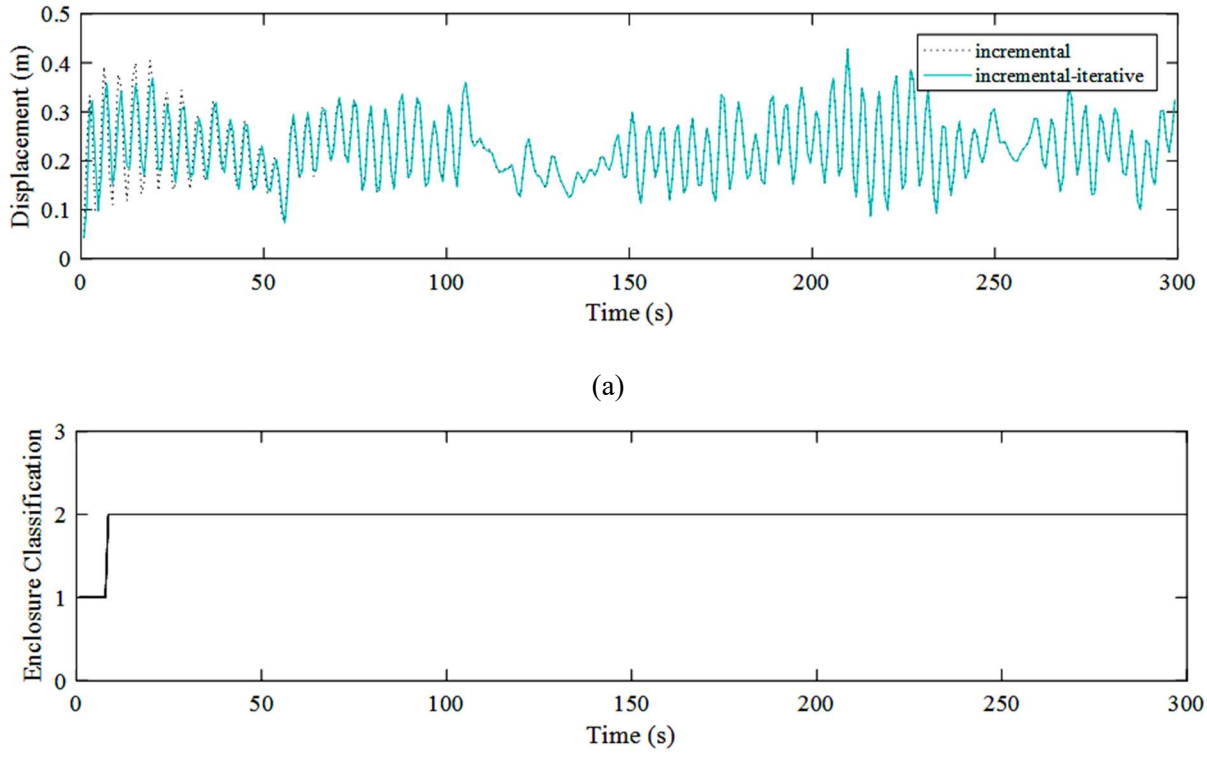


Figure 2-10. Comparison between incremental-iterative and incremental analysis results for building B3: (a) Roof displacement (b) Enclosure classification (1: Enclosed; 2: Partially Enclosed; 3: Partially Open)

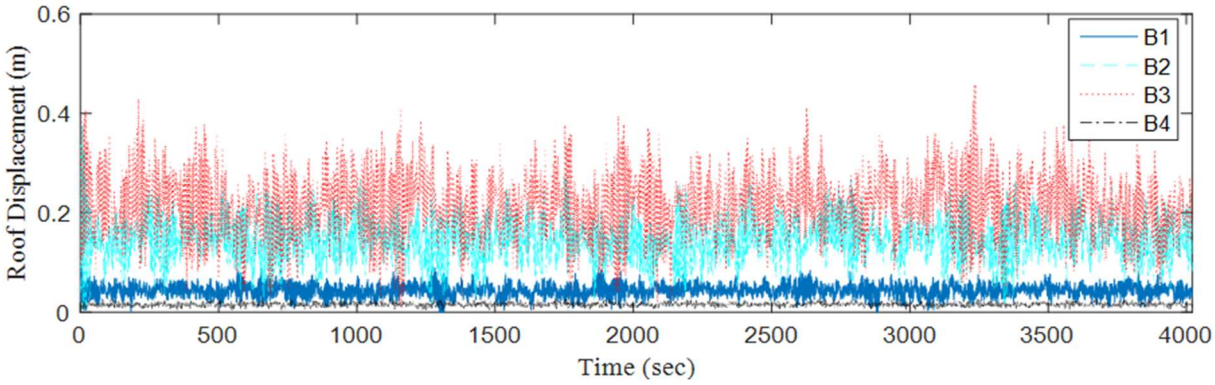


Figure 2-11. Roof displacement responses of four representative buildings

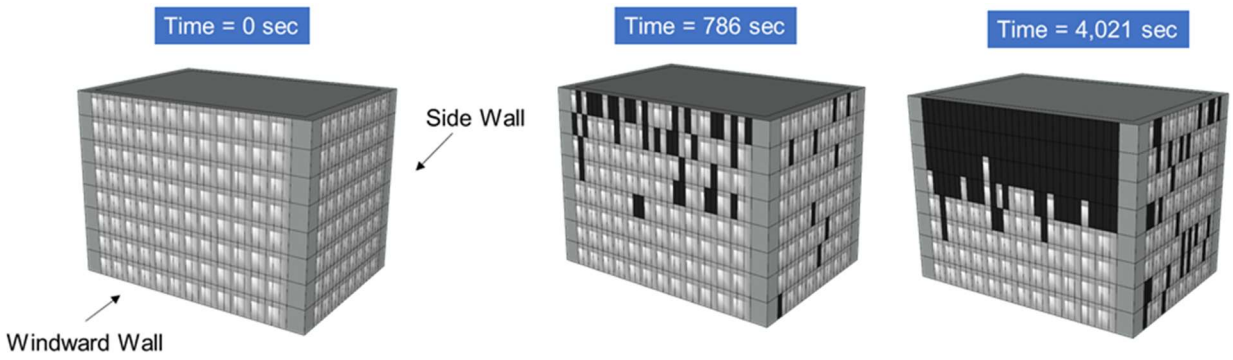


Figure 2-12. 3D rendering of window damage for building B1

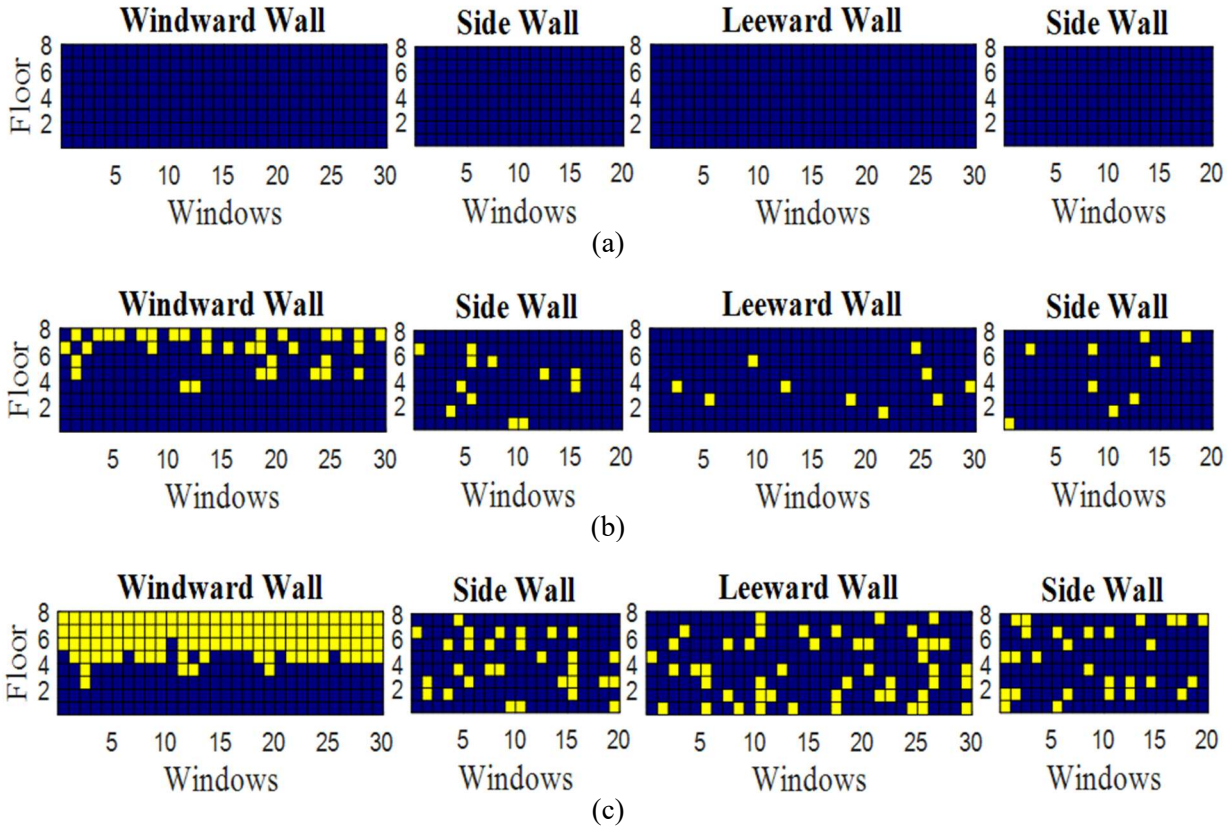


Figure 2-13. Progression of window damage for building B1 at: (a) T = 0 sec; (b) T = 786 sec; (c) T = 4,021 sec

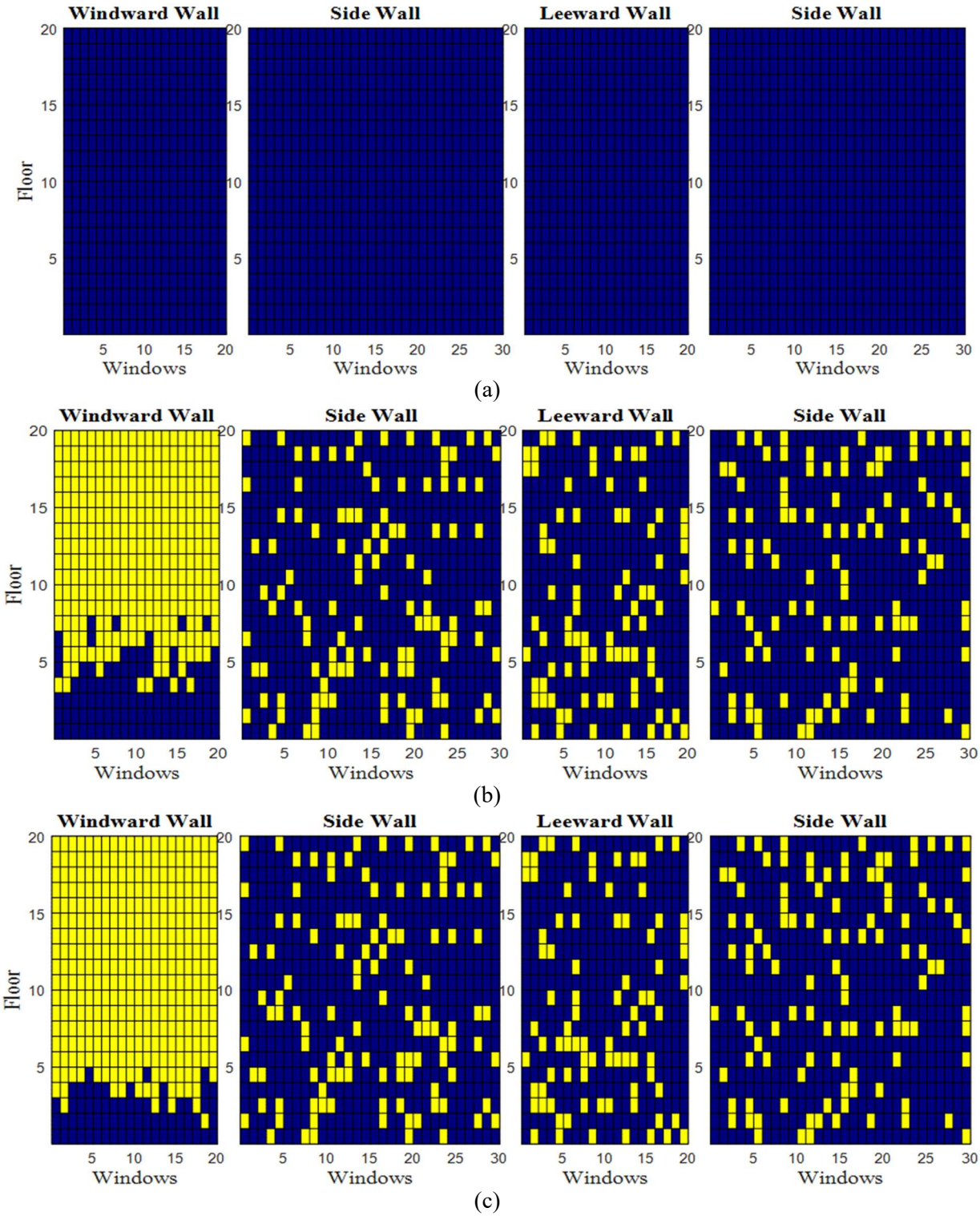


Figure 2-14. Progression of window damage for building B2 at: (a) T = 0 sec; (b) T = 786 sec; (c) T = 4,021 sec

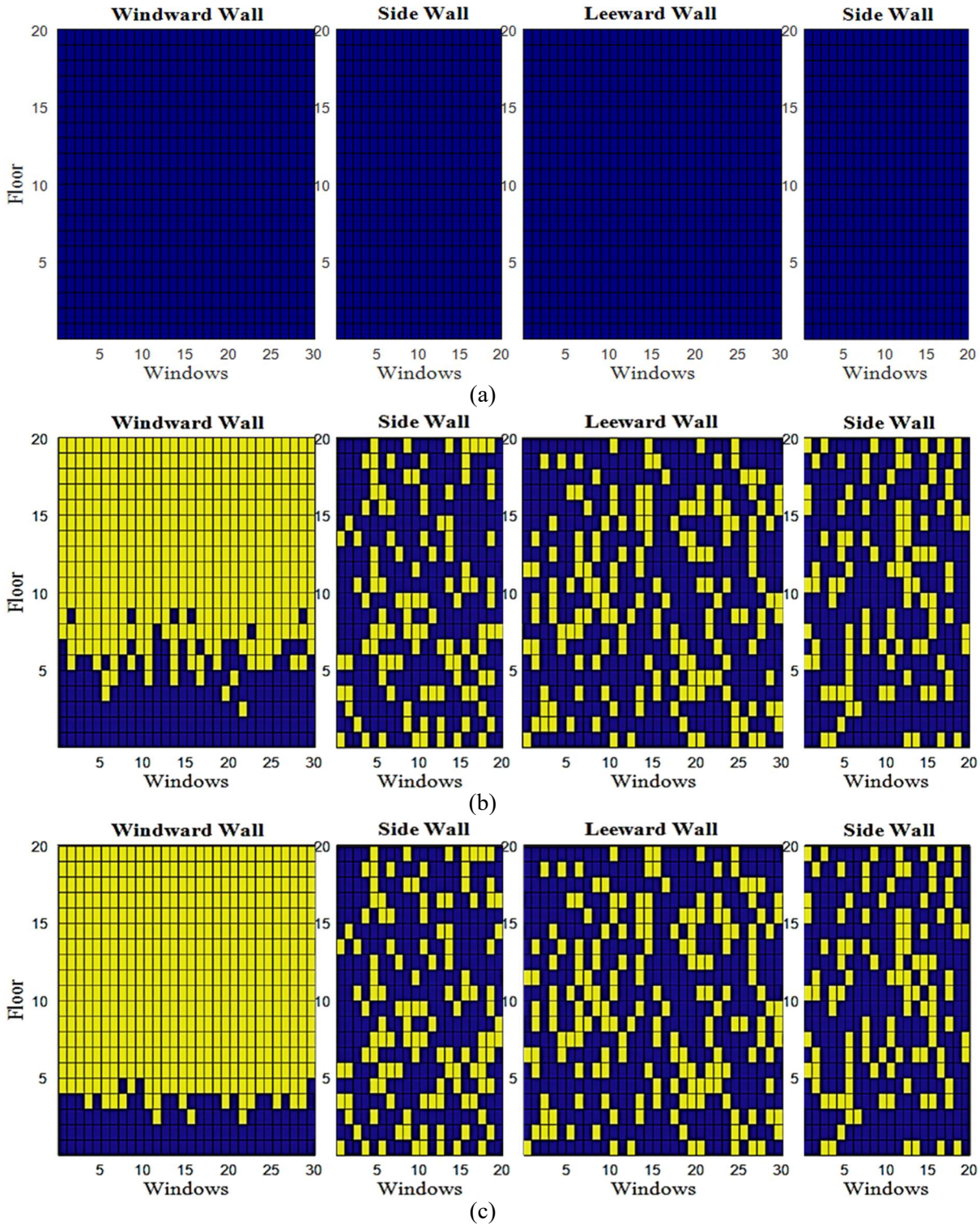


Figure 2-15. Progression of window damage for building B3 at: (a) T = 0 sec; (b) T = 786 sec; (c) T = 4,021 sec

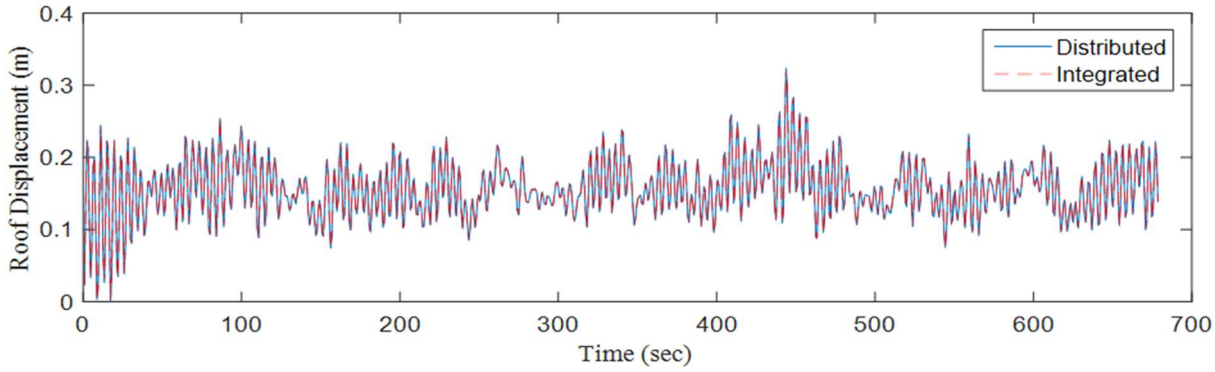


Figure 2-16. Comparison between the integrated and distributed schemes - roof displacement of B2

2.5 Advantages and Potential of the Proposed Framework for Modeling Community Resilience

An important characteristic of the distributed simulation platform is modifiability. Since each simulator is self-contained (a black box, as noted earlier), it can be replaced by another simulator of different fidelity or one that uses another modeling technique or rooted in different theory. For example, the *Structural Analysis Simulator* can be exchanged with one that employs a detailed continuum finite element model instead of the floor-wise lumped mass model discussed earlier. The restriction for successful replacement is that the subscribed data (in this case the applied wind loads) and published data (in this case the horizontal displacement at each floor) remain the same. Similarly, the *Damage Simulator* could be replaced with a more sophisticated *Wind Damage and Loss Simulator* that allows for a more detailed analysis of damage associated with wind pressure, debris impact, and water ingress instead of the simple glass breakage model employed.

The framework is also extensible. The decentralized nature of the platform permits new simulators to be coupled to the simulation, growing it and increasing its complexity. For example, it is conceivable that a *Casualty Simulator* could be connected to account for injuries and fatalities

that occur due to glass breakage (assessed by the *Damage Simulator*) or airborne missiles (represented by yet another simulator). A third characteristic is scalability. The simulation framework permits a set of core simulators to be grouped into larger modules and connected to other coupled simulation frameworks allowing the scope to be scaled, with computational resources as the only limiting factor. The fact that various simulators can run on multiple computers increases the computational efficiency accordingly.

These characteristics suggest that the simulation framework is inherently suitable for modeling problems that are richer in scope than the type of problem demonstrated in this chapter. Instead of modeling just the interactions between a hazard and an infrastructure system as done in the case study presented, the framework could be extended to incorporate the effects of other systems such as power, water and gas networks. It is also conceivable that the effects of recovery after the disaster could be taken into account. For example, a *Debris Simulator* could be coupled to the platform to compute the amount of debris resulting from the damage. A *Recovery Simulator* could then subscribe to this and other data to compute the new components and material required to reinstate the structure to its original condition and then publish recovery trajectories. The simulation platform is therefore well suited for modeling important aspects of community resilience.

2.6 Conclusions

In this chapter, a new distributed simulation platform was developed to simulate the interdependent effects that occur during natural hazard disasters. Each part of the disaster was represented by a simulator that modeled its behavior. Using a publish-subscribe pattern for data management, each simulator published information that it had generated for other connected

simulators to use and subscribed to the data it needed from other simulators. Open source LCM libraries were employed to manage the information traffic. The developed methodology was demonstrated through a case study of wind-induced progressive damage due to both the structural response of a building as well as external wind pressures. The natural hazard scenario was implemented on multiple machines connected to a LAN to demonstrate the distributed capabilities of the proposed method. It was shown that the distributed implementation gave results similar to the traditional integrated application and the advantages of formulating the problem in the developed format were highlighted. The need for iterations during simulation to ensure accuracy was outlined and it was demonstrated that iterations could, indeed, be performed within the platform. Key characteristics of the platform, such as modifiability, extensibility and scalability were discussed and it was suggested that the technique could be used for modeling richer problems, such as community resilience.

CHAPTER 3

Time-Dependent Resilience Assessment of Interdependent Lifeline Systems

3.1 General

This chapter describes a resilience analysis of three interdependent lifeline systems, i.e. water, gas, and electrical power systems, subjected to a series of seismic events. The simulation is comprised of a group of independent simulators, each of which represents one aspect of the disaster scenario (Section 3.4). The simulators are independent of one another, could originate from disparate disciplines, operate at different levels of fidelity and could have differing spatial or temporal scales. The analysis framework addresses the spatial and time-dependent interactions that arise between lifeline systems as the hazard and subsequent restoration processes unfold. Multiscale interdependency within the described hazard event and time-dependent effects of resource allocation strategies on community resilience are discussed. The seismic damage and recovery assessment of lifeline systems in Shelby County, Tennessee are presented as the case study (Section 3.5). The simulation results quantify how operability loss and recovery time may be underestimated if the interdependencies between lifeline systems are not properly taken into account. The effect of insufficient resources on recovery is investigated, and it is demonstrated that among the six resource allocation strategies studied, the time-varying strategies that are responsive to actual conditions on the ground had a better effect on resilience (Section 3.6). This chapter shows the importance of considering multiscale interdependency and time-dependent

effects on community resilience. The ability of the framework to address the linkages that exist between civil and social systems is also presented.

3.2 Motivation and Objectives

Modeling a disaster and the subsequent recovery effort is complicated by the differing time scales of the various phases of the process, that is, seconds or minutes as the hazard unfolds versus days or months as the emergency effort and recovery take place. As a result, studies that model the multiple phases of a disaster within one overarching simulation are rare due to the challenge of integrating different simulation models with disparate temporal and spatial scales.

A common assumption in resilience studies is to assume that a hazard occurs during one analysis step, that is, virtually instantaneously. In reality, the hazard unfolds in a finite amount of time. Accounting for how a hazard unfolds and affects infrastructure systems that can interact with one other can yield new insights into how interdependencies affect community resilience. This is especially important for situations such as a long-period disaster that overlaps with the short-term recovery effort [e.g., emergency response to hurricanes (Schmeltz et al., 2013)], short-period disasters interacting with an ongoing recovery effort (e.g., an aftershock affecting the recovery effort associated with the main shock), or multiple disasters occurring in a specific locale [e.g., an earthquake followed by a tsunami (Moreno and Shaw, 2019)] .

Given the paucity of studies in this area, the objective of this research is to conduct an analysis that explicitly addresses the spatial and temporal progression of earthquake-induced damage and the postdisaster restoration effort. After a review of the literature, the methodology and framework are introduced and a case study of three interdependent lifeline systems subjected

to two successive earthquakes is presented. Last, the applicability and limitations of the framework are discussed.

3.3 Background

There is broad consensus that the interdependencies that exist between the lifeline systems of a society can significantly impact the resilience of communities facing natural and man-made hazards. The classification of interdependencies and the various modeling techniques used to address them have been discussed in detail in Chapter 2 (Section 2.2.1). Numerous studies have been conducted to evaluate the resilience of communities subjected to hazards. The PEOPLES resilience framework (Renschler et al., 2010; Cimellaro et al., 2016) includes seven dimensions for assessing community resilience: population and demographics, environmental and ecosystem, organized governmental services, physical infrastructures, lifestyle and community competence, economic development, and social-cultural capital. Miles and Chang (2011) introduced a simulation model named ResilUS that was built on their previous efforts (Chang and Miles, 2004; Miles and Chang, 2007) and provided an implementation of the 1994 Northridge earthquake. The NIST-funded Center for Risk-Based Community Resilience Planning has developed the Interdependent Networked Community Resilience Modeling Environment (IN-CORE), which some studies have demonstrated on a virtual test bed community called Centerville (Cutler et al., 2016; Ellingwood et al., 2016; Guidotti et al., 2016; Lin and Wang, 2016). The Civil Restoration with Interdependent Social Infrastructure Systems (CRISIS) model (Loggins et al., 2019) mapped services provided by civil infrastructure to the performance of social infrastructure systems and aimed to find restoration schemes that optimize the performance of social systems. As Koliou et al. (2018) concluded, there are only a handful of frameworks that can account for the

multidisciplinary and multiscale nature of community resilience in time-varying resilience analyses. The methodology employed in this research is geared toward addressing these gaps in the literature.

3.4 Computational Framework

Chapter 2 provides a detailed description of the modeling environment and publish–subscribe data transmission pattern used in this Chapter. Figure 2-2 shows how the various simulators employed herein interact together, and Figure 3-2 illustrates the publish–subscribe relationship between the simulators.

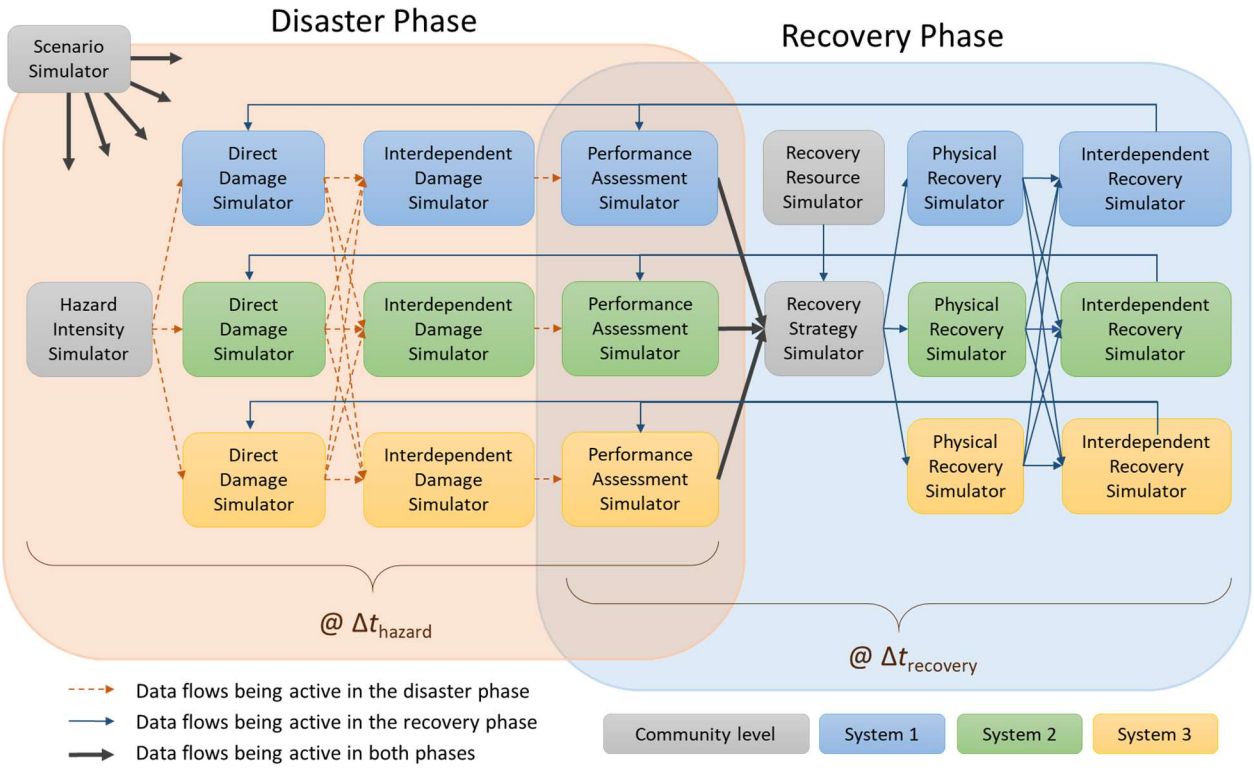


Figure 3-1. Simulation framework and message flow

The messages published during a disaster event are described in Table 3-1 and the corresponding publishers and subscribers are listed in Table 3-2. The run-time interface shown in Figure 3-2 manages the flow of messages, permitting the analysis to proceed in a decentralized and scalable manner. Although Figure 2-2 and Figure 3-2 show the framework for the case study considered herein, which contains three interdependent systems, it can be extended in a straightforward manner to handle other situations with more interacting systems and simulators.

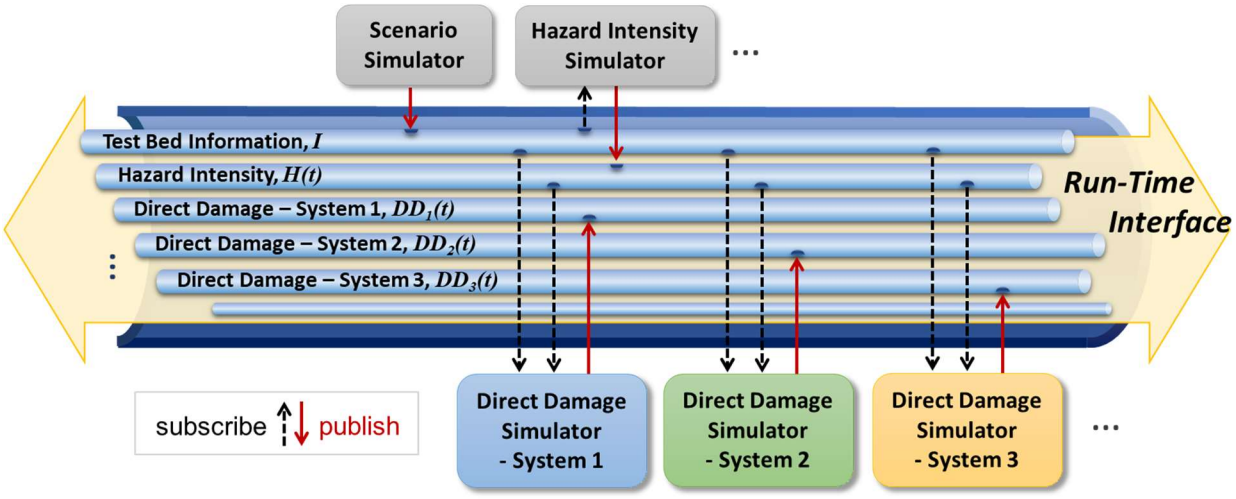


Figure 3-2. Publish–subscribe concept for data exchange

Table 3-1. Message types published during a disaster event

Code	Message description
I	Configuration and information of test bed that does not change with time
$H(t)$	Hazard intensity measures at all locations of interest at time t
$DD_i(t)$	Damage status of components directly induced by the hazard at time t
$ID_i(t)$	Damage status of components considering the interdependency effects at time t
$P_i(t)$	System performance measures at time t
$RS(t)$	Total available recovery resources and constraints at time t
$S(t)$	Allocation strategy of recovery resources at time t
$R_i(t)$	Physical recovery status of components at time t
$IR_i(t)$	Recovery status of components considering the interdependency effects at time t

Note: Subscript i indicates the messages produced by system i.

Table 3-2. Messages published or subscribed to by the simulators

Simulator	Message published	Message subscribed
Scenario Simulator	I	-
Hazard Intensity Simulator	$H(t)$	I
Direct Damage Simulator $_i$	$DD_i(t)$	I, $H(t)$, $IR_j(t)$, $_{j = 1, 2, \ldots}$
Interdependent Damage Simulator $_i$	$ID_i(t)$	I, $DD_j(t)$, $_{j = 1, 2, \ldots}$
Performance Assessment Simulator $_i$	$P_i(t)$	I, $ID_j(t)$, $IR_j(t)$, $_{j = 1, 2, \ldots}$
Recovery Resource Simulator	$RS(t)$	I
Recovery Strategy Simulator	$S(t)$	I, $P_j(t)$, $_{j = 1, 2, \ldots}$
Physical Recovery Simulator $_i$	$R_i(t)$	I, $S(t)$
Interdependent Recovery Simulator $_i$	$IR_i(t)$	I, $R_j(t)$, $_{j = 1, 2, \ldots}$

Note: Subscript *i* refers to system *i*.

The scenario simulator in Figure 2-2 describes the basic configuration information, specifically the location and characteristics of utility facilities and the connectivity between them. Such information is published at the beginning of simulations and assumed not to change with time. Once a disaster occurs (in the disaster phase), the hazard intensity simulator provides information about the hazard, such as the magnitude and epicenter of an earthquake in a seismic disaster. Although this chapter only focuses on seismic events, the hazard intensity simulator could also provide information about storm track and intensity if a hurricane hazard were of concern. The simulator provides specific hazard information at all locations of interest to other simulators—for example, ground motions at a given location.

The direct damage simulator calculates the physical damage of components directly induced by a hazard regardless of the influence of other infrastructure systems. Damage can be evaluated using empirical models, fragility curves, or detailed finite-element models. The fact that the simulation framework does not care about the specific method by which damage is assessed is a key strength of the methodology. The interdependent damage simulator addresses the effects of interdependencies on damage occurrence. Interdependencies come in many varieties. They can be functional, spatial, or both (Zimmerman, 2001); cyber, geographic, and logical (Rinaldi et al.,

2001); physical, geospatial, policy, and informational (Dudenhoeffer et al., 2006); functional, physical, budgetary, market, and economic (Zhang and Peeta, 2011). The performance assessment simulator assesses system performance and is a key determinant for formulating a recovery strategy.

In the recovery phase, the recovery resource simulator estimates the amount of resources, such as labor, equipment, materials, and budget, which can be used for lifeline restoration. The recovery strategy simulator allocates limited recovery resources to the systems based on a given recovery strategy, which may depend on the damage status and performance of the systems. The influencing factors and strategy for the allocation of recovery resources may change during the recovery process. Such time-dependent effects are a key focus of this research; the study of such effects is enabled by the distributed simulation methodology adopted in this work. Once damage occurs, the physical recovery simulator determines the reconstruction priority of damaged components based on their damage situation and degree of importance in the system, and estimates the required time for restoration. During every recovery period, the simulator further distributes recovery resources allocated from the recovery strategy simulator to each damaged component in order of priority, that is, system level to component level. Then, within every recovery step, the physical recovery simulator decides whether reconstruction progress advances forward or pauses according to whether a component has enough allocated resources.

The interdependencies between the various systems must be considered not only as a hazard unfolds but also during the recovery process. For example, one component in a network system may have completely recovered from damage inflicted by a hazard but still cannot function properly due to its dependency on another still-damaged system. Therefore, like the interdependent

damage simulator, the interdependent recovery simulator considers interdependent behaviors across systems and updates the recovery status and functionality of components.

The simulators used in this work span different spatial scales: whole community, infrastructure system, and structural component. Community-level simulators affect large geographic areas (e.g., the scenario simulator and hazard intensity simulator) or represent decisions that address a large part of a community (e.g., the recovery strategy simulator). System-level simulators address physical infrastructure systems such as lifeline networks. The lowest spatial level pertains to components of the various infrastructure systems, such as residential buildings or pumping stations. The times scales considered herein also vary widely. As illustrated in Figure 2-2, the time scale as the disaster phase unfolds Δt_{hazard} is several orders of magnitude smaller than the time step during the recovery phase $\Delta t_{\text{recovery}}$. The framework employed in this work allows for the possibility of subsequent hazards to occur—for example, an aftershock that occurs during an ongoing recovery progress.

Shifts between the disaster and recovery phases are controlled by the performance assessment simulator, which is involved in both phases. This simulator judges the beginning and end of a disaster by interpreting the received damage messages and provides the latest system performance to the recovery strategy simulator. As shown in Figure 2-2, in the disaster phase, the performance assessment simulator calculates system performance based on damage status provided by the interdependent damage simulator, and in the recovery phase, it continues to update system performance according to the recovery status from the interdependent recovery simulator. The direct damage simulator also subscribes to the recovery status provided by the interdependent recovery simulator, although it does not publish anything during the recovery phase. This is

because it needs to know the latest recovery status in order to assess the capacity reduction in components that are not yet fully repaired when the next disaster occurs.

The computational framework handles several types of interdependencies. Most importantly, the interdependencies between system performances and community-level recovery strategy are accounted for in a dynamic sense. In other words, recovery strategy can evolve depending on system performance at a given time. Second, because additional disruptions can occur during an ongoing recovery process, the ability of each component to resist new demands caused by subsequent hazards may be affected by damage from a previous event and unfinished rehabilitation efforts. Last, interdependencies can occur between components of different lifeline systems and must be accounted for. These interdependent relationships are shown in Figure 2-2 by the interlaced lines joining the direct damage simulator and the interdependent recovery simulator or joining the physical recovery simulator and the interdependent recovery simulator.

The extensibility and flexibility of the computational framework for modeling various types of interdependencies between disparate systems are the key strengths of the platform. For example, if new interacting systems are added, the interdependent damage/recovery simulators merely need to subscribe to the new direct damage simulators or physical recovery simulators on which they depend. No changes need to be made to other simulators in the system. The publish–subscribe approach used in this work eliminates the need for using interdependency matrices, which are commonly used to specify the relationships between different pairs of networks. The limitations associated with using interdependency matrices are discussed in Chapter 2.

3.5 Case Study: Seismic Damage and Recovery of Lifeline Systems in Shelby County, Tennessee

Shelby County, Tennessee, which is close to the southwest end of the New Madrid seismic zone (NMSZ) has been used as a test bed in many studies. Dueñas-Osorio et al. (2007), Adachi and Ellingwood (2008), and Hernandez-Fajardo and Dueñas-Osorio (2013) studied the interdependent response of water and power systems in Shelby County under earthquake demands. Adachi and Ellingwood (2009) assessed the performance of its water system under spatially correlated seismic intensities. Song and Ok (2010) analyzed multiscale effects on system reliability of the gas transmission network in Shelby County. González et al. (2016) developed restoration strategies that took into account the interdependencies between the water, power, and gas network systems in Shelby County.

In a departure from previous studies, the computational framework developed in this Chapter was applied to Shelby County, Tennessee in order to demonstrate how it can be used to investigate earthquake-induced damage and the subsequent recovery progress, which itself is interrupted by an aftershock (first shock—short term recovery effort—second shock—long term recovery effort). The framework was applied to three interdependent lifeline systems in order to demonstrate its scalability. Unlike the aforementioned studies, which merely focused on one of the phases in a hazard event, that is, the disaster process or the recovery period, this study presents an overall simulation that addresses the disaster and postdisaster phases in an integrated manner. Another key advantage of the framework is that it naturally combines simulators that have disparate temporal and spatial scales.

