Western Washington University
Western CEDAR

WWU Graduate School Collection
WWU Graduate and Undergraduate Scholarship

Winter 2018

# Discrete-event Processes in Disaster Recovery: A Conceptual Framework and Simulation Model of Household Reconstruction in Pacific County Washington

Derek Huling
*Western Washington University*, derek.huling@gmail.com

Follow this and additional works at: https://cedar.wwu.edu/wwuet

Part of the Geography Commons

## Recommended Citation

Huling, Derek, "Discrete-event Processes in Disaster Recovery: A Conceptual Framework and Simulation Model of Household Reconstruction in Pacific County Washington" (2018). *WWU Graduate School Collection*. 636.
https://cedar.wwu.edu/wwuet/636

This Masters Thesis is brought to you for free and open access by the WWU Graduate and Undergraduate Scholarship at Western CEDAR. It has been accepted for inclusion in WWU Graduate School Collection by an authorized administrator of Western CEDAR. For more information, please contact westerncedar@wwu.edu.

# **Discrete-event Processes in Disaster Recovery: A Conceptual and Simulation Model in Pacific County Washington**

By

Derek Huling

Accepted in Partial Completion
Of the Requirements for the Degree
Master of Science

ADVISORY COMMITTEE

Chair, Dr. Rebekah Paci-Green

Dr. Scott B. Miles

Dr. Nabil Kamel

Dr. Gautam Pillay
Dean of the Graduate School

**Master's Thesis**

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Western Washington University, I grant to Western Washington University the non-exclusive royalty-free right to archive, reproduce, distribute, and display the thesis in any and all forms, including electronic format, via any digital library mechanisms maintained by WWU.

I represent and warrant this is my original work, and does not infringe or violate any rights of others. I warrant that I have obtained written permissions from the owner of any third party copyrighted material included in these files.

I acknowledge that I retain ownership rights to the copyright of this work, including but not limited to the right to use all or part of this work in future works, such as articles or books.

Library users are granted permission for individual, research and non-commercial reproduction of this work for educational purposes only. Any further digital posting of this document requires specific permission from the author.

Any copying or publication of this thesis for commercial purposes, or for financial gain, is not allowed without my written permission

Signature:  Derek Huling

Date: Feburary 20th, 2018

**<u>Discrete-event Processes in Disaster Recovery:</u>**

# A Conceptual and Simulation Model in Pacific County Washington

A Thesis
Presented to
The Faculty of
Western Washington University

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

by
Derek Huling

February 2018

## Abstract

Techniques and frameworks to facilitate modeling, simulation, and visualization of disaster recovery concepts are tools to help researchers unpack the complex web of processes and events undertaken by households and other actors participating in recovery.

This research describes a conceptual framework of owner-occupied housing reconstruction consisting of various events, processes, resources, and their interactions with and among entities (representing owner-occupied households). For example, the process of household reconstruction involves potentially many events: building inspections, fulfilment of financial capital requests, contractual agreements with building contractors, etc.

Elements of this conceptual framework are applied to a discrete-event simulation (DES) to simulate the interactions and outcomes of owner-occupied household reconstruction in a case study area of Pacific County, Washington. This simulation uses the SimPy discrete-event simulation development library for the Python programming language, within a probabilistic structure to monitor, assess, and return outcomes related to household reconstruction. Households interact with shared resources to determine the duration of household reconstruction.

The resulting simulation of owner-occupied household reconstruction shows promise of assembling simulations in a “building blocks” manner, in which researchers can assemble simulations based on their own scenario interests. With the addition of quality and significant parameterization, and increased quantity of resources modeled, this simulation could be used to develop and support pre- and post- disaster decision making and planning activities in the emergency management field.

**Acknowledgements**

I would like to thank my committee: Dr. Rebekah Paci-Green, Dr. Scott Miles, and Dr. Nabil Kamel.
Special thanks to Dr. Scott Miles for allowing me to take on this project, helping extend and improve my analytical and coding skills.

The following individuals kept me sane during the process:
Jacob Lesser, Todd Ellis, Sarah Lindell, Shelby Van Arnam, Andrea Watson, Anthony Wens

Thanks to the SimPy team for creating an easy library to build from.

Thanks to NOAA and Washington SeaGrant for helping fund portions of this project (Award #NA14OAR4170078)

# Table of Contents

**List of Figures and Tables**

## 1. Introduction

Interest in the field of recovery has continued to grow as the frequency, damage, and severity of natural disasters grow, whether due to increasing population density and growth in hazard areas, or as larger, more powerful storms to damage human settlements (Huggel et al., 2015). This interest is often rooted in questions such as: how can we make recovery faster, easier, cheaper, and more effective? How can the impact of improved mitigation, planning, and response techniques be measured if we cannot measure the status of recovery effectively?

Understanding the underlying factors and elements of the complex system of recovery in a community context is necessary to assist in the measurement, analysis, and modeling of disaster recovery outcomes. Researchers have used frameworks and conceptual models to assist them in formulating theories and explanations to common phenomenon across disasters. To help explore and unpack phenomena involving interactions among actors within recovery, simulation modeling is a promising analytical technique applied within the field. Many different techniques exist as to how data is simulated. Nejat & Damnjanovic (2012) uses a LASSO model in combination with decision logic extracted from course surveys, exploring neighborhood reconstruction decisions among neighbors. Miles & Chang, (2011) developed ResilUS, an agent-based, time-iterative model programmed in MATLAB through Simulink, exploring business and housing recovery. More analysis of these simulations, and others, are explored in Chapter Three.

Continuing the progress of simulation and modeling in the field of recovery, the objective of this thesis is to describe a conceptual framework for characterizing processes and events that occur over the course of owner-occupied housing reconstruction. Based on this framework, a modular building block style of simulation is developed, and a test simulation of owner-occupied housing reconstruction based on that framework is conducted. Situating the research in the greater body of modeling in works such as Wisner et al's (2004) work "*At Risk: Natural Hazards, People's Vulnerability and Disasters,*" this research conceptualizes one component of post-disaster owner-occupied housing reconstruction through the lens of actors' access to resources needed for homeowner reconstruction. This conceptual framework is then used as a blueprint to create an applied model of owner-occupied housing reconstruction using discrete-event simulation. This research is driven by two main questions:

1) How can researchers effectively group housing reconstruction functions in a framework to inform a useful model?
2) How can these owner-occupied housing reconstruction functions and processes be integrated into a simulation model that can be used as a modular exploration tool of housing reconstruction?

To situate the questions and objectives in the current literature, the next chapter, Chapter Two, reviews recovery and housing reconstruction literature. The first section in Chapter Two explores processes commonly undertaken by households after disaster to reconstruct their dwellings. The second section of Chapter Two reviews these processes in the context of three different disasters in the United States: the 1994 Northridge Earthquake, 2012 Hurricane Sandy, and 2011 Joplin Missouri Tornado. These events were chosen for their disparity in time, type, geography, and duration of event. The final section of Chapter Two outlines and describes a conceptual framework of elements derived from the previous process breakdown. This is used as a framework to build the simulation and ground it in a conceptual framework.

Chapter Three presents a literature review on theoretical and simulation models used in the field. The first section describes simulation as decision support systems. The second section discusses current simulations used in the wider field of recovery, and those used in housing reconstruction. The third section looks at the methodology of discrete-event simulation. The fourth describes the different techniques used to validate and verify the model outputs and design. The final section provides a summary of the chapter and helps to situate this research into the context of the simulation literature.

Chapter Four is a description of the methods used to both: 1) describe the building of the conceptual framework, 2) describe construction of the simulation model, 3) describe efforts to validate and verify the models, and 4) a short overview of the case study area.

Chapter Five concentrates on results and discussion, and contains three subsections with the following tasks: 1) describing the results of the conceptual framework, including examples of each element of the model and how it might be generalized to model a variety of cases in recovery; 2) detailing the results of the simulation developed from the elements in the conceptual framework, and 3) discussing limitations of both the functionality and the content of the simulation model.

Chapter Six concludes the research by returning to the research questions and objectives, assessing how the simulation model and conceptual framework might be used in the context of modeling disasters. It also discusses the possible further steps in research.

## 2. Housing Reconstruction

This chapter offers a review of common processes that homeowners go through while participating in recovery split into three sections: bureaucratic, financial, and reconstruction processes. Subsequently it situates these processes into a brief review of three disaster events: the 2011 Joplin tornado, the 1994 Northridge earthquake, and the 2012 Hurricane Sandy. These three events all differ considerably in type and size of event, geography, and duration of recovery and provide an overview of housing recovery processes common to disasters, regardless of the natural hazard trigger event.

### 2.1 Housing Restoration Processes

This section reviews groups of processes that homeowners typically go through to rebuild their housing. Some of this description is based in news articles and the case studies listed in section 2.2.3, while the rest is situated in the academic body of literature. The three groups are bureaucratic, financial, and reconstruction processes. Bureaucratic processes -- individuals dealing with the government or a business for non-financial resources -- include inspections, permitting, zoning, and assessments. Financial processes households may experience during recovery include applying for loans, getting financial assistance from FEMA, or making insurance claims. Reconstruction processes, the final group of processes, include finding contractors, building materials, and rebuilding the structure to make habitable. Throughout these groups, households also engage in migration and movement and decisions to buy undamaged housing, also reviewed in this section. Those who rent housing instead of own are not considered in this review nor modeled in the simulation presented in Chapter 4.

### 2.2.1 Bureaucratic Processes

There are several different types of building inspections, often done by different stakeholders with different goals. Some examples include: inspecting buildings for habitability, reconstruction inspections (also called engineering assessments) that ensure adherence to building codes, as well as public (National Flood Insurance Program) and private insurance inspections to determine claim awards.

Post-disaster building inspections are intended to assess the immediate livability status of building stock from a casualty reduction point of view , for example collapse risk from aftershocks, additional flooding, etc. (FEMA, 2015). Building inspections support several other emergency response operations, such as debris removal, demolition of buildings beyond repair, estimation of homeless and shelter needs, and benefit entitlement, as well as for various reconstruction purposes (Dandoulaki, Panoutsopoulou, & Ioannides, 1998), although there are not standard definitions for inspections, as they generally have different end goals.

Several actors are involved in the building inspection process. The local, state, and federal governments often do their own rapid assessment of building stock loss. This rapid assessment is used to assess the need and quantity of emergency/temporary shelter, and to estimate the amount of potential new homeless populations (Dandoulaki et al., 1998; Erdik et al., 2011; Lindell, 2013).

These assessments take many forms depending on the extent of damage and access to the labor necessary to undertake such assessments. If a homeowner is seeking emergency funds from an organization like FEMA or the SBA, that agency will have their own inspection teams to determine livability and claim amount the homeowner receives.

The time-compressed nature of disaster recovery, a concept expanded upon in chapter three, puts a strain on the amount of available building inspectors, causing a bottleneck in the recovery process (Dandoulaki et al., 1998; Gharaati, 2006; Lindell, 2013). Podger (2013) offers suggestions to augment the inspector stock include simplifying the inspection process, for example, by not doing cost estimates, cross-training emergency staff to include inspection knowledge, and deputizing temporary inspectors to complete non-structural habitability inspections. Data is currently unavailable about the efficacy of these interventions.

Insurance companies are another important stakeholder involved in building inspections, though these insurance inspections often go by another name: claim assessments. Claim assessments are about valuating damage, both structure and content, and summing up insurance claim amounts issued to homeowners. Claim assessments are made against hazard coverage obtained prior to the event. In both building inspections, and claim assessments, speed is an important factor (Lindell, 2013). Yet, data about insurance assessments and inspections are largely unavailable to public scrutiny outside of the court system, making insurance assessments and efficacy difficult to study, especially across different disaster types where insurance coverage rates and premiums can vary substantially, e.g.

flood insurance in a flood plain, or earthquake insurance in an earthquake prone area (Weston, 2017). Additionally accusations of fraud and corruption within the inspection and claims industry (Cushman, 2015) may suggest that the opacity of inspection and claims serve to slow the process of accessing insurance as a financial resource by homeowners in a reasonable amount of time..

Once design and structural details are met, the construction company, architect, design engineer, or the homeowner themselves must apply for a building permit from the administrative unit, e.g. city, or county if in unincorporated area. This application process is another factor of time that, in the event of a disaster, compresses and can be a major bottleneck for reconstruction (Stevenson, Emrich, Mitchell, & Cutter, 2010). Factors that may hinder receipt of a building permit include changes in building codes, lack of permitting officials, rezoning, or inability to rebuild due to landscape changes or unsafe conditions. A delay in receiving a building permit can effectively halt construction: one might have a valid building inspection, building materials sourced, a construction crew hired, and an authorized architectural plan, but be stymied in the permit application stage. Yet, reducing the efficacy of building codes by eliminating bottleneck processes such as permitting can increase vulnerability (Alexander, 2004).

### 2.2.2 Financial

Acquiring reconstruction funds is one of the primary hurdles in the ability and decision of a homeowner to rebuild housing (Wu & Lindell, 2004). The funding mechanisms available to homeowners varies incredibly by disaster types, for example flood insurance only available during floods and storm surges, and non-presidentially declared disaster event survivors having no access to individual assistance grants from FEMA. A homeowners' ability to finance their reconstruction may lie at a complex intersection of personal, private, and public funding including: aid and charity, insurance, loans, savings, and retirement investments. Public funding can come from multiple levels of government such as state programs, state emergency management, or from federal sources such as FEMA and the Small Business Administration. Private funding can come from banks in the form of loans and insurance coverage. Personal funding can come from any liquefiable assets owned by the homeowner, such as savings and retirements accounts. Traversing the network of available funding for a homeowner can be a web of qualifications, applications, rules, legal battles, and long durations of waiting. Public, private, and personal funding options vary considerably per-homeowner.