To capture the uncertainty in the seismic damage and the restoration process, Monte Carlo simulations were performed, and the means of the results are presented. Studies with 300, 500, and 1,000 simulations were conducted to select a reasonable number of simulations. The studies showed that the average relative differences of the first to the last were 4.75% (300 runs versus

1,000 runs) and 0.21% (500 runs versus 1,000 runs). Therefore, the number of Monte Carlo runs was set to 500.

The following section describes the details of the simulators shown in Figure 2-2 and discussed previously.

Scenario Simulator

The scenario simulator provides configuration information about the lifeline systems considered herein. The systems of interest include the electric power system (EPS), water distribution system (WDS), and natural gas system (NGS), which are operated by the Memphis Light, Gas, and Water (MLGW) division. The topological configuration of the networks was adapted from Chang et al. (1996), Dueñas-Osorio et al. (2007), and Song and Ok (2010). Figure 3-3 shows the topologies and critical components of the power, water, and gas network systems in Shelby County. The gate stations in EPS and NGS and the elevated tanks and pumping stations in WDS are supply nodes. The 23 kV/12 kV substations in EPS, the intersection nodes in WDS, and the regulator stations in NGS are demand nodes. The intersection nodes in EPS and NGS and all directed arcs represent the transmission components.

Hazard Intensity Simulator

The scenario earthquakes were assumed to have an epicenter at 35°18'N and 90°18'W; the same assumption was made in Adachi and Ellingwood (2009). Ground motions designated RSN-5223 (designated EQ1) and RSN-6536 (designated EQ2) from PEER (2018) were used in this study to represent feasible seismic activity. The ground motion records, which have a 0.01-s time interval, were scaled to peak ground acceleration (PGA) at the center of Memphis (35°08'N and 89°59'W; i.e., 33 km from the epicenter). The PGA values were 0.202 and 0.341 g for EQ1 and EQ2, respectively. These values were chosen based on the United States Geological Survey

(USGS, 2018) for earthquakes with a 10% probability of exceedance in 50 years (10/50) and a 5% probability of exceedance in 50 years (5/50).

Ground motion attenuation was assumed to follow the model proposed by Atkinson and Boore (1995). Although the attenuation relationships were proposed only for the PGA, the model was assumed to be applicable to the entire acceleration record as plotted in Figure 3-4 and to depend only on the distance to the epicenter. Although the assumptions related to the hazard were made for convenience, the hazard intensity simulator can be adjusted in the future once more data or new models become available.

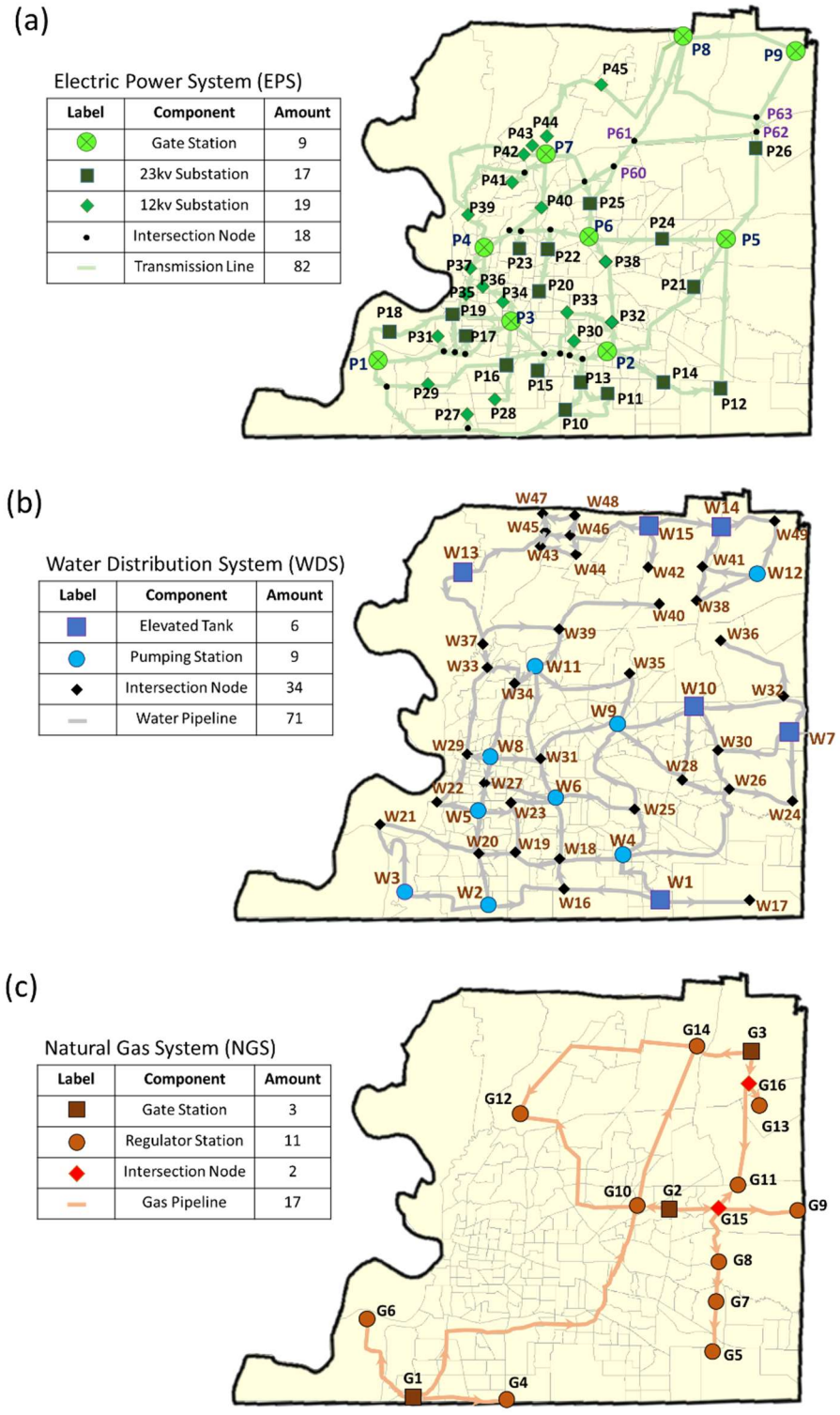


Figure 3-3. Topological configuration of the lifeline systems in Shelby County, Tennessee: (a) electric power; (b) water distribution; and (c) natural gas system. [Adapted (a–c) from Chang et al. (1996); data for (a–b) from Dueñas-Osorio et al. (2007); data for (c) from Song and Ok (2010)]

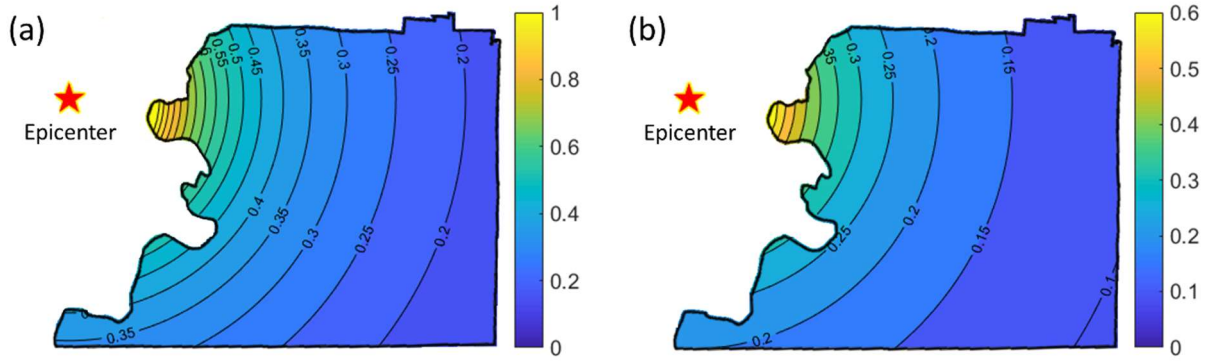


Figure 3-4. Assumed attenuation of ground acceleration (unit: *g*) for the earthquake with: (a) 5% and (b) 10% probability of exceedance in 50 years

Direct Damage Simulator

Direct damage occurs if the hazard intensity, as computed by the hazard intensity simulator, exceeds the capacity of a component. Four damage states are considered: minor, moderate, extensive, and complete. These states are irreversible and occur in sequential order. The capacity of each component is determined at the beginning of each realization in the Monte Carlo simulation. Lognormal fragility functions were used to estimate the capacities associated with different damage states for different types of utility facilities. The fragility functions were adopted from the Hazards US Multi- Hazard (HAZUS-MH) technical manual (FEMA, 2003), and their parameters are listed in Table 3-3. It was assumed that damage to EPS can be assessed from the gate stations and substations, which are the most critical equipment for the functionality of a power system (Shinozuka et al., 2005). The intersection nodes in all networks added for dividing the transmission lines and pipelines were assumed to be not vulnerable to earthquakes (Figure 3-3).

As discussed in FEMA (2003), the rate of occurrence of pipeline failures per unit length is known as the repair rate and is computed via Equation (3-1), in which the unit for peak ground velocity (PGV) is cm/s. The probability that the number of pipe breaks N_B equals b within a

pipeline segment of length L can be expressed as shown in Equation (3-2), and the probability of pipeline breakage is shown in Equation (3-3)

$$R_{\text{rate}}[\text{repairs/km}] \cong 0.0001 \times \text{PGV}^{2.25} \tag{3-1}$$

$$P(N_B = b) = \frac{(R_{\text{rate}} \times L)^b}{b!} \text{e}^{-R_{\text{rate}} \times L} \tag{3-2}$$

$$P(N_B > 0) = 1 - P(N_{\text{B}} = 0) = 1 - \text{e}^{-R_{\text{rate}} \times L} \tag{3-3}$$

Each link in WDS and NGS is divided into several segments of approximately one km length in order to consider the scale effect (Song and Ok, 2010). The PGV that corresponds with a 50% probability of pipeline breakage was used as the capacity of each segment. Buried pipelines may also be damaged by ground failure, for example, by liquefaction. Although such situations were not considered in this case study, they could be considered in the future by adding other specialized simulators.

Table 3-3. Parameters of lognormal fragility functions for utility facilities

System	Components	Minor	Moderate	Extensive	Complete
EPS	Gate Station	0.11 (0.50)	0.15 (0.45)	0.20 (0.35)	0.47 (0.40)
	12kv/23kv Substation	0.15 (0.70)	0.29 (0.55)	0.45 (0.45)	0.90 (0.45)
WDS	Elevated Tanks	0.18 (0.50)	0.55 (0.50)	1.15 (0.60)	1.50 (0.60)
	Pumping Station	0.15 (0.75)	0.36 (0.65)	0.77 (0.65)	1.50 (0.80)
NGS	Gate Station	0.15 (0.75)	0.34 (0.65)	0.77 (0.65)	1.50 (0.80)
	Regulator Station	0.15 (0.75)	0.34 (0.65)	0.77 (0.65)	1.50 (0.80)

Note: All fragility functions for utility facilities are lognormal distributions with peak ground acceleration (PGA) as the engineering demand parameter. The corresponding median and lognormal standard deviation (β) are listed in the table, i.e., median (β); (unit: g).

Component capacities are determined at the beginning of each realization based on the history of seismic activity. In the case of a first seismic event, they are considered to be damage

free. When aftershocks occur during the recovery process, the integrity of the segments has already been compromised by the previous event, and component capacities are assumed to be a function of the previous damage state. In this case, the capacities of discrete components were assumed to be reduced by 40%, 20%, and 10% for extensive, moderate, and minor damage states, respectively, and the reduction ratios for broken pipelines was set to 40%, that is, an extensive damage state. These numbers can be refined in the future if the time between events is specified and the direct damage simulator and interdependent recovery simulator are able to address sequential damage effects.

Interdependent Damage Simulator

Two types of interdependencies are considered at the component-level: functional interdependencies and spatial interdependencies. A functional interdependency indicates the dependence of one system (slave nodes) on the functionality or material flow of another (master nodes). For example, pumping stations in water and gas systems rely on electric power to operate pumping machines; electric power plants rely on the water distribution system for cooling purposes and for controlling emissions of coal-based power generators. In this case study, part of the power grid depended on the natural gas system to fuel generation units. Spatial interdependency is a situation in which components from different infrastructure systems are collocated within the same geographical environment, that is, the components have spatial overlap. There is generally mutual reliance rather than master-slave relationship of functional interdependency; that is, the damage state of both nodes is the same and is governed by the node that has more severe direct damage.

The conditional probability of a slave node being nonfunctional given an inoperative master node can be seen as the degree of interdependency or the coupling strength between the two nodes. Herein, the conditional failure probability of any pair of slave and master nodes is set

to one, but it can be adjusted for other situations. All these interdependent relationships and the nodes involved are listed in Table 3-4. The interdependent damage simulator of each system only needs to know which nodes are the master nodes of its own components and subscribe to their damage conditions.

Table 3-4. Interdependent relationships between EPS, WDS, and NGS

WDS* - EPS	EPS* - WDS	NGS* - EPS	EPS* - NGS	NGS - WDS (Mutual)
W2* - P28	P1* - W21	G6* - P18	P1* - G6	G3 - W12
W3* - P29	P2* - W25	G10* - P24	P5* - G11	G14 - W41
W4* - P14	P3* - W23	G12* - P44	P7* - G12	—
W5* - P17	P4* - W29	G13* - P26	—	—
W6* - P33	P5* - W30	—	—	—
W8* - P36	P6* - W35	—	—	—
W9* - P38	P7* - W39	—	—	—
W11* - P40	P8* - W42	—	—	—
W12* - P26	P9* - W49	—	—	—

Note: The left four columns indicate Functional Interdependency (Slave Node* - Master Node), and the rightmost column indicates Spatial Interdependency.

Performance Assessment Simulator

Ghosn et al. (2016) suggested that the performance measures of a network system can be divided into two categories: flow-based and topology-based measures. Flow-based performance measures are represented by the amount of supplied flow and the proportion of satisfied customer demand. Topology-based measures are calculated based on graph theory. An abstract graph representing a lifeline network consists of supply nodes, demand nodes, and several directed links that indicate the connecting paths from supply nodes to demand nodes.

Due to a lack of information pertaining to flow capacity and demand, a topology-based metric, termed connectivity loss (*CL*), was selected for performance assessment in this case study. *CL* measures the average change in the connectivity of demand nodes to supply nodes after

perturbation and is often used to assess the capability of a network system to withstand disruption (Albert et al., 2004; Dueñas-Osorio et al., 2007). At the beginning of a simulation, each lifeline system is represented as a graph with nodes and links, and the original connectivity is calculated. As the analysis progresses, inoperative components are removed from the graph, then added back when they recover. *CL* of a network system with N_{demand} demand nodes can be computed by Equation (3-4)

$$CL = 1 - \frac{1}{N_{\text{demand}}} \sum_{i}^{N_{\text{demand}}} \left(\frac{P_i}{P_{0,i}}\right) \tag{3-4}$$

where $P_{0,i}$ = original number of supply nodes that connect to the *i*th demand node; and P_i = number of supply nodes connected to the *i*th demand node after a perturbation. The remaining connectivity (*C*) of a network is: $C = 1 - CL$.

Recovery Resource Simulator.

Recovery resources are quantified as a number of resource units. A resource unit is defined as the amount of resources and budget required for an 8-person crew with accompanying repair equipment to work 12 hours (working time per day). In all of the case studies discussed subsequently, the available number of resource units R_{total} for the entire county was assumed to be a fixed value during the recovery process. In general, it is assumed that all crews have unlimited expertise, that is, they can work on all lifeline systems. However, in last case study, the crews were assumed to have different skills, and the maximum number of available crews specializing in the ith lifeline system was denoted as $R_{\text{max},i}$. Clearly, the functionality of the social infrastructure—for example, the availability of able-bodied workers who were not injured or killed in the event—affects $R_{\text{max},i}$. Although not accounted for here due to space and scope limitations, in the future, such a limitation can be accounted for through the addition of a social infrastructure simulator that,

for example, accounts for worker injuries and deaths and for available funding needed to pay for repair crews.

Recovery Strategy Simulator

The recovery strategy simulator interprets the allocation strategy for recovery resources. A feasible recovery strategy is to allocate recovery resources to each system evenly regardless of their damage conditions, as stated in Equation (3-5), where N_s is the number of systems and R_k is the amount of recovery resources allocated to the kth system. This strategy (the EA strategy) could represent a situation in which information about the extent of a disaster is not known. In cases in which R_{total} and N_s are fixed values during the recovery process, the EA strategy is time-independent

$$R_k(t) = \frac{R_{\text{total}}(t)}{N_s} \tag{3-5}$$

Another strategy (the LA strategy) is to assign resources depending on the performance of the systems in terms of connectivity loss. In this case, R_k is computed as

$$R_k(t) = \frac{CL_k(t)}{\sum_i^{N_s} CL_i(t)} \times R_{\text{total}}(t) \tag{3-6}$$

where CL_i = connectivity loss of the ith system. Alternatively, if the number of damaged components in each system ND_i is of concern, then a feasible strategy (the DA strategy) could be as follows:

$$R_k(t) = \frac{ND_k(t)}{\sum_i^{N_s} ND_i(t)} \times R_{\text{total}}(t) \tag{3-7}$$

The LA and DA strategies imply that the amount of recovery resources allocated to each system is not constant and changes over time t during the progress of recovery, reflecting the time-varying characteristic of the recovery process.

The recovery strategies applied in the example assume that systems that are more severely damaged and have worse system performance will receive more recovery resources. However, decision making during an actual disaster may be much more involved and may need to account for other factors, such as economics, politics, and societal values. In such cases, users could refine the algorithm in the recovery strategy simulator without influencing other simulators.

Physical Recovery Simulator

After a system is allocated recovery resources, the physical recovery simulator further distributes them to the damaged components. Once new damage to a system is computed, a normal distributed random variable is generated to estimate the required restoration time for each damaged component based on the restoration functions in the HAZUS-MH technical manual (FEMA, 2003). Uncertainty in the recovery process is considered. The parameters of the restoration function used in the case study are summarized in Table 3-5. The time step of the recovery process is taken as one day, and the required resources for all types of components are assumed to be one unit per day. If a damaged component has been allocated enough resources in the recovery step (day), then its repair progress will advance forward one day. Otherwise, it remains unrepaired.

The physical recovery simulator distributes allocated recovery resources to damaged components using two different strategies: randomly (the R strategy) or in order of their priority (the P strategy). In the latter case, the recovery priority of the components in each network is as follows: supply nodes, demand nodes, and links/pipelines. To simplify the simulation, resources and work crew are assumed available as soon as they are allocated, that is, the effect of

transportation on work crew routing (Morshedlou et al., 2018) is not considered in this study, although it could be incorporated through the addition of other simulators.

Interdependent Recovery Simulator

The same types of interdependencies, that is, functional interdependencies and spatial interdependencies, are considered during the recovery process by the interdependent recovery simulator. The interdependent relationships (master/slave) and involved nodes are listed in Table 3-4. Although slave components may have completely recovered from damage inflicted by a hazard, they may not function until the master components they depend on have fully recovered. For example, the functionality of pumping stations in the water and gas systems depends both on their own repairs and on the availability of electric power. After simulation of the physical recovery, the recovery status and functionality of components is updated depending on the different interdependent behaviors across the systems.

Table 3-5. Parameters of restoration functions for different components

System	Components	Minor	Moderate	Extensive	Complete
EPS	Gate Station	1.0 (0.5)	3.0 (1.5)	7.0 (3.5)	30.0 (15.0)
	12kv/23kv Substation	1.0 (0.5)	3.0 (1.5)	7.0 (3.5)	30.0 (15.0)
WDS	Elevated Tanks	1.2 (0.4)	3.1 (2.7)	93.0 (85.0)	155.0 (120.0)
	Pumping Station	0.9 (0.3)	3.1 (2.7)	13.5 (10.0)	35.0 (18.0)
NGS	Gate Station	0.9 (0.3)	3.1 (2.7)	13.5 (10.0)	35.0 (18.0)
	Regulator Station	0.9 (0.3)	3.1 (2.7)	13.5 (10.0)	35.0 (18.0)

Note: All restoration functions are normal distributions. The corresponding mean and standard deviation are listed in the table, i.e., median (standard deviation); (unit: day).

3.6 Results and Discussion

The simulators described in the previous section were connected together using the computational platform and publish–subscribe data transmission pattern described in Chapter 2. The computational framework was then used to investigate the effects of system interdependencies, multiple shocks, recovery strategies, and allocated recovery resources on the propagation of damage during seismic events and short- and long-term recovery processes.

3.6.1 Interdependencies between lifeline systems

First, a comparison of the performance of the three lifeline systems with and without considering the interdependencies between the systems is presented. Consider a seismic event with EQ2, R_{total} = 45 units/day, and crews with no limitation on their expertise. The allocation of recovery resources is based on the LA and P strategies. Figure 3-5 shows the damage and recovery curves of the three lifeline systems in terms of the average system connectivity performance. The dotted lines in Figure 3-5 reflect analyses that account for interdependencies, while the solid lines reflect simulations that do not account for interdependencies.

Figure 3-5(a), (c), and (e) indicate that EPS is the system most significantly affected by the earthquake out of the three lifeline systems when interdependencies are not considered. Figure 3-5(a) and (b) indicate that the performance of EPS is not significantly affected by interdependencies. Interdependencies are much more influential for WDS and NGS, as shown in Figure 3-5(c) to (e). The computational results show that WDS and NGS are more dependent on EPS, and the overall recovery time, in this case, is controlled by the restoration of EPS. It is clear that the operability loss and recovery times may be significantly underestimated if the interdependencies between lifeline systems are not adequately accounted for.

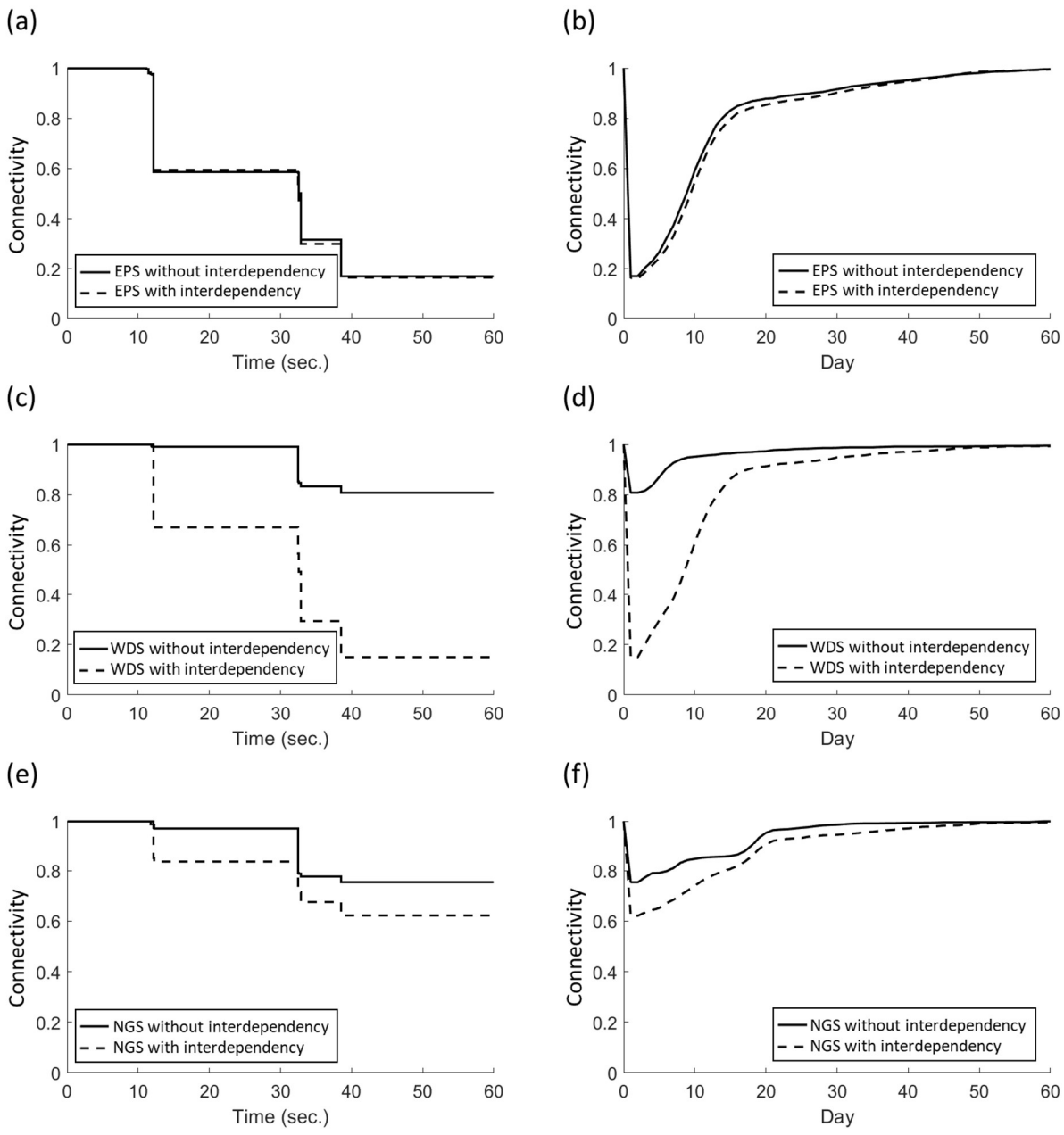


Figure 3-5. Comparison of system performance with and without considering interdependencies during the earthquake and recovery processes: (a) damage curves of EPS; (b) recovery curves of EPS; (c) damage curves of WDS; (d) recovery curves of WDS; (e) damage curves of NGS; and (f) recovery curves of NGS

3.6.2 Influence of foreshock

The influence of the foreshock is evaluated by considering a sequence of seismic events comprised of EQ1 followed by EQ2, R_{total} = 45 units/day, and crews with no limitation on their expertise. The allocation of the recovery resources is based on the LA and P strategies. The performance of the lifeline systems is indicated by the connectivity ratio, as plotted in Figure 3-6. Figure 3-6(a) and (b) pertain only to the main shock, while Figure 3-6(c) and (d) illustrate the outcomes of EQ1 (the foreshock) leading up to EQ2 (the main shock). Figure 3-6(a) and (c) indicate that the worst connectivity ratio of the lifeline systems is governed by the main shock, which is larger than the foreshock. However, the damage inflicted by the foreshock makes the lifeline systems more vulnerable to the later quake. As shown in Figure 3-7, the recovery slows down slightly in the long-term when the foreshock is considered, because the final damage after the main shock is more severe and there are more damaged components in need of repair, which might not be fully reflected in the connectivity loss.

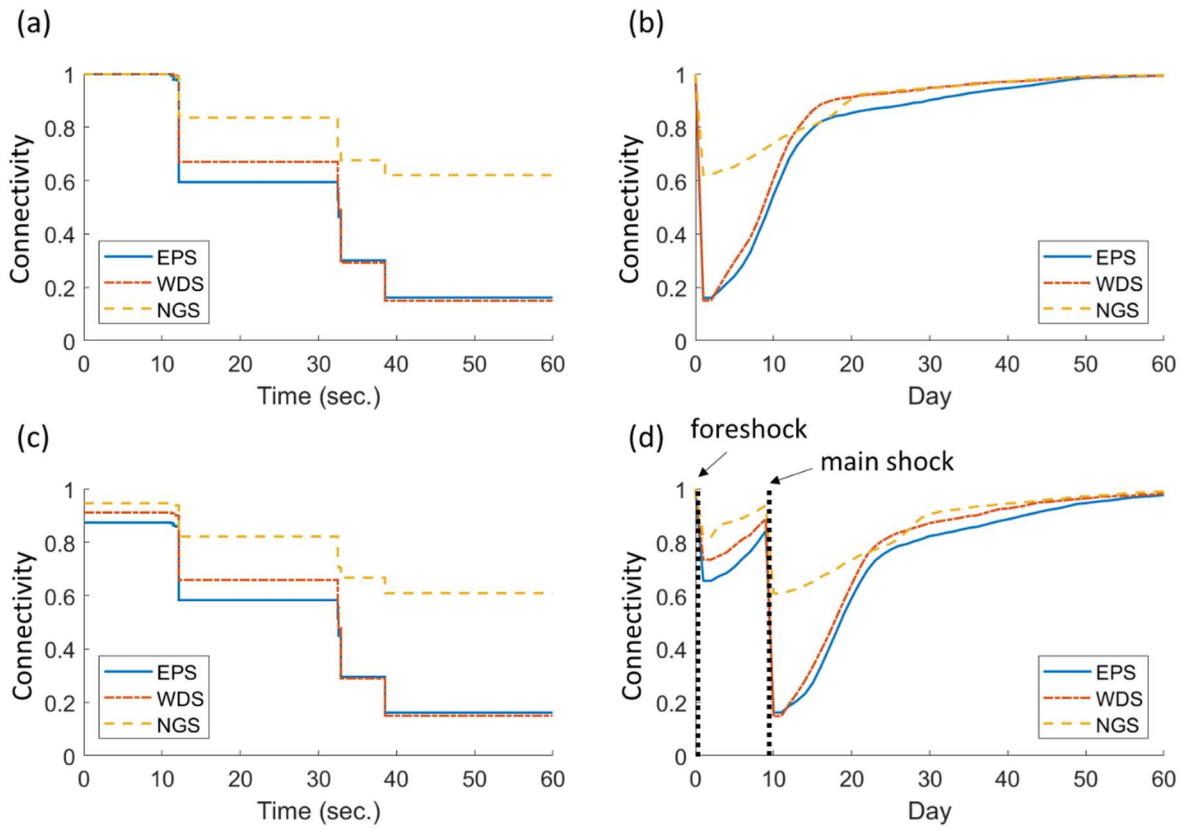


Figure 3-6. Influence of foreshock on the lifeline system performance during (a) main shock without foreshock; (b) overall event without foreshock; (c) main shock affected by foreshock; and (d) overall event with foreshock

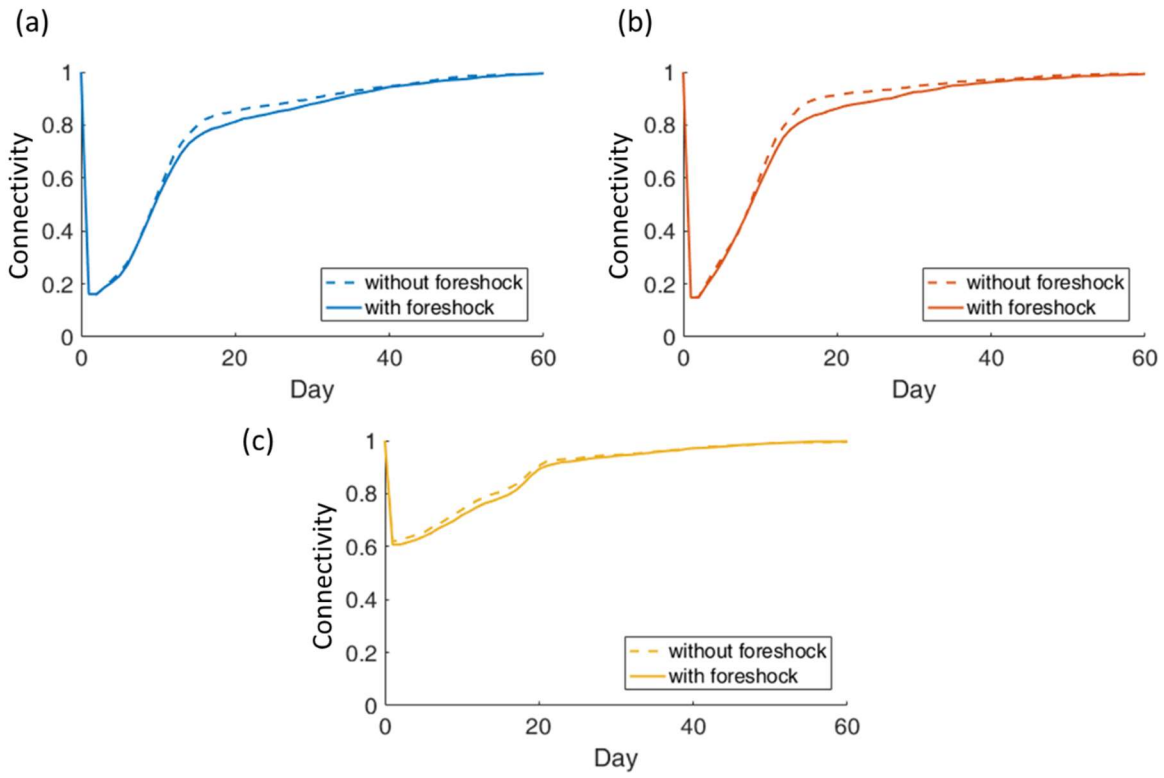


Figure 3-7. Influence of foreshock on recovery after main shock: (a) EPS; (b) WDS; and (c) NGS

3.6.3 Influence of aftershock

Consider a seismic event with EQ2 (main shock) followed by EQ1 (aftershock), $R_{total} = 45$ units/day, and crews with no limitation on their expertise. The allocation of recovery resources is based on the LA and P strategies. Figure 3-8 shows the changes in system performance with an aftershock compared to EQ1 by itself. As shown in Figure 3-8(a) and (b), the aftershock induces additional damage and decelerates the speed of restoration despite being smaller than the main shock. Moreover, by comparing Figure 3-8(a) and (b) with Figure 3-8(c) and (d), it can be seen that the damage due to the aftershock is much more serious than the damage due to a single earthquake with the same magnitude. For example, in Figure 3-8(d), the remaining connectivity of

EPS in the case with EQ1 by itself is about 0.64, but in the case with the aftershock, the connectivity of EPS after EQ1 (aftershock) decreases to 0.42, as shown in Figure 3-8(b).

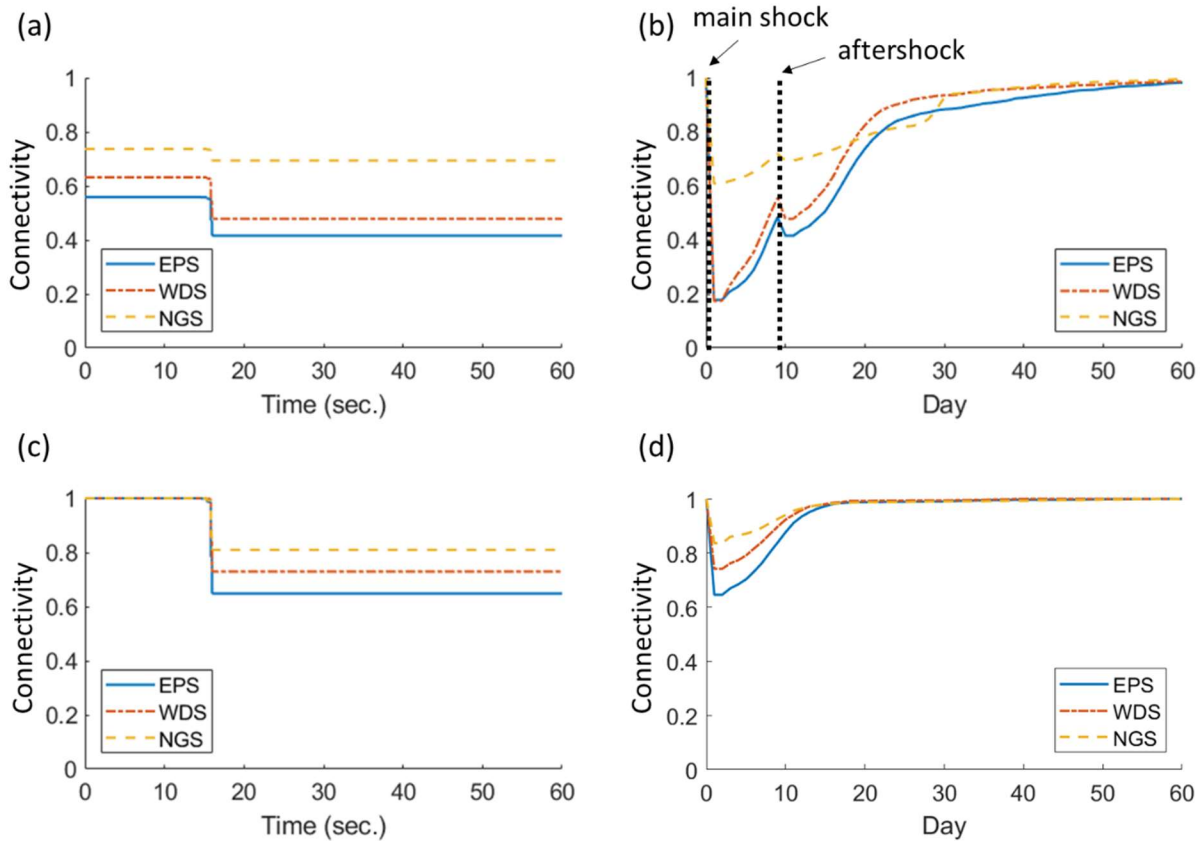


Figure 3-8. Influence of aftershock on the lifeline system performance during (a) aftershock following main shock; (b) overall event (main shock followed by aftershock); (c) aftershock by itself (without considering the effect of the main shock); and (d) overall event involving only aftershock

3.6.4 Effect of recovery strategies

As discussed previously, two levels of recovery strategies are proposed: community to system (EA, LA, and DA) and system to component (R and P). As a result, there are six different combinations of recovery strategies. The schemes are designated by their names—for example, EA followed by R is EA-R and LA followed by P is LA-P.

Consider a seismic sequence of events with EQ1 as a foreshock followed by EQ2 as the main shock. Resources R_{total} = 9 units/day, and repair crews have no limitations on their expertise. The focus is only on the connectivity performance of EPS. It is clear from Figure 3-9 that the three strategies that employ the P allocation have steeper recovery curves in the early stages of reconstruction. They also have better performance in the overall resilience process. Among the six different recovery schemes, the EA-R scheme has the worst recovery performance. The best is DA-P.

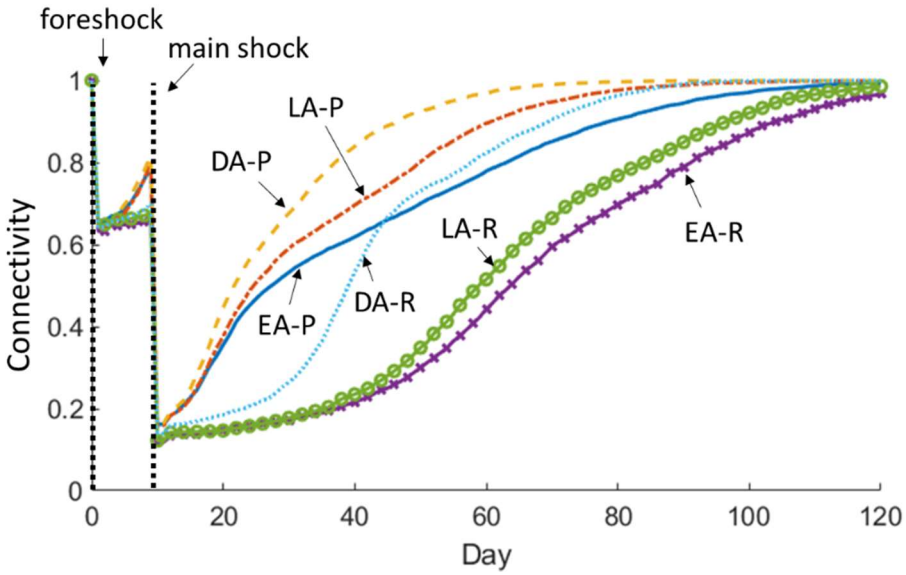


Figure 3-9. Effect of different recovery strategies on the connectivity performance of EPS

3.6.5 Effect of amount of recovery resources

To study the effect of the amount of recovery resources, consider again a case with EQ1 as a foreshock followed by EQ2 as the main shock. In this case R_{total} varies and equals 9, 15, 30, or 45 units/day, and crews have no limitation on their expertise. Focusing again on EPS, the best

(DA-P) and worst (EA-R) strategies discussed previously are employed to maximize the contrast between them, and the results are shown in Figure 3-10(a and b), respectively. As expected, recovery performance improves as more recovery resources are allocated. Figure 3-10 also shows that the effect of limited resources is significantly more pronounced in the lower efficiency scheme. Figure 3-11 compares the effect of the amount of resources on recovery when different strategies are employed. Again, the less efficient schemes suffer more pronounced effects when fewer resources are available.

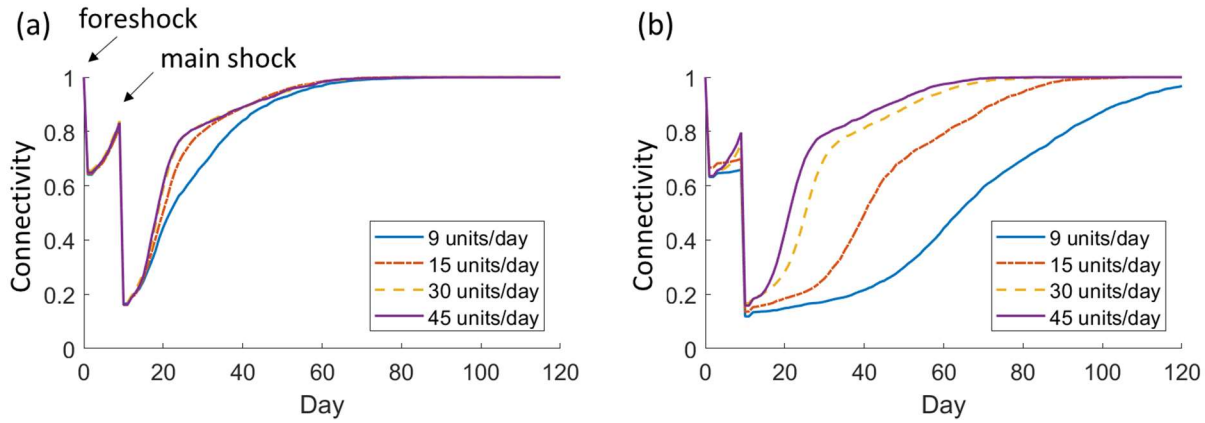


Figure 3-10. Effect of recovery resources on EPS with recovery strategy: (a) DA-P; and (b) EA-R

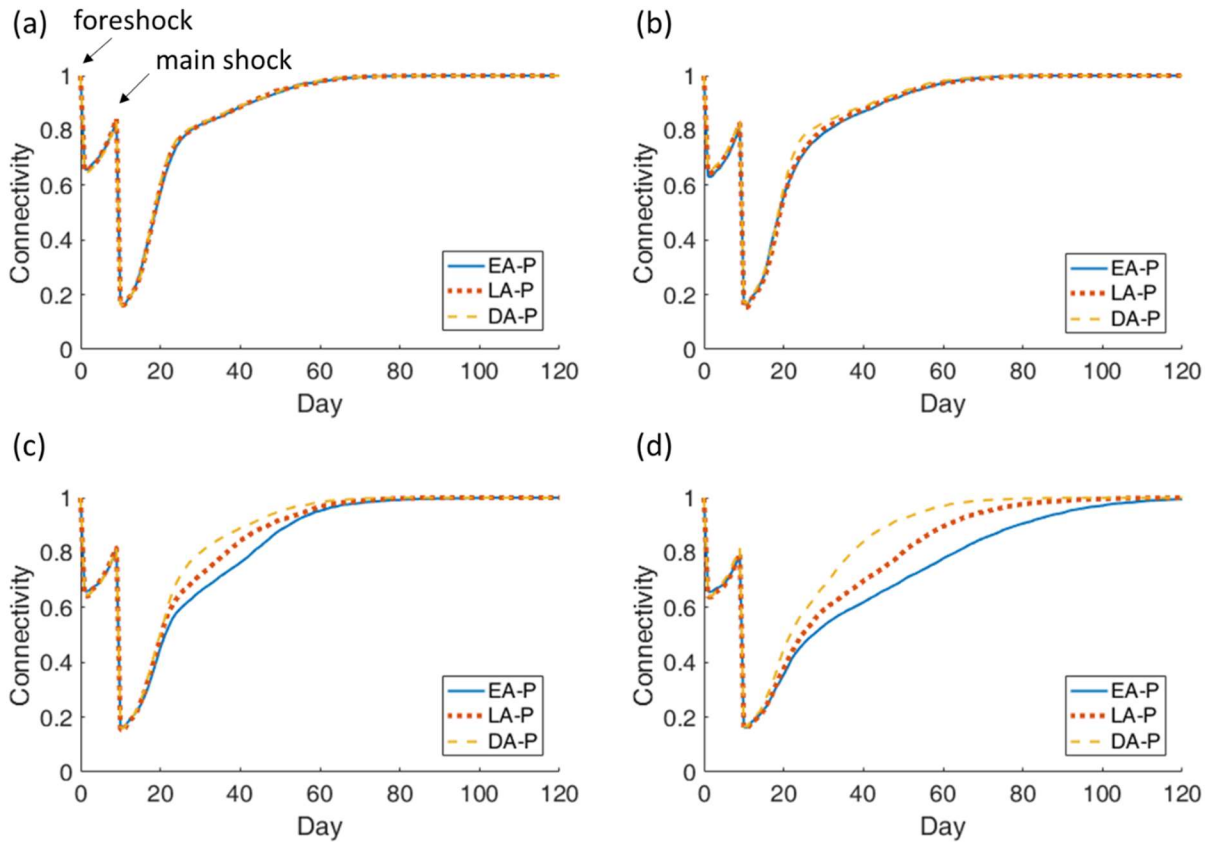


Figure 3-11. The connectivity performance of EPS adopting different recovery strategies with different recovery resources with (a) 45; (b) 30; (c) 15; and (d) 9 units/day

3.6.6 Effect of specialized crews and shortage of workers

Consider a similar study with crews that have specific (not general) expertise—for example, a crew is only able to service a particular lifeline system. In addition, it is assumed that the accessibility of the lifeline service may affect the number of available workers in the area. Consider a seismic sequence of events with EQ1 as a foreshock, followed by EQ2 as the main shock. The DA-P strategy is applied, and three different recovery resource constraints are considered. First, funding is available to pay up to 15 repair crews per day, that is, $R_{\text{total}}(t) \leq 15$ units/day (Constraint 1). Second, the crews are specialized, with up to five crews specializing in

each lifeline system, that is, $R_k(t) \leq R_{max,k} = 5$ (Constraint 2), where $R_k(t)$ is the amount of recovery resources allocated to the kth system. Third, it is assumed that the number of available workers is positively related to the number of males aged 16 to 65 who live in an area with fully functional lifelines (Constraint 3). For example, if the DA strategy is applied, the R_k considering worker shortage is estimated as follows.

$$R_k(t) = \frac{ND_k(t)}{\sum_i^{N_s} ND_i(t)} \times R_{total}(t) \times \frac{male16to65_{funtional}}{male16to65_{total}} \quad (3\text{-}8)$$

where $male16to65_{total}$ denotes the total number of males aged from 16 to 65 who live in Shelby County. $male16to65_{functional}$ is the number of males aged from 16 to 65 who live in the census tracts with access functioning lifeline systems, including electrical power, water and natural gas system. The region of census tracts in Shelby County, as shown in Figure 3-12, and the demographics data of each tract are obtained from Hazus (2015). Table A-1 provides the number of males aged from 16 to 65 living in each census tract and the corresponding demand nodes of each lifeline system that are responsible for the area.

It is worth noting that the third constraint is just an example of considering the mutual-dependency of the recovery of lifelines and the socio-functionality of the community. Equation (3-8) is clearly an assumption made for convenience and could be replaced if detailed information about construction companies and the labor market is available. For example, if the information of casualties is known, the shortage of workers can be predicted more accurately. More importantly, Constraint 3 shows the potential of the proposed framework for coupling physical infrastructure and social systems.

Figure 3-13 compares the performance of EPS with different levels of recovery resource constraints. The figure shows that the resilience of the community is overestimated if crew expertise is not account for, especially in the period between 25 and 60 days. Also, the resilience of the community may be overestimated if the interdependencies between physical infrastructure and social systems are not adequately considered.

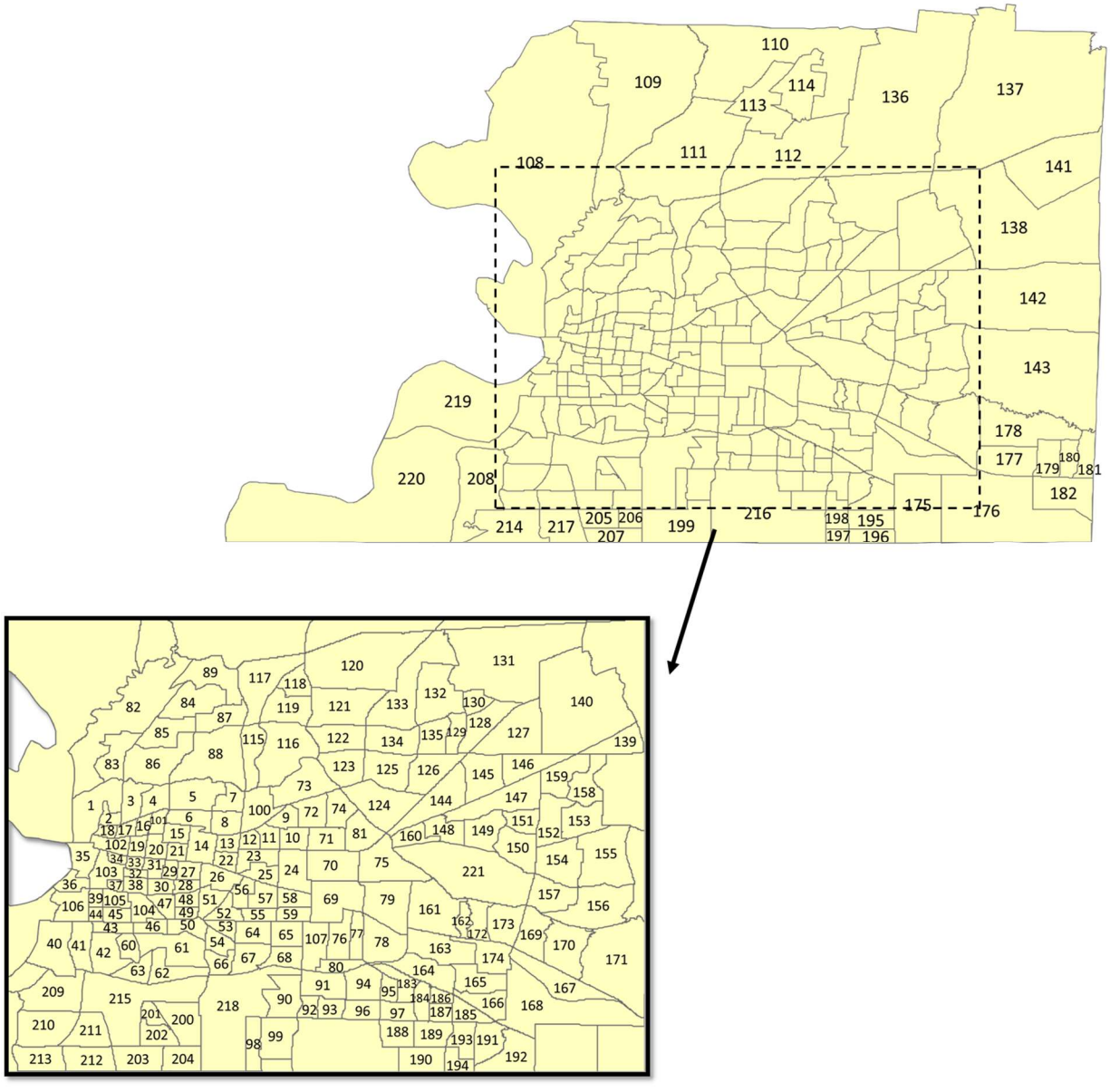


Figure 3-12. The region of census tracts in Shelby County retrieved from Hazus (2015)

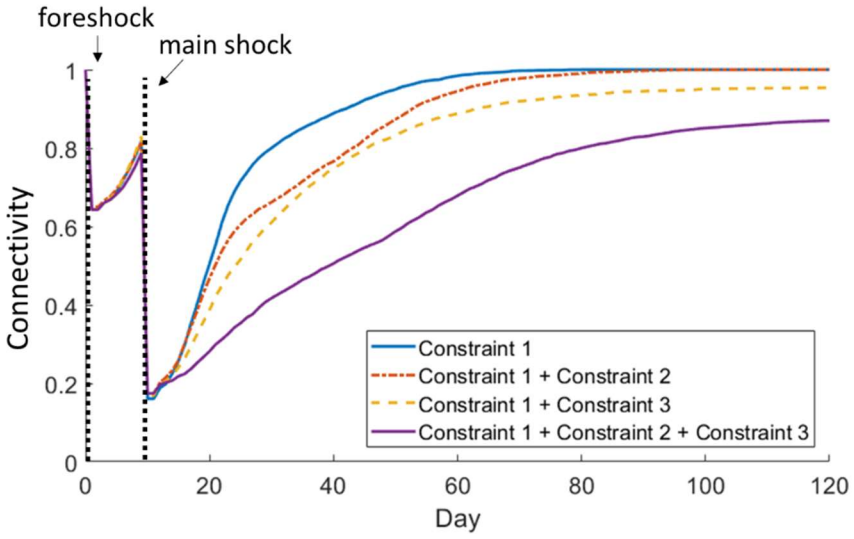


Figure 3-13. Comparison of different levels of recovery resource constraints

3.7 Conclusions

A distributed computational framework was employed to model the interactions that occur between lifeline systems during earthquakes. Various systems were modeled using simulators with disparate temporal and spatial scales. The simulators were connected through the computational platform using publish–subscribe data transmission pattern described in Chapter 2. Shelby County, Tennessee, was used as a case study for demonstrating the ability of the framework to model the interactions between three lifeline systems. The effects of different recovery strategies on system performance were examined as the hazard unfolded and as the recovery process took place. The computational results quantified the influence of the interdependencies between the lifeline systems on the resilience of the community.

Aside from the need to account for multiscale interdependencies, the case study pointed out the necessity of time-varying analysis as the hazard unfolded and during the recovery process.

The seismic hazard considered in this work occurred in just a few seconds. Nevertheless, modeling the interactions that occurred between the lifeline systems during the event provided insights into how interdependencies among infrastructure systems propagate and provided clues as to how to improve their resilience. The ability to handle differences in temporal scales between a hazard and the recovery process is one of the key advantages of the analysis, as evinced by its ability to handle aftershocks that interact with an ongoing recovery effort.

The case study showed that not taking system interdependencies into account will underestimate operability loss and recovery time. It was also shown that that, within the constraints of this research, the strategy of recovery resource allocation had a great impact on community resilience. The impact was exacerbated when resources were insufficient. Among the six resource allocation strategies studied, the ones that adjusted based on damage/reconstruction states enhanced resilience. This points to the necessity of maximizing a community's ability to have good information flow after a disaster. In other words, the hardening of monitoring and communications systems and making them more damage-tolerant is an effective way to increase community resilience. This, of course, can only be achieved by building, prior to the event, institutional relationships that will foster cooperation between the various public and private players that would be involved in response, restoration, and recovery.

A limitation of this work lies in some of the assumptions and simplifications made. For example, the effect of delays due to bad weather conditions and traffic blockages or the effect of limited construction materials on the available number of resource units was not considered. Although these omissions and simplifications may influence the specific results presented in this study, the framework's flexibility and extensibility permit it to address them in the future through the addition of new simulators or the modification of the existing simulators.

CHAPTER 4

Effect of Benefit Fraud in the Wake of Disaster on Community Resilience

4.1 General

The monetary assistance provided for disaster relief creates opportunities for fraudulent behavior. Historical records have shown that the loss of recovery funds due to improper and fraudulent payments could reach hundreds of millions of dollars per event, siphoning support away from those who need it the most and potentially slowing down the economic resurgence of a disaster-stricken community (Section 4.2). In this chapter, an agent-based computational model, which includes a simulation environment of a community facing a natural disaster, fraudsters, and application inspectors, is created to simulate benefit fraud behavior following disasters (Section 4.4). The proposed model considers the effect of both micro-level disaster demands caused by building damages and meso-level social variables on benefit fraud, and estimates the cost to communities associated with these post-disaster crimes. The statistical data from the government reports in the aftermath of Hurricane Katrina and Rita is adopted for the calibration and validation (Section 4.5). Parametric sensitivity analyses quantify how reducing application review errors, decreasing disaster demands, and increasing oversight can help lessen the losses caused by benefit fraud (Section 4.6). They demonstrate how computational simulation can be used to achieve a meaningful balance between the loss of fraudulent payments and the speed of distributing aid in order to improve the overall resilience performance of communities.

4.2 Motivation

The monetary assistance provided by the government and other organizations in the aftermath of large-scale disasters is enormous. While quick disbursement of this aid can reduce human suffering and enable rapid recovery, the abundance of resources combined with emergent conditions create opportunities for disaster-related fraud, which can reach millions of dollars per event. According to the U.S. Government's Accountability Office (GAO, 2006a), the amount of potentially fraudulent assistance in Hurricanes Katrina and Rita through FEMA's Individuals and Households Program (IHP) was estimated to be between $600 million to $1.4 billion, or 10 to 22 percent of the provided aid. For hurricane Sandy, the numbers are smaller, but still significant: $39 million, or 3 percent of the provided monetary assistance (GAO, 2014).

As discussed later on, a number of researchers have studied the influence of disasters and subsequent social conditions on the outcome of crime. However, these studies often addressed criminal behavior at the meso-scale, for example, the change in crime rate at the county level. They also examined the presumed emergence of social solidarity in a community that stopped the majority of crimes from happening, but most did not consider the temporal nature of the problem (Aguirre and Lane, 2019). Importantly, from the perspective of this study, the majority of existing studies did not consider post-disaster crime in the context of long-term community reconstruction and recovery.

Focusing specifically on benefit fraud, the objective of this study is to address the identified gaps in the literature through computational simulation. Benefit fraud is a common type of post-disaster crime where individuals seek to enrich themselves by filing false damage claims. By explicitly considering the temporal nature of the problem and employing an agent-based methodology, this study models and investigates the mechanisms of benefit fraud, taking into

account disaster demands and social variables at both the meso- (community) and micro- (individual) levels. After calibration using data from the Hurricane Katrina and Rita events, the proposed model is used to conduct a parametric study to investigate the effects of key variables on the extent of and propensity for post-disaster benefit fraud.

4.3 Background

Disaster-related crime research is not a central research topic in the literature in disaster science. Zahran et al. (2009) considered baseline demographic variables, i.e., population and economic capital, social order variables, i.e., law enforcement and nonprofit density, and disaster variables, i.e., disaster frequency and presidential disaster declarations, for modeling crime outcomes in Florida. Prelog (2016) used disaster and social indicators, including disaster counts, property damage, crop damage, injuries, income inequality, racial heterogeneity, and ethnic heterogeneity to model the relationship between natural disasters and crime in the United States. Spencer (2017) considered the impact of unemployment and income when estimating the effect of hurricanes on criminal activities. Breetzke et al. (2018) examined the effect of socio-economic deprivation, ethnic heterogeneity, residential mobility, and collective efficacy on crime variability in post-earthquake Christchurch using ten community-level measures. A systematic review of social vulnerability indicators in disasters can be found in Fatemi et al. (2017). The surveyed studies indicate that the current state-of-the-art seeks to quantify the influence of post-impact crime through statistical or equation-based methods that employ meso-level variables. This study takes a different approach based on simulation.

Agent-based modeling (ABM) is selected as the primary computational tool in order to address the desired micro-level variables, i.e. individual-level parameters. ABM is well-

established computational method that has found broad applicability in social science (Epstein, 2006; Fang et al., 2016b; Fang et al., 2016a). The application of ABM in criminology arose in the early 2000s. Groff et al. (2019) provided a systematic review of studies that have used ABM to model urban crime, such as burglary (Johnson et al., 2007; Malleson et al., 2010), drug crime (Dray et al., 2008) , and robbery (Groff, 2007). ABM also has been applied to studying financial crimes, e.g., fraudulent activities in public service delivery programs (Kim et al., 2013) and tax evasion (Hokamp, 2014). To the knowledge of the authors, no studies have used ABM to investigate crime in the wake of disasters as done herein. By necessity, modeling post-disaster crime depends on modeling disaster demands, or the challenges posed by the effects of the disaster, which is another innovation in this study that allows considering the effects of crime.

4.4 Methodology

The process of establishing an agent-based model is divided into the following steps. First, a conceptual model is formalized based on the observed phenomena and the knowledge acquired from previous disaster studies and existing theories in criminology. Second, a computational model, which includes the simulation environment, agents, and their behavior rules, is created for interpreting the conceptual model. After verifying that the constructed computational model adequately represents the properties of the conceptual model, the third step is to calibrate the computational model with empirical data. Finally, artificial experiments (parametric sensitivity analyses) are conducted to study the effects of various variables.

4.4.1 Linking criminology theories to the conceptual model

Before creating a computational model, it is necessary to formalize a conceptual model that adequately represents the situation of interest and captures the theoretical propositions to be explored. The proposed conceptual model builds upon established criminology theories. These theories are briefly reviewed next and key aspects are selected to explain social phenomena observed in disasters. The following discussion is only limited to the main ideas that underlie the proposed model. Additional details about the theories can be found in Miller et al. (2011) and Tibbetts (2018).

The rational choice theory [RCT] (Cornish and Clarke, 1986) is a common theory used to explain criminal behavior. It proposes that the choice to commit crime is based upon the balance of risks, rewards, and efforts perceived by a potential criminal. As a sub-field of RCT, the routine activity theory [RAT] (Cohen and Felson, 1979) is more focused on the available opportunity. In it, criminal activity is conditioned upon convergence of: 1) suitable targets, 2) motivated offenders, and 3) a dearth of guardians. It posits that the absence of any one of these conditions will likely prevent the occurrence of crime. The social disorganization theory [SDT] (Shaw and McKay, 1942) and the therapeutic community hypothesis [TCH] (Fritz, 1996) suggest that crime rate is a function of social conditions. The former suggests that certain ecological characteristics and criminal subcultures of the surrounding area, e.g., high unemployment and low law enforcement density, may increase the likelihood of individuals committing criminal activities. In contrast, the latter theory predicts that illegal activities will decrease in the aftermath of disasters because community cohesion and cooperation rise in the aftermath of disasters. Yet another theory, the social learning theory [SLT] (Jeffery, 1965) proposes that criminal behavior is learned and reinforced by previous actions, so that the propensity of being a repeat offender increases with prior fraud experience.

The most useful summary of these ideas is encapsulated in RAT. According to it, the roles involved in a criminal incident could be divided into three types: targets, offenders, and guardians. In the case of benefit fraud after disasters, the organizations distributing financial assistance are targets that may be scammed by fraudulent applicants (offenders). The government investigators who specialize in detecting fraud crimes are the guardians. Table 4-1 outlines the selected decision factors in a benefit fraud event as inspired by the criminology theories reviewed above and how those factors translate into implementation considerations within the proposed conceptual model.

Table 4-1. **Potential decision factors of benefit fraud and their consideration within the proposed conceptual model**

Criminology theory	Decision factors	Implementation considerations in conceptual model
RCT	Financial demands attributed to the disaster	The repair cost attributed to the disaster. Also of importance is the elapsed time after the disaster without the arrival of assistance from outside the community.
	Effort and skill needed to commit benefit fraud	Probability of an applicant completing a fraudulent application for financial assistance.
	The perceived risks of crime	Number of applicants sanctioned and the extent of punishment for the crime.
RAT	Availability of victims and their vulnerability	Insufficient oversight or weak application review process of victims (such as government or organizations providing aid)
	Guardian strength	Number and diligence of fraud investigators
SDT	Extent of criminal subculture	The criminal subculture influences the propensity of applicants toward committing fraud, where people construct a normative and value consensus that makes criminal behavior appropriate
TCH	Extent of community cohesion	Community cohesion influences the propensity of applicants toward committing fraud.
SLT	Repeat offending	Success in previous fraud efforts may increase the propensity of recommitting benefit fraud. It may also indicate the presence of a criminal career in the deviant, where repeat crime becomes an accepted way of doing things

4.4.2 Constructing the computational agent-based model

Scope and Assumptions

The proposed conceptual model addresses a situation in which benefit fraud may occur when the government or other organizations distribute monetary assistance to homeowners for house reconstruction or relocation after severe natural disasters. Potential claimants are the householders residing in the affected area, and benefit fraud refers to the householders obtaining or attempting to obtain financial assistance from the grantor but having no entitlement to do so, e.g., not having a covered loss as a direct result of the disaster, or duplicating the claims for supernumerary benefits. Benefit fraud committed by non-residential criminals, organized groups and identity thieves is not considered in this study and is left to future research that could include more types of agents.

The inventory of damaged residential buildings and demographic information of homeowners are assumed to be known at the beginning of a simulation. The temporal unit used to update an agent's activity and decision-making is one day. The simulation starts when the first aid application is accepted and ends when all applications have been reviewed and related investigations have concluded.

Agents and Interaction

The simulation of benefit fraud in a disaster is an abstract representation of a hypothesized dynamic relationship between three types of agents: 1) householders in the disaster area, 2) application reviewers in the organizations that distribute financial assistance, and 3) government special investigators, corresponding to the three key components of RAT, i.e., potential offenders, targets, and guardians, in a benefit fraud event. The interactions between the agents is illustrated in Figure 2-2. The opportunity for benefit fraud is present during the interactions between

householders and application reviewers. A householder agent decides whether to submit an application for financial assistance based on the disaster-caused loss and other social impact factors. The probability that a householder agent will submit duplicate/fraudulent applications represents the propensity toward committing fraud. The ability of reviewers to notice suspicious claims represents the vulnerability of the target. Investigator agents provide fraud deterrence and guardianship by investigating suspicious applicants. Claims by fraudulent agents are frozen once they are discovered by investigator agents.

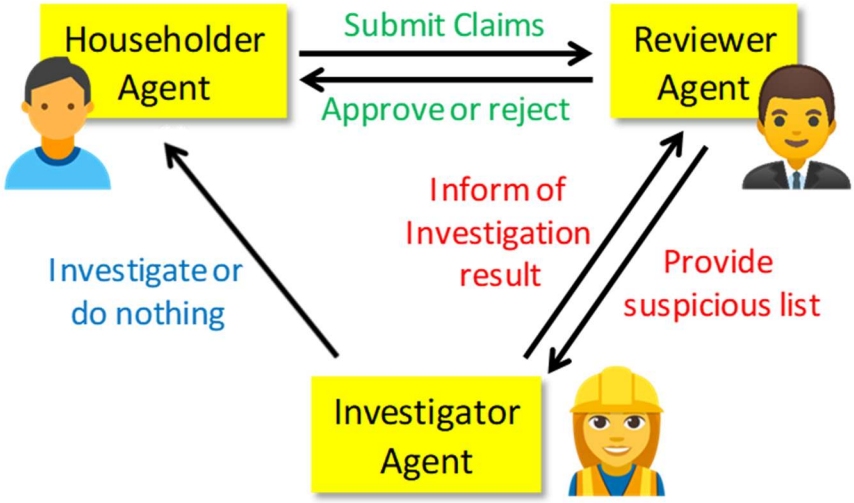


Figure 4-1. Interactions between the agents

The analysis procedure of the ABM simulation is shown in Figure 3-2. The householder's ecological characteristics, e.g. community cohesion and criminal subcultures, and the householder's loss caused by the disaster are the initial inputs to the simulation scenario. This information can be collected by other government agencies in the aftermath of disasters, so investigator agents can make an informed decisions about the likelihood that applications are justified or fraudulent. Applications may have four different results: 1) approved and paid with or

without further investigations, 2) rejected as ineligible or incomplete without investigations, 3) approved and paid by the reviewers but determined as fraud by the investigators in the aftermath, and 4) seen as suspicious by the reviewers and determined as fraud by the investigators before payment is made.

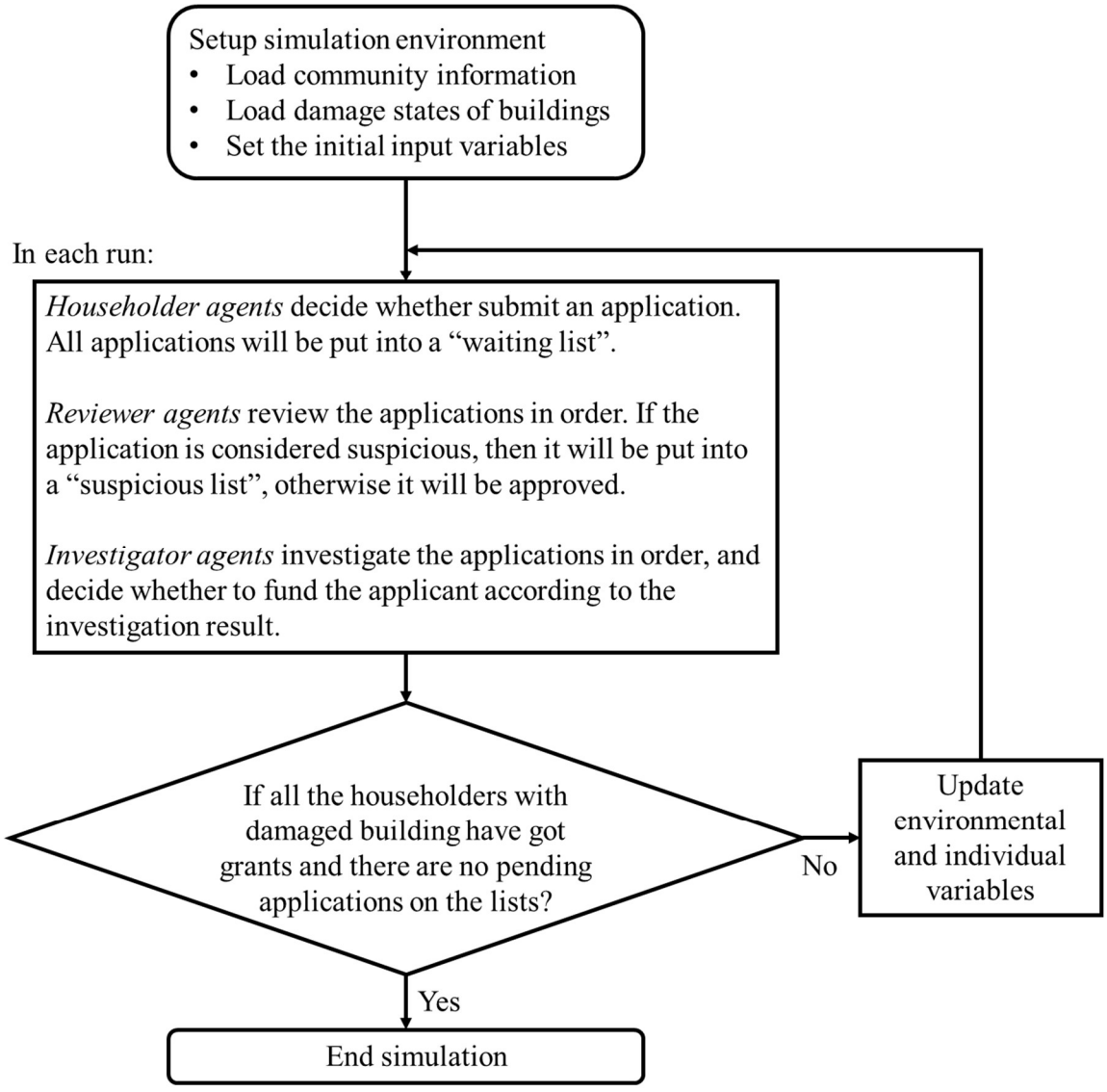


Figure 4-2. Analysis procedure of the ABM simulation

Decision-making rules

The decision process for a householder agent is illustrated in Figure 4-3. Each agent decides whether to submit an application at every step and may submit more than one application during a simulation, i.e., duplicate applications. In Figure 4-3, the probability of a householder agent completing an application at specific time, P_c, characterizes the probability of householders submitting applications in different post-disaster phases. The probability of submitting duplicate/fraudulent applications, denoted as P_f, used to estimate the crime propensity, is described in the next section.

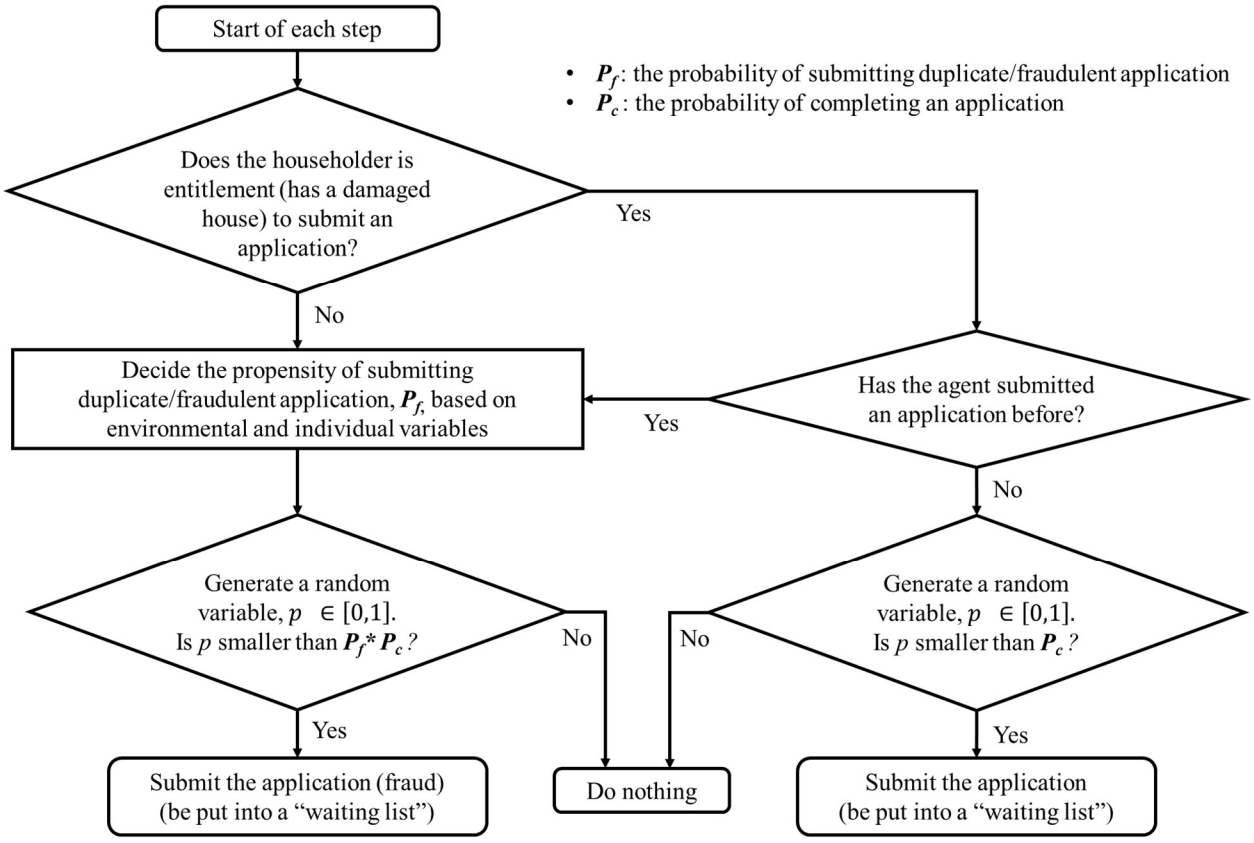


Figure 4-3. Decision-making process of householder agents

The behavior rules for a reviewer agent are shown in Figure 4-4. Reviewers examine applications in the order of submission. Applications that are ineligible or incomplete are rejected and canceled. Eligible applications are either awarded directly or sent to investigators for further investigation. Two types of application review errors are considered here to represent the vulnerability of targets in RAT. Type I error, rv_error_{type1}, refers to judging a justified application to be suspicious, and Type II error, rv_error_{type2}, indicates the case of approving a fraudulent application without further investigations.

The decision rules for investigator agents are presented in Figure 4-5. As with reviewer agents, there may be two types of errors in the investigation results, i.e., judge a justified application as a fraud (inv_error_{type1}) or approve a fraudulent application (inv_error_{type2}). In addition to the suspicious cases listed by the reviewers, investigators may investigate applications already approved by the reviewers. Investigators will provide the results of the investigation to the reviewers and applicants recommended for sanctioning cannot submit further applications. The speed and accuracy of investigations represent the guardian strength in RAT.

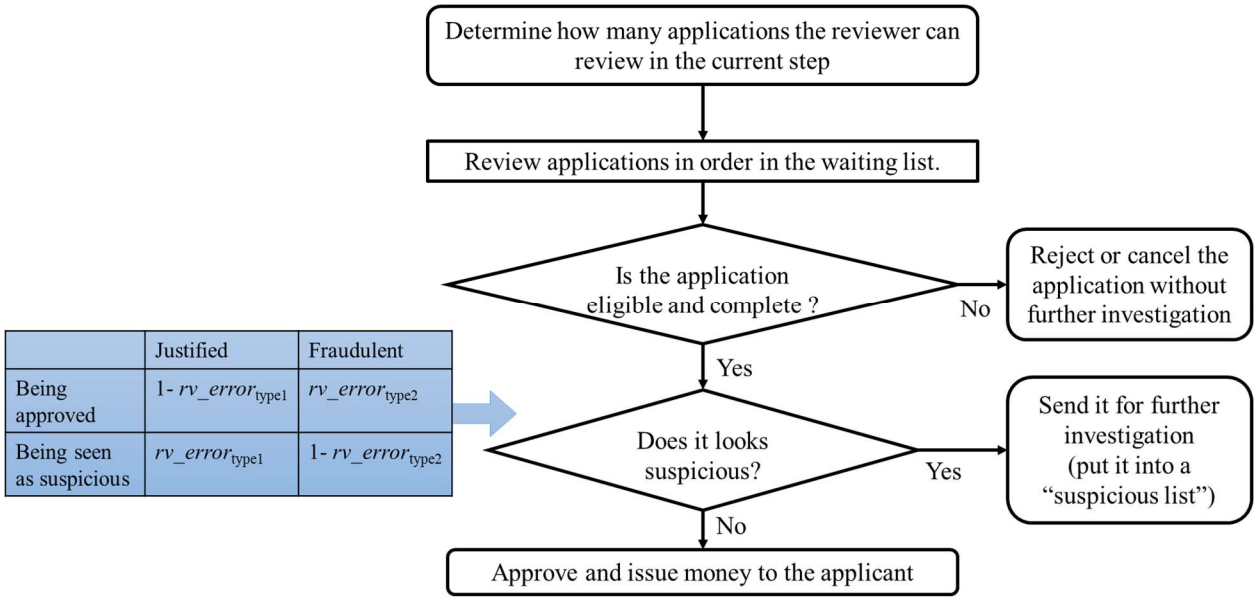


Figure 4-4. Decision-making process of reviewer agents

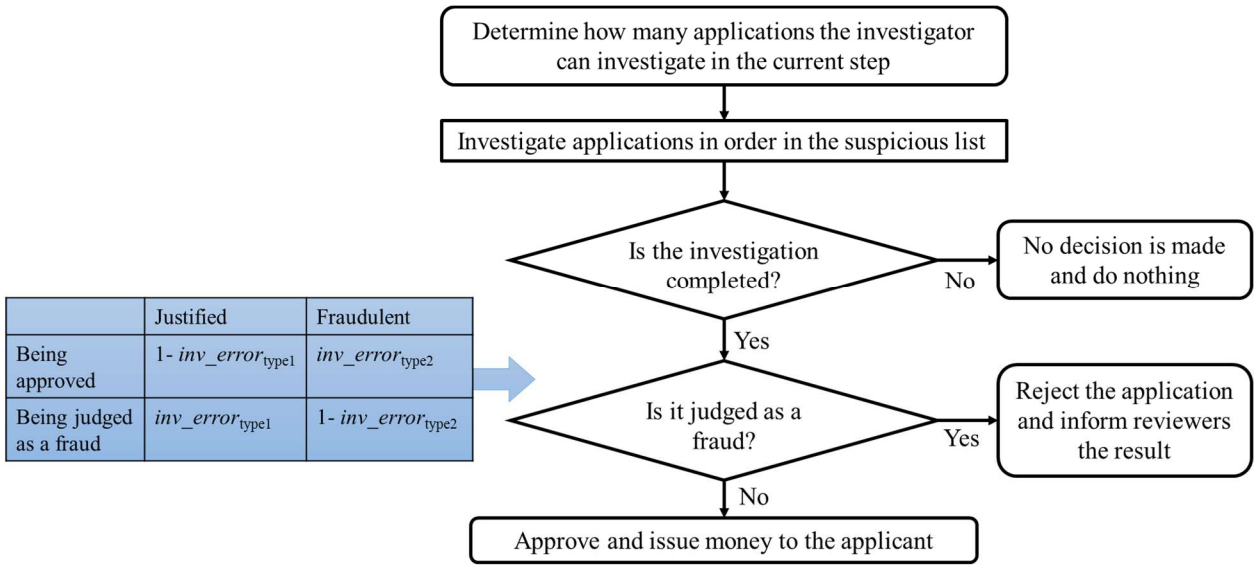


Figure 4-5. Decision-making process of investigator agents

Propensity for committing fraud

The probability that a householder will submit duplicate/fraudulent applications is denoted P_f in Figure 4-3. P_f is a direct function of the social and disaster-related variables at each time step of the analysis, specifically environmental (meso) and individual (micro) level variables. The former pertains to the effect of the environment within which agents take actions and the latter to the individual characteristics of each householder agent. Based on SDT and TCH, two meso-level variables, *CC* and *CS*, are used to consider the effect of community cohesion and criminal subculture, respectively. The measures employed in this work for the quantification of these two variables are listed in Table 4-2 and Table 4-3. The micro-level variables are disaster-caused demands (*DD*) and personal experience with fraud (*E*). *DD* represents the motivation of potential offenders to commit frauds in the wake of disaster (RCT and RAT), and *E* characterizes the learning effect of SLT.