Public sources of financing can be aid-based, in which case grants or other assistance is offered to a homeowner with no penalty or charge (interest, or debt). This aid frequently comes in the form of FEMA Individual Assistance grants, in which claim payouts are limited to $33,000 per household (Stafford, 2013). FEMA requires any insurance policies to be declared before aid decisions are rendered. Therefore, homeowners may need to wait 10 to 14 days to get a complete inspection, generally within 30 days of the event, after they file their FEMA claim. A city or state often erects post-disaster recovery programs with a mixture of funds bolstered by public assistance from the federal government, and payouts vary by the program and its requirements, e.g. New York City's "Build It Back" after

Hurricane Sandy, and Louisiana's "Road Home" program after Hurricane Katrina, both of which funneled funds directly from the federal government. These two programs experienced pitfalls and financing issues, beyond the scope of this research, however they were and are sources of funding available to the rebuilding homeowner. Public financing can also be loan-based, in which a homeowner has access to rebuilding funds that they must repay (such as SBA loans).

Private sources of funding are myriad. NGO relief organizations such as the Red-Cross, Habitat for Humanity, and others provide funding, shelter, and others life-stabilizing resources in the immediate aftermath and longer term, they're unable to meet full need and come with their own sets of qualifications and bottlenecks. Additionally, large and small lending institutions (banks) qualify individuals for home and repair loans.

Disparities can arise between the damage assessment provided by the insurance company and what the contracting company charges for actual reconstruction labor. There are various reasons for this disparity: construction crews are in high demand for the months and even years following large-scale disasters; insurance companies attempt to "low-ball" or provide the least amount of settlement money as possible, due to the strain put on them by the sheer volume of claims; and fraudulent companies can appear, demand up front deposit, and evade completing work.

Homeowners utilizing the financial help of private loans are subsequently thrust into debt, or further into debt. This debt burden may retard their ability to recover adequately and quickly as they recover lost equity in their homes. Home equity is often the largest and most stable asset in the portfolio of an American homeowner (Munnell, Soto, & Aubry, 2007), and reducing the value of that asset often disrupts their ability to retire, or live in the same manner as before the disaster event, often an indicator of being “fully recovered.” Households that cannot reconcile the financial disparity with their own savings typically have several options. If possible, they can sell their property, resulting in a choice between moving out of the community, renting within the community given sufficient rental housing vacancies, or buying a less desirable house in the community. The decision to stay or go from the community, as previously described, can be explained by factors ranging from fear, trauma, employment access, school access, and many other variables unrelated to housing.

### 2.2.3 Reconstruction

The majority of research conducted on household reconstruction and recovery is done at the event level via case studies and comparative studies. Wu and Lindell (2004) found that the process of pre-disaster planning was helpful when comparing the outcomes of the Northridge earthquake and the Chi Chi earthquake in Taiwan, however their analysis of outcomes was focused on maximizing hazard mitigation by examining the extent to which land use planning, comprehensive planning, and disaster recovery programs were included in housing-related policies. The other variable explored was speed of housing reconstruction using building permits as a proxy. Chang, Wilkinson, Potangaroa, & Seville (2010), compared three different disaster events by researching the ways resources are distributed in the reconstruction sector: market driven, government driven, and donor driven resourcing strategies. They found that a mix of all three, multi-stakeholder cooperation, and better pre-disaster planning can assist access to reconstruction resources after a disaster.

Shelter, an umbrella term in the disaster literature, is a place one can as safely as possible sleep and is generally categorized into 4 parts (Quarantelli, 1982): emergency shelter, temporary shelter, temporary housing, and permanent housing. Sometimes these categories are achieved sequentially. Emergency shelter is immediate protection from the elements post disaster e.g. a tent in a yard, a car, mass-care shelters. Temporary shelter includes staying with family or friends, community shelters, and commercial options (motels, hotels) and refers to less than ideal shelter, either by location, size, or lack of desired amenities. Permanent housing is housing in preferred locations, getting back to pre-disaster routines, the result of new home ownership, rebuilding, or renting a new property. The simulation developed for this thesis models the search for, or rebuilding of, permanent housing only; while it does consider decisions about temporary housing, it does not model individual or household decisions about emergency or temporary shelter. Getting to the point of rebuilding requires household actors to undertake many processes in order to gain access to the resources required to obtain viable permanent housing.

Type of event and extent of damage are obvious impediments to reconstruction. Earthquakes often cause significant structural and foundational damage; they may also trigger soil liquefaction, causing former building zones to be unstable and potentially stalling the permitting process. These types of damages are more expensive, take longer, and require more specialty knowledge to fix. Research has shown that the reconstruction sector of the economy booms in response to disasters, taking advantage of the huge amounts of outside aid flowing into the community (Becerra, Cavallo, & Noy, 2010; Singh & Wilkinson, 2008). This boom causes increased prices and stiffer competition among homeowners to get their properties rebuilt (Chang et al., 2010).

When homeowners have the ability to finance home reconstruction, they still may face waiting for available construction crews. This availability differs in every area and every disaster. Homeowners must source both materials and construction labor, directly or indirectly through a contractor. Post-disaster, these resources are limited in both quantity, and quality (bad labor practices, price-gouging, access to heavy machinery), although the literature indicates that often a construction boom brings out of town labor to the scene of the disaster (Chang-Richards, Wilkinson, Seville, & Brunsdon, 2015; Higuchi, Inui, Hosoi, Takabe, & Kawakami, 2012). Due to restrictions on the quantity of labor, homeowners face competition not only with other homeowners but with businesses and governmental organizations. These entities also seek reconstruction of their infrastructure and may have higher access to resources. For example, governments can prioritize building inspection/permitting; businesses may have greater access to financial capital.

Limitations on availability of building materials can also retard reconstruction efforts (Chang et al., 2010; Singh & Wilkinson, 2008). If materials cannot be sourced locally or brought in, post-disaster, construction cannot move forward. This delay may be due to transportation issues such as impaired roads, non-functioning harbors/ports, and damaged railroads. Price gouging, competition, and building codes/zoning may also contribute to reduced building material access. If there are smooth transitions in the reconstruction stages, rebuilding of the structure itself can be done in a short period of time. FEMA's HAZUS module uses general estimations for rebuild time based on building type, ranging from 30 days for mobile homes to 120 days for multi-family dwellings.

## 2.2 Recovery Processes through Events

In addition to the discussion of housing recovery processes, this chapter familiarizes the reader with three disaster events. The short descriptions of these events are intended to show the differences and similarities in terms of geography, scale, and type differences in reference to disaster. This section will explore these case studies through the housing recovery processes.

### 2.2.1 Joplin Tornado, May 22, 2011

On May 22nd, 2011 an EF-5 tornado touched down in the city of Joplin, Missouri. Joplin is located in the southwestern portion of the state and had a population of around 51,000 individuals at the time of the tornado. It was a Sunday, which meant that the vast majority of commuters were at home, and the high schools were empty. Joplin was initially a mining town. However, the largest employers at the time of the event was the healthcare sector, trucking, and manufacturing.

As a result of the tornado 161 people were killed, and extensive damage to one-third of the city was registered (NWS, 2011). An estimated 3 billion dollars in damage was exacted on the city and population of Joplin, and as much as 2 billion dollars was covered by the insurance companies to home and commercial property owners. About 7,000 homes were destroyed or rendered uninhabitable after the event.

The initial building inspection response in the Joplin tornado was mostly provided by the Structural Assessment and Visual Evaluation Coalition (SAVE), a Missouri based organization working with the state emergency management department. SAVE is comprised of over 1,000 volunteers who can be mobilized to perform trained building inspections. From May 26 to May 28, 23 SAVE teams logged more than 45 person-days by working with a City of Joplin representative to inspect damaged buildings. SAVE volunteers inspected more than 6,300 structures in Joplin, evaluating 38% of these buildings as unsafe, 6% as accessible with restrictions, and 56% as safe (Gregg & Lofton, 2011). In the instance of Joplin, while the path of the tornado was devastating, it was not a widespread disaster, therefore many inspectors could be brought in quickly. Damage to outside infrastructure was not a factor, making inspections a small hiccup in the timeline of overall recovery efforts.

Financial assistance for victims of the storm was available quickly for some, yet slower for others. The New York Times (Vigeland, 2012) reported that a close relationship with insurance agents as well as current inventories of household content may have helped some individuals get into temporary housing faster. Those who were too well insured or had too high of income were reportedly denied FEMA aid (Mummey, 2013; Vigeland, 2012).

In Joplin, 11 months after the tornado, the city had issued more than 600 permits for new homes and nearly 3,000 permits for residential repairs and rebuilding projects (Zagier, Clark, & Lieb, 2012). However, one news outlet reported that some homeowners in Joplin were denied permits due to rezoning regulations put into effect to reduce future vulnerability (Mummey, 2013). Homeowners voluntarily made mitigation decisions such as adding safe rooms, and the city adopted new code regulation changes. The city of Joplin required new homes to have both roof straps and foundation anchor bolts to conform to code. However they did not require safe rooms, or storm shelters to be built, in order to speed recovery and reduce the chance of homeowner relocation (Paul & Stimers, 2015).

### 2.2.2 Northridge Earthquake, January 17, 1994

In 1994, a magnitude 6.7 earthquake struck the Los Angeles area, causing widespread damage and disruption. The blind-thrust fault quake lasted between 20 and 30 seconds, but the measured ground velocity was the highest ever recorded in the United States, producing an extensive amount of damage in one of the highest population centers in the United States. Six major freeways, lifelines and infrastructure, as well as 49,000 housing units were damaged or destroyed in Los Angeles and Ventura counties (Comerio, 1995).

While several federal departments were involved in providing financial support and assistance to disaster survivors, the main providers were FEMA and the Department of Housing and Urban Development (HUD). FEMA provided individual assistance to help families get temporary homes and do small-scale repairs. HUD provided 20,000 extra Section 8 housing vouchers, and bumped disaster survivors to the top of the waitlist for housing (Bolin & Stanford, 1998). The breadth of the earthquake caused many insurers to begin canceling home insurance plans in order to cut their losses when it became clear that insurance obligations could run as high as $5 billion (Appleby, 1994; Schwanhausser, 1994). Comerio summed the financial payouts succinctly: "265,000 homeowners received an average of US$30,000 in insurance payments; 74,000 homeowners obtained low interest loans from the Small Business Administration, averaging US$31,000; and 288,000 homeowners received an average of US$3,000 from the Federal Emergency Management Agency's Minimal Home Repair Program" (Comerio, 1997). Additional funds came from a home loan program initiated by the city of Los Angeles Housing Department, funding 300 million dollars in low interest repair loans (D. Smith, 1995). It is likely that homeowners contended with several of these organizations to eventually get enough funding to rebuild.

Post-disaster inspectors inspected 64,000 homes, of which ~90% were labeled with a green tag, meaning they were considered livable. The remaining 10% were yellow and red tagged, meaning they had restricted entry or no entry stipulations. Insurance and claims assessments later showed that these initial inspection estimates were not representative of the entire disaster area, and later estimates showed a much wider geographic area damaged in the quake (Comerio, 1997). According to city inspectors, a home inspection after Northridge could take anywhere from 15 minutes to all day (Castaneda, 1994).

Adopted in 1987, and put into effect five days post-earthquake, Los Angeles was among a few other cities that had developed a long-term recovery and reconstruction plan prior to an event. The plan included Community Development Department and Housing Department plans to expedite building permits after an event. The majority of reconstruction efforts began between three and seven months after the earthquake according to building permits issued (Wu, 2004). Some built in delays slowed recovery, such as access to the Los Angeles Housing Departments loan program being dependent on a SBA loan rejection, a process that could take as long as a year to resolve (D. Smith, 1995).

### 2.2.3 Hurricane Sandy, October 29, 2012

In October of 2011 several storms coalesced off the Atlantic seaboard and pushed inland, affecting seven states, three of which declared major disasters. Beaches eroded, floods destroyed public and private infrastructure, and storm surges knocked out subway service in New York City, the largest city in the United States. The total direct damage was estimated to be in excess of $50 billion (Sullivan & Uccellini, 2013).

Hurricane Sandy survivors experienced bureaucratic processes at a greater scale than Joplin, more akin to the Northridge Earthquake, though with less rapid recovery times. In addition to private insurance companies and non-governmental organizations such as the Red Cross, households had to interact with several tiers of public recovery and emergency management groups (Fugate, 2013). These groups included a city reconstruction program called “Build It Back” in New York City, FEMA, and the SBA. Though the scale of housing reconstruction was similar to Northridge, complaints of mismanagement, fraud, and other abuses have plagued the recovery (as of writing, in its fifth year) (Arvedlund, 2015; Cushman, 2015). Half of the continental United States was affected in some way by the Hurricane. In contrast, Northridge, and even more acutely Joplin, were contained to a smaller localized area compared to the broader, regional effects of Hurricane Sandy. Arguably, New York and New Jersey were among the hardest hit, and much of the current research and lessons learned post-storm uses those locales as case studies. Chandeskhar and Finn’s empirical study on relocation in the Rockaway Beach area of New York City, a small archipelago off the shore of Queens, found that after 9 months, many homes were still severely damaged and uninhabited (Chandrasekhar & Finn, 2013). Similarly, Binder et. al looked at the decision to utilize home buyouts offered by the state to encourage relocation to less storm-prone areas, and its relation to community and individual level factors. They found that contextual community factors, including the history of natural disasters, local cultural norms, and sense of place, contributed to the decision of whether to accept buyouts (Binder, Baker, & Barile, 2015; Binder & Greer, 2016).

The much wider berth of damage inflicted by Hurricane Sandy inherently increased wait times for inspections, permitting, and financing (Cohen, 2013). Reporting by one news source in New York indicated one survivor endured a barrage of home inspections before any progress was made towards rebuilding her house, “after eight home inspections by various city agencies and a promise of a $250,000 grant from the Build it Back program, her house has yet to be repaired“ after two years (Porpora, 2014). An interview with the director of the Housing Recovery Operations reveals a glimpse into the public and private sector cooperation after Sandy, “You have to get certification independently from the insurance companies … There’s really no incentive for them to act quickly in turning around these requests” (G. B. Smith, 2013). While negative experiences and slow responses may not be a common experience, it demonstrates that the bureaucratic processes imposed in the aftermath of disaster can increase the waiting time.

#### 2.2.4 Conclusion

The phase of disaster recovery involves many discrete and continuous processes undertaken by actors throughout its life-cycle. This chapter sought to explore these processes, from a homeowner-centric viewpoint, in order to examine the current literature and nature of reconstruction after a hazard event. These processes can be grouped into bureaucratic, financial, and reconstructive efforts, and they represent substantial barriers to achieving the goal of housing reconstruction across different geographies, disaster types, and time frames, as shown by recovery progress in the three discussed case studies: the Joplin Tornado, the Northridge Earthquake, and Hurricane Sandy. Beyond the case studies, efforts at theorizing and modeling recovery phenomena attempt to delineate and inventory these events and processes. The following chapter discusses the literature surrounding the theoretical and simulation models surrounding the disaster phase of recovery.

### 2.3 Conceptual Framework Development

The underlying development of a simulation requires the development of a conceptual framework to focus and isolate key elements that will interact within a modeled environment. As Chapter Two of this research revealed, there are many processes that household actors must undertake to achieve any degree of recovery after a disaster event. Expanding on that, the conceptual framework that will be presented in Chapter Five is based on both the research presented in Chapter Two, and an iterative, subjective process of grouping like-objects together in as distinct of groups as possible described here.

Reviewing theoretical papers, broadsheet and web news media, case-studies, and comparative analyses, the author developed a framework for understanding disaster recovery in the housing sector. The author grouped like-objects into actors, resources, attributes, events and contextual elements, and further divided these objects into subtypes. These objects become nodes in the recovery framework and allow for a myriad of recovery trajectories experienced in the aftermath of disasters. For example, financial resources is an object in the housing recovery framework, further sub-divided into savings accounts and loans, each with different financial levels, availability, and immediacy. An actor's access to financing shapes the housing recovery trajectory.

Because much of this work is done iteratively and subjectively while reading a wide selection of literature, it is difficult to display the exact process through which the typology arose. Yet, the broad outlines of the process can be described. Google scholar was selected as the search engine of choice, proxied through library holdings from Western Washington University as well as freely available open-access and researcher-hosted holdings. Searches through these documents for keywords e.g. processes, phenomena, wait time, and resources, yielded an array of unique framework elements that differed with each other enough to form a loose typology, then recorded on a scratchpad and added to continually. Further refining this typology involved continually searching for new examples of case studies, newspaper articles, and theoretical research to ground the conceptual framework in the research – literature presented in Chapter Three, as well as whiteboard diagramming with other researchers to reduce and refine the framework. When the framework nodes were of sufficient uniqueness, subdivisions were codified, and examples were found for each of the nodes, differentiating them from each other. Appendix A shows an example scratch pad of one of those iterations classifying literature in preparation for building the conceptual framework.

The conceptual framework developed consisted of six elements (heretofore referred to as nodes), as well as sub-types and examples. Each is described here. The examples provided are non-exhaustive; adequate description of the abundance of concepts have provided a place within the framework to insert other phenomena as needed. This description of the conceptual framework begins by describing entities, those who participate in recovery. Attributes, those data about an entity that controls its access to resources and which processes it may take part in, will follow. The third node, resources, both shared and private, are the necessary external (and some internal) components that aid in recovery. Following resources, the fourth node is processes, which are larger conceptual elements such as searching, rebuilding, etc., made up of a variety of Events, the fourth node. Events are certain triggers that directly change the state of a resource or attribute, multiple event-chains building a process. The final, sixth node in the section is termed ‘context.’ Context contains the supplemental information about the particular disaster, such as regional utility loss. These nodes are characterized by the scope and scale of the research, so are not strict, which will be explained in the following sections.

### 2.3.1 Entities

Shown in Figure 2-1, entities are the participants within the post-disaster recovery realm. What differentiates an entity is its roll within the disaster context. It is a participant that affects the community in recovery and usually makes choices about the processes and events that make up its recovery. It requests and/or releases resources to other entities. There are three sub-types of the entity node: consumption-driven, provider-driven, and a hybrid of the two (both consume and provide). Consumption-driven can be thought of as entities that are driven to consume resources in order to return to or improve upon their previous state. They do not provide any resources.

| Framework Nodes | Sub-Type | Examples (not exhaustive) |
| --- | --- | --- |
| Entities | Consumption-Driven | Homeowners<br>Households<br>Business Owner<br>Renters |
| | Provider-Driven | Federal Government<br>Non-Gov. Organizations<br>Insurance Companies<br>Utility Companies<br>Schools |
| | Hybrid | State Government<br>City Government |

**Figure 2-1: Entities node, subtypes, and examples**

Examples may include homeowners, households, business owners, renters. The examples given are merely examples; the factors that control their categorization are not based on rigid types, e.g. businesses are always providers, but are based on the context and phenomena being modeled. For example, some might say that businesses are inherent providers, but that might only be if one is looking at economic factors such as a business providing jobs, materials, or some other service. However, if the structure of the model is such that businesses are simply consumers of government services, reconstruction resources, and perhaps utilities, they could logically be categorized as consumption driven. It is often implied that the recovery of consumption-driven entities are indicators of health of the community, enhancing overall recovery of the community, even though they may not provide any resources, other than labor, directly (Dwyer & Horney, 2014; Hayashi, 2007; Johnson & Olshansky, 2013).

Provider-driven entities are providers of resources but must make decisions themselves as to when and how to dole out such provisions. The behavior of providing resources varies among organizations: FEMA has financial limits placed upon it by congress as well as surge capacity for extreme disaster cases; the Red Cross generally provides emergency and temporary shelter, and food, but not longer-term recovery services. The behavior aspect is important; without it, why not just model the resources themselves directly and not worry about who is providing them? Such as it is, the provision-driven entities make use of available resources and make decisions about who has access to aid, and who does not. They may choose to limit the resource, append to the resource, or delay access to a resource. Another way to delimit provider-driven entities is that they are generally not affected directly by the disaster (they do not incur damage), thus they need not compete or consume for local resources. Examples of provision-driven entities may be the federal government, non-governmental organizations (NGOs), and insurance companies.

A hybrid version of these two is a third sub-type of entities and may be, in fact, the most common entity sub-type in models of disaster recovery. The hybrid entity both consumes, as well as provides, resources. It has attributes that may restrict it to certain resources. For example, if a presidential disaster declaration is not made, a state government may not qualify for the additional resources that can help in recovery. If a state doesn’t have enough credit to borrow money for a disaster, its recovery efforts may be hindered. State agencies may compete for resources with home and business owners by nature of limited availability of construction workers and inspectors. Federal aid often flows through state agencies; thus, it becomes a provider itself. The Red Cross provides institutional grants to organizations that have established recovery goals in a disaster area. These organizations are often small, such as churches, and may be the benefactor of recovery resources themselves if they are based in the disaster area (Red Cross, 2013).

In a complex model of recovery, it is necessary to recognize that the entities themselves maintain the ability to be complex and may not be solely provision or consumption driven, but some combination of the two. However, a simpler model, not looking at certain indicators, may wish to use a parsimonious approach. Thus, entities are defined as to how they function in the particular model environment in which they are used, not solely by their real-world counterparts, but by the scope, scale, and complexity of the model environment.

### 2.3.2 Attributes

The attribute node of the framework are the elements that describe entities. The attributes themselves will change with the entity being described. These attributes contribute to an “attribute profile” which in turn grants or inhibits access to a variety of resources. For example, a grant to homeowners below a certain income would not be available to entities who were not a) homeowners, or b) above the income threshold for aid. In this way, it also controls which processes in which an entity might take part. If an entity does not own property, it would not take part in the process of household reconstruction. If that entity’s shelter was destroyed, it would necessarily need to search for new housing, embarking on a different process entirely. These attributes can be subtyped into qualifications, and quantifications. Qualifications are generally non-numerical or binary. They include things like sex, race, ownership status, religious affiliation, etc. Quantifications are numerical representations of qualities, such as income, age, number of dependents, or outstanding debt. This allows for the model to be complex, encompassing a wide range of attributes of which creates a very unique entity. A parsimonious model can also be created, with a few choice attributes that help inform a simpler model.

### 2.3.3 Resources

Resources within the framework have 4 subtypes: physical, financial, human, and internal, as shown in Figure 2-2. Physical resources are consumable inanimate material. They may act as bottlenecks to processes, such as roofing material availability holding up reconstruction efforts. The implication of bottlenecks correctly shows that these resources are shared. This is the case for all the subtypes except internal resources, covered in the following paragraph. Examples of the physical resource type include building materials (lumber, plumbing, electrical), and natural resources (timber). Including this type of resource is useful for modeling efforts, as material stockpiles could be planned, or supply networks improved to increase the availability of such resources. Financial resources are relatively self-explanatory, and act as containers of monetary quantities. Some of these may empty and never be refilled, others may be refilled by acts of government or donations. Examples of financial resources include loans, government assistance, and non-governmental organization aid. Human resources encompass individuals or groups of individuals that provide some kind of service, and their labor is the bottleneck. Examples of the human resource type are home inspectors, structural engineers, construction workers, loan officers, and insurance adjusters. These are “public” jobs that are competed for among entities. The final resource type, the internal resource, is unique among the types in that it is made up of the other types, but it is not shared or competitive. The idea of an internal resource is that it is still a container or amount of something that can be consumed or requested but is usually available immediately because it is a personal resource. These could be thought of alternatively as individual resources. Examples might be a savings or checking accounts, insurance payout money, and other personal financial resources.

| **Framework Nodes** | **Sub-Type** | | **Examples (not exhaustive)** |
|---|---|---|---|
| Resources | *Shared* | Physical | Environmental (e.g. timber)<br>Building Materials<br>Infrastructure |
| | | Human | Builders/Contractors<br>Loan Approvers<br>Building Inspectors<br>Insurance Adjusters |
| | | Financial | Grants (FEMA, SBA)<br>Loans (FEMA, Bank)<br>Direct Aid (donations) |
| | Internal | | Business Inventory<br>Savings<br>Insurance payouts |

**Figure 2-2: Resources node, subtypes, and examples**

### 2.3.4 Processes

The next node of the conceptual framework belongs to Processes, as shown in Figure 2-3. Processes are actions taken by entities to achieve the goal of recovery. They are ongoing phenomena starting and resulting in events. This framework delineates 4 different subtypes of Processes crucial to recovery of an entity. The first subtype, Search, contains elements that involve the search for some service, infrastructure, or physical resource. Examples within Search are housing search, job search, looking for employees, etc. The second subtype is Rebuild. As the name suggests, it involves rebuilding shelter, infrastructure, utilities, public institutions, and businesses. Primarily conceived of as the rebuilding of physical infrastructure, it might also be operationalized to include institutional rebuilding as well. Migration is the third subtype of processes. It acknowledges the unique decisions and actions that lead to moving either within the community/disaster area or outside of it. The decision may be based on many factors, such as schools being out of commission for an extended period, loss of employment, loss of shelter, mental or physical well-being (fear, injury), or various economic reasons.

The final identified subtype in the Processes node of this framework is the process of healing. Healing refers to human well-being of physical and mental health. Damage to physical health, post-disaster, could be minor, only taking a few weeks, or major, resulting in permanent disability. Likewise, for mental health, notions of fear and post-traumatic stress disorder may affect an entities ability to work, attend school, or remain in the community.

| Framework Node | Sub-Type | Examples (not exhaustive) |
|---|---|---|
| Processes | Search | Housing<br>Job<br>Employees<br>Renters |
| | Rebuild | Shelter<br>Infrastructure<br>Utilities<br>Public institutions<br>Business locations |
| | Migrate | Within Community<br>Outside Community |
| | Heal | Physical<br>Mental |

**Figure 2-3: Processes node, subtypes, and examples**

### 2.3.5 Events

The concept of Events is the next node in the framework. An Event is a point in time in which a process begins or terminates. For example, the process of selling a house terminates in the sale of that house. Because events are most often tied to processes, expanding on them is unnecessary here. That discrete point in time in which you sell your house is often the salient data point to collect as researchers. Events can then trigger other processes, or can be used as interruptions, e.g. if a household finds a new house to buy, they might cease reconstruction efforts. This event of finding and purchasing a new house triggers a process that might be "selling house" or "moving." It is almost an implied piece of processes, but the interruption value earns it a node in the framework.

### 2.3.6 Context

The final node in the framework is Context. Context provides for environmental level variables that affect all entities. Examples of this could be widespread infrastructure loss (electricity, water, gas), rezoned areas that forbid certain types of additional development, public institutional damage, size and duration of event that affects damage and loss models, and any other community wide disaster effects. Essentially this constitutes the disaster scenario.

Together these 6 framework nodes identify the potential attributes, resources, and activities taken in recovery, and how they can be grouped in order to model recovery scenarios and phenomenon adequately. The next section will describe the use of this framework to model housing reconstruction in a discrete-event simulation.

## 3. Theoretical and Simulation Models of Recovery

This Chapter looks at the current literature on simulation used in the context of recovery. The first section discusses the use of simulation as decision support systems, allowing practitioners and researchers to explore outcomes to inform and influence policy and research decisions. The second section looks at current simulations in recovery contexts, though there are few focused on housing reconstruction. The third section examines discrete-event simulation as a technique and method for building simulations. Section Four then describes some of the validation and verification techniques common in simulation methodology. The chapter concludes with a summary, situating the sections into the literature.