The probability that a householder agent will submit a duplicate/fraudulent application at time *t*, $P_{f,t}$, is proposed to be an additive combination of these variables (weighted) as shown in Equation (3-1).

$$\begin{aligned} P_{f,t} &= P_i \cdot (w_{CC} \cdot CC_t + w_{CS} \cdot CS_t + w_{DD} \cdot DD_t + w_E \cdot E_t) \\ \text{s.t.} \quad & 0 \le P_{f,t} \le 1 \\ & 0 \le P_i \le 1 \\ & -1 \le CC_t, CS_t, DD_t, E_t \le 1 \\ & w_{CC}, w_{CS}, w_{DD}, w_E \ge 0 \end{aligned} \tag{4-1}$$

where P_i is the given initial probability for a householder to submit a duplicate/fraudulent claim (input parameter); CC_t, CS_t, DD_t, E_t are the quantified values for the effects of community cohesion, criminal subculture, disaster-caused demands, and the agent's personal experience at time *t*, respectively. The positive quantified values indicate the stimulative influences on the propensity of committing a crime, and the negative ones represent the debilitating effects. w_{CC},

w_{CS}, w_{DD}, w_E are the corresponding weights for each variable. Calibration of these variables is discussed later on in the following section. It is worth noting that there are no interaction terms in Equation 1, i.e. assuming no correlation between CC_t, CS_t, DD_t, E_t. However, different forms of equation or model can be adopted to estimate the $P_{f,t}$ in the proposed ABM.

Table 4-2. **Meso-level community cohesion variables**

Variables	Measures
Racial heterogeneity	Herfindahl index (Kwoka, 1985): the sum of squared proportions of different *racial* categories.
Ethnic heterogeneity	Herfindahl index (Kwoka, 1985): the sum of squared proportions of different *ethnic* categories.
Economic status	GDP Per capita
Income inequality	Gini coefficient (Gini, 1997)
Unemployment	Unemployment rates of population (15 years or older)
Nonprofit organization density	Total number of nonprofit organizations divided by the population size and then multiplied by 10,000.
Religion-linked organization density	Total number of religion-linked organization divided by the population size and then multiplied by 10,000.
Law enforcement density	Total number of law enforcement personnel divided by the population size and then multiplied by 10,000.

Table 4-3. **Meso-level criminal subculture variables**

Variables	Measures
WCC loss	Average losses caused by white collar crime (WWC)
WCC rate	Percent of white collar crime over the total crime taking place
Percent repeat offenders	Percent of all arrestees who are repeat offenders
Percent WWC guardians	Percent of police force devoted to control white collar crime
Percent dangerous area	Percent city subunits as defined by U.S. Census recognized by police as areas populated by criminals, i.e. dangerous places.
Gang density	Total number of gangs divided by the population size and then multiplied by 10,000.

4.5 Calibration

The main purpose of this section is to show that the proposed agent-based model is able to capture the key features of benefit fraud in the wake of disaster and produce reasonable results. The procedure used here is empirical validation as defined by Groff (2014), where empirical knowledge and data is used to build and calibrate a model. The statistical data of the individuals and household disaster relief provided by FEMA for Hurricanes Katrina and Rita is used to calibrate the ABM model in a control experiment.

4.5.1 Individuals and Households Program (IHP) for Hurricane Katrina and Rita

On August 29 and September 24, 2005, Hurricanes Katrina and Rita caused over 1,800 deaths and more than 1.2 million people were evacuated or displaced throughout the Gulf Coast region (NHC, 2006). The Federal Emergency Management Agency's (FEMA) Individuals and Households Program (IHP) is one of the main disaster relief programs that provided financial assistance to the victims, including rental, repair, replacement, property and expedited assistance. The information and data on the benefit fraud of FEMA's IHP assistance is obtained from the U.S. Department of Justice Hurricane Katrina Fraud Task Force (DOJ, 2005, 2006a, 2006b, 2007, 2010, 2011), the U.S. Department of Homeland Security Office of Inspector General (DHS-OIG, 2006; DHS, 2006; DHS-OIG, 2011), and the U.S. Government Accountability Office (GAO, 2006b, 2006a, 2006c, 2006d, 2007, 2014).

Distribution of Aid

After the hurricanes, FEMA IHP provided expedited assistance of $2,000. The program also offered rental assistance funds (based on area fair market rent) and additional housing assistance with repair and replacement assistance capped at $5,200 and $10,500, respectively. The

combination of all forms of IHP financial assistance had a maximum cap of $26,200 (DHS-OIG, 2006). In 2006, the expedited assistance was adjusted to $500, and the other housing assistances were increased to $5,400, $10,900 and $27,200, respectively, to reflect increases in the Consumer Price Index (GAO, 2006c).

According to DHS-OIG (2006), by September 30, 2005, FEMA had received 1,557,937 registrations for IHP assistance from residents in all affected areas and awarded over $2.4 billion. By October 18, 2005, there were 1,645,784 total registrations. Among those, about 19 percent were cancelled due to ineligibility or flagged as potential duplicates (DHS-OIG, 2006). By November 16, 2005, the registrations increased to 1,680,516 of which 984,432 (or 59 percent) were eligible and worth approximately $3.5 billion in assistance (DHS-OIG, 2006). By mid-December 2005, mid-February 2006, and mid-May 2006, IHP payments totaled about $5.4 billion, $6.3 billion and $6.7 billion, respectively (GAO, 2006a, 2006b). By August 2006, among the total 2.4 million applications for IHP, only 67% of household assistance and 41% of other needs assistance applications were found eligible and had been approved, and approximately 9% of applications were pending or appealing the assistance decision. Approximately $7 billion of IHP assistance was distributed by October 2006 (GAO, 2006c, 2007).

The approximate number of total registrations and awarded aid of IHP assistance for hurricanes Katrina and Rita are plotted in Figure 4-6 and Figure 4-7. About 65% of the registration were received in the first month, assuming that the total number of registrations by Oct 2006 is 100% (Figure 4-6). Figure 4-7 shows that about 50% of aid was awarded in the first three months, and over 90% was distributed in the first six months.

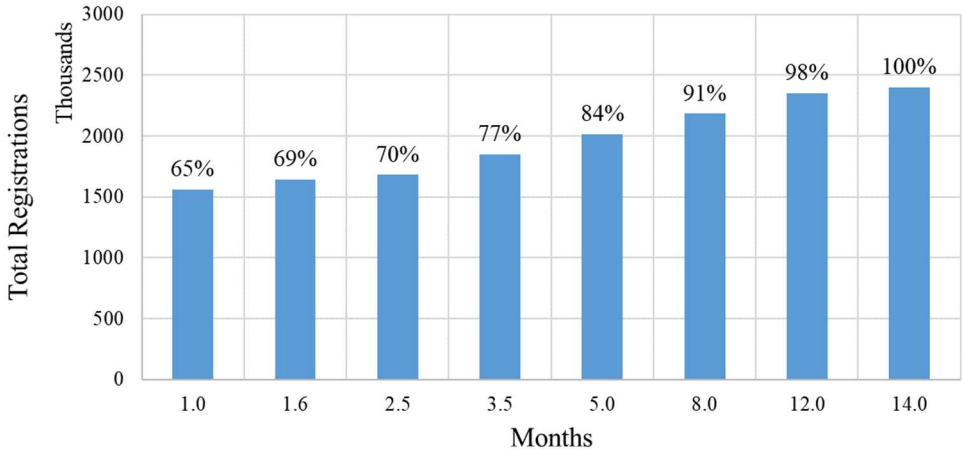


Figure 4-6. Approximate number of registrations for IHP assistance

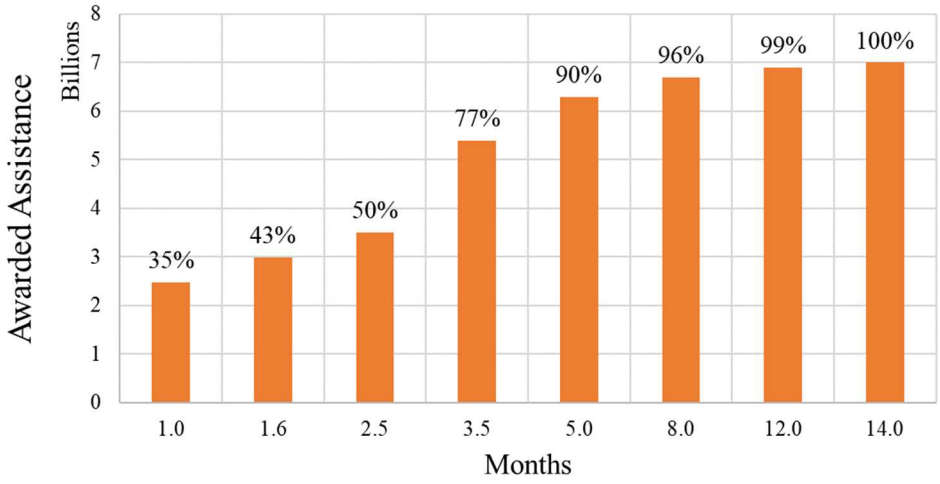


Figure 4-7. Awarded assistance for IHP (USD)

Improper and potentially fraudulent payments

Although a considerable proportion of total registrations had been determined as ineligible applications and canceled before being paid, many improper and potentially fraudulent payments among the awarded applications were recognized, including duplicate registrations, invalid primary residences, and bogus damaged addresses. The GAO selected a random sample of 250

payments of the 2.6 million IHP payments made to hurricanes Katrina and Rita registrants by February 2006 for further examination (GAO, 2006a). An estimated 16 percent of payments (95 percent confidence interval of 12 percent to 21 percent of payment or from $600 million to $1.4 billion) totaling approximately $1 billion were improper and potentially fraudulent due to invalid applications (GAO, 2006a). On the other hand, using a different sampling and analysis method, DHS reported an estimated improper payment rate of 8.56% for an estimated improper payment amount of $450 million by March 1, 2006 (DHS, 2006). FEMA reported that it had overpaid about $290 million to nearly 60,000 registrants but it had recollected only about $7 million by November 2006 through its own internal means (GAO, 2006d). FEMA's recoupment activities were suspended from June 2007 through January 2011 due to a lawsuit and other challenges. As a result, even though more than $621.6 million of potentially improper IHP payments have been identified, much of this improper disaster assistance disbursed since Hurricane Katrina was uncollected for several years. This evidence shows the weakness of the benefit distribution process that exposed IHP assistance to fraud, and served to emphasize the importance of preventive controls (DHS-OIG, 2011).

4.5.2 Control experiment

Model environment and parameters

In the control experiment, the number of householder agents is set to 1.3 million, the approximate number of victims of hurricanes Katrina and Rita. The distribution of home damage levels and the number of householder agents for each damage level are listed in Table 4-4, which is based on the estimated number of damaged houses in Mississippi after hurricanes Katrina and Rita (DHS-OIG, 2006). The damage level of houses are randomly assigned to the householder

agents at the beginning of each simulation according to the distribution in Table 4-4. The definition of different damage levels and the corresponding eligible assistance are documented in DHS-OIG (2006). It is assumed that the claimed and awarded amounts are equal to the eligible assistance for the householder agent, as listed in Table 4-4, but householder agents may submit more than one claim in the overall simulation, i.e., duplicate/fraudulent applications.

Table 4-4. **Distribution of house damage states and corresponding eligible IHP assistance**

Damage level	None	Minor	Major	Destroyed
Percentage	7.7%	81.0%	9.8%	1.5%
Number of householder agents in ABM	100,000	1,053,000	127,000	20,000
Eligible house assistance (USD/ householder)	0	2600	5,200	10,500
Eligible assistance for other needs (USD/householder)	2,000	2,000	2,000	2,000
Total eligible IHP assistance (USD/ householder)	2,000	4,600	7,200	12,500

The other parameters used in the control experiment are listed in Table 4-5, where the number of applications reviewed and investigated per day, and the initial probability for householders to submit a fraudulent claim are set to normal random variables to capture the uncertainty and variations among individuals. To cater to the fact that most of the registrations had been made in the first few months, the probability of a householder completing an application in a given day, P_c, varies in different post-disaster phases (see Table 4-5). Also, it was often observed that very few ineligible and suspicious applications were detected due to the felt need for immediate support and the lack of inspection staff during the emergency phase right after the disaster. It is assumed that the vulnerability of targets being exposed to fraud will decrease with the recruitment of more staff and enhancing fraud-related training. Therefore, the eligible rate of applications and the errors of review vary with time to represent the changes in target vulnerability.

To simplify the problem, it is assumed that neither type I error nor type II errors of investigation occurred in the control experiment.

Table 4-5. Parameterization of the model for control experiment

Parameter	Value
Number of householders	1.3 million
Number of application reviewers	3,000
Number of special investigators	1,000
Number of applications reviewed per day by per reviewer, mean (standard deviation)	5 (1)
Number of applications investigated per day by per investigator, mean (standard deviation)	2.4 (0.5)
Probability of householder completing an application in a given day, P_c	30% [a], 25% [b], 8% [c]
Initial probability for householders to submit a fraudulent claim, P_i , mean (standard deviation)	1% (0.1%)
Probability of application being eligible and complete, P_e	95% [a], 75% [b], 60% [c]
Probability of type I error of review (judging a justified claim to be suspicious)	0% [a], 2% [b], 5% [c]
Probability of type II error of review (approving a fraud claim without further investigations)	100% [a], 80% [b], 60% [c]
Probability of type I error of investigation (judging a justified claim as a fraud)	0%
Probability of type II error of investigation (approving a fraud claim)	0%
The deadline of submitting applications (number of days after the disaster occurred)	426 days (14 months)

[a] Day 1 – Day 14; [b] Day 15 – Day 45; [c] After Day 46;

Due to lack of pertinent records, the mean values of *CC* and *CS* are assumed to be time-dependent index functions as plotted in Figure 4-8. The initial negative value of *CC* in the early stages of the disaster reflects the cooperative behavior of the community during the immediate post-impact period as predicted by the TCH theory. Eventually, as modeled by the index function, the helpful behavior disappears and returns to neutral. In accord with SDT, the downhill trend of *CS* reflects the dearth of guardians and the negative effect of social disorganization during the post emergency phase. It is also a function of the enhanced defenses devoted to control benefit crime in the latter phase of disaster.

To consider the variation of community cohesion and criminal subculture impacts on individuals, normal random variables with the time-varying mean values in Figure 4-8 and standard deviation of 0.25 are employed for the *CC* and *CS* indices for each householder agent at every time step. The values and trends indicated by Figure 4-8 and discussed above are initial estimates that guided by relevant theories of criminology. These estimates can be refined in the future as better information becomes available.

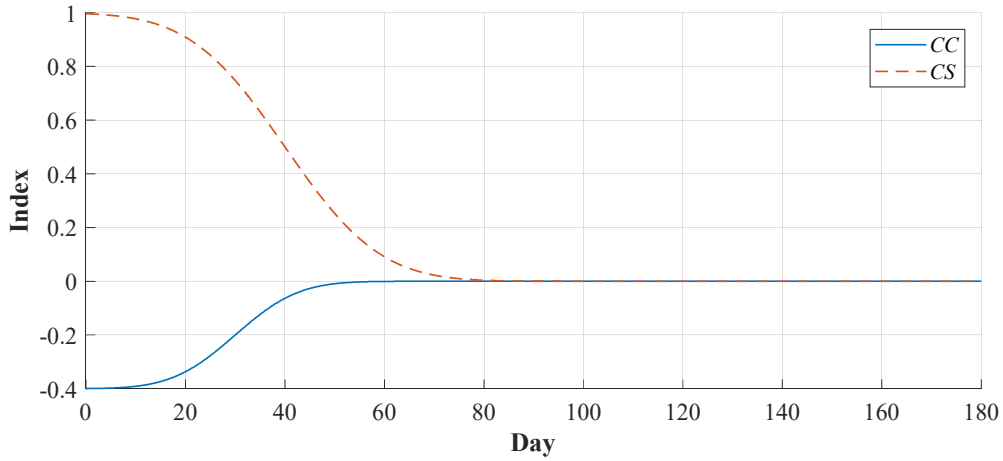


Figure 4-8. Time varying community cohesion and criminal subculture effects as inspired by TCH and DST criminology theories

Variable DD_t (at time t) is calculated for individual householder agent at each step as shown in Equation (3-2). The equation models the effects of disaster-caused demands and the period of time experienced without financial support, where increases in both enhance the effects of the disaster demands and vice versa.

$$DD_t = \frac{DL \times T_{without\ support}}{8} \quad \text{s.t. } 0 \le DD_t \le 1 \tag{4-2}$$

where DL = 4, 3, 2, and 1 for destroyed, major, minor, and none damage levels of the householder agent's home, respectively; $T_{witho\ \ support}$ is number of weeks after the disaster that the householder agent has not received monetary support. DD_t is capped to 1.0. For a householder agent with a destroyed home, DD_t equals to 1 if their $T_{witho\ \ support}$ is longer than two weeks.

Based on SLT theory, the success in previous applications may increase the propensity of recommitting benefit fraud. Therefore, the fourth meso-level variable, E_t, is used to represent the effect of an agent's personal experience. E_t is assumed to be 0.2 times the number of approved applications $N_{approved}$, submitted by the agent, with an upper bound of 1.0, as expressed in Equation (4-3). The corresponding weights w_{CC}, w_{CS}, w_{DD}, w_E in Equation (3-1) are all equal to one. As with *CC* and *CS*, these weights are assumed values at present that can eventually be refined as more information becomes available in the future.

$$E_t = 0.2\, N_{approved}\,, \qquad \text{s.t. } 0 \le E_t \le 1 \tag{4-3}$$

Monte Carlo simulations are conducted to capture the effects of variability in the main parameters. The estimated improper payment rate obtained from a different number of Monte Carlo realizations, i.e., *n*, are compared in Figure 4-9. It is clear that the results converge when the number of realizations is more than 500. Therefore, the mean value and standard deviation of one thousand Monte Carlo realizations (n = 1000) are used in the following calibration and parametric studies.

Figure 4-9 shows that the crime response of the community occurs in three distinct phases. In Phase I, there is an initial gentle increase in fraud for the first two months. Phase II, which spans about a month, sees a sharp rise in improper payments at the beginning of the third month. In Phase

III, the crime rate ameliorates. These distinct phases fit the observation in Aguirre and Lane (2019) that crime level exhibits distinct phases in the aftermath of disasters.

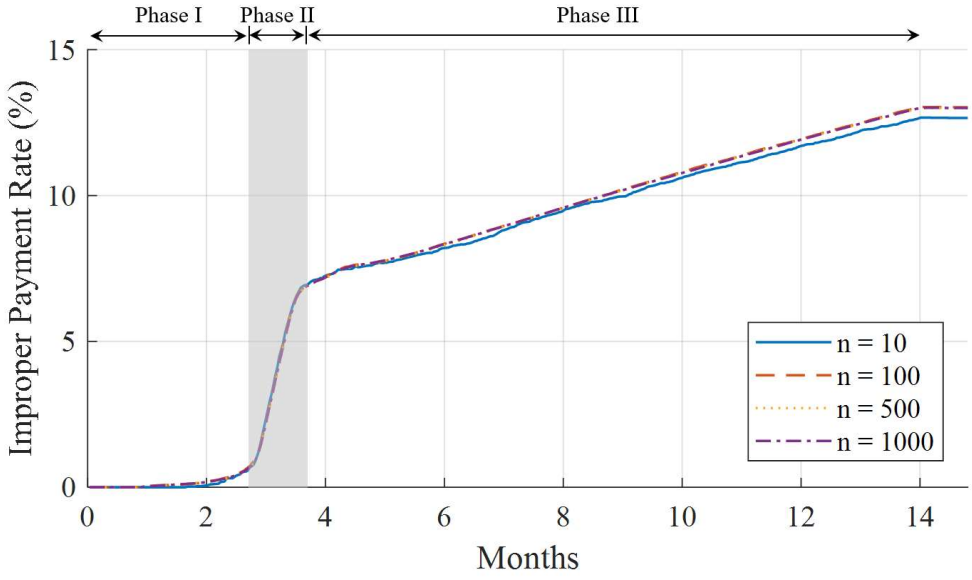


Figure 4-9. Estimated improper payment rate obtained from Monte Carlo simulations with a different number of realizations

Adjustment of parameters

One of main strength of ABM is the ability to model the desired meso-level phenomena by properly adjusting individual-level parameters. Figure 4-10 shows how the probability of a householder completing an application, P_c, affects the average number of received applications during the first 14 months, where the shaded area represents the range between standard deviations, i.e., $\pm\sigma$. The actual data from the Hurricane Katrina and Rita case study is represented as circles. With P_c constant at 10% and the other parameters kept the same as shown in Table 4-5, the figure shows that the number of applications after the third month matches well with the actual data, but the results in the first two months are much fewer than the reported data (dotted line in Figure

4-10). Increasing P_c to 30% (dashed line in Figure 4-10) makes the number of applications rise faster in the beginning, but causes significant deviations at later times. Therefore, to represent the fact that most of the registrations had been made in the first few months and became much fewer later, P_c is made to vary from 30% to 8% with the post-disaster phase as listed in Table 4-5 (solid line in Figure 4-10).

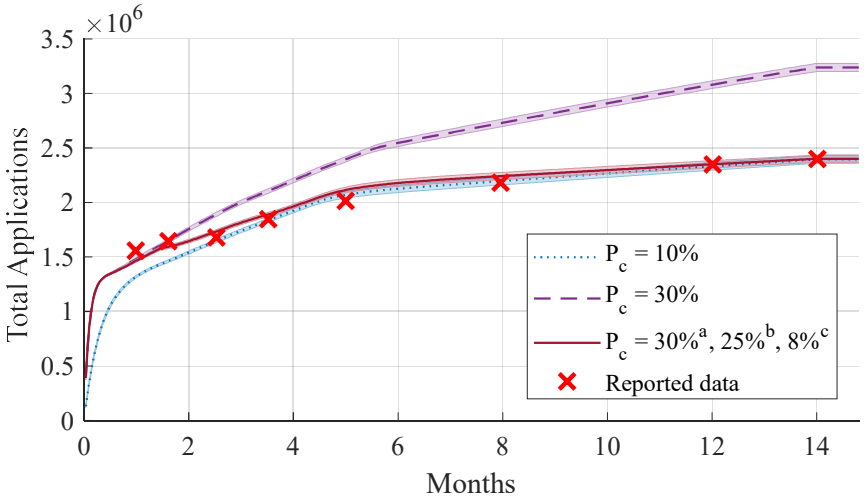


Figure 4-10. Number of received applications in the control experiment

Using the distribution of P_c in Table 4-5, Figure 4-11 shows the change in the amount of assistance granted versus the number of application reviewers. Clearly, the results of the case with 3000 reviewers match the actual data well, lending some credence to the selected parameters. Figure 4-12 shows the improper payments rate over time as a function of P_i, the initial probability that a householder will submit a fraudulent claim. The estimated final improper payment rates are approximately 8.3%, 13.0%, and 18.3% when P_i equals 0.6%, 1.0% and 1.5% (with 0.1% standard deviation) respectively. The numbers compare favorably to the estimates provided in GAO (2006a) (12% to 21%) and DHS (2006) (8.56%). It is observed from Figure 4-12 that the improper payment

rate had a shaper rise during the third month in the control experiment, which is a direct function of the selected parameters and trends. Judging the veracity of this prediction is not feasible because the collected statistics are not available at the given time intervals (only the final numbers are available). Nevertheless, Figure 4-12 indicates that the parameters and trends can be broadly tuned by changing the initial agents' propensity toward committing frauds, P_i, and the other parameters.

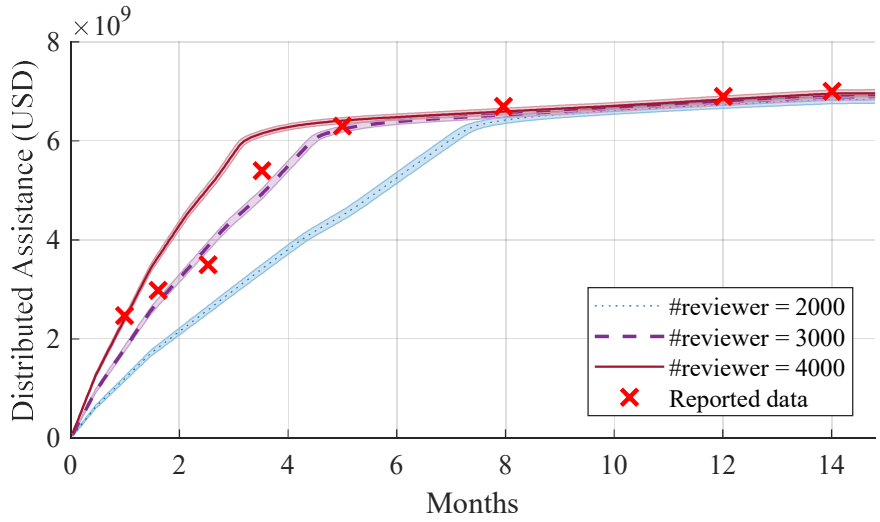


Figure 4-11. Distributed assistance in the control experiment

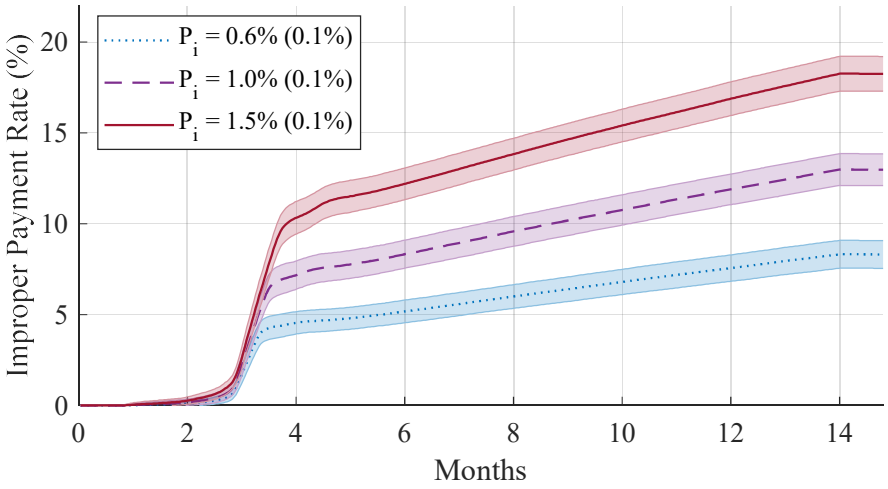


Figure 4-12. Improper payment rate

4.5.3 Results and discussion

Results of control experiment

The parameters of the model used in the control experiment are listed in Table 4-5. Figure 4-13 shows the proportion of ineligible applications canceled by reviewers before payment as a percentage of total reviewed applications over time. According to DHS-OIG (2006), 19% of the registrations were cancelled or flagged as potential duplicates by October 18, 2005, and only about 59% were eligible by November 16, 2005 (41% were found ineligible). By August 2006, only 67% of household assistance were discovered eligible, i.e., 33% were ineligible, (GAO, 2006c, p. 18). The reported data is not enough for calibration, but it can be tuned as needed or when detailed information becomes available by adjusting the probability of applications being eligible and complete, P_e, as listed in Table 4-5.

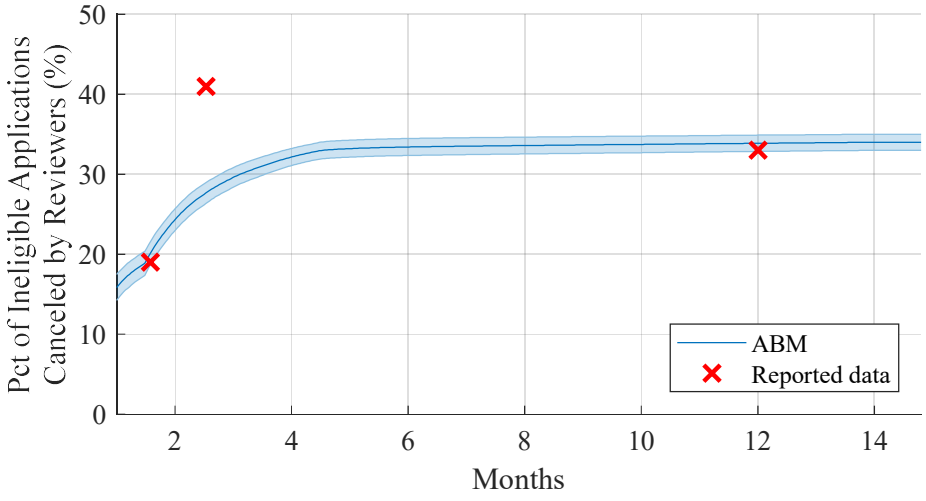


Figure 4-13. Ineligible applications canceled by reviewers

The FEMA investigation detected about $290 million in overpayments by November 2006, about 15 months after the hurricane (GAO, 2006d, p. 4). The simulation shows that the average improper payment rate by the 15th months is $312 million, which matches well with the given data, as shown in Figure 4-14. It should be noted that, in the actual scenario, it took about 23 months to detect all improper payments, but the rate of detection is not known. Figure 4-15 shows the proportion of justified and improper payments of total distributed assistance estimated in the control experiment.

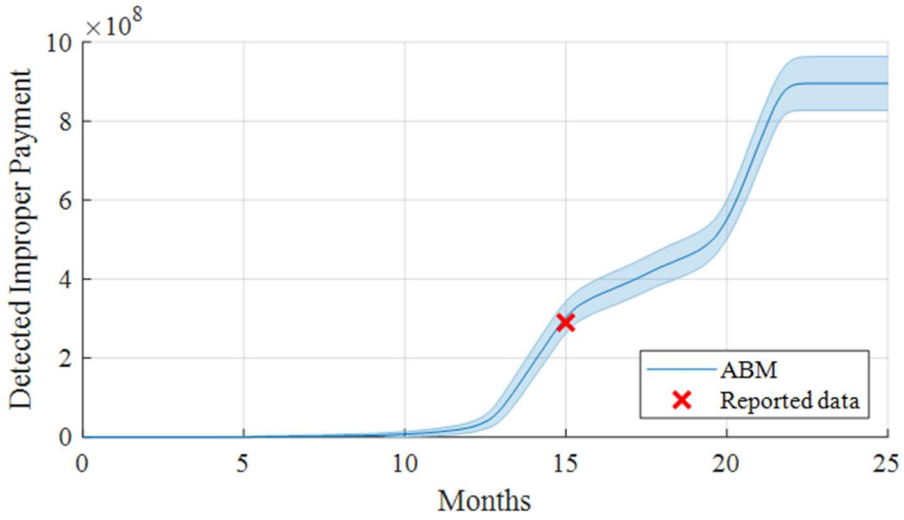


Figure 4-14. Detected improper payments

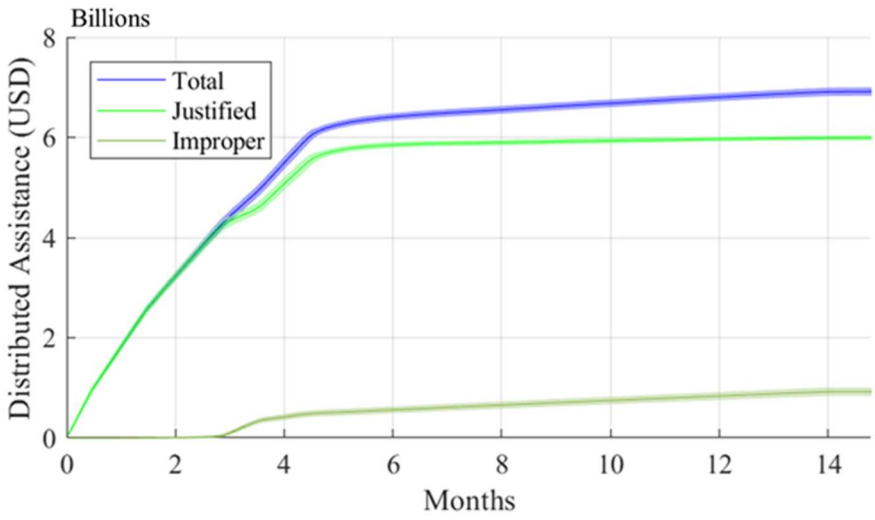


Figure 4-15. Comparison of justified and improper payments

Figure 4-16 examines the relationship between number of months from occurrence of disaster to date of fraud being detected and the Log10 of thousand dollars lost to benefit fraud, where the crosses represent the result of one realization of Monte Carlo simulation. The ABM result is compared to the plot in Aguirre and Lane (2019), which showed the cost of various types of white collar crime (WCC) in the aftermath of the Hurricane Katrina and the British Petroleum Gulf Coast oil spill versus the number of months to arrest. The difference between the control experiment in this study and Aguirre and Lane (2019) is that the control experiment only considers the benefit fraud of IHP in Katrina, and the loss is interpolated by the number of months to complete the investigation. The dollars lost to benefit fraud is comparatively lower because of the amount of financial assistance had a maximum cap. It is needed to build ABMs addressing different WWC and distinguish different crime types from the data in Aguirre and Lane (2019) for further comparisons.

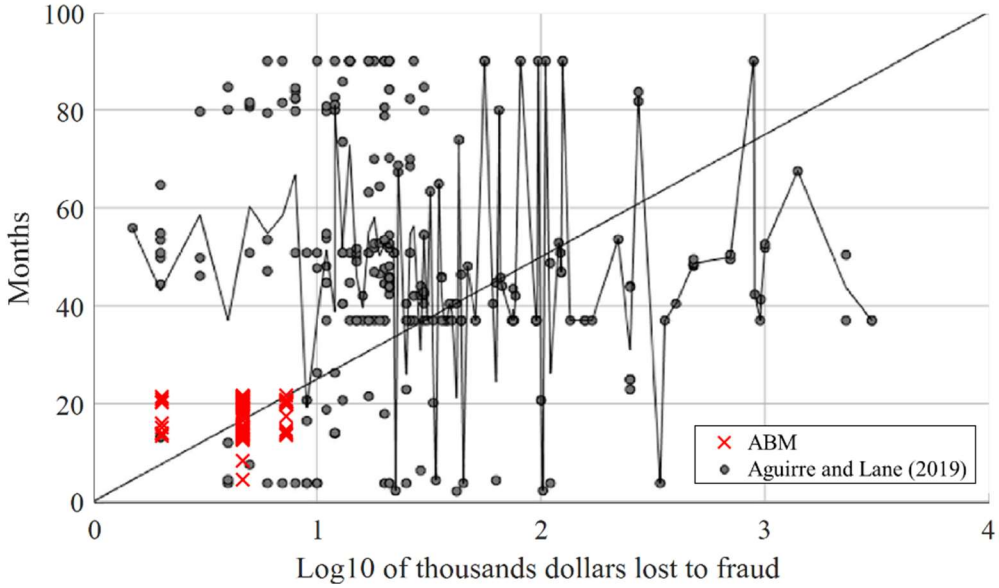


Figure 4-16. Relationship between dollars lost to benefit fraud and month of investigation [reprinted from Aguirre and Lane (2019)]

Limitations of the calibrated model

It is commonly accepted that validation of ABMs is challenging due to the complexity and uncertainty of modeling human behavior and limitations in obtaining meaningful calibration data. Even if more calibration data were available, the resulting calibrated model will still only be valid within the range of parameters for which it was calibrated. Moreover, while any specific favorable comparison to experimental or observed data strengthens confidence in the model, the high-dimensional nature of the research problem suggests that the model will not necessarily yield correct answers under all conditions. The control experiment conducted above shows that the proposed model can reasonably capture the general characteristics of post-disaster benefit fraud in a manner that is consistent with well-known theories of criminology. Most importantly, it clearly points to areas that need additional data and provides motivation for researchers to collect this type of information during future events.

Moreover, in this study, the modeled benefit fraud refers to the householders obtaining or attempting to obtain financial assistance but having no entitlement to do so, e.g., not having a covered loss as a direct result of the disaster, or duplicating the claims for supernumerary benefits. It is an important future research to include different behavioral pattern of benefit fraud, e.g., the fraud committed by criminals outside of affected areas or syndicated criminal activities, for estimating how crime varies in the aftermath of disasters.

4.6 Parametric Sensitivity Analyses

The agent-based model described and calibrated in the previous section is used to conduct a parametric sensitivity analysis to gain deeper insight into the benefit fraud problem. Two key dependent variables are considered here. The first is the percentage of overall improper payments, Pct_{Fraud}, which indicates the proportion of funding that could be used more effectively. The second dependent variable is the speed of distributing financial assistance, and the time required for 95% of victims with disaster-caused loss to receive the grants, $T_{95\%}$, is adopted. Both dependent variables are important and should be evaluated at the same time. The effect of meso/micro impact factors and disaster-caused demand on crime propensity are discussed. The other independent variables investigated in this section include the number of special investigators, the type II error of application review, and the number of applications reviewed per day per reviewer.

4.6.1 Model environment and parameterization of the base model

An artificial environment is designed to represent a community of 1 million households that is subjected to disasters with varying intensity levels. Three different distributions of house damage states are adopted to represent different level of the disaster impact, as shown in Table 4-

6 and Figure 4-17. The replacement cost of a residential building is assumed 253,728 USD, which corresponds to the replacement cost of an average class, two story, single family dwelling model with a typical size of 1,600 square feet (FEMA, 2003). The repair cost ratios of residential buildings, including structural and non-structural components, for five damage states (none, slight, moderate, extensive, and complete) are defined as 0%, 2%, 10%, 50% and 100% of building replacement cost, respectively (FEMA, 2003). To capture the uncertainty of house damages, the repair cost of each damage state is set to a log-normal random variable with μ (mean of logarithmic values) equal to the logarithm of repair cost ratio times replacement cost, i.e., log(repair cost ratio $\times$ replacement cost) and σ (standard deviation of logarithmic values) equal to 0.35. It is assumed that the amount of grants claimed by householder agents and approved by reviewer agents equal to a quarter of the repair cost, with cap of $34,900 USD, which is the maximum amount for IHP declared by FEMA on October 1, 2018 (FEMA, 2018). For criminals with undamaged houses, the requested amount is randomly assigned according to the distribution of building damage states in the whole community, exclusive of non-damaged houses. Table 4-7 lists the other parameters of the base model, and the independent variables to be adjusted in the parametric studies are indicated in bold.