### 3.1 Simulating Social Processes as Decision Support Systems

Simulation as a technique in evaluating and examining complex interactions of social systems is a methodology that begins as early as electronic computers were accessible to researchers, one such early example being an analysis of a social clique network to show that surveys and computer analysis could be stitched together in a simulation to analyze clique behavior (Coleman, 1961). However, simulation as a decision support system (DSS) is more recent. Decision support systems are tools used to help practitioners of any given field make decisions in a quantitatively informed way (Snediker, Murray, & Matisziw, 2008). A concise, early history of DSS is offered by Power, in which he describes the evolution of DSS from IBM punch-card machines, through mainframes, and finally to personal and web-based computational solutions (Power, 2007). While decision support systems in business intelligence and information technology systems are widely adopted, the use of DSS – particularly simulation – has not been frequently realized by practitioners of social problems, such as disaster planning.

Simulation has been used in various social science applications, most commonly using agent-based designs to simulate interaction among agents or entities, however there are examples of Monte-Carlo and discrete-event, which are reviewed in the following section. Axelrod gives five key uses of simulation in a social scientific context: prediction, performance, training, education, and proof and discovery (Axelrod, 1997). Using simulation as prediction in social science can be dubious, as social systems are so complex that a simulation cannot provide an accurate enough image for true predictive use (Srbljinovic & Škunca, 2003).

Analyzing efficacy of DSS is much less common in the literature than the design, building, and utilization of them (Power & Sharda, 2007). Effective DSS methods help impart new perspectives or information to the user about the system. It is commonly implied in the literature that usefulness and practicality of simulation models are inherent. However, rigorous study of the efficacy of simulation as a DSS method to help inform practitioners has been the focus of several studies. Ben-Zvi evaluated the value of business game simulation in learning core concepts, in this case management information systems, and found a positive correlation to student learning and retention (Ben-Zvi, 2010). Interestingly, he also found that an increase in complexity often offered negative correlation in user experience and understanding, suggesting a multi-faceted system may not provide better results.

Simulation for use in social sciences, to help understand and interact with complex systems involving participation among actors, is well established with agent-based models. In the field of disaster recovery, there are fewer models to choose from, and while several are agent based, none are discrete-event based, as the next section will detail.

### 3.2 Current Simulations in Recovery

The most mature area of recovery-oriented simulation modeling is for lifeline infrastructure restoration. Discrete-event simulations (DES) have been developed to model the disruption and restoration of electricity after a hazard event (Cagnan & Davidson, 2015; Çagnan, Davidson, & Guikema, 2004; Xu, Guikema, Davidson, Nozick, & Zehra, 2007). Restoration simulations for water networks (Luna, Balakrishnan, & Dagli, 2011; Tabucchi, Davidson, & Brink, 2010) and manufacturing supply chains (Melnyk, Rodrigues, & Ragatz, 2009; Schmitt & Quantifying, 2009) have also been researched. Tabucci et al. (2010) describe their model explicitly as a post-event short-term utility simulation, allowing inspection, rerouting, and repair functions to be examined. Schmitt et al. (2009) combine a Monte Carlo simulation to develop risk profiles with DES to study the flow of material and network interactions among a supply chain. While potentially the most sophisticated simulation models related to recovery, infrastructure restoration models ignore or are not explicitly linked to other processes of recovery such as home reconstruction or interacting with aid and other resource providers.

Post-disaster reconstruction is a recent area of simulation research. Nejat and Damnjanovic (2012) developed a conceptually driven, agent-based simulation to model homeowner decisions regarding whether to reconstruct damaged buildings. Within the simulation, time is fixed based on an assumed total reconstruction period of 18 months. An alternative approach by Nejat (2011) uses a multinomial logistic regression model that relates variables such as the availability of utility infrastructure and the ratio of available funds to required expenses to an ordinal dependent variable having the categories of (1) reconstruct immediately, (2) wait six-months and observe the reconstruction in the neighborhood, and (3) take the insurance money and buy a new house elsewhere. These models do not account for events that can interrupt rebuilding efforts, such as another disaster striking, nor the access limitations of its agents, such as income, school access for children, or employment.

Some of the most recent simulation modeling of post-disaster housing reconstruction comes from Kumar, Diaz, Behr, & Toba (2015), who use a system-dynamics model to simulate labor shortages that may occur during reconstruction. Using an extensive set of variables that contribute to the dynamics of labor in housing reconstruction, they build an effective model, such as hiring and firing, searching for employees, and attrition rates. It does not, however, factor in homeowners' decisions as entities, nor does it take into account interruptions in the reconstruction effort that may originate outside the system.

To date, the most comprehensive model of disaster recovery is ResilUS (Miles & Chang, 2006, 2011). ResilUS models a wide range of recovery phenomena, such as building reconstruction, employment, debt repayment, and business failure, using a Markov Chain Monte Carlo (MCMC) simulation approach.

ResilUS represents recovery as continuously progressing input and output variables at weekly intervals. ResilUS models recovery progress as the result of a comparison between a random number at each week to joint probabilities derived from the multiplication of normalized input or intermediate variables. ResilUS does not allow for representation of discrete events, nor explicit requesting and access of recovery resources. ResilUS is written in MATLAB and SimuLink, which is a proprietary, licensed-based software, unavailable to those able to pay for the software, reducing its accessibility.

### 3.3 Discrete-event Simulation

Discrete-event simulation, or DES, is a method of simulation that uses events instead of time-steps to move the simulation forward and reveal interactions in the system being studied. Entities interact with resources instead of each other – though entities could act as gatekeepers for resources – based on qualifications and attributes assigned to the entity and enforced by the resource. Time is not measured until a desired event is triggered, making it asynchronous. The advantage over other simulation types is the focus on entities interacting with resources, not each other. This is sufficient and useful in the study of household reconstruction, as the processes demanded of the homeowner to reconstruct their household or find a new dwelling involves accessing needed resources by navigating the logistical web of applications, qualifications, and attributes required by resources, of which themselves are stymied by ability to provide a resource, e.g. labor, money, or time limitations.

The design of a particular DES consists of the specification of the elements—entities, events, and resources—and their interactions to represent the phenomena being modeled (Karnon et al., 2012). DES has seen very little use outside of manufacturing and medical logistics, and as such there is often only one entity being processed at a time. The key difference between it and other simulation types is the event-driven nature of the simulation. Events can trigger different processes, for example, in an emergency room simulation, the event of checking in puts an entity in line to be seen by the doctor. Once their turn arrives, they are able to receive the doctor, at which point they consume the resource of a room. Once the doctor has treated them or moved them to more intensive care, the entity releases the room and the doctor, so they may be consumed by another entity. In this example, a doctor and a room can only accommodate one entity at a time, and it is a fairly straightforward conceptual framework. Operationalizing this model, one might add many more resources, such as lab technicians, where the doctor can move between waiting rooms while awaiting test results. These notions are event driven. Each time step – for example days – is not considered or recorded. Only the time of an event occurring is recorded and analyzed post-simulation. That event then often triggers other events e.g. when the labs come back, initiate some new process e.g. ordering procedures, requesting an operating room, prescribing some treatment, etc. The functionality of the system is important, not the time-steps. However, when monitoring a simulation, access to the current simulation time is important, as this is where one may identify bottlenecks. Thus, a discrete-event simulation of disaster recovery must represent associated phenomena in terms of entities, events, and resources.

A python package – a grouping of modules consisting of related functions, classes, and other parameters around a common namespace – called SimPy, is a discrete-event simulation specific package extending DES capabilities to python users. SimPy is a DES framework written specifically for the Python programming language. Figure 3-1 shows a general overview flowchart of how processes are scheduled. While discrete-event simulations in other languages (Java, .NET, etc) utilize parallel computing functionality to pause and interrupt process functions to mimic discrete events, SimPy uses Python's built-in generator functions. The use of the *yield* keyword (generator specific) allows processes to be "paused," suspending a function while retaining its execution state, such that resumption of the function picks up where it left-off (Scherfke & Lünsdorf, 2015). This is useful for SimPy: it is fundamentally an asynchronous event-dispatcher that schedules events at run-time based on definitions provided by the designer. Defined processes can be paused, resumed and interrupted, and other processes may be yielded. SimPy provides abstract elements of processes, and resources that are combined with an environment to build the simulation.

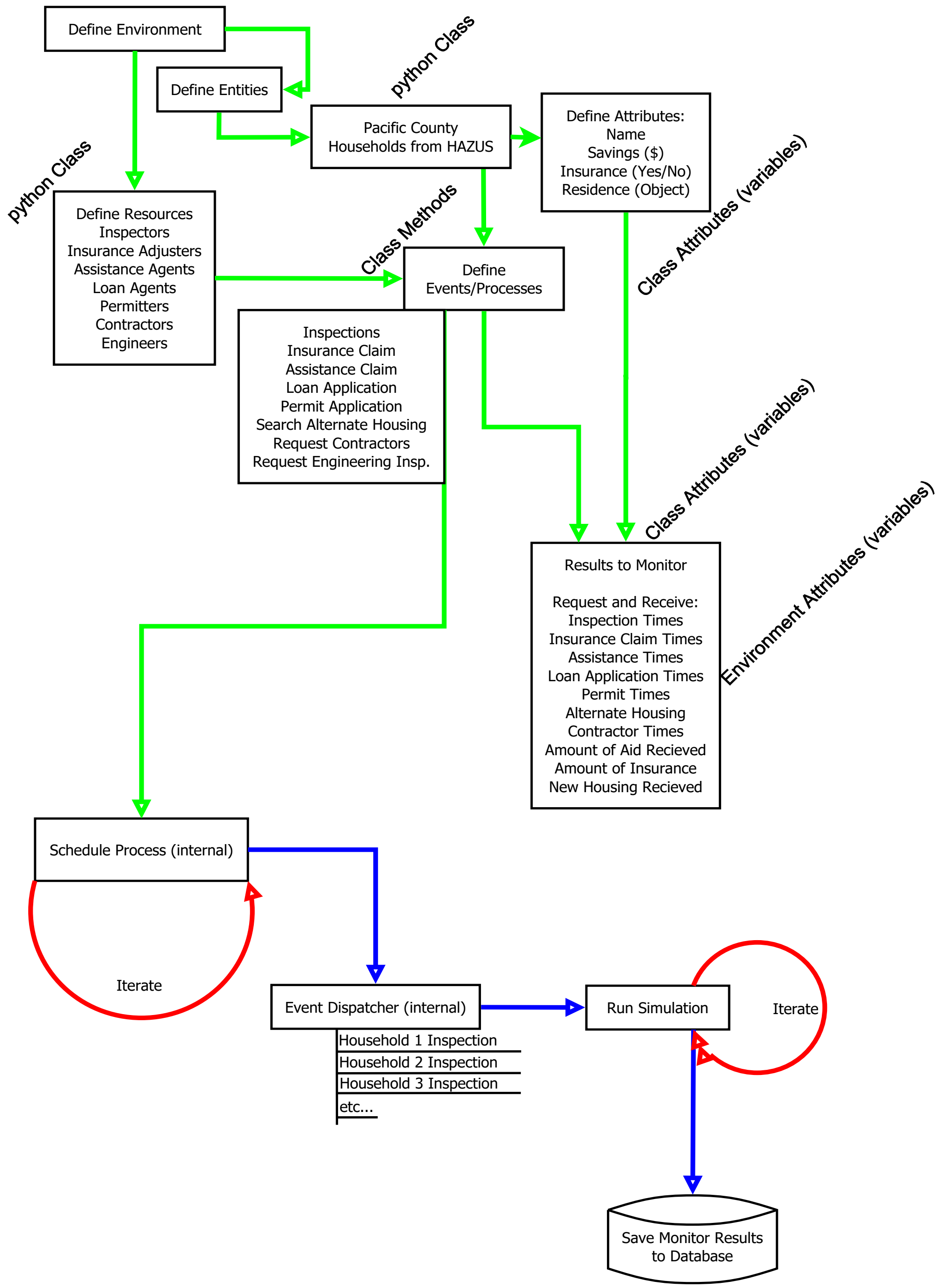


**Figure 3-1: Diagram of SimPy's event scheduler**

Processes in SimPy can be used to model active components like customers, vehicles or agents (Scherfke & Lünsdorf, 2015). They can also be used to model a conceptual process that have discrete start and end points (e.g. refueling a car, accessing a web service, rebuilding a house). SimPy provides the idea of a time-out process, pausing the process for a defined duration, continuing after the duration subsides. This can account for time-delays that are unable, too simple, or not critical enough to be modeled by resource or entity interaction behavior, e.g. quarantine, post-dive decompression, airplane travel times.

SimPy also provides various types of shared resources to model limited capacity bottleneck points, (e.g. gas pumps, network sockets, or building materials). This allows models to have entities that interact with resources via request and release functions. Resources function in much the same way as the standard DES concepts: a container with a variable quantity of whatever resource assigned to it. SimPy provides three different forms of the resource type:

- Resource, which can be requested and released, and queue up processes while they wait for an available resource.
- Container, which can hold a discrete or continuous variable amount of something (e.g. apples or water, respectively) as well as be refilled at will by another process or after an amount of time.
- Store and FilterStore to manage Python objects that act as consumables, SimPy provides the store.

SimPy maintains a queue of requests, so the next in line can then receive the open resource. Code within the process will determine the duration of the use of resources.

### 3.4 Model Validation and Verification

To ensure that the model built adequately describes the system researchers are attempting to model, they must use verification and validation techniques to attempt an objective analysis of the finished model. Given the relative dearth of simulations of this nature, however, there are no agreed upon or industry standard techniques with which to accomplish this task, and some have referred to the process of validation and verification as an art as much as a science. Each model is unique, and while some methods may work well for one model, they are inappropriate or not feasible for another (Sargent, 2011).

While the methods of validation and verification are used in tandem with one another on models, they are fundamentally different. *Verification* refers to the process of testing the workability, efficacy, and function of the model, such that it works as expected and intended, functionally. The effort of testing code, inputs, outputs, and debugging all help verify that it is working as intended. Some techniques are design based, while others are meant to be applied after, or even during, the process of actively programming the model. *Validation*, conversely, assures that the simulation appropriately models the represented system, and that the inputs and outputs make sense in comparison to reality. Much of the validation techniques can be construed as subjective, or as necessary for the scope of the project, as no model can fully emulate a complex system such as recovery.

### 3.5 Summary

Research on the modeling and simulating of social processes, including the recovery and reconstruction phases of disasters, is an emerging trend. This chapter described some of the salient techniques, literature, and methods involved in simulating recovery processes. While many agent-based models exist in the literature looking at interactions between actors in a system, this pivots on the idea that such an interaction meaningfully changes behavior. The research described in this work takes an alternative view that the important facet of recovery is not actors' access to each other, but to resources enabling a return to normalcy after a disaster event. Discrete-event simulation offers the ability for actors to interact with resources, form a queue, and move through the web of recovery based on attribute and qualification profiles.

Viewing recovery through the lens of modeling and simulation offers a perspective for educational use, as well as to display alternative and 'what-if' scenarios to planners and policy makers. The following chapter explores the methods of development of the simulation and conceptual models formed during this research, as well as the case-study area used as a test for the simulation model.

# 4. Simulation Model

This chapter details the methods used to develop a simulation model of owner-occupied housing reconstruction following disaster. The first section describes the development of the discrete-event simulation model of owner-occupied household reconstruction in the Python programming language, as well as the inputs and parameterization schemes used to propagate the model. Section 4.2 provides an overview of the case study area used to test the simulation model on a "real world" example.

## 4.1 DES Simulation Model of Owner-Occupied Household Reconstruction

This section describes the discrete-event simulation model and how it was coded and designed. It details the way the "stage was set" for the simulation to run, with results and discussion in the following chapters. This section begins by describing the structure and software used in forming the simulation. The next sections contain descriptions of the conceptual framework nodes: entities, resources, attributes, processes, and context.

### 4.1.1 Structure and Software

The structure of the prototype simulation developed as part of this thesis is based on the conceptual framework, outlined in section 2.1 and detailed section 2.3, following the basic structure of core discrete-event simulation elements. The aim was to construct a simple model of the recovery phenomenon of homeowner reconstruction. A sole entity of "homeowners" was used, as well as four resources they need to request and fulfill before reconstruction is completed.

The Python programming language was chosen for use due to its accessible syntax, numerous scientific computing libraries, strong community support, open-source ethos, and low cost. Programs can be mocked up quickly, and due to its interpreted nature, requires no compiling between executions, shortening development time. A discrete-event simulation library – SimPy (Simulation for Python) - was chosen to implement the recovery simulation prototype. SimPy facilitates rapid prototyping, robust and in-depth documentation, and an object-oriented approach to design, while abstracting away the more nuanced intricacies of creating a DES framework ground-up.

The simulation model implements all of these examples in various constructs. SimPy resources model building inspectors, loan officers, and contractors. SimPy containers model FEMA funds that are immediately available following the disaster event and can only be refilled via another process. The Store and FilterStore construct is used to hold building objects – houses – that are swapped and searched for while household actors search for a new residence.

### 4.1.2 Entities

The simulation was designed to be a combination of procedural and object-oriented design, a programming paradigm based on the concept of objects. These objects have attributes (variables) and methods (functions). Objects are instantiated based on a Class, which can be thought of as a blueprint of an object. Each object is an encapsulated *thing* with attributes and any functions, in the case of this simulation, these objects are households.

The entity objects were then instantiated by looping over an input file containing a list of names, household savings, coordinate pairs, damage levels, type and replacement value of home, the parcel number, and insurance coverage to create the objects, henceforth referred to as entities. These entities are what interacted with the simulation and served as households needing to rebuild their shelter. The code for the entities class is included in Appendix 1, under the “entities.py” subsection. Computationally, these entities were contained within a Python dictionary, named “household,” and accessed via python list notation e.g. household[“name”], where household is the object list container, and “name” is the object contained within. These objects represent conceptual owner-occupied household entities, and allow for the dynamic addition of any number of entities as objects. Much of the input data, shown in Figure 4-1 was obtained from FEMA Region X, which has run HAZUS loss models on the study area of Pacific County, Washington.

| ID Number | US000001 | US000002 | US000003 |
|---|---|---|---|
| Name* | KRISTY PAGE | CHRISTOPHER HIDALGO | JAMES YU |
| Value | 106400 | 40600 | 69200 |
| Area | 2080 | 1728 | 1456 |
| Content_Value | 53200 | 20300 | 34600 |
| Damage State | Extensive | Extensive | Extensive |
| Latitude | 46.62482 | 46.623327 | 46.624823 |
| Longitude | -123.655925 | -123.656808 | -123.657676 |
| Parcel_ID | 72035000021 | 72035000029 | 72035000038 |
| Occupancy | Mobile Home | Mobile Home | Mobile Home |
| Has Insurance* | 0 | 0 | 0 |
| Savings* | 4336.008054 | 3038.477733 | 3767.752137 |
| Insurance* | 0 | 0 | 0 |

*Attributes not derived from HAZSUS loss estimation model of Pacific County

**Figure 4-1: An example of 3 input households to the model**

The data used to run the HAZUS scenario is 3 years old and was derived from tax assessor data provided by the county to FEMA. FEMA then ran the subduction zone earthquake model and provided the results of this scenario to the public via HAZUS Hazus Packaged Region files. Specific data provided by the HAZUS 9.0 subduction zone earthquake model for use in the simulation included coordinates (latitude/longitude) of house, occupancy type (single family, mobile home, etc.), replacement cost, damage state, floor area (in square footage), and the parcel number. Floor area and occupancy type were specifically used in the search for permanent housing of similar type to the original house. Coordinates were reported in latitude/longitude pairs in the WSG 1984 reference system. Replacement cost comes from HAZUS, and was based on a percentage of the house’s value at the time of the module run. Damage state was used to determine how much of the replacement cost was applicable to reconstruction/rebuilding. Parcel number was used to incorporate any other geographic data or tax data provided by the county in later simulations or for further analysis.

Other elements incorporated into the entity not provided by the HAZUS were estimates, assumptions, or derived from distributions around a mean. Figure 4-2 shows some of the numerical assumptions used to generate these data. Household savings were derived from a mean of $2,629, obtained from Census Bureau wealth statistics, along a normal distribution of savings. As an assumption for this simulation, those households with higher value replacement costs get the larger tail of savings. Whether a household was insured or not was decided under the assumption of 10% insured, and the majority of those with insurance are going to be higher income households with housing that is more expensive to replace. Names were the final input created. HAZUS does include names of owners in order to respect anonymity and as such a list of unique fictitious last and first names was created for tracking purposes and for ease of human readability of outputs.

| | Derivation Scheme |
|---|---|
| Savings | Gaussian Distribution (Mean=2629, St.Dev = 1) |
| Insurance | Assumed 10% insured |
| Names | Census list of most popular first and last names |

**Figure 4-2: How the non-Hazus derived data was generated**

### 4.1.3 Attributes

Contained within each entity object were the attributes that factored into its access to certain resources. Attributes that factor into an entity's access to resources were coded into each entity object. Conveniently, terminology in object-oriented programming uses the term attributes to refer to private variables owned by an object. Each individual entity was given different attributes from the input file. Three variables were chosen to include as entity attributes: savings, insurance, and residence. Savings represented a raw dollar number derived from the national average of savings. Insurance-policy holder rates vary by state and population center (III, 2016). The Insurance Information Institute reports a figure of 14 percent coverage in western United States households (*Homeowners Insurance: Understanding, Attitudes and Shopping Practices*, 2017), while the National Association of Insurance Commissioners puts the figure between seven percent and 28 percent of households obtaining adequate earthquake insurance (*Earthquake Insurance*, 2016). Insurance was programmed as a binary variable applied to 10% of the homeowners, a reasonable approximation for households located in the study area. Residence was a household object with a description (number of bedrooms, bathrooms, home value, and damage state). Any attributes could be added, before and during the simulation. These attributes belong to each individual entity, and could therefore be queried post-simulation to track changes, examine states, and help verify and otherwise monitor the model.

As mentioned in Chapter 3, monitoring is important in a complex simulation, as errant emergent behavior can interrupt the flow of designed interactions among entities and resources. Tracking key attributes back through the program gave the ability to trace faulty behavior to where it originated in the simulation. One of the approaches used to help monitor the simulation and explain behavior was an attribute called “story,” which contained human readable sentences of events that happen to that specific entity as shown in Figure 6. To create the “story” attribute, a sentence was added to each process recording the time, amount, or other state change. The result, detailed in the following chapter, was a paragraph structure showing the progression of the household through and factors explaining their result.

#### 4.1.4 Resources

After construction of the entities, resources were programmed into the simulation. There are three types of resources provided by the SimPy library that we utilized in the simulation – see chapter 3.2.3 – *containers*, *resources*, and *stores*. In accordance with the conceptual framework, resources used in the simulation were delineated as different forms of capitals available to entities. Three types of capitals were available in the simulation: Financial capital was provided money in a *container*; Human capital, initiates human resources such as building inspectors inside *resources*; and Built capital, which controls structures, buildings, and residences, stored in a *store*. Financial capital included two elements: FEMA Aid funds, parameterized to 200 million dollars; and Building Materials value, which was set to 2 million dollars, but reloaded with 30 million dollars between 6 months and 1 year after the simulation began. Human capital had 7 elements included in the simulation: Contractors, Engineers, FEMA processors, Insurance adjusters, Loan processors, Inspectors, Permit processors. Human capital was all parameterized to have between 1 and 100 individuals available, at random. Because the simulation was run for 1000 iterations, a large spread of data was obtained to show the effects of low and high parameters. Built capital was used in two different instances: BuildCaptial and Residence. A general BuiltCapital class was implemented that was a parent of Residence, which all homes in the simulation are based on. The BuiltCapital class could be anything built with a location on the earth, such as a bridge, dam, or house, allowing in the future to include businesses or other structures to be added to the simulation. However, in the case of housing, more precision beyond the BuiltCapital class was required in terms of attributes, such as bathrooms, bedrooms, square footage, value, and post-disaster damage state. Each entity in the simulation “owns,” as one of its attributes, a residence object, which is passed around to processes to rebuild or sell the home. The other use of Built capital, residences specifically, was set up as a filter store. Recall that a filter store allows an entity to request an object from the store based on certain criteria (e.g. number of bedrooms, square footage). According to the 2010 census, 72 percent of the homes in the study area are owner occupied. Thus, the assumption was made that the other 28 percent of homes are vacation homes or long-term rentals, as much of Pacific County is a tourism destination and home to many vacation homes. The homes are therefore put on the “open” market and for sale, for those entities that could not afford to rebuild their damaged house or did not want to wait.

These parameters are all set as constant variables in the main program, but can be changed as “what-if” variables to monitor the behavior of the underlying interactions. None of these variables are immutable. Any of them can be changed by a process or a delay. For example, the building materials value resource was set at 2 million dollars. Realistically, this was not nearly enough for reconstruction purposes, so most homes won't be able to rebuild. A delayed process can do is introduce a process that might mimic the opening of a crucial transportation system (a highway, port, train terminal) at some period after disaster, and make a certain quantity of new building material available. This can be a recurring or one time "deposit." The following sections explains processes and which processes were included in this simulation.

#### 4.1.5 Processes

The design of the process functions of the simulation relied on SimPy's underlying basis of classes and methods. Each process function was programmed independently and fed to a "process" class constructor that turned the function into a scheduled process in the SimPy environment. This is accomplished in the background: the scheduler finds all the processes it can at execution time and schedules them, or puts callbacks on them (processes to activate if triggered by an event). The simulation then 'begins' and processes are free to make requests and otherwise interact with the environment. This section will showcase the variety of processes we have determined are important and interesting to simulate in the phenomenon of household settlement (reconstruction or migration), and the decisions that were made to parameterize these processes.

The list of processes chosen to represent home reconstruction in this simulation are as follows: home inspection, insurance claim application, FEMA aid application, engineering assessment, building permit application, loan application, rebuild home, rebuild housing stock, and search for new home. The processes and their parameterization in the model can be seen in Figure 4-3. Many of the processes sample from a distribution to obtain a “random” number – random in that they change every subsequent iteration run of the simulation. In the case of most of the parameters they are Gaussian distributions, meaning they are normally distributed around a mean, and that 67 percent of the numbers fall within 1 standard deviation. Therefore, when the standard deviation is set to 1, the function only samples numbers from that 67 percent. The range is shown in the “Min” and “Max” columns, units being approximate days.

| Processes | Duration distribution (mean, std) | Min (days) | Max (days) |
|---|---|---|---|
| Inspection | Gaussian (1, 1) | 0 | 3 |
| Insurance Claim | Gaussian (15, 1) | 10 | 20 |
| FEMA Assistance | Gaussian (20, 1) | 16 | 24 |
| Loan Application | Gaussian (30, 1) | 26 | 34 |
| Engineering Assessment | Gaussian (25, 1) | 20 | 29 |
| Building Permit Application | Gaussian (35, 1) | 31 | 39 |
| Contractor Time | Depends on damage | 2 | 180 |
| Search for New Home | Depends on stock availability | 0 | 500 |

**Figure 4-3: Parameterization of the resource durations**

Not every entity started or completed every process. Many might start and never complete a process, such as if a household was unable to acquire finances to rebuild their home, and they decide to move, they would never start the rebuilding process, or it would be interrupted. The processes are not necessarily listed in order, as the ‘master’ process is the final determinant for order of processes.