Table 4-6. Percentage distribution of house damage state for designed disaster scenarios

Damage State	None	Slight	Moderate	Extensive	Complete	Total
Level 1	50%	34%	13%	3%	0%	100%
Level 2	25%	27%	31%	13%	4%	100%
Level 3	0%	11%	34%	40%	15%	100%

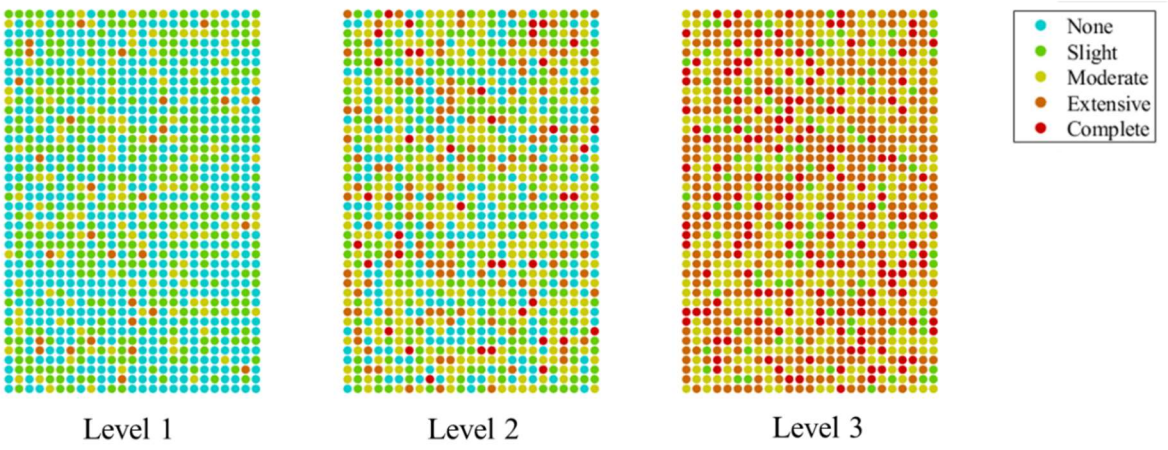


Figure 4-17. Visual representation of house damage states of the disaster scenarios listed in Table 4-6

The effect of disaster-caused demands is quantified considering the damage state of the housing stock and the length of time experienced after the impact without financial support, as expressed by Equation (3-2), where DL = 4, 3, 2, and 1 for complete, extensive, moderate, and slight damage levels of the householder agent's home, respectively. The damage state of houses are randomly assigned based on the distributions in Table 4-6 at the beginning of each Monte Carlo realization. Namely, the damage state of the housing stock is known at the beginning of each realization, and the spatial correlation which exists due to the geo-clustering effect or the intensity of disasters is ignored in the case studies. The effect of the agent's personal experience is expressed by Equation (4-3). The corresponding weights w_{DD}, w_{CC}, w_{CS}, w_E in Equation (3-1) are all set to unity, if no values are specified.

Table 4-7. Parameterization of the model for sensitivity analysis

Parameter	Value
Number of householders	1 million
Number of application reviewers	3000
Number of special investigators	**2000**, 3000, **4000**
Number of applications reviewed per day by per reviewer, mean (standard deviation)	**2(1), 3 (1)**, **4(1),** 5 (1)
Number of applications investigated per day by per investigator	1
Probability of householder completing an application in that day, $\boldsymbol{P_c}$	30% [a], 25% [b], 8% [c]
Initial probability for householders to submit a fraudulent claim, $\boldsymbol{P_i}$, mean (standard deviation)	1% (0.1%)
Probability of application being eligible and complete	95% [a], 75% [b], 60% [c]
Probability of type I error of review (judging a justified claim to be suspicious)	0% [a], 2% [b], 5% [c]
Probability of type II error of review (approving a fraud claim without further investigation)	100% [a], 80% [b], 60% [c] (E1) **75% [a], 60% [b], 45% [c] (E2)** **50% [a], 40% [b], 30% [c] (E3)** **25% [a], 20% [b], 15% [c] (E4)**
Probability of type I error of investigation (judging a justified claim as a fraud)	0%
Probability of type II error of investigation (approving a fraud claim)	0%
The deadline of submitting applications (number of days after the disaster occurred)	426 days (14 months)

4.6.2 Effect of meso/micro impact factors on crime propensity

The influence of the four meso/micro variables is investigated by adopting five different combinations of weights in Equation (3-1): the base case, $W_{Base,}$ and four other cases (W1 to W4), as listed in Table 4-8. Consider the base model subjected to a Level 2 disaster scenario. Figure 4-18 shows the improper payment rate over time for cases with different weights. By comparing the results of case W1 with the ones for case W2 in Equation (3-1), the effect of an agent's personal experience (E) increases with time as more applications are reviewed. Secondly, by comparing the results of W1 with W3 and W4, community cohesion (CC) reduces the fraud rate while the

criminal subculture (CS) has an opposite effect, matching the THC and SDT theories, respectively, used to formulate the model. While influential, varying the weights results in a variation in fraudulent applications that is somewhat modest, i.e. between 6.9% and 10.9% (see Figure 4-18).

Table 4-8. Combination of weights used in Equation (4-1)

Combination	w_{DD}	w_{CC}	w_{CS}	w_E
W_{Base}	1	1	1	1
W1	1	0	0	0
W2	1	0	0	1
W3	1	1	0	0
W4	1	0	1	0

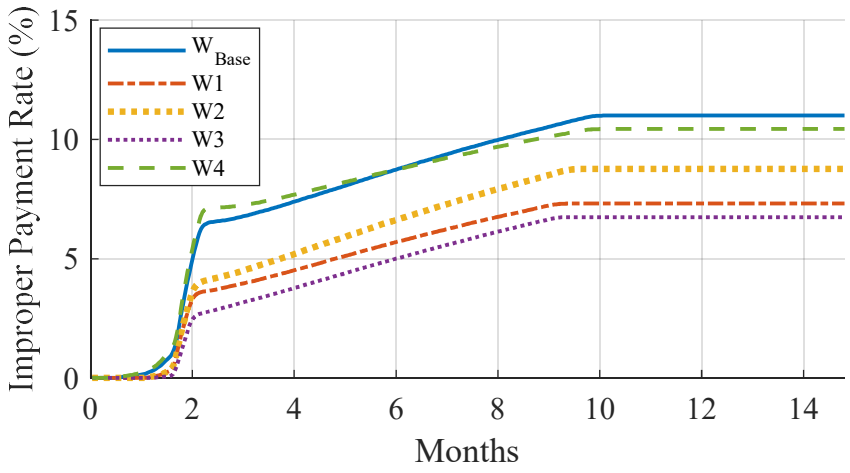


Figure 4-18. Effect of weight values for the key meso and micro variables

4.6.3 Effect of disaster-caused demand

The effects of disaster intensity (see Table 4-6 and Figure 4-17) are shown in Figure 4-19, which depicts the progress of payment distribution and the improper payment rate over time as a

function of disaster demand. It is intuitive that the more severe the disaster, the more loss caused and the more financial assistance will be needed. However, what is not obvious is that it takes longer to distribute the aid (Figure 4-19 and Figure 4-20) and that the total amount of improper payment and the overall improper payment rate become higher (Figure 4-21 and Figure 4-22)as a disaster becomes more severe. This is directly attributed to limitations in staffing, which directly correlate with a reduction in oversight (guardianship) as well as the desire to expedite the distribution of aid.

Figure 4-21 shows that the timing of the increase in the improper payment is slightly different for different levels of disaster. As can be seen in Figure 4-22, the rise of the fraudulent rate in the Level 1 case is earlier than the other two. That is because the effect of *CS* is greater than *DD* in the Level 1 case, and the used *CS* index (Figure 4-8) is higher in the first few weeks. This demonstrates that, with properly calibrated parameters, the proposed model can be used not only to assess the overall fraudulent results, but also to investigate the timing of the increase of improper payments.

Figure 4-22 reproduces the three distinct phases observed in Figure 4-9, although the span for each phases changes with the level of disaster demand. The model suggests that an increasing disaster demand pushes the crime spike later on in time. Those most motivated to seek benefits due to their dire situation apply early. Influenced by the strong community cohesion right after the disaster, the so-called Gemeinschaft feelings, these applications are mostly legitimate and represent Phase I of the process. When the disaster demand is high, Phase I will take longer than when the disaster demand is low because the review speed is assumed constant. As the opportunity for fraud becomes more broadly recognized and as the Gemeinschaft feelings abate, fraudulent applications for undamaged houses or duplicate claims after a first successful claim start coming

in, pushing the fraud sharply upwards (Phase II). The rate then flattens out as the guardianship efforts kicks in (Phase III).

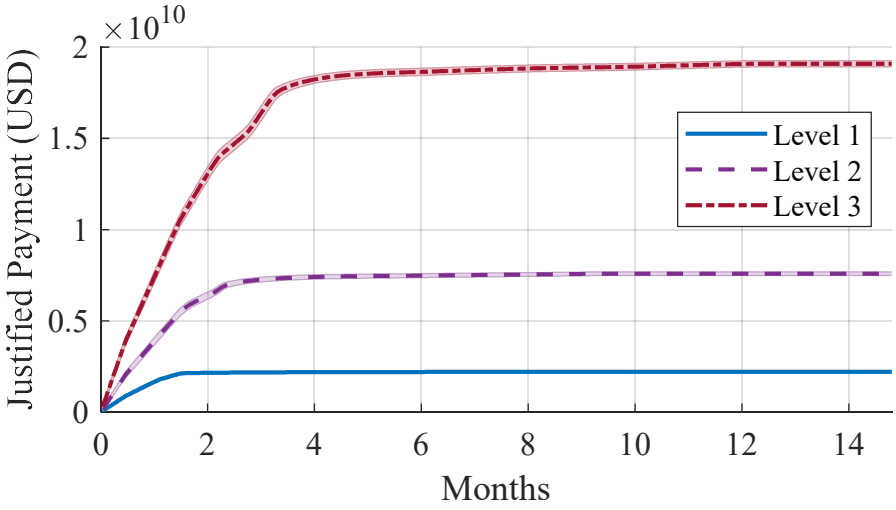


Figure 4-19. Effect of disaster-caused demand on justified payment

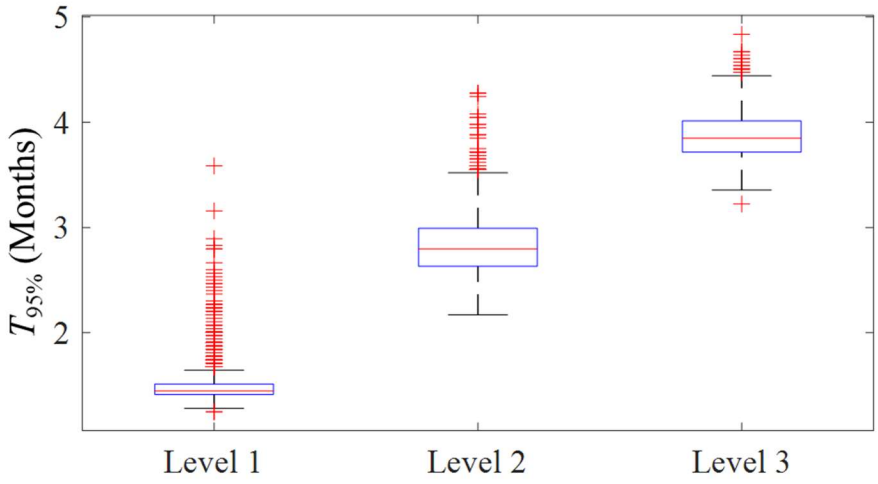


Figure 4-20. Effect of disaster-caused demand on $T_{95\%}$

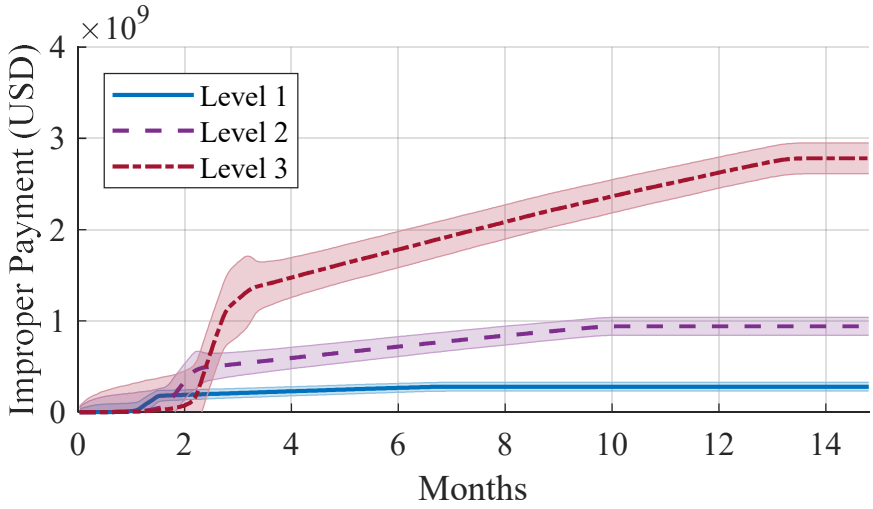


Figure 4-21. Effect of disaster-caused demand on improper payment

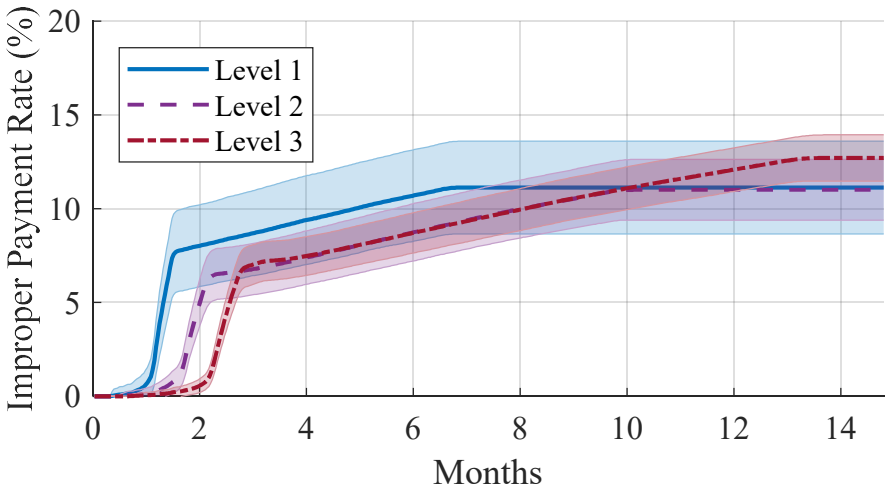


Figure 4-22. Effect of disaster-caused demand on improper payment rate

4.6.4 Effect of community cohesion

In the previous control experiment and parametric studies, the community cohesion effect is assumed as the time-dependent function plotted in Figure 4-8 to represent the cooperative behavior of the community in the early stages of the disaster. However, how long such gemeinschaft feelings last is still a question desired to be answer in social science and disaster

studies. To model different levels of community cohesion effect, normal random variables with the time-varying mean values in Figure 4-23 and standard deviation of 0.25 are employed as *CC* indices. Figure 4-23(a) shows the *CC* indices with the same maximum absolute value but changing differently along time, where CC1 is the case that the effect of community cohesion last shortest, and CC4 the longest. Besides the duration, Figure 4-23(b) consider the impact degree of *CC* effect, where the larger absolute value of CC_t indicate the larger negative effect of crime propensity, i.e., lower probability to commit crimes. Figure 4-24 (a) and (b) show the changing of improper payment rates from Day 1 to Day 450 (15 weeks) corresponding to the CC indices in Figure 4-23 (a) and (b), respectively, and the other parameters are kept unchanged. It is found that higher community cohesion could help to reduce the fraud rate in the long run.

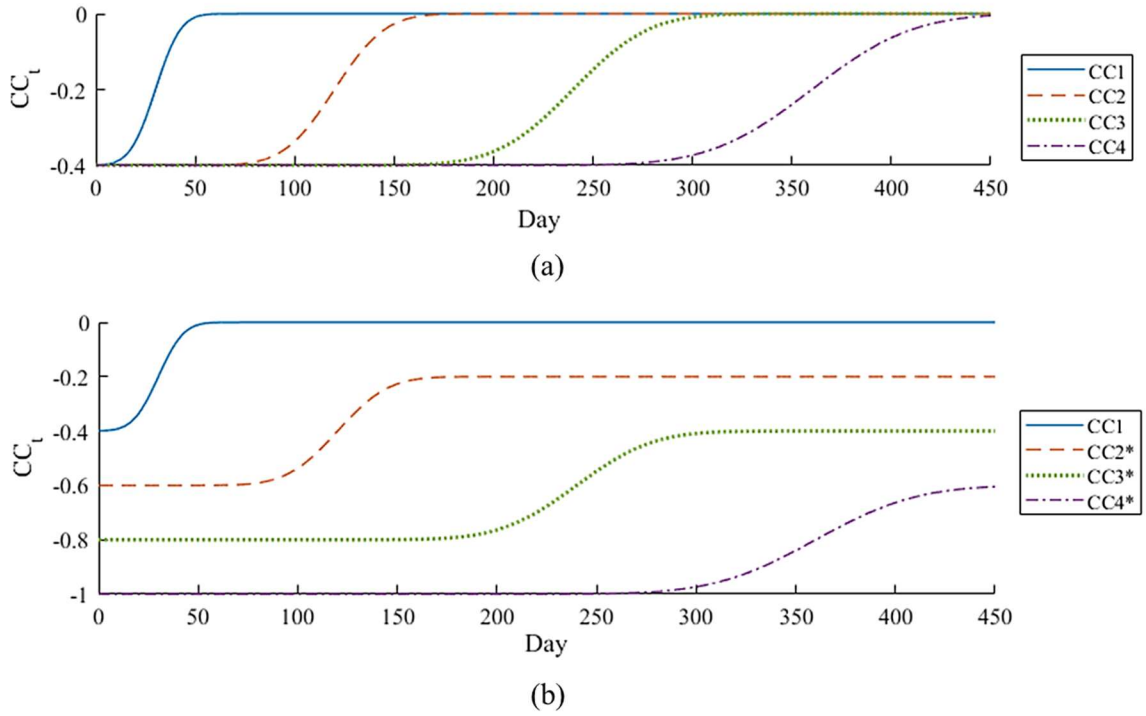
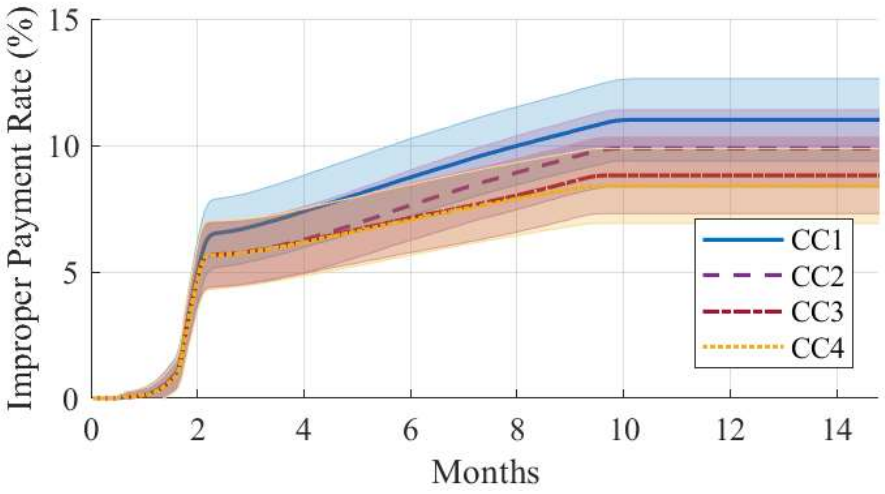
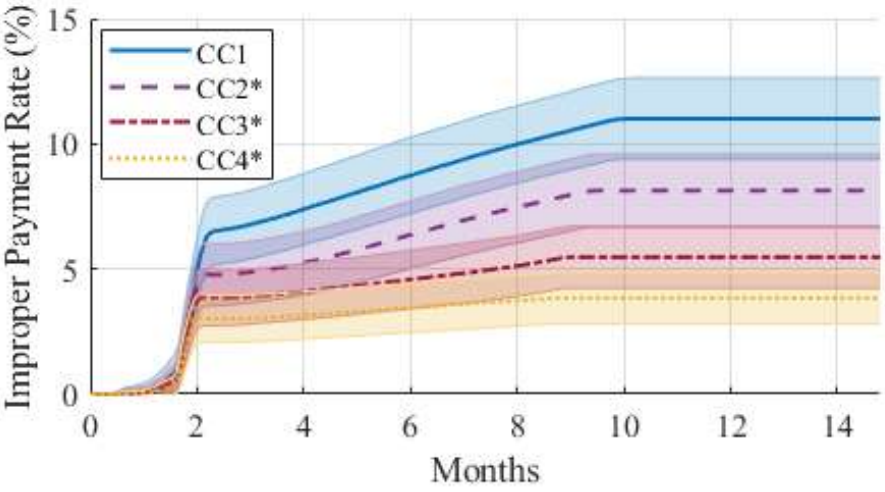


Figure 4-23. Time varying functions for different levels of community cohesion effect



(a)



(b)

Figure 4-24. Improper payment rates under different levels of community cohesion effect corresponding to Figure 4-23 (a) and (b), respectively

4.6.5 Effect of guardian strength

The effect of using more government special investigators (2000, 3000, and 4000) is investigated to determine the influence of guardian strength in a Level 2 disaster scenario. The

distribution of justified payment is plotted in Figure 4-25 for this example. Since the number of investigators does not affect the speed of application review, the progress and time required for distributing the justified aid is barely affected, as shown in Figure 4-26. From Figure 4-27, the increase in the amount of improper payments can be slowed down if the fraudulent behaviors are detected earlier. The final improper payment rate can also be reduced, as shown in Figure 4-28.

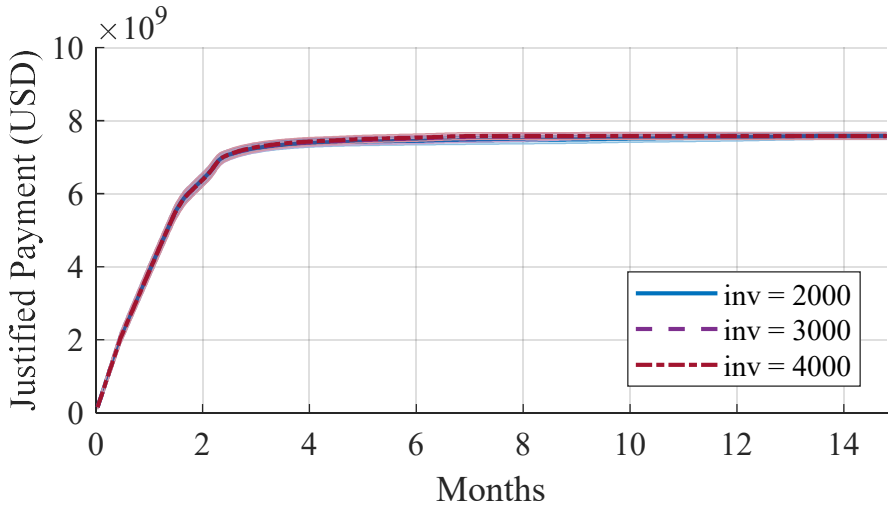


Figure 4-25. Effect of guardian strength on justified payment

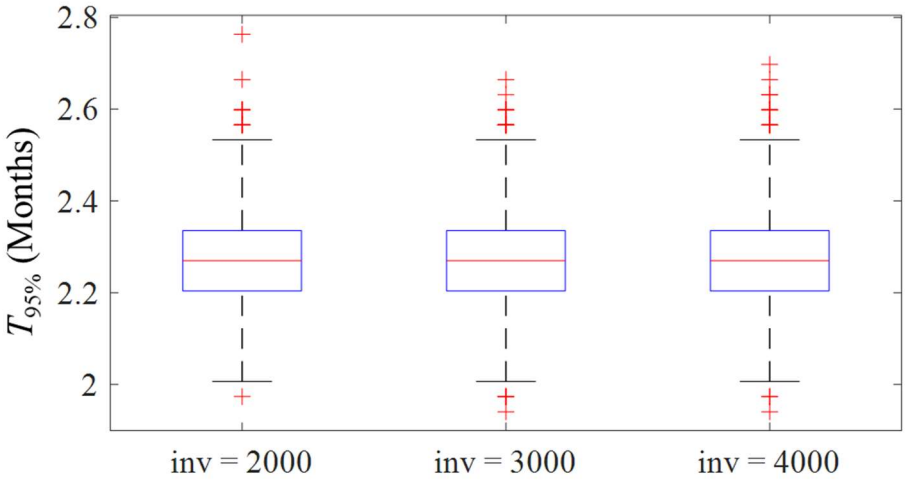


Figure 4-26. Effect of guardian strength on $T_{95\%}$

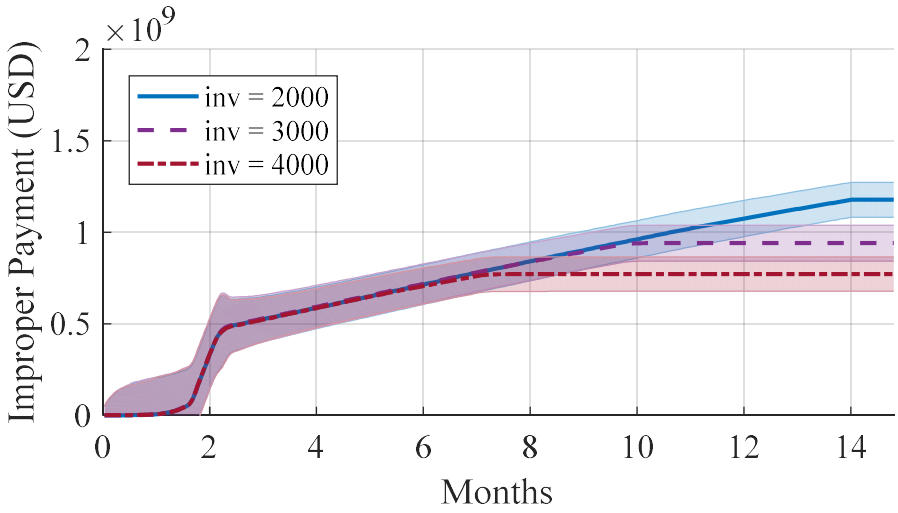


Figure 4-27. Effect of guardian strength on improper payment

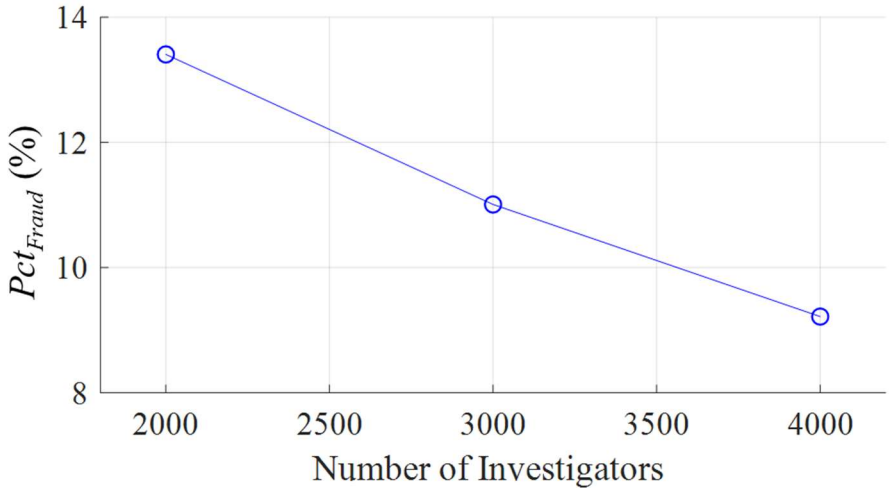


Figure 4-28. Effect of guardian strength on Pct_{Fraud}

4.6.6 Effect of target vulnerability

The effect of target vulnerability is evaluated by considering different probabilities of Type II review error, i.e., E1, E2, E3, and E4 listed in Table 4-7, where E1 indicates the highest probability of review errors and E4 is the lowest. As can be seen in Figure 4-29 and Figure 4-30, because the speed of review and the other parameters are kept unchanged, the progress and the

time required for all justified assistance being paid are close for these four cases. However, the improper payment rate is significantly reduced with the decrease of review error, as shown in Figure 4-31 and Figure 4-32. Reducing the vulnerability of targets is an effective way to decrease the chance of success of fraudulent behaviors, but it might not be easy to achieve. For example, in order to decrease the review error, additional specialized training of reviewers must be completed and a more comprehensive standard reviewing procedure has to be established. Moreover, this is a first order sensitivity analysis, and only one independent variable is adjusted in the experiment. The interactions between different parameters are not considered, e.g., a stricter review of an application could reduce the errors, but it may take more time to review the same application.

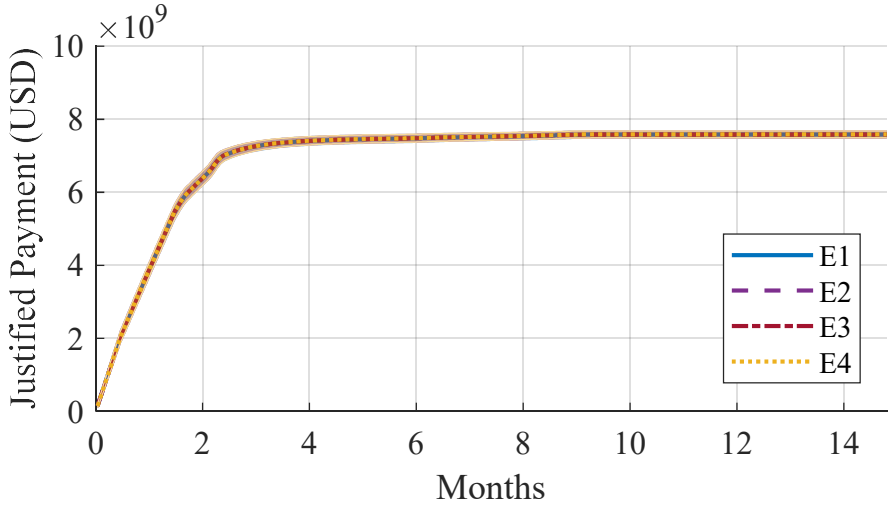


Figure 4-29. Effect of target vulnerability on justified payment

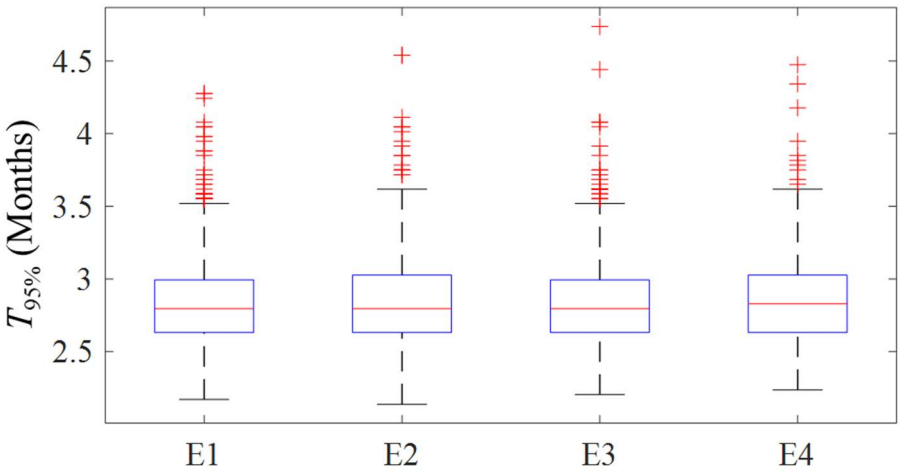


Figure 4-30. Effect of target vulnerability on $T_{95\%}$

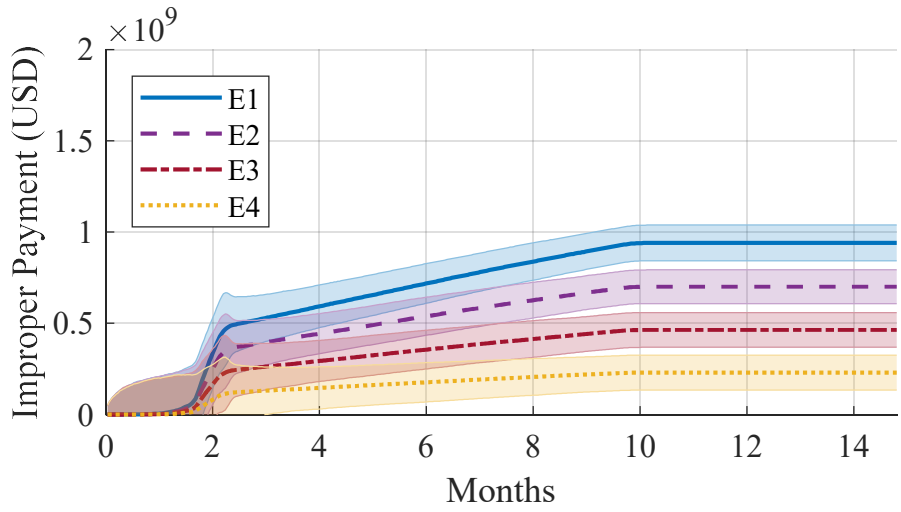


Figure 4-31. Effect of target vulnerability on improper payment

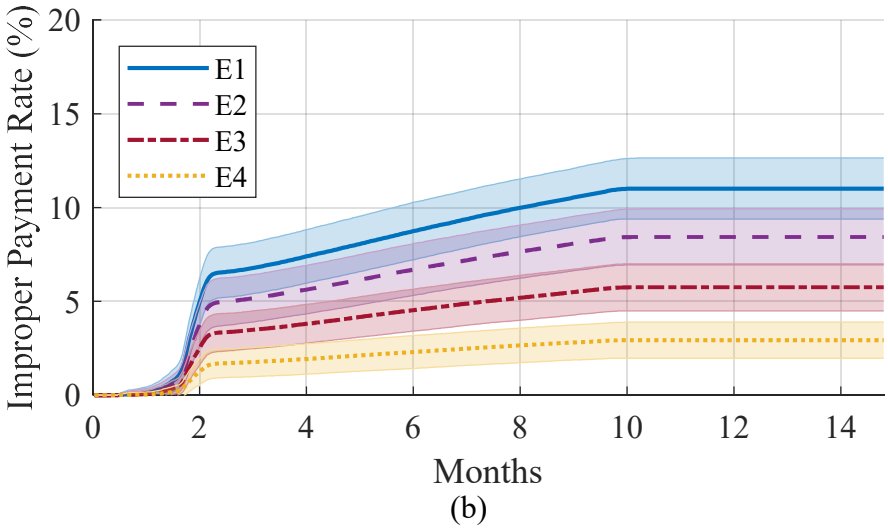


Figure 4-32. Effect of target vulnerability on improper payment rate

4.6.7 Effect of review speed

The effect of review speed (number of applications reviewed by a single reviewer) is investigated via the base model subjected to a Level 2 disaster. The number of applications handled by a single person is considered to be 2, 3, 4, and 5 with a standard deviation of 1. The other parameters are kept the same as the base model shown in Table 4-7. Figure 4-33 shows the distribution of justified payment over time, and Figure 4-34 shows the estimated $T_{95\%}$. It is clear that the faster the applications are reviewed the more payments could be made in the first few months, but the overall improper payment increases slightly with the increase in the review speed, as shown in Figure 4-35 and Figure 4-36. This implies that distributing the financial assistance to the affected people faster will help the community recover faster, but it must be balanced with the anticipated increase in the cost associated with improper payments. Figure 4-35 also suggest that a faster review speed contributes to shorter Phase I and more pronounced Phase II. This is because

the opportunity for fraudulent duplicate applications occurs earlier as fraudsters, encouraged by the quick release of funds, submit more duplicate applications.

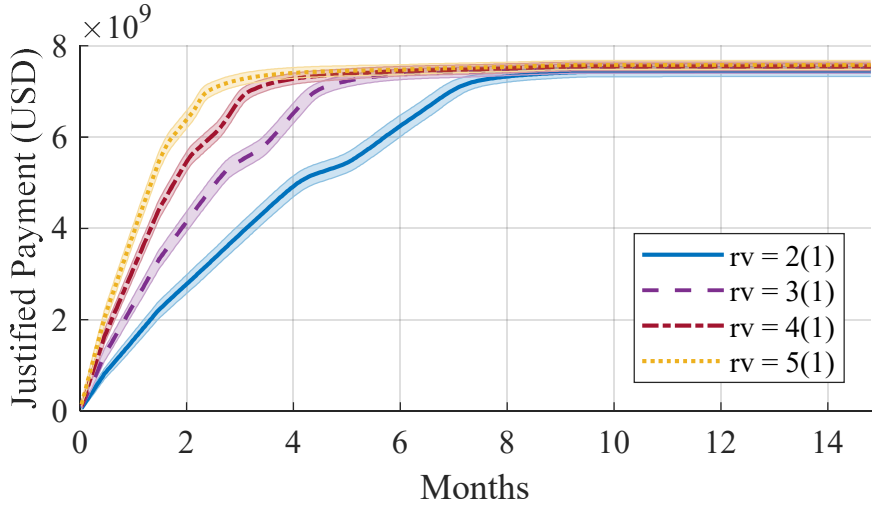


Figure 4-33. Effect of review speed on justified payment

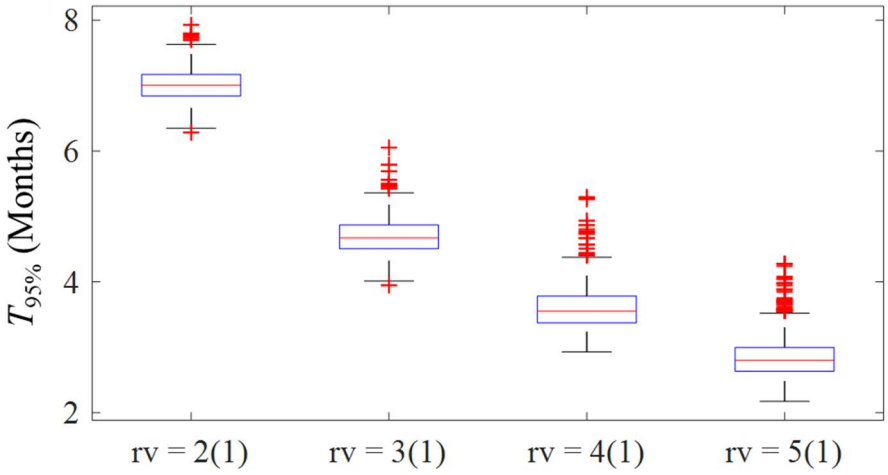


Figure 4-34. Effect of review speed on $T_{95\%}$

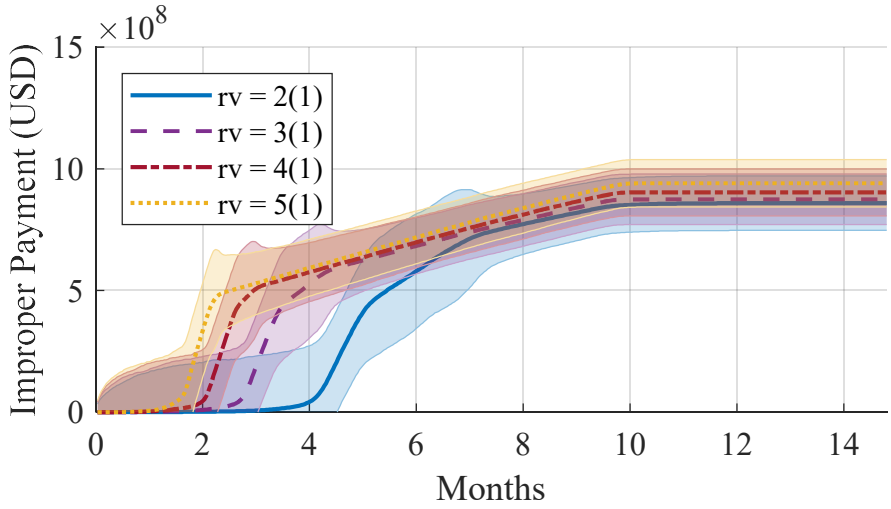


Figure 4-35. Effect of review speed on improper payment

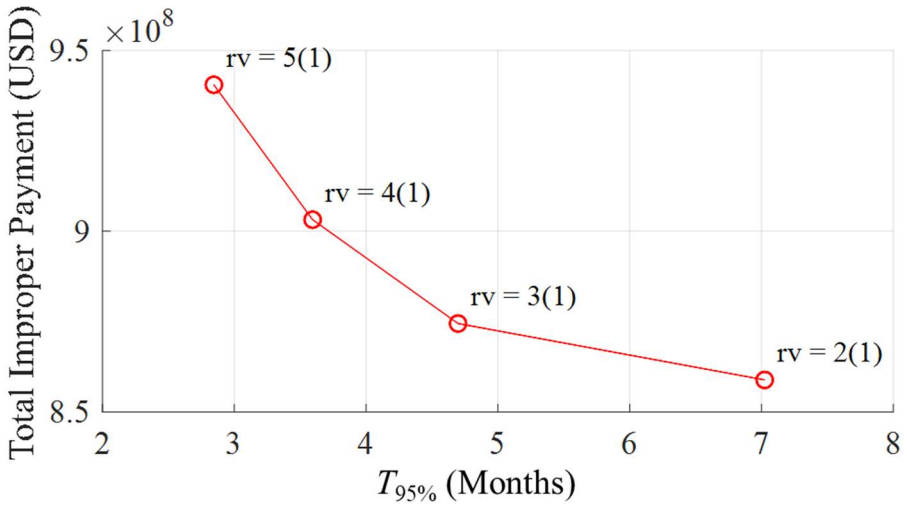


Figure 4-36. Effect of review speed on overall improper payment and $T_{95\%}$

4.7 Conclusions

A computational tool was proposed to simulate the dynamic process of criminal activities, specifically benefit fraud, in the wake of disasters. The formulation, which uses an agent-based simulation, considers observed phenomena, relies on established criminological theories and

addresses disaster demands and social characteristic at both the micro- and meso-levels. Statistical data from FEMA's Individuals and Households Program (IHP) for Hurricanes Katrina and Rita was used to calibrate the simulation model and to show that it can capture the basic features of benefit fraud while producing sensible results.