The first process in the simulation was a homeowner request for a home inspection. This home inspection was necessary before an insurance claim or FEMA assistance could be filed and settled, or the damage amount in dollars would not be known. The process made a request to the inspector resource established earlier, and queued up to receive an inspection. As soon as the household's position was next in line and an available inspector was unoccupied, the process locked down that inspector for a variable amount of time. The inspection did not do any actual calculation of damage, as the damage value was imported from the HAZUS Scenario results, provided by FEMA Region X, however it did occupy the time of one inspector for one day.

The next process, the insurance claim application, checked to see if the household had insurance, and if so payed the household the damage value of its property. The household then requested an insurance adjuster to process their claim, having to wait in line behind hundreds or potentially thousands of other policyholders. Again, this process could be altered in any number of ways, by assigning policies to households or limiting claim awards. In the simulation, the insurance adjuster time sampled a Gaussian (normal) distribution with a mean of 15 days and a standard deviation of 1, meaning that the max time was approximately 20 days and the minimum time was approximately 10 days for claim processing.

Concurrent to the insurance claim, the entity requested individual assistance from FEMA. When this process was called, it first verified that the entity had not already received rebuilding funds from insurance. If the entity already had enough funds to rebuild, it did not ask for any more funds from FEMA, and exited the process. If it did not have enough funds, it requests a FEMA claim processor, holds it for a random sample of time drawn from a Gaussian distribution with a mean of 20 days and a standard deviation of 1, and releases it. Then it calculates a refund level based on damage, not to exceed the currently mandated maximum reward offered by FEMA of $33,000. If FEMA had no money left for assistance, the request is not paid out and the entity received no individual assistance.

Entities then entered the process of attempting to secure a loan (presumably from the Small Business Administration, but that isn't specified in our simulation). The loan algorithm bridges the gap between the remaining damage value to the household and what they were unable to secure via FEMA assistance and insurance claims. When an entity applied for a loan, they received it. Limits could be placed on this, as well as qualifications e.g. income or down payment, but in this simulation the entity simply had to wait for it. The parameterization of loan wait duration, as seen in Figure 4-3, was a sample from a Gaussian distribution of 30 days from 1 standard deviation from the mean. If they become too impatient waiting for their rebuilding loan, they will exit the simulation (emigrate).

If the damage state of the household is "complete," then in conjunction with starting the search for capital to rebuild, the household started the search for new housing. Recall that 28% of households were set aside as vacant households. This list was then loaded into a feature store that could be requested by the process. When the request was made by the entity, it specified the desired attributes of the permanent house (value, damage state, bedrooms, bathrooms, etc.). The wait time in the search process was wholly dependent on the attributes being requested. That is, a homeowner may never find a home if one is not in the store consisting of the desired attributes. If the homeowner took a home out of the store, they then deposited their existing home, which is then available for “sale,” albeit damaged.

A pre-rebuild engineering assessment was the next process. Similar to the other processes, after a household had received their financial assistance, they requested an engineer from the resource object, occupying it for a random sample of a Gaussian distribution with a mean of 25 days, then returned it to the pool of engineers.

Following the engineering assessment, the household applied for building permits. Permitting, after a building permit official was requested and occupied, took a duration from a sample Gaussian distribution of 35 days. The final process, after all the other processes completed, was the construction of the physical structure. For completion durations, figures provided by FEMA in the HAZUS manual were used. Figure 4-4 shows the repair duration matrix. Therefore, if a household had a damage state of “slight,” its repair time was two days after it secured contractors.

| Occupancy | None | Slight | Moderate | Extensive | Complete |
|---|---|---|---|---|---|
| Single Family Dwelling | 0 | 2 | 30 | 90 | 180 |
| Mobile Home | 0 | 2 | 10 | 30 | 60 |

**Figure 4-4: Repair duration matrix. Provided by HAZUS, it is the estimate of repair times (days) for a given occupancy type**

The described processes do not give the simulation the order in which to run the processes. That was done by a master process, signaling the environment to schedule the processes and triggers. Figure 4-5 shows the chronological arrangement of the simulation from event to steady state. The order of the processes can be changed, such as running processes concurrently, or waiting for two or more processes to conclude before moving ahead. SimPy offers the ability to hard code into each of the processes *the next process* in a chain. However, a master process was written that calls one process after the other (or concurrently) in an attempt to simplify the model, to make it more modular, and to ease our ability to experiment with different patterns and model structures.

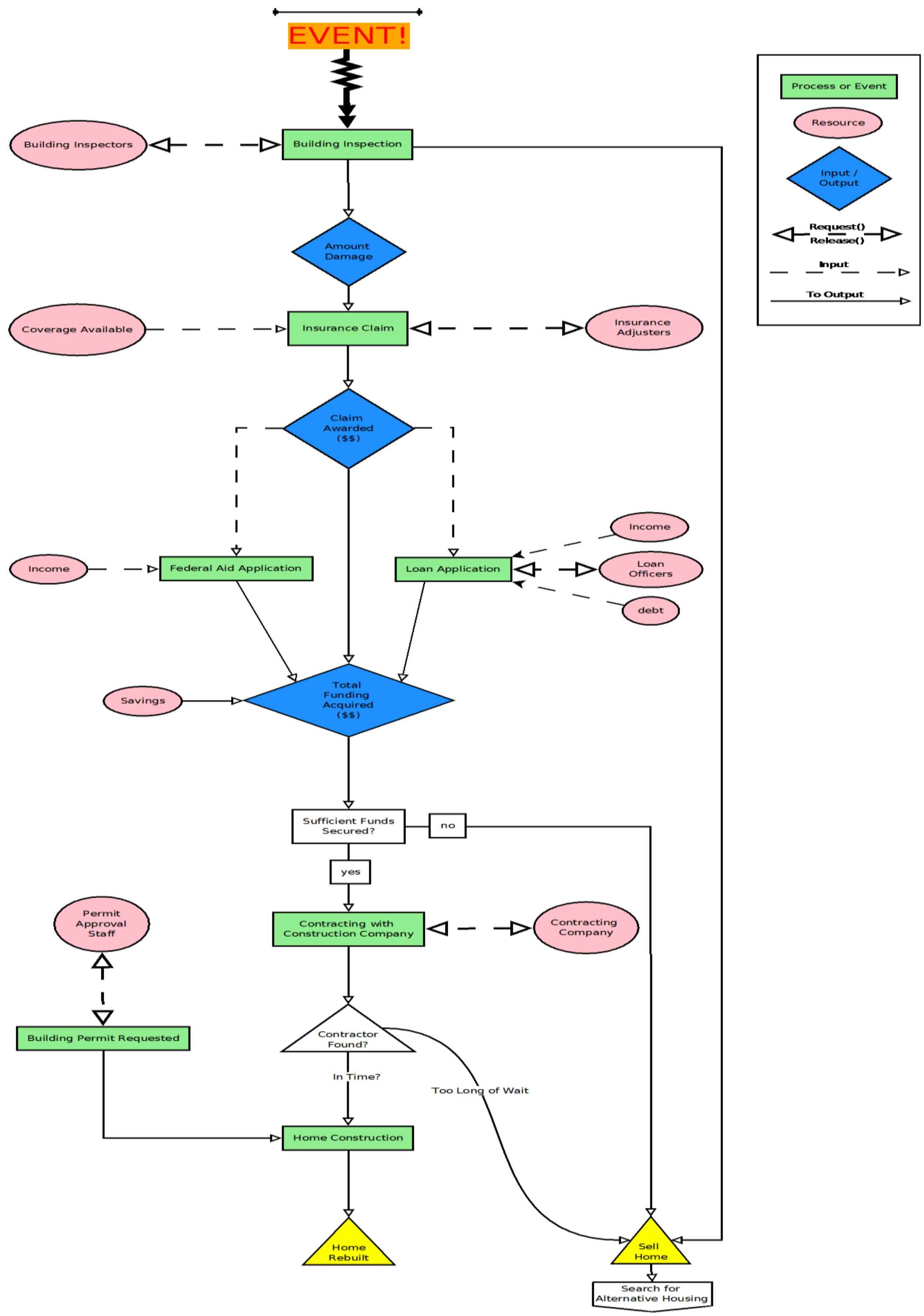


**Figure 4-5: Sequential decision diagram tree of simulation model**

### 4.1.6 Context

The final element in the conceptual framework that was translated to the simulation is context. Context provides all the elements of a hazard scenario that might not be modeled directly, but are needed to assess the scope and limitations of the simulation. Context contains the design decisions and parameters necessary for modeling a system as complex as household reconstruction.

In terms of context for the disaster scenario modeled here, a M9.0 Subduction Zone Earthquake scenario in Pacific county, Washington. The simulation's geographic bounds are Pacific county, meaning other cities or counties are not present in the simulation. Our unit of measurement is the household, the phenomena being modeled is household reconstruction. The decision not to model sectors like utilities, transportation, and other public entities was made as a matter of scope. Ownership status is not taken into account, all buildings are assumed owner occupied, unless vacant, then they are folded into the available vacant housing stock. This decision was based on lack of data.

SimPy provides an "environment" class, in which all interactions take place and time is accounted for, however this is more of an internal construct than a contextual tool. The environment contains the method "run." which begins the simulation. It will run indefinitely, until all requests are fulfilled, or until a predetermined time set by the user. This simulation runs until all processes are completed.

### 4.1.7 Verification and Validation

While there exist many methods to help verify the model, as described in section 3.3, a select few were chosen that fit within the size, scope, and nature of the model. The first was designing the simulation in a modular way by breaking up complex tasks into modules. This modular design was undertaken not only while designing the conceptual framework, but also while coding the simulation using Python modules, and assembling the final simulation into a master process. Instead of one long code file of several thousand lines, six different files – or modules – were used and assembled in a seventh, allowing an easy way to find bugs by grouping like processes. The second method was using deterministic, or “constant” parameters to test the outputs. When the model was working as intended for constants, then we ran simplified cases, the third method of verification used. Simplified cases involved running the model with only five inputs to track the individual entities and see that they are behaving as expected. This also involved adding resources and processes slowly to make sure they are not introducing errors. One of the most useful methods of verifying the behavior of the model is tracing, allowing a trace back of the model elements step by step to see where unexpected results occurred. This trace was done in the simulation programmatically by introducing variables specifically for monitoring purposes. For example, an entity should never have a rebuilt house before they have a home inspection. Monitored variables reveal exactly when a home was rebuilt and search through the order of code operation to investigate why the error may have occurred.

### 4.1.8 Outputs

The monitor variables were saved to a SQLite3 database in which multiple complex structured query language (SQL) statements could be tested. To test against a wide range of potential parameters, each of the human capitals were initialized with a random quantity of "workers," between 1 and 100 for the duration of the simulation. To get a large sample of each randomization, the simulation was run 1000 times. This allowed later for analysis to be conducted on different combinations of parameters, as well as exploring "fringe" cases in the model.

The variables chosen to save for analysis and results are in Figure 4-6. There are three sections to the table. The first section is variables involving Get and Put functions, that is, processes requesting and receiving something. In this case, Get is a synonym for receiving, Put is a synonym for requesting. The second section is for search variables, namely searching for money to rebuild and searching for a new house to purchase. The third section is other attributes deemed worthy to save and account for such as coordinate pairs and damage state, and importantly sim_run, which is a incremented count of each simulation run, 1 – 1000, which is the common link among all the database tables.

**1: Get and Put Variables, in Days**

| Variable | Meaning | | Variable | Meaning |
|---|---|---|---|---|
| assessment_get<br>assessment_put | Engineering Assessment Times | | inspection_get<br>inspection_put | Inspection Times |
| Assistance_get<br>Assistance_put | FEMA Assistance Times | | loan_get<br>loan_put | Loan Application Times |
| claim_get<br>claim_put | Insurance Claim Times | | materials_get<br>materials_put | Building Materials Times |
| home_get<br>home_put | Home Repair Times | | permit_get<br>permit_put | Building Permit Times |

**2: Search Variables**

| Variable | Meaning | | Variable | Meaning |
|---|---|---|---|---|
| money_search_start<br>money_search_stop | When a household began and ended the search for repair funds. (days) | | home_search_start<br>home_search_stop | When a household began and ended the search for a NEW house to buy. (days) |

**3. Other Attributes (unit in parentheses)**

| Variable | Meaning | | Variable | Meaning |
|---|---|---|---|---|
| insurance | Was household insured (True/False) | | money_to_rebuild | Total money available to household to Rebuild (dollars) |
| latitutde<br>longitude | Coordinates of household (WGS 1983) | | name | Name of entity/household (text) |
| loan_amount | Amount secured in loan application (dollars) | | savings | Amount the household had in savings (dollars) |
| moved | Did the homeowner move? (True/False) | | story | Structured collection of sentences detailing aspects of reconstruction (text) |
| sim_run | Which simulation iteration (integer) | | damage_state | How damaged was the household at the start of the simulation (text) |

**Figure 4-6: Variables monitored for outputs in the simulation**

Figure 4-7 shows two other database tables used to monitor simulation outputs: the human capitals parameters, and process times for processes. Because these parameters change to a random quantity with every simulation run, it is necessary to account for the changes in order to explain simulation outputs. Section 1 of Figure 4-7 shows the parameters with human capitals, while Section 2 shows the process times used in the simulation. The only common variable among the three database tables is “sim_run,” so that the three tables can be joined when making selections from the database.