The computational results show that fraudulent activity occurs in three distinct phases. In Phase I, where most of the applications come from people who are in a dire situation and the so-called Gemeinschaft feelings are still strong, the rate of fraudulent applications is small and increases gently. As the opportunity for fraud become more broadly recognized and encouraged by the release of an initial round of funding, fraudsters become more active, leading to a sharp spike in the improper payment rate in Phase II. In Phase III, the guardianship mechanisms kick in and the improper payment rate plateaus out. The observed phases match in spirit observations in actual disasters.

The effects of community cohesion, criminal subculture, disaster demands and a criminal's personal experience on crime propensity are examined in light of extensive parametric simulations. The results of the analysis show that increasing the accuracy of review, decreasing disaster demands, and enhancing guardian strength can help lessen the loss due to post-disaster benefit fraud. However, improving the accuracy of review may lead to an increase in review time, i.e. there is a trade-off between different objectives, quantified in this research. A higher-order sensitivity analysis is needed in future studies to find the optimal balance between the loss of fraudulent payments and the speed of aid distribution for improving the overall resilience of communities.

This study addresses an existing research gap regarding post-disaster benefit frauds in the context of long-term community reconstruction and recovery. The proposed agent-based methodology provides the ability to model post-disaster crime in great detail. For example, as shown in this study, it is possible to model the spatial and temporal nature of the criminal process and its interactions with the unfolding disaster and subsequent recovery efforts. While experience with previous disasters and addressing logical gaps in the process can help improve in the aid-disbursement process, simulations of the sort presented here offer another systematic and powerful way to achieve this.

Due to lack of information about a number of key variables, which were assumed out of necessity, the simulation results are not truly predictive. As such, one of the key aspects of this study is that it provides insights into the type of information and its level of detail that should be sought by researchers and aid officials in future disasters. In the current multi-agent model, benefit fraud is only captured in terms of repeatedly submitting application or submitting application when the property is not damaged. It is an important future research to include different behavioral pattern of benefit fraud, e.g., inflating damaged property, frauds committed by criminals outside of the affected areas, identity theft, or syndicated criminal activities, for better understanding the mechanisms of benefit fraud in the aftermath of disasters.

Moreover, the proposed agent-based model can be linked and cooperate with the other simulators through the distributed computing platform described in the previous chapters for investigating more complex socio-technical interactive behaviors. For example, it can be connected to a physical recovery simulator to consider the dynamic effect of recovery efforts on householders' disaster-caused demands. On the other hand, the status of receiving financial aid may affect the householders' decision of repairing their house, rebuilding, or relocating, which

affects the reconstruction plans, the population distribution in the region, and even the local economy. All of these are potential research topics in community resilience and can be studied in the future by integrating the proposed ABM with the other discipline specific computational models.

CHAPTER 5

Simple Run-Time Infrastructure (SRTI) for Distributed Simulation of Community Resilience

5.1 General

This chapter introduces a new, scalable, versatile, and user-friendly distributed computing software solution, termed Simple Run-Time Infrastructure (SRTI), for simulation of community resilience. The motivations, target users and development history of SRTI are provided (Section 5.2). The initial version of SRTI, codenamed v1.00.00, is described from the high-level architecture to its fundamental components (Section 5). Then, an alternative version of SRTI (codenamed v2.00.00) that aims to reduce the usage barrier and enable a plug-and-play approach to composing simulations is introduced (Section 5.3.2). The case study of wind-excited buildings suffering progressive damage in Chapter 2 is repeated using SRTI v1.00.00 to demonstrate the capabilities and compare its performance to LCM (Section 5.4). Also, a cross-platform resilience analysis of a lifeline system subjected to an earthquake is built using a graphical user interface (GUI) and executed through the SRTI v2.00.00 to show its scalability and flexibility (Section 5.5). Lastly, the time management and user experience of different versions of SRTI are discussed. The limitations and potential features to develop in the future are outlined (Section 5.6).

5.2 Background and History of SRTI

A review of the existing distributed simulation standards and tools was provided in Chapter 2. All existing platforms require extensive user knowledge in order to operate them and therefore offer a high barrier to entry for non-specialized users. In this Chapter, a new distributed computational solution, termed Simple Run-Time Infrastructure (SRTI), is proposed to address the limitations of existing platforms. SRTI permits natural hazards simulations to be modifiable, extensible and scalable. It is designed to transfer any type of information between simulators thereby maximizing its generality. By not requiring recompilation when new simulators are added or data format changes, SRTI is geared to decrease users' effort to get different simulators to interact in a meaningful manner so that users can create wide ranging, large-scale simulations using a variety of available and user developed simulators of varying fidelities. The data communication in the SRTI employs a middleware for data exchange and uses a publish-subscribe pattern. The tool is available as a pre-compiled application for quick deployment and integration. Treated as a black box accessed through API function calls, a user can edit the open-source code or rewrite components to support additional programming languages. Since the SRTI is primarily a tool for sharing data, it can also be used for applications outside of simulation, including Internet of Things (IoT) devices, remote sensing and smart cities.

The concept generation and technical development of the SRTI were conducted under the auspices of a project funded by the University of Michigan and the US National Science Foundation: Interdependencies in Community Resilience (ICoR) (ICoR, 2019) since 2016. The initial version of the SRTI, designated v1.00.00, was purely a data-transmission system based on the JSON format and did not apply mandatory rules on simulators for time management. While highly flexible for composing simulations, this approach required users to embed time-related

information within the transmitted messages. This made the barrier for use of v1.00.00 relatively high, as users needed to be familiar with the inner workings of the platform.

To lower the entry barrier and enable an easy to use, plug-and-play approach for composing simulations, a new version of the SRTI that is fundamentally different in its approach is developed. The new version is code named, v2.00.00, and designed to have substantially higher functionality by supporting time-dependent simulations, for which time synchronization between multiple simulators and phase changes are indispensable.

Unlike the earlier version of SRTI which required users to add code into their simulators to allow them to access SRTI, the new version simply requires users to prepare a configuration file, which can be edited in any common ASCII text editor, for providing the definition of simulators, messages, and their publish/subscribe relationships. An optional application, the SRTI graphical user interface (GUI) is provided to reduce a user's effort and time for constructing such configuration files. By using the SRTI GUI, users can describe their simulator system through an interactive graphic interface and export the configuration files according to their design. It also allows users to launch and execute the RTI Server and simulators through the same interface. Moreover, the SRTI GUI helps users inspect the content of messages and system timestep during the execution of the simulation.

SRTI is open-source and free to use. The full source code, pre-compiled libraries, documentation, and other information are available at SRTI (2019). The following sections describe in detail the high-level architecture of different version of the SRTI, their essential components, and data structure. Implementation examples are provided. In addition to the user experiences, a summary of limitations and potential features to develop in the future are discussed in the end.

5.3 Introduction of SRTI

5.3.1 SRTI v1.00.00

As shown in Figure 2-2, the initial version of SRTI consists of three components: 1) RTI Server (middleware), 2) RTI Lib API, and 3) clients (simulators provided by the user).

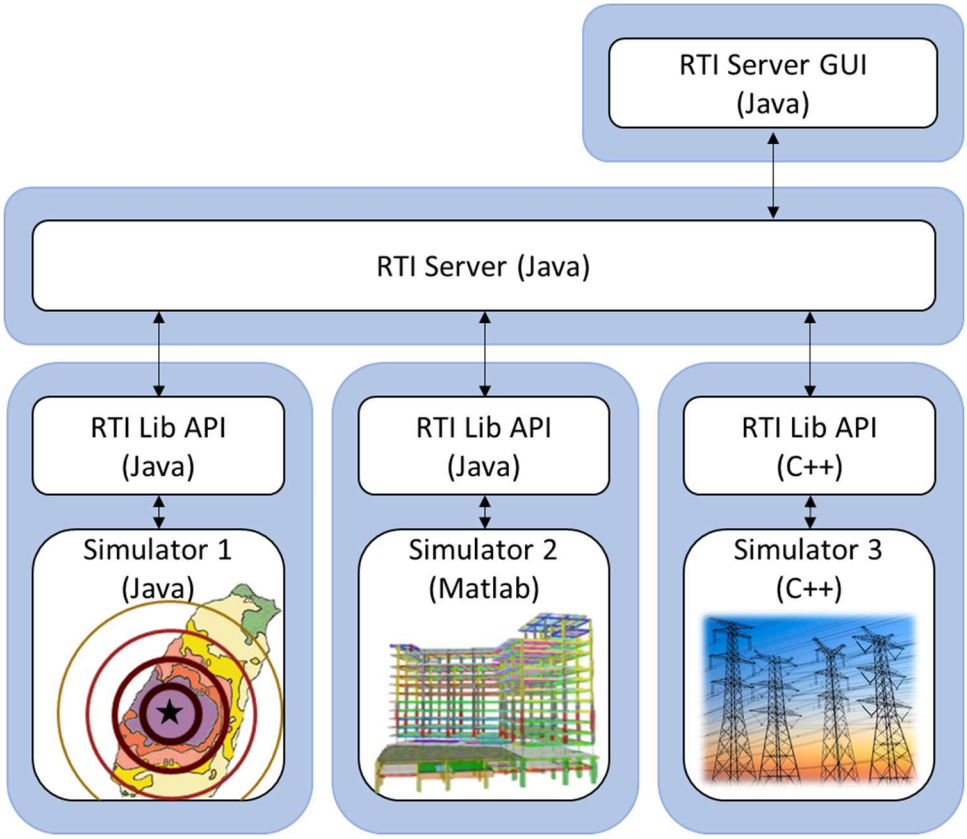


Figure 5-1. High-level structure of the SRTI v1.00.00

The RTI Server is a server-side application that manages the deployment of messages throughout the SRTI system. All simulators in the SRTI system must connect to the RTI Server for sending or receiving messages. These connections use the transmission control protocol (TCP) "socket" communication. The pre-compiled RTI Server is written in Java to make it easily portable

and compatible with different operating systems, and it can be run either on the same machine or a separate machine from simulators in a system. Once launched, the RTI Server opens a server socket and displays its “hostname” and “port number” for the user to identify it. Then, a simulator can connect to the RTI Server based on the associated hostname and port number.

The RTI Server GUI is an optional graphical user interface that displays the information of the RTI Server, such as hostname, port number and a list of connected simulators. The RTI Server GUI acts as a simple simulator that automatically connects to the RTI Server, and subscribes to every message, but publishes nothing. Therefore, users can inspect a live feed of connected simulations and the data messages presently received by the RTI Server. Figure 3-2 is an example of the RTI Server GUI.

After connecting to the Server, a simulator must notify the RTI Server regarding what messages (by name) it wants to subscribe to for performing the “publish-subscribe” message distribution. A proprietary SRTI message is sent through the RTI Lib API to the RTI Server to manage the subscription. The RTI Lib API is a pre-compiled library of functions used to assist and manage the interactions between the RTI Server and simulators. A Java and C++ version of the RTI Lib API have been prepared to be used instantly with simulators written in a compatible language.

The RTI Lib API currently provides over fifty functions to assist in connecting to the server, publishing/subscribing messages, and parsing data. A simulator may need lines of code to be added to call upon specific functions available in the RTI Lib API, which may affect its design. A complete list of the available APIs and detailed sequence diagrams describing the order of events between the RTI Server and the RTI Lib API is available from the SRTI Developer Documentation (SRTI, 2019).

Once told about the subscription, the messages received by the RTI Server for a given name will be sent to the simulator through its RTI Lib API instance. When a simulator publishes a message, it sends the message through the socket to the RTI Server. The RTI Server immediately checks the name of the message and forwards the message to all simulators subscribing to that message. In the SRTI v1.00.00, the RTI Server does not have control of who joins the simulation system or when they execute. By letting the RTI Server take charge of message deployment, each simulator can be independent in order to attain a plug-and-play system design.

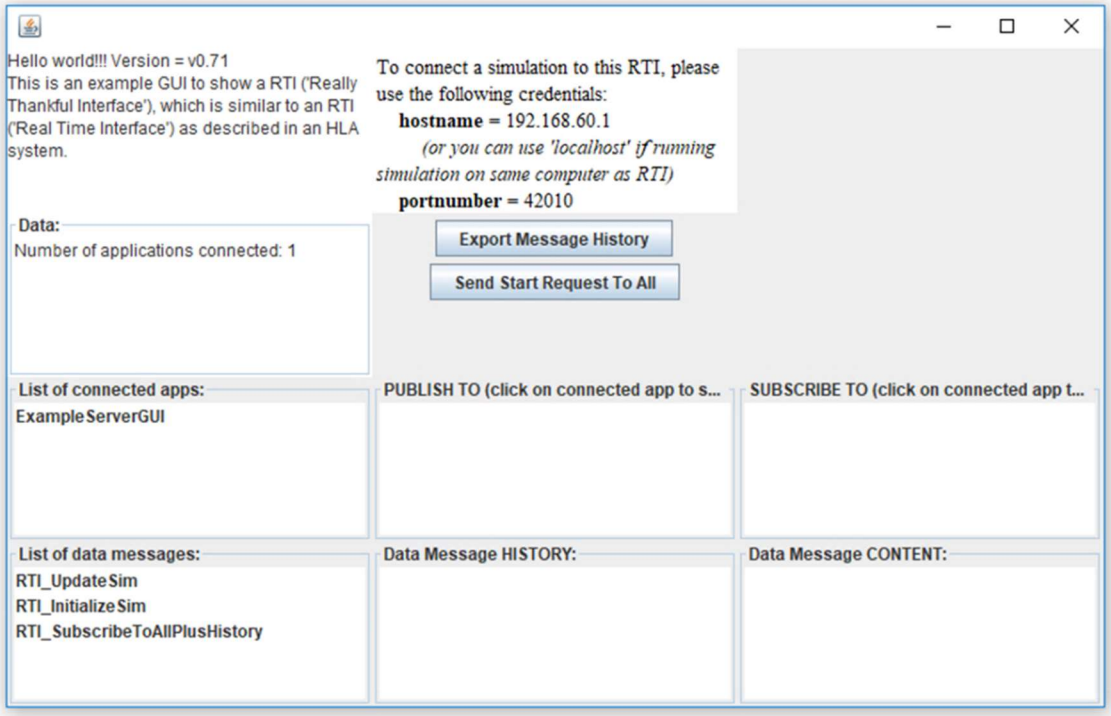


Figure 5-2. Example of the RTI Server GUI

5.3.2 SRTI v2.00.00

As discussed above, the SRTI v1.00.00 is a flexible and simple solution to pass data between different programs, but it requires users to embed time-related information within the

transmitted messages and use RTI Lib API functions for composing simulations. To lower the entry barrier and reduce the user programming efforts, an alternative version of the SRTI that is built upon v1.00.00's foundation was developed. The new version, code named, v2.00.00, was designed with the purpose of managing simulator actions and time synchronization between multiple simulators inside a larger complex system, while maintaining user accessibility to easily replace or edit components of the system.

A high-level view of the architecture of the SRTI v2.00.00 is shown in Figure 4-7. To facilitate referral to the software, the version (v2.00.00) will be dropped in subsequent discussions. In addition to the RTI Server and RTI Lib API, the pre-compiled components added in the SRTI include the RTI Management Server, RTI Wrapper, and GUI (optional).

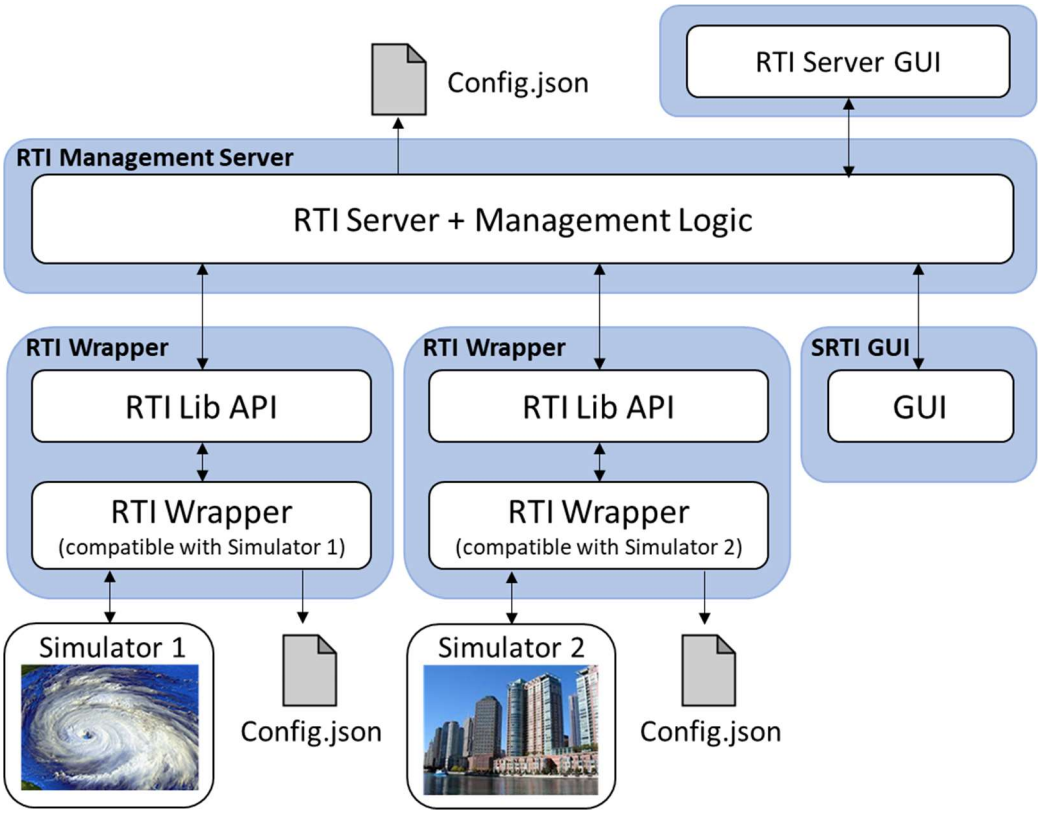


Figure 5-3. A high-level view of the architecture of the SRTI v2.00.00

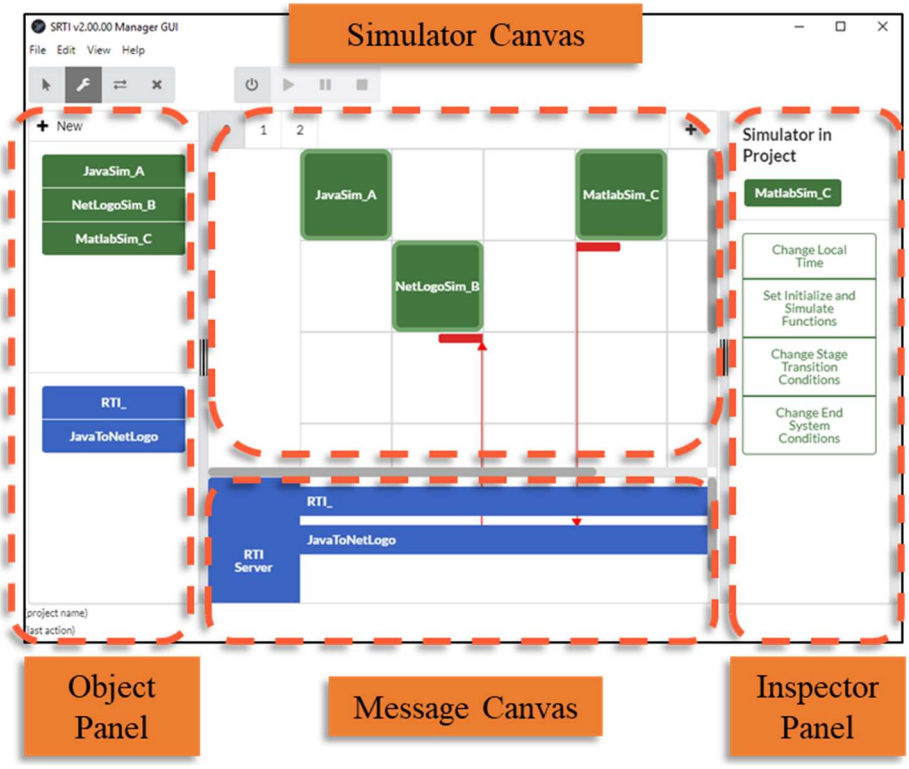
SRTI v2.00.00 employs the same socket connection and publish-subscribe data transmission pattern as v1.00.00. The difference is that the RTI Management Server and the RTI Wrapper will manage the system and convey data in the expected order based on specific key-value pairs embedded in each message. These internal key-value pairs, including "timestamp" (the computer system's time when the message was sent), "vTimestamp" (the virtual timestep for the simulation system), and "source" (the name of the simulator it came from), are clearly defined in the SRTI Developer Documentation (SRTI, 2019).

The RTI Management Server keeps track of an internal virtual timestep that is shared in each message. Once the RTI Management Server receives a "finish" confirmation from every active simulator, the virtual timestep is increased by one, and the Server sends a message to request to each RTI Wrapper to "start" the next step. This process repeats, preventing any single simulator from proceeding to the next step too early, while allowing for some computational parallelization where the user deems it safe.

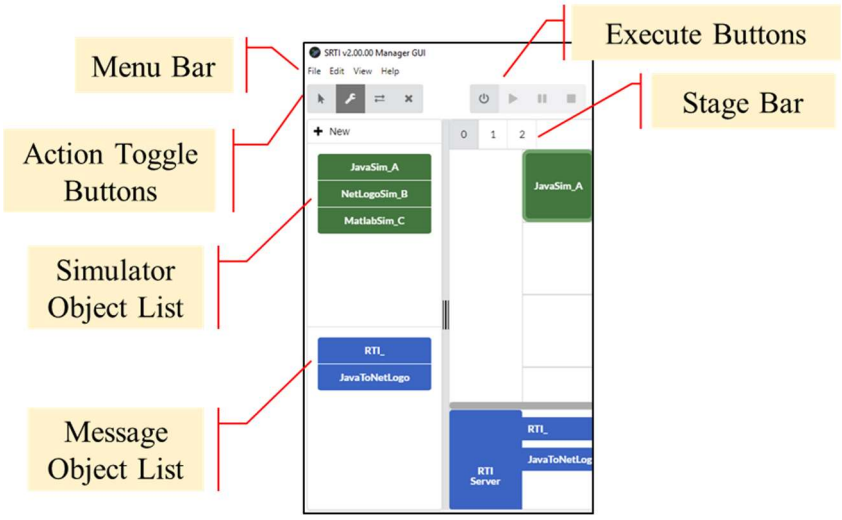
The RTI Wrapper handles both the connection to the RTI Management Server, and to the user's simulator code itself. It reads and writes variable data from the simulator during execution and shares it with the larger simulation system. At the beginning of the RTI Wrapper's execution, it will read a configuration file, which defines the relevant variables and functions inside the user's simulator, as input. The configuration file is saved in JSON format, and consists of a large set of parameters that can be defined by the user. A complete list of these parameters including value type and definition can be found in the documentation on the SRTI GitHub site (SRTI, 2019).

The configuration file can be edited by hand in a text editor, or through the SRTI GUI. The SRTI GUI is a more advanced version of the interface with two primary functions: to help define the configuration file for each RTI Wrapper, and to allow execution of the simulation system from

a single place (within the GUI itself). An overview and important details of the SRTI GUI are shown in Figure 5-4. Users may need to study the documentation of the SRTI (SRTI, 2019) carefully to understand what properties exist and how to define them.



(a)



(b)

Figure 5-4. Screenshot of simple project inside the SRTI GUI: (a) overview; (b) important details

5.3.3 Data structure

One important objective of the SRTI is providing the freedom and flexibility of message definition in simulations. To reduce data size and speed up the transmission, some other distributed simulation solutions encode a message as byte code, but this requires defining the data format in advance and compiling it as a specific data package of the data-transfer software to handle it. For the SRTI, there are fewer limitations. From the user's perspective, a simulator generally only needs to provide a "message name" and "message content," each represented in String format (ASCII text). When sent over the network to the RTI Server, the entire message (including name and content) is converted as a single String through the RTI Lib API. Because the message is encoded as a String, no recompiling of the SRTI is required to drive a simulation system. SRTI's implementation may be expected to be slower in comparison with the ones using byte values, but its versatility without relying on complex compiling instructions is a significant advantage.

Each message, hidden from the user, is sent as a JSON object within the SRTI system. The SRTI adds the following elements to the top level of the JSON object: "name" (the name of the message), "content" (data the simulators share and receive), as well as system "timestamp", "vTimestamp", and "source", which have been described before. Aside from "name" and "content," any other values are proprietary to the internal function of the SRTI and are typically not the responsibility of the user, but these elements can be accessed by the user's simulator if deemed necessary.

Although the RTI Lib API provides many useful functions for converting data format, the responsibility to parse and generate "content" is entirely given to the simulators, providing the most freedom possible, as well as improved ease-of-use regarding message transfer. It is suggested to use the JSON format for "content," but it is not a strict requirement. The exact content format

must be known by simulators in order to parse and utilize the data and must be convertible to "String" format. With these two rules met, the "content" can be in any format a simulator can output. Such design of data structure is a major factor allowing the use of the SRTI without recompiling when new data formats are required.

5.4 Implementation of SRTI v1.00.00: Application to Wind Engineering

More implementation details and user experiences for different versions of the SRTI are demonstrated in the following sections. First, the case study of wind-excited buildings suffering progressive damage in Chapter 2 is repeated using the SRTI v1.00.00 to demonstrate the capabilities and compare its performance to LCM. Then, a cross-platform resilience analysis of a lifeline system subjected to an earthquake is modified from the framework in Chapter 3 and executed through the SRTI v2.00.00 using the GUI to show its scalability and flexibility.

5.4.1 Problem description

The same case study of wind-excited buildings suffering progressive damage in Chapter 2 is conducted using the SRTI v1.00.00 to demonstrate its capabilities and computational performance. In this example, each simulator representing one part of the wind disaster event is written in MATLAB and run on a separate processor. The difference between both simulations is the method of connecting the simulators. In Chapter 2, the data transmission was done through LCM, whereas the SRTI v1.00.00 is used to conduct the same simulation herein. Comparing to the data flow of the illustrative example in Chapter 2 (Figure 2-2), Figure 4-5 shows the SRTI architecture of the case study. The detail descriptions of simulators can be found in Chapter 2. The messages published or subscribed to by each simulator are listed in Table 2-2, and an example

content of a *StoryDisplacement* message, *D(t),* is shown in Table 5-1. There are two variables, *step* and *step_time*, included in every message to indicate the current step and simulation time of the message. Example Matlab code of the *Structural Analysis Simulator* is shown in Figure 5-6 to illustrate the use of the RTI Lib API functions.

To compare the results between the SRTI and LCM implementations, an identical wind speed history with a 3,927 sec duration and a sampling frequency of 1.27 Hz is then applied to a group of prototype buildings. The affected buildings all have the same rectangular building plan but different heights and orientations, as shown in Figure 2-6. Special moment resisting steel frames are used for the lateral resistance, and the location and the member sizes of the frames can be found in Figure 2-5. The solution is incremental as done in Chapter 2.

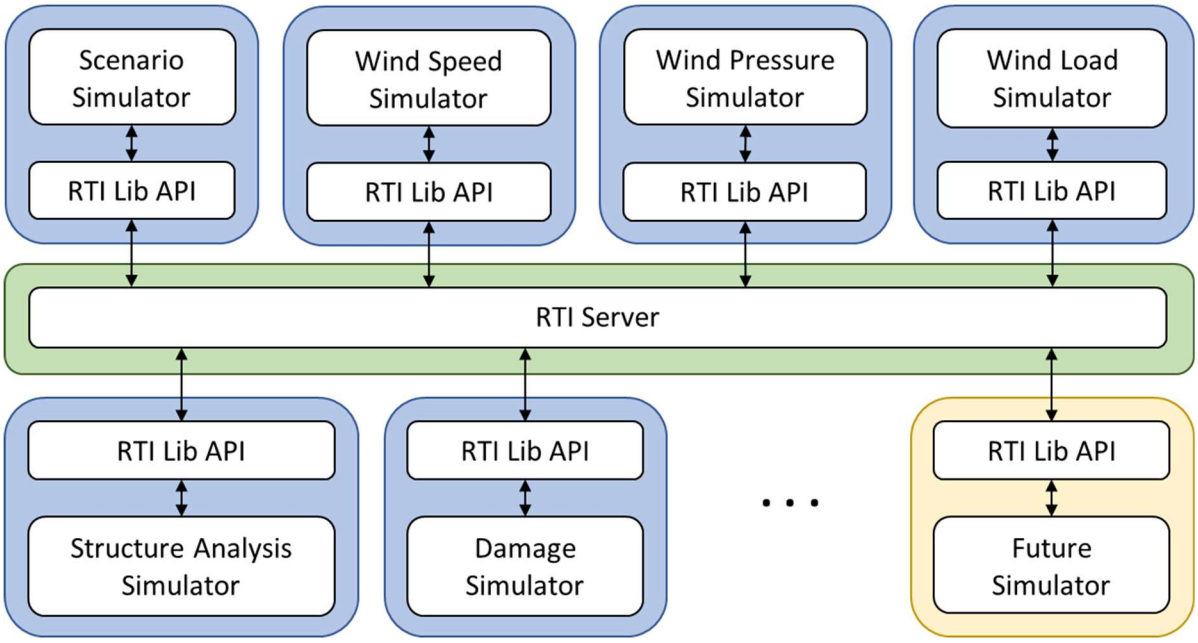


Figure 5-5. Architecture of the same case study in Chapter 2 using the SRTI

Table 5-1. Example content of StoryDisplacement message

StoryDisplacement: *D*(*t*)		
Variable	Data Type(*)	Description
step	int	Index for the current step
step_time	double	Simulation time of the current step
nbuildings	int	Total number of buildings
max_story	int	Number of stories of the tallest building
story_disps	double (*max_story*, *nbuildings*)	A matrix containing story displacements of all buildings

*: where (m, n) describes the dimensions of a matrix if needed

```
%======== Add the pre-compiled SRTI file to the MATLAB classpath==========
javaaddpath SRTI_v1_00_00.jar;
import mainServer.*;

%======== Connect to the SRTI Server ====================================
rtiLib = RTILib();
hostname = "192.168.60.1";
portnum = "4200";
rtiLib.setSimName("StructureAnalysis");
rtiLib.connect(hostname, portnum);

%======== Subscribe messages ============================================
rtiLib.subscribeTo("Scenario");
rtiLib.subscribeTo("WindLoad");

%======== Receive Subscribed Messages ===================================
msg_scenario = rtiLib.getNextMessage("Scenario");
m_sce = rtiLib.getMessageContent(msg_scenario);
msg_windload = rtiLib.getNextMessage("WindLoad");
m_wl = rtiLib.getMessageContent(msg_windload);

%======= decode messages from Json format ===============================
nbuildings = int32(str2double(rtiLib.getJsonObject("nbuildings",m_sce)));
buildinginfo = str2num(rtiLib.getJsonObject("buildinginfo",m_sce));
step = int32(str2double(rtiLib.getJsonObject("gstep",m_wl)));
windloads = str2num(rtiLib.getJsonObject("gloads",m_wl));


%======= Calculate displacement =========================================
displacements = StruAnaModel(nbuildings,buildinginfo,step,windloads);

%======= Encode messages to Json format =================================
msg = rtiLib.setJsonObject("", "step", rtiLib.getJsonObject("step",m_wl));
msg = rtiLib.setJsonObject(msg, "step_time", rtiLib.getJsonObject("step_time",m_wl));
msg = rtiLib.setJsonObject(msg, "nbuildings",
rtiLib.getJsonObject("nbuildings",m_sce));
msg = rtiLib.setJsonObject(msg, "story_disps", mat2str(displacements));

%======= Publish messages ===============================================
rtiLib.publish("StoryDisplacement",msg);

%======= Disconnect to the SRTI Server ==================================
rtiLib.disconnect();
```

Figure 5-6. Example Matlab code of Structural Analysis Simulator (not a complete code)

5.4.2 Results and discussion

As shown in Figure 4-6, which plots the roof displacement of B1, conducting the simulations using the SRTI and with the LCM (Chapter 2) yielded numerically equivalent results thus validating the SRTI implementation. A key advantage of the SRTI is in user effort, which is significantly less than the LCM effort documented in Chapter 2. The SRTI users in the above example downloaded the pre-complied SRTI applications and wrote a few additional lines of code to call the RTI Lib API functions, as shown in Figure 5-6, in order to connect the server and publish/subscribe the desired messages. The RTI Lib API automatically converted the messages into String and JSON objects and the simulation was then ready to proceed.

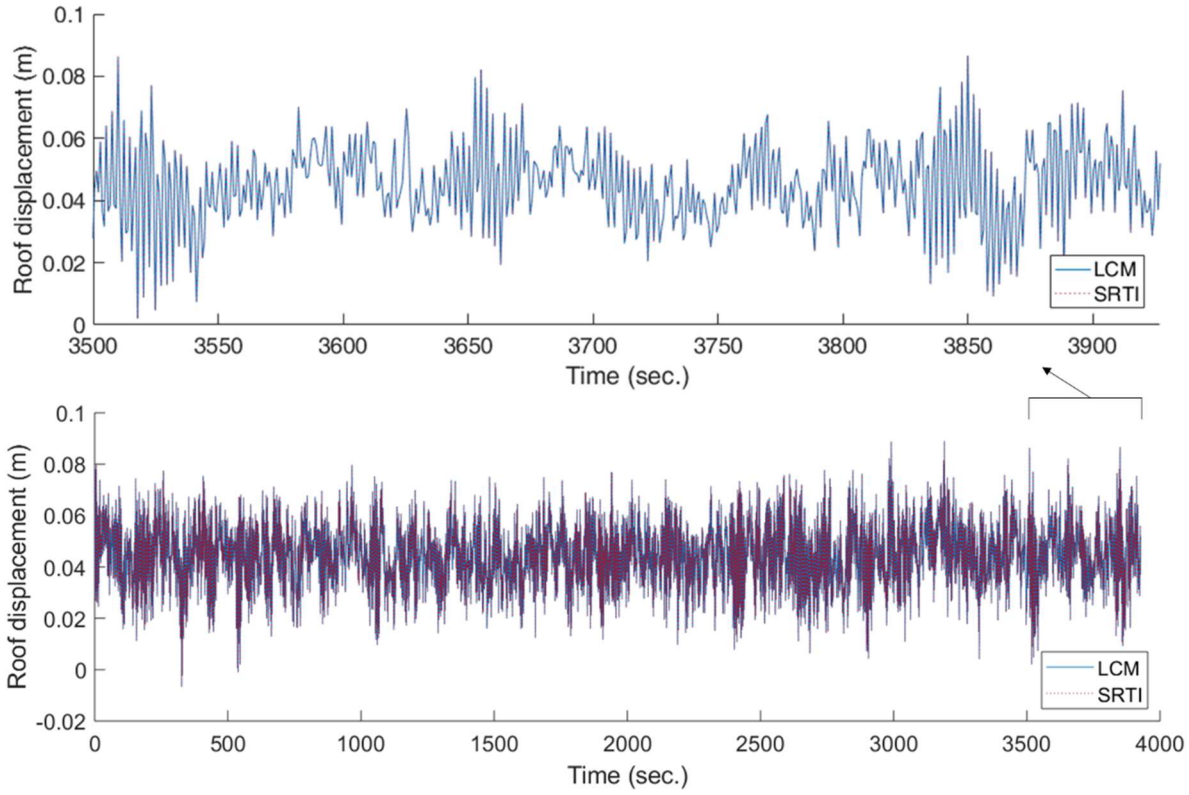


Figure 5-7. Roof displacement responses of B1

In contrast, users of the LCM version had to download the most current version of the LCM and then compiled it on their local machine. Users then needed to create a type definition file (.lcm) for each message following the LCM data structure syntax, and then run the build the LCM tool, i.e., lcm-gen, to marshal well-defined data into language-specific bindings. These bindings had to be re-generated if the original message definitions (.lcm) had any changes, e.g., the type of a variable was changed from integer to double, the size of a matrix was changed, or additional variables were added. Also, similar to the SRTI, the users had to write several lines of code for publishing and subscribing the messages to the corresponding the LCM channels.

The above discussion indicates that there are two main differences between the SRTI and the LCM regarding user efforts. The first is that the former is pre-complied and ready-for-use after being downloaded while the latter requires local compilation. The second and more important difference is the flexibility of message definition in the SRTI simulations. The entire message within the SRTI system is converted into a single String. Unlike the LCM, the user of the SRTI does not need to re-generate language-specific message bindings every time the definition of a message changes. As long as the "name" and "content" of the message are converted to a String, the value and the data type of the variables within the messages are independent of the SRTI system. This characteristic gives the users great freedom for message definition and the flexibility to easily add new variables or change their format. In addition, as is clear from Figure 4-5, SRTI has better extensibility to allow additional simulators to join the simulation and therefore increase its complexity.

The total simulation and computation time for each simulator are compared in Table 5-2 and Table 5-3, for the SRTI and the LCM trials, respectively. The data in the tables are the average of five complete simulation exercises each of which is 5,000 steps conducted on a computer with

an Intel Xeon CRU-E3-1240 v5, 3.50 GHz, 32 GB, 64-bit machine. The computation time of a simulator indicates the time spent in the main calculation function which computes output results based on the input values extracted from the subscribed messages. It does not include time for data transmission, connecting to the server, processing data format, and subscribing to and publishing messages. It is also important to note that the case study is a sequential one, where each simulator does not proceed until it receives the messages to which it subscribes. Since both the SRTI and the LCM cases use the same computation functions for all simulators, the total simulation time minus the sum of the computation time for all simulators is the extra time spent on distributed communication, including connecting to the server (for the SRTI), parsing data formats, setting up publish-subscribe relationships, and passing messages.

Table 5-2 and Table 5-3 show that the computation times are consistent among the five trials. On average, the whole simulation took 1.2 times longer to execute with SRTI than with the LCM. The communication time was 1.5 times longer in the SRTI compared to the LCM. This outcome can be attributed to three main reasons, all of which the authors are working to address in future versions of the SRTI without sacrificing the advances in terms of interface available to developers. The first reason is a difference in the networking architecture, i.e., simulator-to-simulator in the LCM versus simulator-to-middleware in the SRTI. Thus, messages in the SRTI are transmitted twice more than the ones in the LCM case, first from publisher to the server, then from the server to the subscribers. The second reason pertains to the differing transfer speeds of different data types. SRTI currently uses “string” format to allow messages to be sent without compilation, thus requiring more bytes for the same amount of information that is marshaled in binary format as done in the LCM. The last reason pertains to the computational expense associated

with parsing messages. In the SRTI, messages have to be encoded/decoded to/from a JSON object through the RTI Lib API.

Although the slower speed is a disadvantage of the current version of SRTI compared to LCM, it is believed that facilitating the ease of use by users who are potentially not well versed in programming or computation is a higher priority goal. Improving the interfaces to developers can be much harder in future versions than improving performance, since interfaces tend to be hard to change once they develop a user base. Another crucial advantage of the SRTI is that its architecture preserves the ability to manage complex, large-scale simulations. While increasing the freedom of the message format definition, SRTI uses RTI Server as a middleware to manage message communication, thus retaining the ability to control the overall simulation and has the potential to foster the development of new features in the future as discussed in the later section.

Table 5-2. Performance time of the SRTI (v1.00.00) trials

SRTI	Computation time of each simulation						Total Simulation Time	Time Spent on Distributed Communication
Trial	Scenario	Wind Speed	Wind Pressure	Wind Load	Structural Analysis	Damage		
1	0.117	73.017	1.557	0.622	1917.007	1545.904	6806.148	3267.924
2	0.042	74.386	1.526	0.625	1963.711	1597.581	6973.225	3335.354
3	0.010	72.207	1.534	0.649	1845.327	1434.615	6520.056	3165.714
4	0.087	70.745	1.509	0.632	1829.499	1431.861	6425.632	3091.299
5	0.070	80.054	1.738	0.705	2084.790	1654.067	7301.648	3480.224
Avg.	**0.065**	**74.082**	**1.573**	**0.647**	**1928.067**	**1532.806**	**6805.342**	**3268.103**

(Unit: seconds)

Table 5-3. Performance time of the LCM trials

LCM	Computation time of each simulation						Total Simulation Time	Time Spent on Distributed Communication
Trial	Scenario	Wind Speed	Wind Pressure	Wind Load	Structural Analysis	Damage		
1	0.007	77.222	1.512	0.617	1981.982	1592.495	5882.098	2228.263
2	0.008	75.038	1.612	0.637	1862.710	1459.244	5465.294	2066.045
3	0.005	76.174	1.516	0.621	1862.548	1433.553	5437.788	2063.371
4	0.006	75.761	1.572	0.653	1891.179	1463.391	5536.242	2103.680
5	0.005	75.533	1.465	0.627	1885.574	1461.992	5516.550	2091.354
Avg.	**0.006**	**75.946**	**1.535**	**0.631**	**1896.799**	**1482.135**	**5567.594**	**2110.543**

(Unit: seconds)

5.5 Implementation of SRTI v2.00.00: Cross-Language Simulation

5.5.1 Problem description

In this section, a cross-platform simulation of an electric power system subjected to an earthquake is conducted through the SRTI Server and Wrapper (release "v2.22.02") and v2.00.00 Manager GUI (release "v0.80.03"). To demonstrate the scalability and flexibility of the SRTI, the time-dependent analysis of interdependent lifeline systems in Chapter 3 has been simplified to consider the electric power system only, as illustrated in Figure 5-8. In addition, a *Visualization Simulator* written in NetLogo has been added to show the ability of the SRTI to connect simulators across multiple languages. Figure 5-9 shows the simulation framework of the cross-language example. Except for the *Visualization Simulator*, which is implemented using NetLogo, the other simulators are written in Matlab scripts.

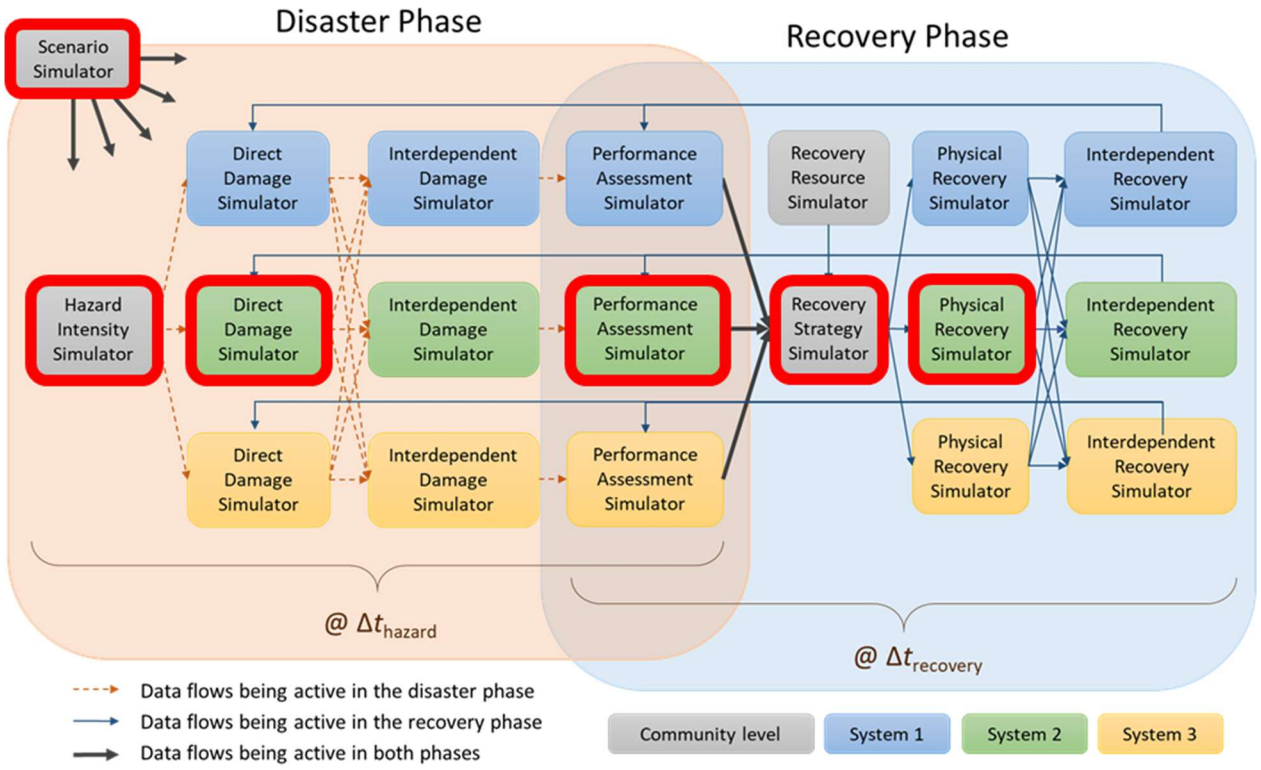


Figure 5-8. The simplified framework considering only one system in Chapter 3

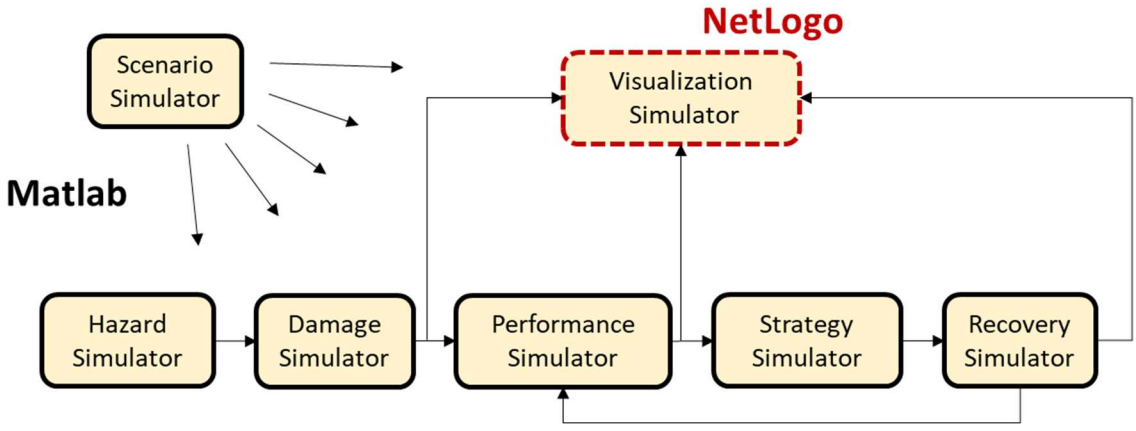


Figure 5-9. Simulation framework of cross-language example

Using the SRTI and GUI to conduct the simulation, the disaster event and post-disaster recovery effort are represented by seven simulators that may run at one or both of the disaster and recovery phases, named Stage 0 and Stage 1, as shown in Figure 5-10 and Figure 5-11. The time step of the disaster phase and recovery process is taken as 0.01 second and one day, respectively.

In Stage 0, (i.e. disaster phase), the *Scenario Simulator* provides the coordinates and connectivity of the electric power system. The *Hazard Simulator* estimates ground motion magnitudes at the location of components, where the EQ2 case in Chapter 3 is adopted for considering an earthquake with a 5% probability of exceedance in 50 years. The *Damage Simulator* obtains the physical damage of components using fragility functions from HAZUS-MH (FEMA, 2003). The *Performance Simulator* assesses the connectivity loss of the electric power system. The *Recovery Simulator* and *Visualization Simulator* subscribe the messages from *Damage Simulator* and *Performance Simulator* to update the damage state of components and the system performance, but they neither carry out calculations nor publish messages in this stage.

In Stage 1, (i.e. recovery phase), the *Strategy Simulator* distributes limited recovery resources (15 units/day) to the damaged components in order of their priority, namely, the P strategy in Chapter 3. The *Recovery Simulator* estimates the physical reconstruction progress of the damage components based on whether they are allocated recovery resources at the current time step. The *Performance Simulator* keeps updating the system performance of lifelines based on the recovery progress and the *Visualization Simulator* subscribes to the messages from the *Recovery Simulato*r and *Performance Simulator* to update the reconstruction progress and system performance.

The active simulators and messages in each stage are listed in Table 5-4 and Table 5-5. The message “RTI_” referenced in the tables refers to a handful of private variables (including current

timestep and stage) inside the "RTI Wrapper," which has been adapted to use the same subscription pattern as normal messages for simplification. Simulators can decide whether to subscribe the "RTI_" message depending on their needs, e.g., dependency with timestep or stage. The detailed topological configuration of the lifeline system and the methodology of each simulator have been well documented in Chapter 3. This section focuses primarily on the description of the procedure using the SRTI v2.00.00 and GUI.

Table 5-4. Active simulators and messages in Stage 0

Simulator	Published Message	Subscribed Message
Scenario	*Scenario*	*RTI*
Hazard	*GroundMotion*	*Scenario, RTI*
Damage	*Damage*	*Scenario, GroundMotion, RTI*
Performance	*Performance*	*Scenario, Damage, RTI*
Recovery	-	*Scenario, Damage, RTI*
Visualization	-	*Scenario, Damage, Performance, RTI*

Table 5-5. Active simulators and messages in Stage 1

Simulator	Published Message	Subscribed Message
Performance	*Performance*	*Recovery, RTI*
Strategy	*Strategy*	*Performance, RTI*
Recovery	*Recovery*	*Strategy, RTI*
Visualization	-	*Recovery, Performance, RTI*

Figure 5-10. Simulators running in Stage 0: disaster phase

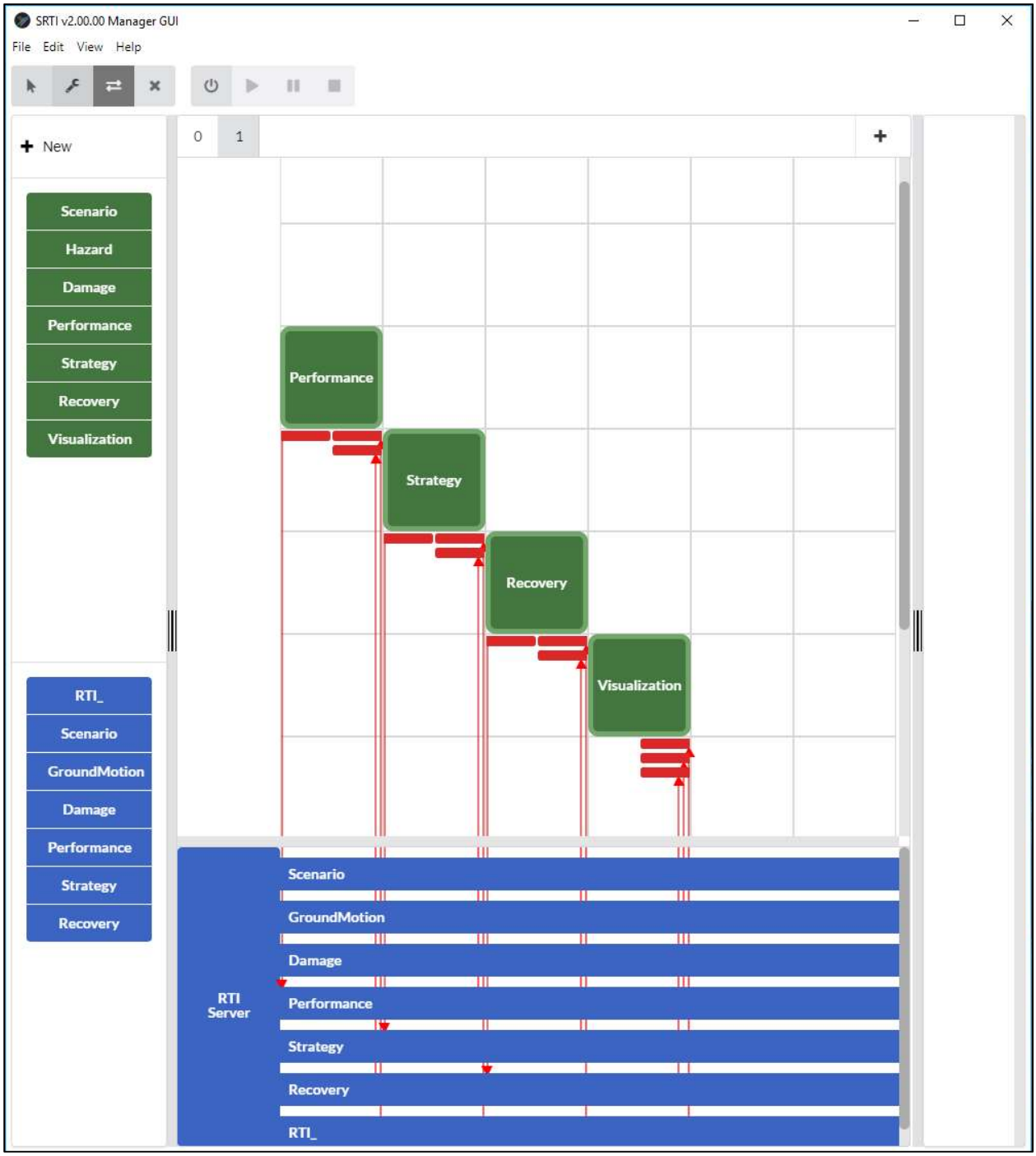


Figure 5-11. Simulators running in Stage 1: recovery phase

5.5.2 The procedure using the RTI Wrapper and GUI

As described in the documentation of the SRTI (SRTI, 2019), the pre-compiled SRTI files can be downloaded from the public GitHub Site. The download should include the SRTI Server, RTI Wrapper, and SRTI GUI, all of which can execute without explicitly installing (uncompressing files and setting system paths, etc.). In addition to saving the SRTI Server and GUI in a file directory on the local machine, user needs to prepare individual folders for each simulator, which contains the executable file of simulator itself and the corresponding Wrapper. For instance,

- **/Simulator_A/**
  - SRTI_JavaWrapper_v2_00_00.jar
  - Simulator_A.jar
- **/Simulator_B/**
  - SRTI_NetLogoWrapper_v2_00_00.jar
  - Simulator_B.nlogo
- **/Simulator_C/**
  - SRTI_Wrapper_v2_00_00.m
  - Simulator_C.m
- **…**

After preparing the file system, the next step is to launch the SRTI GUI.exe, for which layout is shown in Figure 5-4. Simulators/messages can be created and defined through the GUI. Then, users can click the objects on the Object List to add well-defined simulators or messages to the Canvas for a given stage. They could use action toggle buttons to define the publish/subscribe relationships between them, as shown in Figure 5-10 and Figure 5-11. All the defined simulators, messages, and publish/subscribe relationships can then be saved as a .project file to be imported and edited in the future. User would find the other files with extensions as described in Table 5-6 after saving a project. These files are coded in ASCII-text data format, typically representing a JSON object and can be read and edited in most text editors. Among them, the definition file of a

simulator or message (i.e. .simdef and .mesdef) can be imported into a new project independently to prevent re-defining from scratch.

Table 5-6. Description for different file extension

File Extension	Description
.project	Represents a single large JSON object that defines the full project. 1 of these files exists per project.
.simdef	Optional output file that describes a single simulator referenced in the .project file.
.mesdef	Optional output file that describes a single message referenced in the .project file.

User needs to click "Export Execute Files" from Menu Bar first to generate the configuration files for a Wrapper to read. The example Matlab code of the *Scenario Simulator* ("Scenario.m") written by the user and the auto-generated configuration files ("Settings.json" and "Scenario_config.json") exported from the GUI are shown in Figure 5-12 to Figure 5-14.

```
classdef Scenario < handle
    properties
        stage
        timestamp
        node
        connectivity
    end

    methods
        function obj = Scenario()
        end

        function Initialize(obj)
            [obj.node,obj.connectivity] = ScenarioSim(obj.stage);
        end

        function Simulate(obj)
        end

        [node,connectivity] = ScenarioSim(stage);
    end
end
```

Figure 5-12. Example Matlab code of Scenario Simulator ("Scenario.m")

```
{
    "global": "Global.json",
    "configuration": "Scenario_config.json"
}
```

Figure 5-13. Example configuration files ("Settings.json")

```
{
    "hostName": "localhost",
    "portNumber": "42012",
    "simulatorName": "Scenario",
    "simulatorRef": "Scenario",
    "stageChannels": [
        {
            "stage": 0,
            "order": 0,
            "timestepDelta": 1,
            "timestepMul": 1,
            "timestepVarDelta": ""
        }
    ],
    "initializeChannels": [
        {
            "functionName": "Initialize",
            "stage": 0
        }
    ],
    "simulateChannels": [
        {
            "functionName": "Simulate",
            "timestepDelta": 1,
            "stage": 0
        }
    ],
    "subscribedChannels": [
        {
            "messageName": "RTI_",
            "oneTime": false,
            "mandatory": true,
            "relativeOrder": 0,
            "maxTimestep": 0,
            "timestepDelta": 1,
            "stage": 0,
            "varChannel": [
                {
                    "valueName": "stage",
                    "varName": "stage"
                },
                {
                    "valueName": "stageVTimestep",
                    "varName": "timestamp"
                }
            ]
        }
    ],
    "publishedChannels": [
        {
            "messageName": "Scenario",
            "initial": true,
            "timestepDelta": 1,
            "stage": 0,
            "varChannel": [
                {
                    "valueName": "node",
                    "varName": "node"
                },
                {
                    "valueName": "connectivity",
                    "varName": "connectivity"
                }
            ]
        }
    ],
    "endConditions": [],
    "stageConditions": []
}
```

Figure 5-14. Example **configuration file ("Scenario_config.json")**

The user can either run the simulation system within the SRTI GUI, or separately using a command line prompt to open each simulator one by one. To run the simulation directly within the GUI, user can click on the "Power On" button (in the Execute Buttons section) to launch an instance of the RTI Server and the individual simulators. After allowing a few seconds for everything to finish opening and ensure all simulators are connected to the Server successfully, the user can then inspect the "List of connected apps" panel in the RTI Server graphic interface (Figure 3-2). At that point, the other buttons in the Execute Buttons section are possible to click. "Play" will start/resume the simulation system, "Pause" will pause the simulation system, and "Stop" will close the RTI Server and the simulators.

Other features of the SRTI GUI include being able to display the content of messages and system timestep in the inspector panel to better trace the system's execution outside the individual simulators. In addition, while the simulation is running, the color of the simulator objects on the Canvas will change according to their status, as shown in Figure 5-15. Blue ones indicate the simulators waiting for the messages that they subscribed to, and red ones are the simulators running their internal calculation. Figure 5-16 shows the progress of seismic damage and post-disaster recovery of the power system plotted by NetLogo *Visualization Simulator*, where blue, green, yellow, orange, and red color indicate the non-, minor, moderate, extensive, and complete damage state of lifeline components. Figure 5-17 shows the time history of system performance plotted by the NetLogo *Visualization Simulator*.

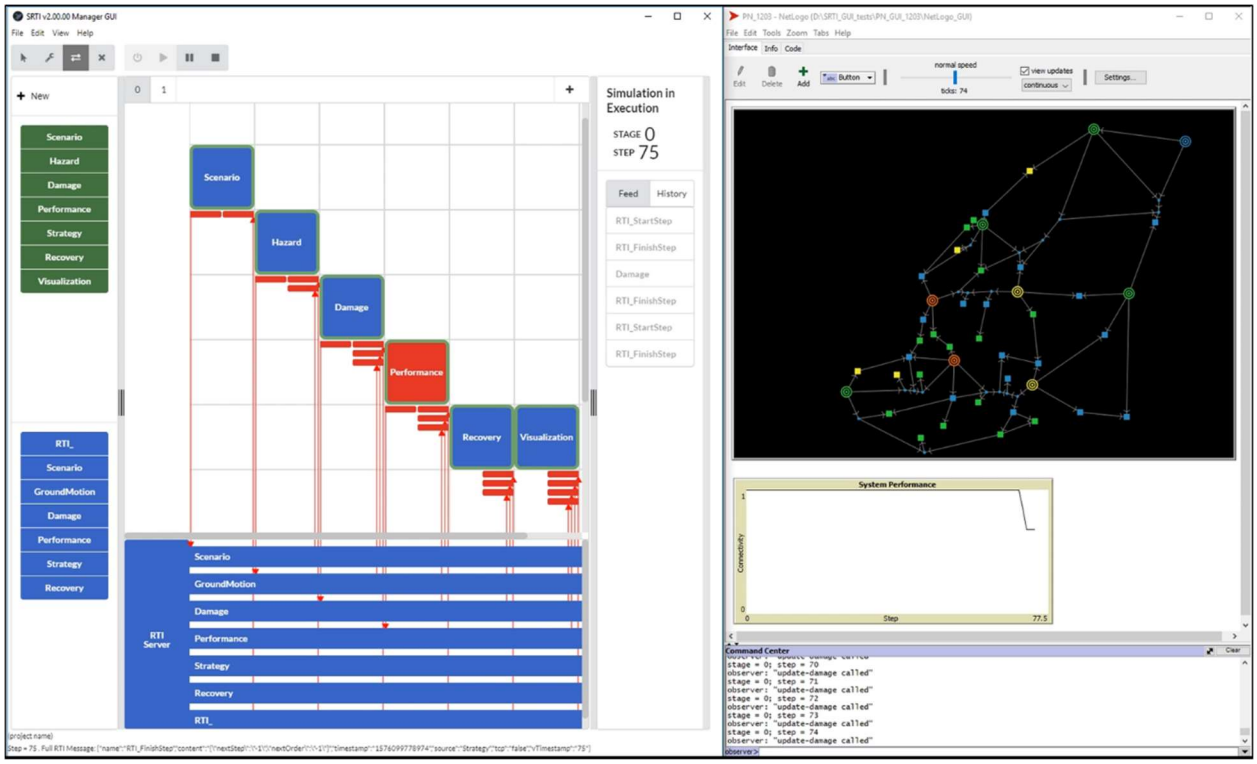


Figure 5-15. Screenshot of SRTI GUI and NetLogo Visualization Simulator while the simulation is running

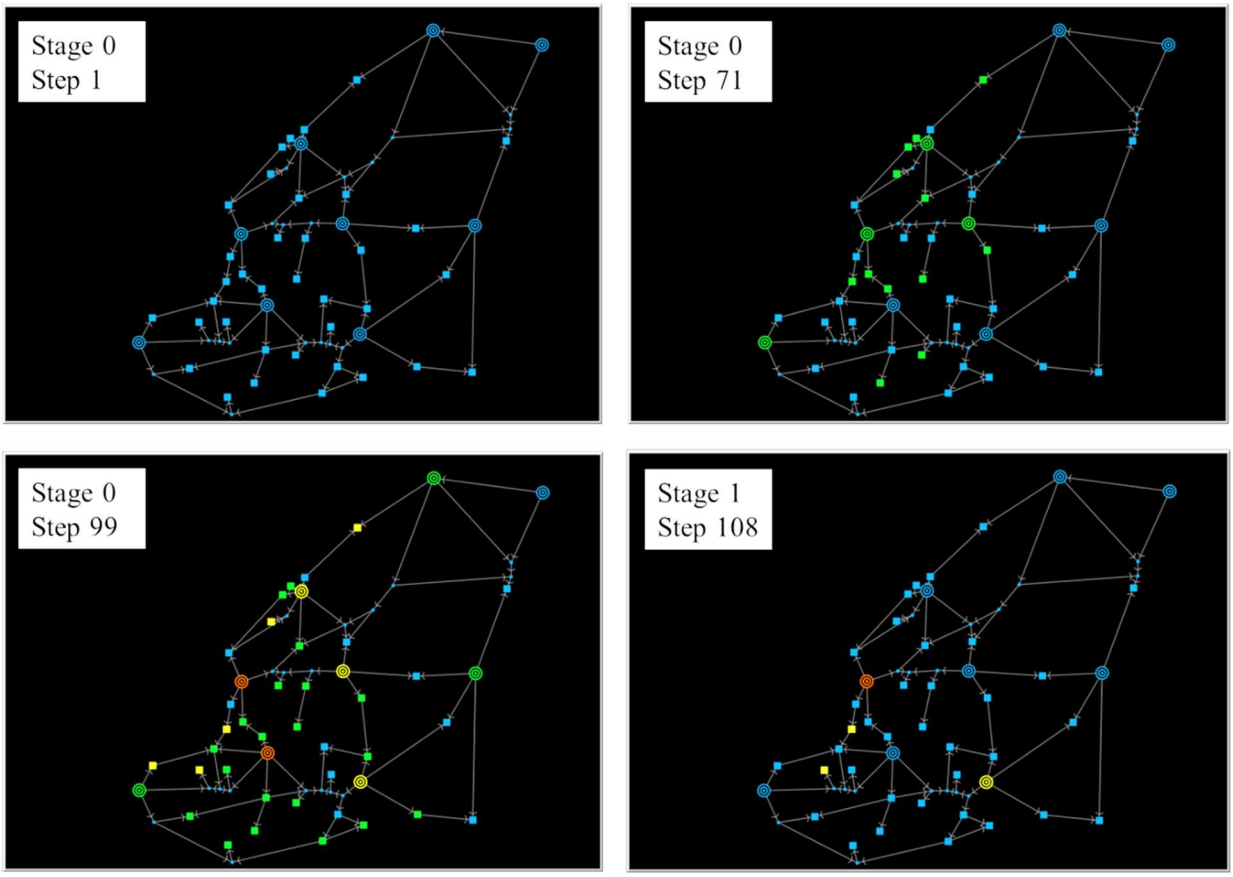


Figure 5-16. Progress of seismic damage and post-disaster recovery of electric power system

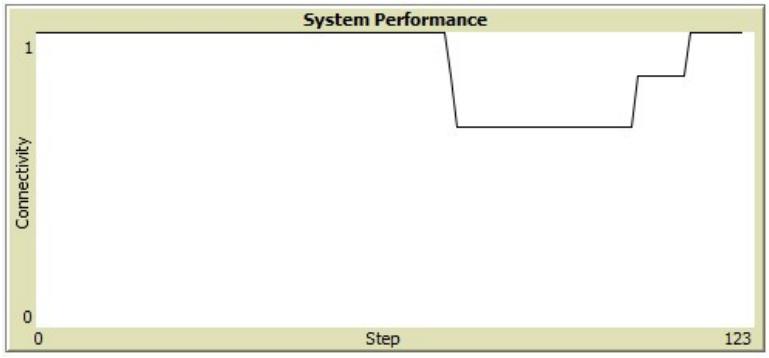


Figure 5-17. System performance plotted by NetLogo Visualization Simulator

5.6 Discussion

5.6.1 Time management and user experience

In SRTI v1.00.00, the RTI server does not apply any mandatory rules on simulators for time management. If simulation time is a key concern, users need to embed this kind of information within transmitted messages, and create their own rules regarding time. However, the SRTI v2.00.00 permits some time management features to be controlled at the RTI Management Server-level. For example, the controlling of time step and simulation stage can be automated through the RTI Management Server by many controlling parameters, either defined inside a configuration file by users before the start of a system execution, or inside subsequent messages during execution. A complete list of these controlling parameters are provided in the documentation on the SRTI GitHub site (SRTI, 2019).

To conduct a simulation using the SRTI v1.00.00, the user must first start the RTI Server so that simulators can connect to it. Then, to connect simulators to the RTI Server through the RTI Lib API, the user must assign the corresponding host name and port number. They can either type these values manually into the simulator's source code or let a pre-compiled simulator read these from an external source, such as launch parameters or from an external data file. In order to run an entire simulation system, each simulator must be launched individually by the user. After running the simulator, the RTI Server should react to confirm it has connected successfully, and the users can check the list of connected simulators on the RTI Server GUI (see Figure 3-2). Since the RTI Server has no control of when simulators execute, users are responsible for ensuring the logic of the calculations within the simulators. They will need to design their simulators to either wait for a specific message to start, or else start the simulators in a specific order.

On the other hand, the RTI Wrapper and the GUI in the SRTI v2.00.00 greatly improve the workflow. Rather than writing additional lines of code in the simulator to call upon specific RTI Lib API functions, users can execute a RTI Wrapper to handle the connection to the RTI Management Server and data parsing, as long as a configuration file is provided in the corresponding file directories. A simulation system's configuration files can be edited by hand, and individual RTI Wrapper instances (for each simulator) can be started manually. Alternatively, the GUI can assist users defining the configuration file for each RTI Wrapper, and allow execution of the whole simulation system from a single place. Project files for the GUI, and individual simulators and messages in the project can be saved and reloaded for the later use. In addition, the GUI provides system feedback when the simulation is running, such as portraying which simulator is currently in the process of finishing a step, the content of recent messages, and current system timestep.

Overall, the SRTI v2.00.00 provides useful time management features and improves the workflow, but the downside is the difficulty of creating a generalized management system. Although the SRTI v2.00.00 can handle most design preferences, it does not allow unlimited control like earlier versions of the SRTI (v1.00.00) can. Users can choose either version depended on the desired usability.

5.6.2 Limitations and future opportunities

The SRTI's different versions have each been designed with usability as its primary goal, ignoring the common goal of execution speed. Some common strategies to increase computation performance may require local compilation to optimize data formats, or require that simulators run on the same local machine (not allowing network communication), or require simulators be

defined in the same language. While, these may conflict with the SRTI's goal of being scalable and flexible. Ultimately, it is unlikely that the SRTI will be able to match the performance of similar tools without losing the original goals. Therefore, systems that require frequent transfer of large packets of data should not use the SRTI, unless they are capable of absorbing the extra execution time from this new bottleneck.

On the other hand, the data content from each simulator is not absolute, and cannot represent everything that an additional (future, as-of-yet unknown) simulator might require. For certain research groups, creating a standard, strictly defined format might be necessary to help designers who want to extend a system with new simulators. This is against the open concept of the SRTI, but such rules can be defined and enforced while using the SRTI as middleware. Finding a balance between these lines of thought will invoke further discussion across different fields.

Despite a potential trade-off of computational efficiency, providing ease of use to users who may not be knowledgeable in programming is a significant advantage of SRTI. At the same time, SRTI's architecture preserves the ability to manage complex, large-scale simulations. While having high flexibility in the message format definition, SRTI uses the RTI Server as a middleware to manage message communication, thus retaining the ability to control the overall simulation and has the potential to foster the development of new features, such as sophisticated time management or specific quality of service (QoS) policies for complex simulations.

The RTI Wrapper plays an important role to communicate these time management features to the RTI Management Server in the SRTI v2.00.00. As discussed in Xu et al. (2020), in most existing distributed simulation platforms and data passing tools, the user must modify the internal code of their simulator to connect to a larger system using specific instructions. While this feature remains in the SRTI to allow the most customizability for experienced users, most users will utilize

the RTI Wrapper. The RTI Wrapper is pre-compiled and distributed online for users to simply download and run locally. However, a RTI Wrapper compatible with the simulator must be used. At the time of this writing, RTI Wrapper versions for Java, Matlab and NetLogo (Wilensky, 1999) have been prepared for public access. It may be re-written in a new language for the other computer language in need of support.

The time management and synchronization of virtual simulations is commonly associated with artificial time, but there is no reason a 'real-time' simulator (utilizing sensors in real space) could not be linked to the system. This type of simulation would require the assumption that certain messages are not necessary to be sent once every time step, an option that the SRTI fully supports. Therefore, investigating a hybrid system that consists of both virtual and real simulators can be a future research opportunity for the SRTI.

5.7 Conclusions

A new distributed computing software solution, Simple Run-Time Infrastructure (SRTI), is described. SRTI is able to integrate discipline-specific models in natural hazards simulation. SRTI is based on a client-server structure and uses a publish-subscribe pattern to achieve data communication between simulators executed on separate processes or machines. It is a scalable and versatile solution that does not require recompilation when new simulators are added or data format changes, thereby significantly facilitating its deployment by non-specialized users. As such, SRTI provides a unique service for researchers in the natural hazards field who seek to integrate a variety of simulators into overarching simulations of disaster scenarios to study how these events and the subsequent recovery efforts unfold.

The high-level structure of the initial version of the SRTI (v1.00.00) and its key components are introduced in detail firstly. Next, an improved version of the SRTI (v2.00.00), which better supports time-dependent simulations and provides a low barrier to entry to user with limited programming experience, is described. The data structure of the SRTI is also introduced. The full source code, pre-compiled libraries, documentation, and other information are available at SRTI (2019).

Two implementation examples for different version of the SRTI are provided. A case study of wind-excited buildings suffering progressive damage described in Chapter 2 is presented to demonstrate the capabilities of the SRTI v1.00.00. The user experience and performance are discussed and compared with the LCM. The result of trials represents the performance of the SRTI where computational efficiency is sacrificed to some extent in the quest to allow users to compose their distributed simulations more easily and quickly. Besides, the scalability and usability of the SRTI v2.00.00 are demonstrated through a cross-language simulation of seismic damage and post-disaster recovery of a lifeline system, exploiting the SRTI GUI. It shows that the RTI Wrapper and GUI greatly improve the workflow.

Finally, the time management and user experience of different versions of SRTI are discussed. Some other enhancements of the SRTI that feasible in the future are outlined, including improvement of performance and better compatibility with other programming languages. As a free and open source software, the implementation of SRTI's design will continue to be improved and optimized in the future.

CHAPTER 6

Summary, Conclusions, and Future Research

6.1 Summary

The interdependencies that arise between the infrastructure systems of society can profoundly influence community resilience. Studying the interactions that occur between infrastructure systems is challenging because of the high disciplinary boundaries between fields in which each system is rooted and due to the disparate temporal and spatial scales of these systems. To address these technical challenges, simulation frameworks for modeling and investigating the interactions between infrastructure systems during extreme natural disasters and the post-disaster recovery process were developed. In particular, this dissertation explored the time-dependent effect of interdependencies on community resilience.

Existing distributed simulation standards and tools were systematically reviewed first. Based on the survey, the LCM distributed simulation framework, which connects discipline specific simulators via a publish-subscribe data transmission pattern, was selected for modeling interdependencies. Each simulator in the system was seen as a black box subscribing to data from other simulators and/or publishing its results for others to use. An analysis of a collection of multistory buildings that suffered progressive damage during a wind event was presented to demonstrate the capabilities of the selected framework.

A time-dependent resilience assessment of three interdependent lifeline systems subjected to a series of seismic events was conducted utilizing the publish-subscribe data transmission

pattern and distributed simulation concepts in LCM. A group of discipline specific computational models that have disparate temporal and spatial scales were linked together to study the multiscale and time-dependent interdependency during a series of seismic events that include short- and long-term recovery processes. The time-dependent effects of resource allocation strategies on community resilience were investigated. It was noted that the analysis results could be used to assist decision-makers to determine emergency relief and post-disaster recovery policies.

To address broader socio-technical considerations in resilience research and explore the coupling effects between civil and social systems from both hazards engineering and social science perspective, an agent-based computational model simulating benefit fraud behavior in the wake of disaster was created. The effect of both micro-level disaster demands caused by building damage and meso-level social variables on benefit fraud were considered. The statistical data from the government reports in the aftermath of Hurricane Katrina and Rita was used for the calibration and validation. The proposed model, which includes a simulation environment of a community facing a natural disaster, fraudsters, and application inspectors, was then used to estimate the losses associated with these post-disaster crimes and their impact on community resilience. A parametric sensitivity analysis was performed to gain deeper insight into the benefit fraud problem.

To provide a scalable, versatile, and user-friendly solution for natural hazards simulations, a new distributed computing software called Simple Run-Time Infrastructure (SRTI) was introduced. The high-level structure, data structure, and fundamental components of the SRTI were comprehensively described. The same case study of wind-excited buildings suffering progressive damage in Chapter 2 was conducted using an initial version of SRTI (v1.00.00) to demonstrate its capabilities and compare its performance to the case using LCM. Then, to reduce a user's effort of composing a complex distributed simulation and better handle time management, an alternate

version of the SRTI (v2.00.00) was designed and described. By means of the improved or additional pre-compiled components of SRTI v2.00.00, including the RTI Management Server, RTI Wrapper, and GUI, a cross-language simulation of time-dependent resilience analysis of an electric power system was conducted for showing its scalability and flexibility. The choice between different versions of the SRTI was discussed, and useful features to develop in the future were outlined.

6.2 Conclusions

Within the scope of the studies conducted in this dissertation, the following conclusions can be drawn:

6.2.1 Distributed disaster simulation

- Distributed implementation gives results similar to the traditional integrated application, but the computing efficiency may be traded off for distributed communication, including connecting to the server (for the SRTI), parsing data formats, setting up publish-subscribe relationships, and passing messages.
- By modularizing each part of disaster as a simulator and using a publish-subscribe pattern for data management, the proposed distributed simulation frameworks have better modifiability, extensibility and scalability for modeling complex problems, such as hazard and resilience assessment of a community.
- In the example selected, the communication time is about 1.5 times longer in SRTI v1.00.00 compared to LCM due to different data transmission mechanisms. However, SRTI allows users to compose their distributed simulations more easily and quickly. It does

not require recompilation when new simulators are added or data format changes, thereby significantly simplifying its deployment by non-specialized users. The employment of RTI Wrapper and SRTI GUI lowers the entry barrier and coding efforts for users with limited programming experience.

- As a real-time data transmission solution for distributed computational simulations that supports time-dependent simulations, SRTI provides a unique service for researchers in the natural hazards field who seek to integrate a variety of simulators into overarching simulations of disaster scenarios to study these events and the subsequent recovery efforts unfold. The open source nature of SRTI's code and design will encourage future improvement and optimization.