**1. Parameters**

| Variable | Meaning | | Variable | Meaning |
|---|---|---|---|---|
| Contractors | Number of contractors available | | Engineers | Number of engineers available |
| FEMA_Processors | Number of FEMA processors available | | Inspectors | Number of inspectors available |
| Insurance_Adjusters | Number of insurance adjusters available | | Loan_Processors | Number of loan processors available |
| Permit_Processors | Number of permit processors available | | Building_Mat_ Reload_Time | Amount of time of building materials becoming available |

**2. Durations (Days)**

| Variable | Meaning | | Variable | Meaning |
|---|---|---|---|---|
| Inspection_Time | Time to complete inspection | | Adjuster_Time | Time to complete insurance claim |
| FEMA_Process_ Time | Time to complete FEMA aid application | | Loan_Process_ Time | Time to complete a loan and receive money |
| Engineering_ Assessment_Time | Time to complete an engineering assessment | | Permit_Process_ Time | Time to complete and receive a building permit |

**Figure 4-7: Human capital output parameters durations and their meanings**

Validation methods try to ensure that the built model reasonably reflects the real-world system being modeled. The easiest way to validate the model is to look at its constituent parts, and validate them on an individual basis. Validation techniques used in this simulation include comparing inputs and assumptions to real world measurements. The validity is also shown in the way the conceptual framework was designed modularly with many constituent parts of the whole.
The next chapter, Chapter Five describes the structure of the conceptual results and output from the simulation runs, as well as a discussion of implications, limitations, and further work, followed by a chapter concluding the work.

**4.2 Case Study Area**

Earthquakes and tsunamis pose significant threats to Pacific Northwest coastal regions, harbors, and communities (Parsons et al., 1998). These communities may be subjected to substantial damage to infrastructure, utilities, and loss and general obstruction of daily life as a result of these disasters.

The study area, Pacific County (shown in Figure 4-8), Washington, was chosen for its proximity to the ocean, and notably for its particular vulnerability in the event of a Magnitude 9.0 Cascadia Subduction Zone earthquake. Recent modeling efforts conducted by governmental and private organizations have estimated heavy losses in this area in the event of such a disaster.

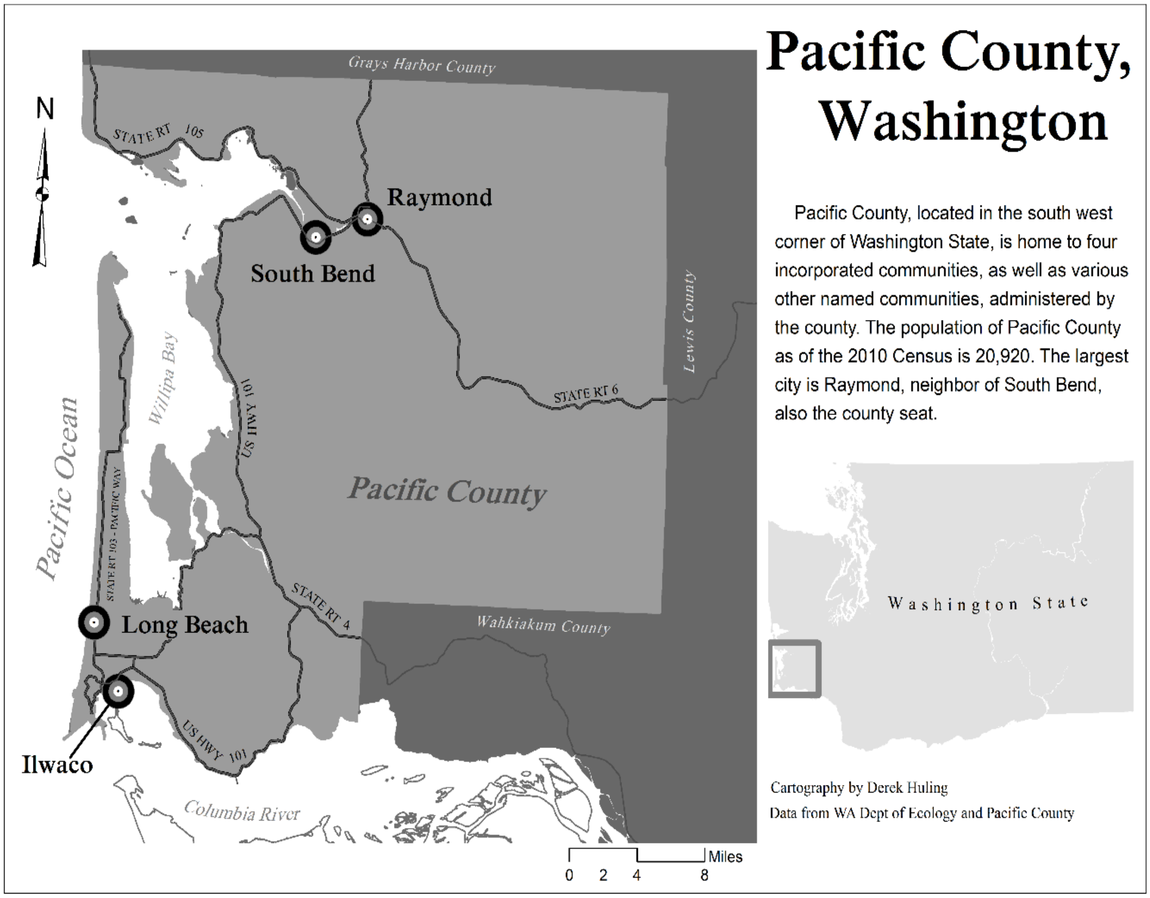


**Figure 4-8: Pacific County, Wa – Case Study Area**

Washington State's location at the convergence of the Pacific and Juan de Fuca tectonic plates increases risk of subduction zone, Benioff zone, and shallow crustal earthquakes. In the last 125 years, it has experienced 20 damaging earthquakes, and is considered the state with the second highest susceptibility to economic loss by earthquakes (FEMA, 2008). At least 7 tsunamis have been triggered by these events over the past 3,500 years. There is an estimated 10-14% chance of another tsunami occurring in the next 50 years (Wood & Soulard, 2008). Among the seismic events possible, an earthquake along the Cascadian Subduction Zone (CSZ) has been modeled extensively, and inundation levels have been calculated for a tsunami following a M9.0 earthquake. The nearby towns of Seaview and Ocean Park (both located in Pacific County) show 100% inundation from a CSZ tsunami (Venturato, Arcas, & Utku, 2007).

Pacific County, Washington is situated north of the border between Oregon and Washington State, about 125 miles southwest of Seattle. Figure 4 shows a map of the proposed study area. It is bounded by the Columbia River to the south, the Pacific Ocean to the west, Grays Harbor County to the north, and Lewis County to the east. The four largest cities in the county are Raymond, Long Beach, South Bend (county seat), and Ilwaco. According to the 2010 U.S. Census, the population of Pacific County was 20,920.

Timber, tourism, and fisheries are the noted industries of the area (Vleming, 2012). While timber was formerly the largest employer in Pacific County, it has since declined, leading to smaller populations in the area (Gable, 2011). The estuaries surrounding the study cities include the Columbia River and Willapa Bay, both of which house significant aquaculture and fishing activity, most notably oyster farms, which account for a large percentage of the total economy. These estuaries also contain unique ecosystems and recreational areas that draw tourists to the county. While the residential population is relatively low, coastal tourism may draw many times the average population number during the warmer months, which must be accounted for when planning for any hazard scenario.

## 5. Results and Discussion

Chapter 5 details the results of both the construction of the conceptual framework of disaster recovery, as well as the simulation model developed from it focusing on household reconstruction. Section 5.1 begins with the conceptual framework and all 6 elements of the framework. Section Two details and discusses the results of the simulation model. Section Three provides a discussion on the constraints and limitations of this work.

### 5.1 Simulation Results

Raw results from the simulation originate from the variables and attributes chosen to be examined at the outset of our simulation design – see Figures 4-6 and 4-7 for descriptions of these variables. Outputs from the model generally involve the tracing of interactions through various discrete processes and events of household reconstruction identified as being salient. This necessarily includes bottlenecks at each point where competition exists for a resource, as well as the resource constraints themselves. In this section, results are presented from the iterative runs of the model, chosen at 1000.

The simulation, as the conceptual framework suggested in Chapter Four, is very modular. Only one “master” flow control was used in this thesis due to the simplicity of the processes used. While the modular nature of the conceptual framework meant that other flow control was possible. For example, if FEMA decided that it didn’t require insurance verification before issuing financial support, those processes could be run concurrently instead of sequentially. Similarly, another building block of "USDA Agricultural Grant" could be added as a resource, cutting into the FEMA benefits of the homeowner. While this could have been practically demonstrated in this paper, it would not have matched empirical housing recovery processes documented in actual disasters. It would not have made sense to arrange the simulation having certain processes go before the others. Yet, the modularity means that as additional processes and sub-processes are added to the simulation, the ordering of processes could be varied to explore how process order impacts simulation outcomes like time to reconstruction.

### 5.2 Output Results

The simulation run contained over 2860 households, equating to a finished database size of 2.86 million rows of data spanning 34 rows, plus 1000 rows of nine parameters – different parameters for each simulation run – and 1000 rows for each duration randomization. In total, the simulation contained three database tables and just over 97 million records. These numbers do not reflect the simulation outputs.

The output of the simulation is a derivative tabulation that calculates total rebuilding wait-times, ranges of wait-times, as well as queries of different parameter counts. Figure 5-1 shows four columns containing derivative data: Average wait-time, minimum wait-time, maximum wait-time, the range of wait-times in that particular simulation run, and the simulation run count itself. Included are the first 10 results out of 1000, with the 11$^{th}$ row being the average of the entire set of 1000.

| Average Wait | Min Wait | Max Wait | Range of Waits | Simulation Iteration # |
|---|---|---|---|---|
| 966.02 | 62 | 2592 | 2530.48 | 1 |
| 683.74 | 62 | 1343 | 1281.05 | 2 |
| 921.58 | 62 | 1814 | 1751.79 | 3 |
| 2359.14 | 63 | 4681 | 4618.12 | 4 |
| 3058 | 62 | 10068 | 10006.22 | 5 |
| 3641.59 | 63 | 7230 | 7167.39 | 6 |
| 1733.51 | 62 | 3404 | 3341.25 | 7 |
| 1146.03 | 62 | 2238 | 2176.48 | 8 |
| 5876.4 | 62 | 11692 | 11629.5 | 9 |
| 1787.81 | 64 | 4478 | 4414.06 | 10 |
| Mean over 1000 records | | | | |
| 1920.695 | 415.5 | 4226.5 | 3811.33 | |

**Figure 5-1: Derivative wait-time matrix with simulation iteration. Wait times in Days.**

The derivative wait-time matrix describes wait-times in each simulation run. The “Range of Waits” column represents the difference of the maximum and minimum wait times of that simulation run. Each simulation run is made up of 2860 households, so this number represents the average of each household in that simulation number. The wait times are the difference between when a household requested an inspection and when that household’s home repairs were completed. The final row, average, is an average of averages, since each of the records in the table is an average from a simulation run of 2860 households. The results are further expanded upon in Figure 5-2, which displays the parameter data for the same subset of data – the first 10 results and an average of all 1000 results. As expected, because of the random parameterization between 1 and 100, the average lies around 50 for the human capital parameters. To expand a bit, it is useful to explore the outputs of Figure 5-1 with Figure 5-2. The simulation number with the highest and lowest range of wait times in Figure 5-1 is simulation iteration number nine and two, respectively.

| Contractors | Engineers | FEMA Processors | Inspectors | Simulation Iteration |
|---|---|---|---|---|
| 85 | 70 | 98 | 78 | 1 |
| 81 | 93 | 86 | 51 | 2 |
| 62 | 43 | 70 | 95 | 3 |
| 92 | 16 | 73 | 31 | 4 |
| 41 | 58 | 77 | 85 | 5 |
| 64 | 30 | 93 | 30 | 6 |
| 60 | 22 | 67 | 50 | 7 |
| 57 | 42 | 34 | 87 | 8 |
| 75 | 11 | 8 | 65 | 9 |
| 14 | 47 | 50 | 20 | 10 |
| Mean over 1000 records | | | | |
| 48.391 | 50.429 | 51.769 | 49.168 | - |

| Insurance Adjusters | Loan Processors | Permit Processors | Simulation Iteration |
|---|---|---|---|
| 13 | 21 | 97 | 1 |
| 10 | 50 | 75 | 2 |
| 99 | 77 | 75 | 3 |
| 55 | 14 | 73 | 4 |
| 37 | 3 | 20 | 5 |
| 16 | 29 | 13 | 6 |
| 42 | 28 | 71 | 7 |
| 75 | 91 | 43 | 8 |
| 63 | 44 | 8 | 9 |
| 55 | 37 | 52 | 10 |
| Mean over 1000 records | | | |
| 50.004 | 51.138 | 49.079 | - |

**Figure 5-2: Parameters of human capital for first ten records, mean for all records**

Examining the parameters of those iteration runs helps explain the large variance seen between those two runs. The difference between the parameters is shown in Figure 5-3. It is apparent that a large discrepancy exists between the two simulation run averages and their access to human capital resources, with the largest difference being engineers, followed closely by FEMA and Permit processors. Engineers and permit processors are needed by every entity that enters the rebuilding phase, meaning they have money and just need to move through the processes. FEMA processors are needed as well, yet homeowners will give up and get loans if they need to accumulate the necessary financial capital to rebuild. This large discrepancy is highlighted especially in the difference between the total number of human capital available during each simulation run, with simulation 9 having 172 fewer total human capital resources. This demonstrates the reality that when access to resources are diminished or limited during housing recovery, wait-times increase substantially.

| Resource | Sim Run 2 | Sim Run 9 | Difference |
|---|---|---|---|
| Contractors | 81 | 75 | -6 |
| Engineers | 93 | 11 | -82 |
| FEMA_Processors | 86 | 8 | -78 |
| Inspectors | 51 | 65 | 14 |
| Insurance_Adjusters | 10 | 63 | 53 |
| Loan_Processors | 50 | 44 | -6 |
| Permit_Processors | 75 | 8 | -67 |
| Sum | 446 | 274 | -172 |

**Figure 5-3: Resource parameter amounts for two simulation runs, as well as their difference, and the sum of the human capital parameters**

Totals and averages are interesting results for trends and to verify that the model is functioning as intended. However, another unit of analysis, the homeowner, also provides useful and interesting results. To expand on the example in Figure 5-3, individual stats for both simulation runs – two and nine – show the manner in which these interactions propagate. Instead of looking at the average of thousands or millions of rows, viewing a ‘story’ from a single homeowner from a single simulation is useful.