6.2.2 Interdependencies in community resilience

- The dynamic coupling behavior and interdependencies in disaster events can be simulated through either incremental-iterative or incremental analysis, depending on whether the computational results reach equilibrium within each simulation step through iterations. With a reasonable length of time step, these two analytical schemes yield similar results, but more computational studies of the effect of local iteration loops are required to explore the limits of simulation accuracy.
- The computational results in Chapter 3 quantified the influence of the interdependencies between the lifeline systems on the resilience of the studied community. The results showed that the disaster-caused damage, operability loss, and required recovery time can be greatly underestimated if system interdependencies were not considered. It was also found that the time-varying resource allocation strategies, which adjusted based on system

connectivity and damage states, could achieve better resilience performance. The impact of different resource allocation strategies was more significant when resources were insufficient.

- The ability to handle differences in temporal scales between a hazard and the recovery process is one of the key advantages of the proposed simulation framework in Chapter 3. Aside from the capability of accounting for multiscale interdependencies, the case study pointed out the need of time-varying analysis as the interactions occur during hazard and recovery process.
- The proposed agent-based methodology in Chapter 4 can capture the basic features of benefit fraud while producing sensible results. Due to lack of information about some key variables, which were assumed out of necessity, the simulation results are not truly predictive. However, it provides the ability to model post-disaster benefit frauds in detail, including the spatial and temporal nature of the criminal process.
- From the simulation results, it is found that fraudulent activity occurs in three distinct phases in the aftermath of a disaster which matches the observations in actual disasters. The results of the ABM analysis also show that increasing the accuracy of review, decreasing disaster demands, and enhancing guardian strength can help lessen the loss due to post-disaster benefit fraud.

6.3 Recommendations for Future Research

The following research topics are recommended for future research in the area of interdependencies in community resilience and distributed disaster simulation:

- To better quantify the impact on society due to loss of lifeline serviceability, different indicators can be used to assess the system performance such as the actual amount of service flow and the number of people who are able to access to the lifeline systems service. The occupancy type of the demand nodes, e.g. hospitals and refugees, should be considered when deciding the priorities of system components to be recovered. These can be adopted into the existing framework once the information is available.
- Through the addition of new simulators or the modification of the existing simulators, various types of recovery resources and possible recourse constrains can be considered in the future research, including the limited construction materials, available budget, and shortage of workers owing to injuries. The effect of delays due to bad weather conditions and traffic blockages on these recovery resources should be taken into account as well. Moreover, in reality, when decision makers make decisions such as recovery strategies, they may not be able to get comprehensive and accurate information in real time. Hence, the quality of information flow should be considered while doing time-dependent decision-making simulations.
- The damage assessment performed in this study are mostly based on computational simulation results or empirical data such as fragility curves. Yet, owing to the scalability and flexibility of the proposed distributed simulation framework, various damage detecting techniques can be applied. For example, the applications of machine learning techniques on post-disaster reconnaissance can be used to increase the reliability of simulation. In addition, a hybrid simulation of both virtual and real simulators can be conducted. By connecting the sensors on smart buildings and infrastructure to the current simulation framework, the extracted real-time data can be used to evaluate the actual damage situation.

- A higher-order parametric sensitivity analysis of the proposed agent-based model is needed in future studies to consider the interdependency existing between multiple independent variables. For example, it usually takes longer to review an application in order to reduce the judgement errors. However, it relies on more comprehensive data collection and further studies from the social science perspective to better understand these correlation effects.
- Although a improved computational performance can be achieved by optimizing the SRTI's implementation, users could also consider designing their simulation system in a parallel or time-sliced manner, which is similar to the concept of data "pipelining" in computer science. Rather than executing simulators in series, i.e. some simulators may be idling while the others are doing simulations, conducting a simulation with a certain degree of pipeline data pattern can better take advantage of distributed simulation by means of computing resources on discrete machines at the same time.

APPENDIX A

Table A-1. Demographics data in Shelby County retrieved from Hazus (2015)

ID	Population	Male16to65	Households	Responsible demand nodes		
				EPS	WDS	NGS
1	5394	2214	3054	P37	W29	G10
2	1033	259	356	P37	W29	G10
3	1060	314	380	P37	W29	G10
4	1727	541	658	P37	W29	G10
5	2152	665	885	P23	W29	G10
6	4861	1325	1991	P23	W29	G10
7	2684	677	992	P23	W31	G10
8	2884	777	1148	P23	W31	G10
9	3666	1250	1217	P22	W31	G10
10	3733	1235	1457	P22	W31	G10
11	3817	1129	1299	P22	W31	G10
12	1648	525	631	P22	W31	G10
13	1745	500	724	P20	W31	G10
14	3158	1163	997	P36	W29	G10
15	3848	1286	1666	P36	W29	G10
16	1601	458	576	P37	W29	G10
17	1969	539	677	P37	W29	G10
18	1192	312	553	P37	W29	G10
19	1839	708	688	P36	W29	G10
20	2877	1019	1227	P36	W29	G10
21	2511	918	1221	P36	W29	G10
22	2018	633	710	P20	W31	G10
23	3762	923	1492	P20	W31	G10
24	4637	1505	2282	P33	W31	G10
25	2938	778	1309	P20	W31	G10
26	3410	1366	1709	P34	W31	G10
27	3856	1440	2198	P34	W27	G6
28	2141	787	1103	P34	W27	G6
29	2099	950	1186	P36	W27	G6
30	2996	1111	1470	P34	W27	G6
31	2039	974	1024	P36	W27	G6
32	1345	459	859	P36	W27	G6
33	1048	532	800	P36	W29	G6
34	1184	505	791	P35	W29	G6
35	2348	1258	1622	P35	W29	G6
36	1353	592	880	P35	W22	G6
37	956	178	287	P35	W29	G6

38	1498	378	634	P35	W27	G6
39	1074	305	383	P35	W22	G6
40	3945	1080	1483	P31	W22	G6
41	2731	746	1025	P19	W22	G6
42	3802	1025	1529	P17	W22	G4
43	2633	813	955	P19	W27	G4
44	1235	360	448	P19	W22	G6
45	2730	821	1016	P19	W27	G6
46	2030	580	794	P19	W27	G4
47	2242	694	789	P34	W27	G6
48	2405	931	1112	P34	W27	G6
49	1585	467	677	P34	W23	G6
50	2747	745	1064	P20	W23	G4
51	2286	827	1172	P34	W23	G10
52	3559	1014	1357	P20	W23	G10
53	2538	729	984	P20	W23	G10
54	3011	905	1116	P20	W23	G4
55	3324	1155	1323	P20	W23	G10
56	2549	762	1067	P20	W31	G10
57	2322	838	1294	P20	W31	G10
58	4127	1406	1488	P33	W31	G10
59	3094	1490	1458	P33	W23	G10
60	1581	440	600	P17	W27	G4
61	2802	787	1161	P17	W23	G4
62	5903	1307	2261	P17	W23	G4
63	1681	498	650	P17	W22	G4
64	5528	1505	2091	P30	W23	G10
65	5174	1626	2085	P30	W23	G10
66	2725	823	951	P15	W23	G4
67	4393	1232	1893	P30	W23	G4
68	4840	1485	1587	P30	W23	G4
69	4116	1055	1827	P33	W31	G10
70	5886	1819	2388	P33	W31	G10
71	4768	1602	1961	P22	W31	G10
72	6854	2295	2202	P22	W31	G10
73	4078	1520	1333	P22	W31	G10
74	2567	843	854	P38	W31	G10
75	7200	2149	3174	P38	W31	G10
76	4495	1470	2125	P30	W18	G10
77	3128	1021	1605	P30	W18	G10
78	6346	1988	2984	P32	W25	G7
79	4830	1260	1990	P32	W25	G7
80	2908	877	1006	P30	W18	G10
81	3110	1005	1229	P38	W31	G10
82	3783	1147	1372	P39	W33	G12
83	2851	799	1034	P37	W33	G12
84	7653	2007	2453	P41	W33	G12
85	7191	1712	2358	P41	W33	G12
86	5399	1331	1895	P37	W33	G12

87	5706	1479	1852	P41	W34	G12
88	6810	1793	2314	P23	W34	G12
89	1478	388	467	P41	W33	G12
90	2413	665	778	P15	W19	G4
91	7051	2183	2211	P30	W18	G4
92	4026	1110	1386	P13	W18	G4
93	4452	1316	1519	P13	W18	G4
94	4831	1382	1513	P30	W18	G4
95	3206	923	1187	P11	W18	G4
96	5865	1673	1946	P13	W18	G4
97	4146	1208	1473	P11	W18	G4
98	4116	1077	1420	P15	W19	G4
99	1615	525	659	P15	W19	G4
100	1892	572	726	P22	W31	G10
101	1720	501	692	P37	W29	G10
102	1850	797	771	P37	W29	G10
103	5954	3467	1543	P35	W29	G6
104	2914	857	1132	P19	W27	G6
105	2743	755	953	P35	W27	G6
106	1693	400	578	P19	W22	G6
107	5645	1763	1983	P30	W18	G10
108	4676	1336	1526	P39	W37	G12
109	3084	1003	1178	P42	W45	G12
110	6292	2025	2217	P45	W46	G14
111	2828	898	1094	P43	W39	G12
112	3536	1018	1106	P44	W39	G14
113	5393	1497	2292	P45	W44	G14
114	1118	459	239	P45	W42	G14
115	2128	610	793	P40	W34	G12
116	5837	1531	2607	P40	W34	G12
117	3693	905	1215	P40	W39	G12
118	3547	837	1136	P40	W39	G12
119	4909	1309	1543	P40	W34	G12
120	5803	1582	1701	P25	W39	G14
121	6290	1787	2222	P25	W39	G14
122	5158	1419	1761	P25	W39	G14
123	5313	1310	2179	P25	W35	G10
124	4164	1155	1684	P38	W35	G10
125	7124	2099	2868	P25	W35	G10
126	4026	1141	1580	P25	W35	G10
127	5522	1675	1955	P24	W35	G11
128	3347	942	1233	P24	W35	G14
129	2912	879	1162	P25	W35	G14
130	2840	866	996	P24	W35	G14
131	10075	3051	3217	P24	W40	G11
132	8492	2593	3103	P25	W35	G14
133	9469	2744	3384	P25	W35	G14
134	6442	1870	2394	P25	W35	G14
135	5340	1609	2037	P25	W35	G14

136	2584	779	924	P45	W41	G14
137	3274	1038	1157	P26	W49	G13
138	8206	2364	2798	P26	W36	G11
139	3889	1182	1600	P24	W32	G22
140	8393	2572	2825	P24	W35	G11
141	9886	2867	3010	P26	W36	G13
142	14912	4700	5590	P26	W32	G9
143	6205	1891	2108	P21	W24	G8
144	3713	1137	1306	P38	W35	G10
145	6724	2119	2380	P24	W35	G10
146	3877	1183	1391	P24	W35	G10
147	4901	1495	2404	P24	W28	G10
148	5626	1839	2300	P38	W28	G10
149	7801	2369	3020	P24	W28	G10
150	4375	1332	1901	P24	W28	G8
151	4070	1455	1975	P24	W28	G10
152	6334	1900	2734	P24	W30	G9
153	4925	1497	1773	P24	W30	G9
154	8037	2547	3381	P21	W28	G8
155	3387	1058	1323	P21	W30	G8
156	4263	1249	1420	P21	W26	G8
157	4787	1564	2010	P21	W28	G8
158	7301	2294	2764	P24	W30	G9
159	3440	1027	1514	P24	W30	G9
160	4526	3807	3	P38	W28	G10
161	4778	1165	2218	P32	W25	G7
162	2206	774	1229	P32	W25	G7
163	5924	1581	2464	P32	W25	G7
164	3247	899	1328	P32	W25	G7
165	3993	1156	1585	P14	W25	G7
166	3710	1061	1615	P14	W25	G7
167	5055	1481	1920	P14	W17	G7
168	9004	2906	3970	P14	W17	G7
169	4726	1216	2130	P21	W26	G7
170	6867	1876	2485	P21	W26	G7
171	9295	2834	3046	P21	W26	G7
172	2890	820	1199	P32	W25	G7
173	3244	954	1334	P32	W25	G7
174	4138	1216	1809	P14	W25	G7
175	7992	2392	3238	P14	W17	G5
176	10210	3157	3931	P12	W17	G5
177	5361	1645	1838	P12	W26	G7
178	11474	3447	3694	P12	W26	G7
179	6111	1871	2141	P12	W26	G7
180	4599	1395	1573	P12	W26	G7
181	4644	1423	1510	P12	W26	G7
182	3089	858	1120	P12	W17	G5
183	3114	881	1230	P11	W18	G4
184	3974	1109	1574	P11	W18	G4

185	2312	653	872	P14	W25	G5
186	4073	1324	1931	P11	W18	G4
187	4156	986	2145	P11	W18	G4
188	2845	812	821	P11	W16	G4
189	6966	2007	2551	P11	W16	G4
190	7888	2263	2519	P11	W16	G4
191	5879	1758	1946	P14	W17	G5
192	7718	2243	2599	P14	W17	G5
193	4588	1259	1791	P11	W17	G5
194	4035	1188	1264	P11	W17	G5
195	7066	2102	2320	P14	W17	G5
196	4661	1325	1670	P14	W17	G5
197	3323	941	1163	P11	W17	G5
198	4736	1371	1610	P11	W17	G5
199	5031	1467	1716	P10	W16	G4
200	3377	916	1124	P16	W20	G4
201	2757	836	978	P16	W20	G4
202	3490	1038	1317	P16	W20	G4
203	5742	1337	2166	P28	W20	G4
204	6014	1659	2292	P28	W20	G4
205	4436	1300	1594	P28	W16	G4
206	3995	1104	1485	P28	W16	G4
207	5920	1667	2079	P28	W16	G4
208	4615	1236	1583	P29	W21	G6
209	4099	1102	1507	P31	W20	G6
210	5520	1316	1917	P29	W20	G6
211	4052	1069	1475	P27	W20	G6
212	4197	1170	1558	P27	W20	G6
213	4744	1202	1611	P27	W20	G6
214	6011	1690	2190	P27	W16	G6
215	4641	1410	1697	P16	W20	G4
216	3881	1072	1308	P13	W16	G4
217	7166	1841	2488	P27	W16	G6
218	63	47	1	P15	W19	G4
219	3	1	1	P18	W21	G6
220	0	0	0	P18	W21	G6
221	293	4	9	P38	W28	G8

REFERENCES

Adachi, T., and Ellingwood, B. R. (2008). Serviceability of earthquake-damaged water systems: Effects of electrical power availability and power backup systems on system vulnerability. *Reliability Engineering and System Safety, 93*(1), 78-88. doi:10.1016/j.ress.2006.10.014

Adachi, T., and Ellingwood, B. R. (2009). Serviceability assessment of a municipal water system under spatially correlated seismic intensities. *Computer-Aided Civil and Infrastructure Engineering, 24*(4), 237-248. doi:10.1111/j.1467-8667.2008.00583.x

Aguirre, B. E., and Lane, D. (2019). Fraud in disaster: Rethinking the phases. *International Journal of Disaster Risk Reduction, 39*, 101232.

Albert, R., Albert, I., and Nakarado, G. L. (2004). Structural Vulnerability of the North American Power Grid. doi:10.1103/PhysRevE.69.025103

ASCE. (2017). Minimum Design Loads and Associated Criteria for Buildings and Other Structures (Vol. ASCE 7-16). Reston, VA: ASCE (American Society of Civil Engineers).

Atkinson, G. M., and Boore, D. M. (1995). Ground-motion relations for eastern North America. *Bulletin of the Seismological Society of America, 85*(1), 17-30.

Barton, D. C., Eidson, E. D., Schoenwald, D. A., Stamber, K. L., and Reinert, R. K. (2000). *Aspen-EE: an agent-based model of infrastructure interdependency*. Retrieved from

Beckmann, K., and Dedi, O. (2015). *sDDS: A portable data distribution service implementation for WSN and IoT platforms.* Paper presented at the IEEE Conferences.

Breetzke, G. D., Fabris-Rotelli, I., and King, M. (2018). The impact of the Canterbury earthquakes on the temporal and spatial patterning of crime in christchurch, New Zealand. *Australian and New Zealand Journal of Criminology, The, 51*(1), 135-156.

Casalicchio, E., Galli, E., and Tucci, S. (2007). *Federated Agent-based Modeling and Simulation Approach to Study Interdependencies in IT Critical Infrastructures.* Paper presented at the Proc., 11th IEEE Symposium on Distributed Simulation and Real-Time Applications.

Cayirci, E. (2013). *Modeling and simulation as a cloud service: a survey.* Paper presented at the Proc., 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World.

Chang, S. E., Seligson, H. A., and Eguchi, R. T. (1996). *Estimation of the economic impact of multiple lifeline disruption: Memphis Light, Gas and Water Division case study*. Retrieved from

Chang, S. E., and Miles, S. B. (2004). The Dynamics of Recovery: A Framework. In Y. Okuyama and S. E. Chang (Eds.), *Modeling Spatial and Economic Impacts of Disasters* (pp. 181-204). Berlin, Heidelberg: Springer.

Chuang, W. C., and Spence, S. M. J. (2017). A performance-based design framework for the integrated collapse and non-collapse assessment of wind excited buildings. *Engineering Structures, 150*, 746-758. doi:https://doi.org/10.1016/j.engstruct.2017.07.030

Cimellaro, G. P., Solari, D., and Bruneau, M. (2014). Physical infrastructure interdependency and regional resilience index after the 2011 Tohoku Earthquake in Japan. *Earthquake Engineering & Structural Dynamics, 43*(12), 1763-1784. doi:doi:10.1002/eqe.2422

Cimellaro, G. P., Renschler, C., Reinhorn, A. M., and Arendt, L. (2016). PEOPLES: A Framework for Evaluating Resilience. *Journal of Structural Engineering, 142*(10), 4016063.

Cohen, L. E., and Felson, M. (1979). Social Change and Crime Rate Trends: A Routine Activity Approach. *American Sociological Review, 44*(4), 588-608.

Cornish, D. B., and Clarke, R. V. G. (1986). *The reasoning criminal : rational choice perspectives on offending*.

Cutler, H., Shields, M., Tavani, D., and Zahran, S. (2016). Integrating engineering outputs from natural disaster models into a dynamic spatial computable general equilibrium model of Centerville. *Sustainable and Resilient Infrastructure, 1*(3-4), 169-187. doi:10.1080/23789689.2016.1254996

Deodatis, G. (1996). Simulation of Ergodic Multivariate Stochastic Processes. *Journal of Engineering Mechanics, 122*(8), 778-787.

DHS-OIG (2006). *A Performance Review of FEMA's Disaster Management Activities in Response to Hurricane Katrina*. (OIG-06-32). U.S. Department of Homeland Security's Office of Inspector General. Retrieved from https://www.oig.dhs.gov/assets/Mgmt/OIG_06-32_Mar06.pdf.

DHS-OIG (2011). *Assessment of FEMA's Fraud Prevention Efforts*. (OIG-11-84). U.S. Department of Homeland Security's Office of Inspector General. Retrieved from https://www.oig.dhs.gov/assets/Mgmt/OIG_11-84_May11.pdf.

DHS (2006). *Performance and Accountability Report for Fiscal Year 2006*. U.S. Department of Homeland Security. Retrieved from https://www.dhs.gov/sites/default/files/publications/cfo_par2006_fullreport.pdf.

DOJ (2005). *Hurricane Katrina Fraud Task Force: A Progress Report to the Attorney General*. U.S. Department of Justice, Hurricane Katrina Fraud Task Force. Retrieved from

https://www.justice.gov/sites/default/files/criminal-disasters/legacy/2012/07/30/KatrinaProgressReport10-18-05.pdf.

DOJ (2006a). *Hurricane Katrina Fraud Task Force: A Progress Report to the Attorney General (February 2006)*. U.S. Department of Justice, Hurricane Katrina Fraud Task Force. Retrieved from https://www.justice.gov/sites/default/files/criminal-disasters/legacy/2012/07/30/katrinareportfeb2006.pdf.

DOJ (2006b). *Hurricane Katrina Fraud Task Force: First Year Report to the Attorney General (September 2006)*. U.S. Department of Justice, Hurricane Katrina Fraud Task Force. Retrieved from https://www.hsdl.org/?view&did=466863.

DOJ (2007). *Hurricane Katrina Fraud Task Force: Second Year Report to the Attorney General*. U.S. Department of Justice, Hurricane Katrina Fraud Task Force. Retrieved from https://www.justice.gov/sites/default/files/criminal-disasters/legacy/2012/07/30/09-04-07AG2ndyrprogrpt.pdf.

DOJ (2010). *Hurricane Katrina Fraud Task Force: Fifth Anniversary Report to the Attorney General*. U.S. Department of Justice, Hurricane Katrina Fraud Task Force. Retrieved from https://www.justice.gov/sites/default/files/criminal-disasters/legacy/2012/07/30/09-13-10katrinaprogress-report.pdf.

DOJ (2011). *Disaster Fraud Task Force: Report to the Attorney General for Fiscal Year 2011*. U.S. Department of Justice, National Center for Disaster Fraud. Retrieved from https://www.justice.gov/sites/default/files/criminal-disasters/legacy/2013/04/04/ReportDFTF2011.pdf.

Dray, A., Ritter, A., Perez, P., and Mazerolle, L. (2008). Policing Australia's 'heroin drought': Using an agent-based model to simulate alternative outcomes. *Journal of Experimental Criminology, 4*(3), 267-287.

Dudenhoeffer, D. D., Permann, M. R., and Manic, M. (2006). *CIMS: A Framework for Infrastructure Interdependency Modeling and Analysis.* Paper presented at the Proceedings of the 2006 Winter Simulation Conference.

Dueñas-Osorio, L., Craig, J. I., and Goodno, B. J. (2007). Seismic response of critical interdependent networks. *Earthquake Engineering & Structural Dynamics, 36*(2), 285-306. doi:doi:10.1002/eqe.626

Ellingwood, B. R., Cutler, H., Gardoni, P., Peacock, W. G., van de Lindt, J. W., and Wang, N. (2016). The Centerville Virtual Community: a fully integrated decision model of interacting physical and social infrastructure systems. *Sustainable and Resilient Infrastructure, 1*(3-4), 95-107. doi:10.1080/23789689.2016.1255000

Epstein, J. M. (2006). *Generative social science : studies in agent-based computational modeling*: Princeton University Press.

Eusgeld, I., Henzi, D., and Kröger, W. (2008). *Comparative Evaluation of Modeling and Simulation Techniques for Interdependent Critical Infrastructures*. Retrieved from Laboratory for Safety Analysis, ETH Zurich, Zurich, Swiss:

Eusgeld, I., and Nan, C. (2009). *Creating a simulation environment for critical infrastructure interdependencies study.* Paper presented at the Prod. of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM).

Eusgeld, I., Nan, C., and Dietz, S. (2011). "System-of-systems" Approach for Interdependent Critical Infrastructures. *Reliability Engineering and System Safety, 96*(6), 679-686. doi:https://doi.org/10.1016/j.ress.2010.12.010

Fang, J., El-Tawil, S., and Aguirre, B. (2016a). New Agent-Based Egress Model Allowing for Social Relationships. *Journal of Computing in Civil Engineering, 30*(4), 4015066.

Fang, J., El-Tawil, S., and Aguirre, B. (2016b). Leader–follower model for agent based simulation of social collective behavior during egress. *Safety Science, 83*, 40-47. doi:https://doi.org/10.1016/j.ssci.2015.11.015

Fatemi, F., Ardalan, A., Mohammadfam, I., Mansouri, N., and Aguirre, B. (2017). Social vulnerability indicators in disasters: Findings from a systematic review. *International Journal of Disaster Risk Reduction, 22*, 219-227.

FEMA (2003). *Earthquake loss estimation methodology: Technical manual*. National Institute of Building for the Federal Emergency Management Agency, Washington, DC.

FEMA (2006). *Multi-hazard Loss Estimation Methodology, Hurricane Model, HAZUS, Technical Manual*. Federal Emergency Management Agency (FEMA), Washington, D.C.

FEMA (2018). *Notice of Maximum Amount of Assistance under the Individuals and Households Program*. Federal Emergency Management Agency, Washington, DC. Retrieved from https://www.federalregister.gov/d/2018-22884

Fiedrich, F. (2006). *An HLA-based multiagent system for optimized resource allocation after strong earthquakes.* Paper presented at the Proc., 2006 Winter Simulation Conference.

Fritz, C. E. (1996). *Disasters and mental health : therapeutic principles drawn from disaster studies*. Newark, DE: University of Delaware Disaster Research Center.

Fujimoto, R. (2015). *Parallel and distributed simulation.* Paper presented at the 2015 Winter Simulation Conference (WSC).

GAO (2006a). *Hurricanes Katrina and Rita Disaster Relief: Improper and Potentially Fraudulent Individual Assistance Payments Estimated to Be Between $600 Million and $1.4 Billion*. (GAO-06-844T). U.S. Government Accountability Office. Retrieved from https://www.gao.gov/assets/120/114055.pdf.

GAO (2006b). *Expedited Assistance for Victims of Hurricanes Katrina and Rita: FEMA's Control Weaknesses Exposed the Government to Significant Fraud and Abuse*. (GAO-06-403T). U.S. Government Accountability Office. Retrieved from https://www.gao.gov/assets/260/250473.pdf.

GAO (2006c). *Hurricanes Katrina and Rita: Unprecedented Challenges Exposed the Individuals and Households Program to Fraud and Abuse; Actions Needed to Reduce Such Problems in Future*. (GAO-06-1013). U.S. Government Accountability Office. Retrieved from https://www.gao.gov/new.items/d061013.pdf.

GAO (2006d). *Hurricanes Katrina and Rita Disaster Relief: Continued Findings of Fraud, Waste, and Abuse*. (GAO-07-252T). U.S. Government Accountability Office. Retrieved from https://www.govinfo.gov/content/pkg/GAOREPORTS-GAO-07-252T/pdf/GAOREPORTS-GAO-07-252T.pdf.

GAO (2007). *Hurricanes Katrina and Rita Disaster Relief: Prevention Is the Key to Minimizing Fraud, Waste, and Abuse in Recovery Efforts*. (GAO-07-418T). U.S. Government Accountability Office. Retrieved from https://www.gao.gov/assets/120/115223.pdf.

GAO (2014). *Hurricanes Sandy: FEMA Has Improved Disaster Aid Verification but Could Act to Further Limit Improper Assistance*. (GAO-15-15). U.S. Government Accountability Office. Retrieved from https://www.gao.gov/assets/670/667469.pdf

Garcia-Valls, M., and Basanta-Val, P. (2013). Usage of DDS Data-Centric Middleware for Remote Monitoring and Control Laboratories. *IEEE Transactions on Industrial Informatics, 9*(1), 567-574. doi:10.1109/TII.2012.2211028

GCEER. (2011). *Grand Challenges in Earthquake Engineering Research: A Community Workshop Report*. Washington, DC: The National Academies Press.

Ghosn, M., Duenas-Osorio, L., Frangopol, D. M., McAllister, T. P., Bocchini, P., Manuel, L., Ellingwood, B. R., Arangio, S., Bontempi, F., Shah, M., Akiyama, M., Biondini, F., Hernandez, S., and Tsiatas, G. (2016). Performance indicators for structural systems and infrastructure networks. *Journal of Aerospace Engineering, 29*(4).

Gini, C. (1997). Concentration and dependency ratios. *Rivista di Politica Economica (Italian), 87*, 769-789.

González, A. D., Dueñas-Osorio, L., Sánchez-Silva, M., and Medaglia, A. L. (2016). The Interdependent Network Design Problem for Optimal Infrastructure System Restoration. *Computer-Aided Civil and Infrastructure Engineering, 31*(5), 334-350. doi:10.1111/mice.12171

Groff, E. R. (2007). Simulation for Theory Testing and Experimentation: An Example Using Routine Activity Theory and Street Robbery. *JOURNAL OF QUANTITATIVE CRIMINOLOGY, 23*(2), 75-103.

Groff, E. R. (2014). Agent-Based Modeling for Understanding Patterns of Crime. In G. Bruinsma and D. Weisburd (Eds.), *Encyclopedia of Criminology and Criminal Justice* (pp. 32-41). New York, NY: Springer New York.

Groff, E. R., Johnson, S. D., and Thornton, A. (2019). State of the Art in Agent-Based Modeling of Urban Crime: An Overview. *JOURNAL OF QUANTITATIVE CRIMINOLOGY, 35*(1), 155-193.

Guidotti, R., Chmielewski, H., Unnikrishnan, V., Gardoni, P., McAllister, T., and van de Lindt, J. (2016). Modeling the resilience of critical infrastructure: the role of network dependencies. *Sustainable and Resilient Infrastructure, 1*(3-4), 153-168. doi:10.1080/23789689.2016.1254999

Hazus. (2015). *Federal Emergency Management Agency.* Accessed Nov. 19, 2015. https://www.fema.gov/hazus-software/hazus-quick-reference-guide

Hernandez-Fajardo, I., and Dueñas-Osorio, L. (2011). Sequential Propagation of Seismic Fragility across Interdependent Lifeline Systems. *Earthquake Spectra, 27*(1), 23-43. doi:10.1193/1.3544052

Hernandez-Fajardo, I., and Dueñas-Osorio, L. (2013). Probabilistic study of cascading failures in complex interdependent lifeline systems. *Reliability Engineering and System Safety, 111*, 260-272. doi:10.1016/j.ress.2012.10.012

Hokamp, S. (2014). Dynamics of tax evasion with back auditing, social norm updating, and public goods provision – An agent-based simulation. *Journal of Economic Psychology, 40*, 187-199.

Hollenbach, J. W. (2009). *Inconsistency, neglect, and confusion; A historical review of DoD distributed simulation architecture policies.* Paper presented at the Simulation Interoperability Standards Organization - Spring Simulation Interoperability Workshop 2009.

Huang, A. S., Olson, E., and Moore, D. C. (2010). *LCM: Lightweight Communications and Marshalling.* Paper presented at the Proc., 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems.

ICoR. (2019). Interdependencies in Community Resilience (ICoR) Project. Accessed Dec. 15, 2019. https://icor.engin.umich.edu/

IEEE. (1993). IEEE Standard for Information Technology - Protocols for Distributed Interactive Simulations Applications. Entity Information and Interaction. *IEEE Std 1278-1993*. doi:10.1109/IEEESTD.1993.115125

IEEE. (2010a). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules (Vol. IEEE 1516-2010). New York, NY: IEEE.

IEEE. (2010b). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Federate Interface Specification (Vol. IEEE 1516.1-2010). New York, NY: IEEE.

IEEE. (2010c). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Object Model Template (OMT) Specification (Vol. IEEE 1516.2-2010). New York, NY: IEEE.

IEEE. (2012). IEEE Standard for Distributed Interactive Simulation--Application Protocols. *IEEE Std 1278.1-2012 (Revision of IEEE Std 1278.1-1995)*, 1-747. doi:10.1109/IEEESTD.2012.6387564

IEEE. (2015). IEEE Standard for Distributed Interactive Simulation (DIS) -- Communication Services and Profiles. *IEEE Std 1278.2-2015 (Revision of IEEE Std 1278.2-1995)*, 1-42. doi:10.1109/IEEESTD.2015.7459689

Jain, S., and McLean, C. R. (2006). *An Integrated Gaming and Simulation Architecture for Incident Management Training*. Retrieved from Gaithersburg, Maryland

Jeffery, C. R. (1965). Criminal Behavior and Learning Theory. *The Journal of Criminal Law, Criminology, and Police Science, 56*(3), 294-300.

Johnson, S. D., Bernasco, W., Bowers, K. J., Elffers, H., Ratcliffe, J., Rengert, G., and Townsley, M. (2007). Space–Time Patterns of Risk: A Cross National Assessment of Residential Burglary Victimization. *JOURNAL OF QUANTITATIVE CRIMINOLOGY, 23*(3), 201-219.

Khaefi, M. R., Jin-Yong, I., and Dong-Seong, K. (2015). *An efficient DDS node discovery scheme for naval combat system.* Paper presented at the 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA).

Kim, Y., Zhong, W., and Chun, Y. (2013). Modeling sanction choices on fraudulent benefit exchanges in public service delivery. *JASSS, 16*(2).

Koliou, M., van de Lindt, J. W., McAllister, T. P., Ellingwood, B. R., Dillard, M., and Cutler, H. (2018). State of the research in community resilience: progress and challenges. *Sustainable and Resilient Infrastructure*, 1-21. doi:10.1080/23789689.2017.1418547

Kwoka, J. E., Jr. (1985). The Herfindahl index in theory and practice. *Antitrust Bulletin, 30*(4), 915–947.

LCM. (2018). Lightweight Communications and Marshalling (LCM). Accessed Nov. 1, 2018. https://lcm-proj.github.io/

Lee, E. E., Mitchell, J. E., and Wallace, W. A. (2007). Restoration of Services in Interdependent Infrastructure Systems: A Network Flows Approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37*(6), 1303-1317. doi:10.1109/TSMCC.2007.905859

Lin, P., and Wang, N. (2016). Building portfolio fragility functions to support scalable community resilience assessment. *Sustainable and Resilient Infrastructure, 1*(3-4), 108-122. doi:10.1080/23789689.2016.1254997

Loggins, R., Little, R. G., Mitchell, J., Sharkey, T., and Wallace, W. A. (2019). CRISIS: Modeling the Restoration of Interdependent Civil and Social Infrastructure Systems Following an Extreme Event. *Natural Hazards Review, 20*(3), 04019004. doi:doi:10.1061/(ASCE)NH.1527-6996.0000326

Malleson, N., See, L., and Heppenstall, A. (2010). Crime reduction through simulation: An agent-based model of burglary. *Computers, Environment and Urban Systems, 34*(3), 236-250.

Miles, S. B., and Chang, S. E. (2003). *Urban Disaster Recovery: A Framework and Simulation Model*. Retrieved from http://dx.doi.org/10.1193/1.2192847

Miles, S. B., and Chang, S. E. (2007). *A simulation model of urban disaster recovery and resilience: implementation for the 1994 Northridge earthquake*. Retrieved from

Miles, S. B., and Chang, S. E. (2011). ResilUS: A Community Based Disaster Resilience Model. *Cartography and Geographic Information Science, 38*(1), 36-51. doi:10.1559/1523040638136

Miller, J. M., Schreck, C. J., and Tewksbury, R. A. (2011). *Criminological theory : a brief introduction*.

Moreno, J., and Shaw, D. (2019). Community resilience to power outages after disaster: A case study of the 2010 Chile earthquake and tsunami. *International Journal of Disaster Risk Reduction*. doi:10.1016/j.ijdrr.2018.12.016

Morshedlou, N., González, A. D., and Barker, K. (2018). Work crew routing problem for infrastructure network restoration. *Transportation Research Part B, 118*, 66-89. doi:10.1016/j.trb.2018.10.001

NER (2011). *National Earthquake Resilience: Research, Implementation, and Outreach*. (978-0-309-18677-3). NER, Washington, DC. Retrieved from https://www.nap.edu/catalog/13092/national-earthquake-resilience-research-implementation-and-outreach.

NHC (2006). *Tropical Cyclone Report Hurricane Katrina*. National Hurricane Center. Retrieved from https://www.nhc.noaa.gov/data/tcr/AL122005_Katrina.pdf.

NIST (2010). *Evaluation of the FEMA P-695 Methodology for Quantification of Building Seismic Performance Factors*. Retrieved from Gaithersburg, MD:

NIST (2016). *Community resilience planning guide for buildings and infrastructure systems, Vols. I and II*. NIST.

OMG. (2015). *Data Distribution Service (DDS), Version 1.4* O. M. Group (Ed.) Retrieved from http://www.omg.org/spec/DDS/1.4/

Ouyang, M. (2014). Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability Engineering and System Safety, 121*, 43-60.

PEER. (2018). PEER Ground Motion Database. *Pacific Earthquake Engineering Research Center.* Accessed Jan., 2018. https://ngawest2.berkeley.edu/

Postel, J. (1980). "User Datagram Protocol" (Vol. STD 6): RFC 768.

Powell, E. T., and Noseworthy, J. R. (2012). The Test and Training Enabling Architecture (TENA). In A. Tolk (Ed.), *Engineering Principles of Combat Modeling and Distributed Simulation*. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Prelog, A. J. (2016). Modeling the Relationship between Natural Disasters and Crime in the United States. *Natural Hazards Review, 17*(1), 4015011.

Reilly, A. C., Guikema, S. D., Zhu, L., and Igusa, T. (2017). Evolution of vulnerability of communities facing repeated hazards. *PLOS ONE, 12*(9), e0182719. doi:10.1371/journal.pone.0182719

Renschler, C. S., Frazier, A. E., Arendt, L. A., Cimellaro, G. P., Reinhorn, A. M., and Bruneau, M. (2010). *Developing the 'PEOPLES' resilience framework for defining and measuring disaster resilience at the community scale.* Paper presented at the Proceedings of the 9th US national and 10th Canadian conference on earthquake engineering (9USN/10CCEE), Toronto.

Rinaldi, S. M., Peerenboom, J. P., and Kelly, T. K. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems, 21*(6), 11-25. doi:10.1109/37.969131

Schmeltz, M. T., González, S. K., Fuentes, L., Kwan, A., Ortega-Williams, A., and Cowan, L. P. (2013). Lessons from Hurricane Sandy: a Community Response in Brooklyn, New York. *Journal of Urban Health, 90*(5), 799-809. doi:10.1007/s11524-013-9832-9

Schoenwald, D. A., Barton, D. C., and Ehlen, M. A. (2004). *An agent-based simulation laboratory for economics and infrastructure interdependency.* Paper presented at the Proceedings of the 2004 American Control Conference, June 30 2004-July 2 2004.

Shaw, C. R., and McKay, H. D. (1942). *Juvenile delinquency and urban areas, a study of rates of delinquents in relation to differential characteristics of local communities in American cities*.

Shinozuka, M., Dong, X., Xianhe, J., and Cheng, T. C. (2005). *Seismic Performance Analysis for the LADWP Power System.* Paper presented at the 2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific, 18-18 Aug. 2005.

Sivanerupan, S., Wilson, J. L., Gad, E. F., and Lam, N. T. K. (2014). Drift Performance of Point Fixed Glass Façade Systems. *Advances in Structural Engineering, 17*(10), 1481-1495. doi:10.1260/1369-4332.17.10.1481

Song, J., and Ok, S.-Y. (2010). Multi-scale system reliability analysis of lifeline networks under earthquake hazards. *Earthquake Engineering and Structural Dynamics, 39*(3), 259-279. doi:10.1002/eqe.938

Spencer, N. O. (2017). Look what the hurricanes just blew in: analyzing the impact of the storm on criminal activities. *Journal of Crime and Justice, 40*(4), 417-429.

SRTI. (2019). Simple Real Time Interface (SRTI). Accessed Jan. 15, 2019. https://github.com/hlynka-a/SRTI

Tibbetts, S. G. (2018). *Criminological Theory: The Essentials*: SAGE Publications.

USGS. (2018). Memphis, Shelby County Seismic Hazard Maps and Data. Accessed Feb., 2018. https://earthquake.usgs.gov/hazards/urban/memphis/grid_download.php#Deterministic

Wilensky, U. (1999). NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL. Accessed http://ccl.northwestern.edu/netlogo/.

Xu, L., Lin, S.-Y., Hlynka, A. W., Lu, H., Kamat, V. R., MenassaC.C., El-Tawil, S., A., P., Spence, S. M. J., and McCormick, J. (2020). Distributed Simulation Platforms and Data Passing Tools for Natural Hazards Engineering: Reviews, Limitations, and Recommendations. *Advanced Engineering Informatics, under review*.

Zahran, S., Shelley, T. O. C., Peek, L., and Brody, S. D. (2009). Natural Disasters and Social Order: Modeling Crime Outcomes in Florida. *International Journal of Mass Emergencies and Disasters, 27*(1), 26-52.

Zhang, P., and Peeta, S. (2011). A generalized modeling framework to analyze interdependencies among infrastructure systems. *Transportation Research Part B: Methodological, 45*(3), 553-579. doi:10.1016/j.trb.2010.10.001

Zimmerman, R. (2001). Social Implications of Infrastructure Network Interactions. *Journal of Urban Technology, 8*(3), 97-119. doi:10.1080/106307301753430764