Figure 5-4 is a direct output from one of the households in simulation two. This household has the longest recovery wait time, and was selected to show the raw output of the simulation. The ‘story’ is provided to display human-readable sentences of the homeowner’s journey through the simulation, and is also part of the output. In this example query from the results database, the model returns 35 variables, each with a value with different units. The units of these are shown in Figure 4-6, in Chapter 4. The majority, however, are in days since the simulation began. To begin to explain the results of the reconstruction wait-time of several years – 1344 days – one can look at the raw data and review where bottlenecks occurred. Another option is to read the ‘story,’ of the fictitious homeowner “Brittany Pierce” in the paragraph style cell at the bottom of Figure 5-4. Shown under the “damage_state” record, Brittany’s house was damaged Completely, meaning full value needs to be replaced, in her case $144,100. Her house was inspected quickly, within 3 days of the event. She then needed money to rebuild, which came from both loans and insurance, though it took several years to secure enough financing to rebuild. After securing engineering, permitting, and building materials, her house was rebuilt in 60 days due to it being a mobile home, 3.5 years after the event. The main bottleneck in her processes were finding rebuilding money, after which the availability of human capital in the form of permit processors, engineers, and construction workers were not a hindrance. Thus, viewing the outputs first in-aggregate, then examining specific cases within the model becomes a powerful exploratory tool from a simulation model standpoint as well as a reconstruction effort and logistics standpoint.

| Variable | Value | Variable | Value |
|---|---|---|---|
| Reconstruction_wait | 1343.26233 | latitude | 46.63524 |
| assessment_get | 1086.84255 | loan_amount | 0 |
| assessment_put | 1060.78499 | loan_get | 1060.78499 |
| assistance_get | 22.91689 | loan_put | 22.91689 |
| assistance_payout | 0 | longitude | -123.9266 |
| assistance_put | 3.45444 | materials_get | 1283.26233 |
| assistance_request | 30000 | materials_put | 1283.26233 |
| claim_get | 264.58298 | money_search_start | 3.45444 |
| claim_payout | 144100 | money_search_stop | 1060.78499 |
| claim_put | 3.45444 | money_to_rebuild | 148771.1745 |
| home_get | 1343.26233 | moved | - |
| home_put | 1283.26233 | name | BRITTANY PIERCE |
| home_search_start | - | permit_get | 1283.26233 |
| home_search_stop | - | permit_put | 1086.84255 |
| inspection_get | 3.45444 | savings | 4671.17452 |
| inspection_put | 0 | sim_run | 2 |
| Insurance | 144100 | damage_state | Complete |

| story |
|---|
| Brittany Pierce lives in a 5 bedroom Mobile Home at None worth $144,100. Its damage level from the event was Complete. Brittany Pierce's house was inspected 3 days after the event and suffered $144,100 of damage. Brittany Pierce submitted an insurance claim 3 days after the event. Brittany Pierce submitted a request to FEMA 3 days after the event. Brittany Pierce received no money from FEMA because of inadequate funding. Brittany Pierce submitted a loan application 23 days after the event. Brittany Pierce received a $144,100 insurance payout 265 days after the event. It took Brittany Pierce 1057 days to receive enough financial assistance and now has $148,771 to rebuild. Brittany Pierce received an engineering assessment 1087 days after the event. Brittany Pierce received permit approval 1283 days after the event. Construction materials were received at 1283. Brittany Pierce's home was repaired 1343 days after the event, taking 60 days to repair. |

**Figure 5-4: Example output from one household in the simulation.**

One of the key motivations for this research was exploring recovery as a complex system, and designing and implementing a method to do so. Some of the results in this section are very exploratory. Running statistical analysis on a model in which the processes are deterministic, even if the parameters are stochastic, would be a fruitless endeavor because correlated variables were designed that way. However, exploring the data and explaining the reasons for this output are interesting results, regardless of the lack of statistical rigor.

The variable parameterization in the model is where the behavior of the model is evaluated. Changing the quantities or durations of the shared resources predictably alters the final model results (as it should according to the verification techniques). Durations of the model are shown in Figure 5-5, and can be queried along with all the other data, as well as appended to any other query. Each step represents a potential bottleneck in the simulation, making logical decisions and case switches meaningful (a homeowner receiving adequate insurance payouts is unlikely to wait for loan approval before contracting with builders). The modularity of the model ensures that a user can quite easily add a process function to enhance validity or represent a particular step they are interested in exploring. The model succeeds in that respect quite well.

| Time (Days) | Inspection | Adjuster | FEMA Process | Engineering Assessment | Loan Process | Permit Process |
|---|---|---|---|---|---|---|
| Average | 1.1499 | 15.0305 | 19.9941 | 24.9992 | 30.0663 | 34.9995 |
| Minimum | 0.0002 | 12.4198 | 16.5630 | 20.6308 | 27.3622 | 31.4448 |
| Maximum | 3.8802 | 18.3183 | 22.8712 | 29.0748 | 32.7613 | 38.2898 |

**Figure 5-5: Durations of processes, in days.**

## 6. Conclusions

This chapter concludes the work presented in this thesis. Section One reiterates and concludes the research questions and objectives. Section Two describes the limitations present in the work. Section three lays a groundwork for future development in simulation and modeling of disaster recovery phenomena.

### 6.1 Research Questions and Objectives

This research set out to answer the questions, how can like concepts of recovery be grouped into a framework; and how can such a framework be used to effectively simulate housing reconstruction? These questions led to: 1) building a conceptual framework that describes elements of owner-occupied household reconstruction, and allows new elements to be included as they're hypothesized; 2) operationalizing this framework in a simulation of owner-occupied household reconstruction, using the elements and constructs provided in the conceptual framework.

The first objective was accomplished by grouping like-concepts of recovery into a typology, generally depending on their function to provide, receive, grant, or otherwise alter access to resources after a disaster or hazard event. These constructs were labeled as: entities, processes, attributes, resources, events, and context. The framework was able to include various phenomenon of recovery and had the flexibility to be adapted to the needs of the researcher.

Applying the concepts derived from the framework to a working, programmed simulation constituted the second objective, and was accomplished using the Python programming language and various other tools, including SimPy, developed for scientific research. Over 1000 lines of code, and several hundred development hours went into writing, testing, verifying, and validating the simulation code and output.

### 6.2 Limitations

Many of the limitations of the model stem from the technology and method of simulation chosen for the research. Discrete-event simulations are not continuous, time-step simulations where we can look at each day of the simulation and trace "progress" of reconstruction. By some measure this can be viewed with a positive light, as a partially reconstructed house does not mean much for a homeowner. However, if a user of the simulation wanted to step through every day of reconstruction or some other recovery scenario used with the model, a continuous, perhaps agent-based, design would be more appropriate.

Limitations of the content of the model are myriad. Human behavior was not simulated with any further complexity than attempting to rebuild and, where rebuilding proves impossible or takes too long, to search for new shelter. This avoids modeling behavior such as fear or trauma, or behavior based on other reasons such as loss of job for adults, or school availability for households with children. Household makeup is also left out of the model, an attribute which could be highly useful when homeowners are making decisions. While the above are limitations in this implementation of the simulation model, the conceptual framework provides for these circumstances, and could be added later as attributes.

Data limitations were also present in the model. The HAZUS data was at least three years out of date. At the time of design and run of this research, the tsunami module of HAZUS was not yet operational, so the model was unable to account for damage influenced by a tsunami. Insurance data is not easy to come by, especially in a geographic community the size of Pacific County, so accuracy could have been increased with respect to insurance claims and payouts.

Each process in homeowner reconstruction, and in recovery, is an incredibly complicated procedure of qualifications, paperwork, access to resources, and behavioral system that likely can never be fully or adequately modeled. Additionally, there are many geographic, political, and economic variables and decisions to be made at every level. Content limitations reflect the above complexity; however, it offers exciting avenues of future research – examined in the following section.

Validation was a weakness of the model. No substantive validation was performed. Expert elicitation, comparison to historical data, other basic validation techniques was not performed during the development of the simulation. The case study data was to show the use and utility of the model, not to show the reality of a reconstruction effort in Pacific County, Washington.

### 6.3 Future Work in Recovery Simulation

Simulation as an analytical and exploratory tool within disaster recovery is a bourgeoning technique, and its creation presents a challenge of concept, implementation, and skillset among disaster researchers. The simulation presented here offers a new approach in the use of discrete-event simulation and is a convincing use case that simulation as a whole is a useful method for viewing and understanding behaviors, interactions, and outcomes from planning and resilience perspectives.

Future use and further research of simulation models within the field are immense, and almost every new publication cycle brings with it new models and concepts in recovery. Combining different simulations into a common simulation or using outputs of one simulation as inputs to another simulation are simple ways to combine complex research while using their results. For example, Nejat (Nejat & Damnjanovic, 2012) simulation outputs could be used as a decision matrix for entities in a discrete-event simulation’s decision behavior. Research continues into combining elements of agent-based simulations with discrete-event, which may be more effective than one or the other alone. The addition of validation would strengthen the model results and give a more robust validity to the simulation. Any future study using the building blocks of the simulation, such as the parameterization, inputs, and flow control of the model should include validation as well as verification techniques.

One of the use-cases of simulations of this type are the ability to turn a diagram of events into a working simulation. For example, in a parliamentary paper from the New Zealand auditor general’s office, diagrams are used to show the complex decisions that have to be made by homeowners following disasters (Provost, 2012). These diagrams can be explored by constructing a simple simulation from the pieces already modularly coded in this research, in an arrangement that reports claim are valid. This use transforms these ideas from simple paper diagrams to fully interactive and explorable simulations with outputs that can be displayed and analyzed.

As development continues on simulation and other modeling techniques in recovery, increasing levels of complexity, size, and scope will improve these models for use in tools for planning, exploring, and researching effective ways to minimize recovery times, increase recovery quality, and mitigate future roadblocks to effective recovery. Simulation will likely be a useful, active tool in the pocket of emergency management, planning professionals, and recovery mitigation specialists of the future, guiding decisions and increasing awareness of the milieu of concepts needed to effectively manage and precipitate a successful recovery.

## References

Alexander, D. (2004). Planning for post-disaster reconstruction. *I-Rec 2004 International Conference Improving Post-Disaster Reconstruction in Developing Countries*.

Appleby, T. (1994, October 18). First good news, then bad news on Los Angeles SON OF EARTHQUAKE / The Sequal to L.A.'s terrifying disaster in January is the horrifying new reality of costly and inadequate home insurance. *The Globe and Mail (Canada)*.

Arvedlund, E. (2015). N.J. payout to homeowners for Hurricane Sandy slow - philly-archives. Retrieved March 22, 2015, from http://articles.philly.com/2015-03-02/real_estate/59647208_1_hurricane-sandy-contractors-grant-process

Axelrod, R. (1997). Advancing the art of simulation in the social sciences. In *Simulating social phenomena* (pp. 21–40). Springer. Retrieved from http://www.agsm.edu.au/bobm/teaching/SimSS/AdvancingArtSim2003.pdf

Becerra, O., Cavallo, E., & Noy, I. (2010). In the Aftermath of Large Natural Disasters, what happens to foreign aid?

Ben-Zvi, T. (2010). The efficacy of business simulation games in creating Decision Support Systems: An experimental investigation. *Decision Support Systems*, *49*, 61–69. https://doi.org/10.1016/j.dss.2010.01.002

Binder, S. B., Baker, C. K., & Barile, J. P. (2015). Rebuild or Relocate? Resilience and Postdisaster Decision-Making After Hurricane Sandy. *American Journal of Community Psychology*, 180–196. https://doi.org/10.1007/s10464-015-9727-x

Binder, S. B., & Greer, A. (2016). The Devil Is in the Details: Linking Home Buyout Policy, Practice, and Experience After Hurricane Sandy. *Politics and Governance*, *4*(4), 97. https://doi.org/10.17645/pag.v4i4.738

Bolin, R., & Stanford, L. (1998). The Northridge Earthquake: Community-based Approaches to Unmet Recovery Needs. *Disasters*, *22*(1), 21–38. https://doi.org/10.1111/1467-7717.00073

Cagnan, Z., & Davidson, R. A. (2015). Discrete event simulation of the post-earthquake restoration process for electric power systems. *International Journal of Risk Assessment and Management*. Retrieved from http://www.inderscienceonline.com/doi/abs/10.1504/IJRAM.2007.015298?journalCode=ijram

Çagnan, Z., Davidson, R., & Guikema, S. (2004). Post-earthquake restoration modeling of electric power systems. *… of the 13th World Conference on …*, (109). Retrieved from http://www.iitk.ac.in/nicee/wcee/article/13_109.pdf

Castaneda, C. (1994, January 20). Now, building inspectors make people tremble. *USA Today*, p. 3A.

Chandrasekhar, D., & Finn, D. (2013). POST-DISASTER RELOCATION A Case Study of The Rockaways , NY after Hurricane Sandy, (September), 2–22.

Chang-Richards, Y., Wilkinson, S., Seville, E., & Brunsdon, D. (2015). A systems approach to managing human resources in disaster recovery projects, 15–17.

Chang, Y., Wilkinson, S., Potangaroa, R., & Seville, E. (2010). Resourcing challenges for post-disaster housing reconstruction: a comparative analysis. *Building Research & Information*, *38*(3), 247–264. https://doi.org/10.1080/09613211003693945

Cohen, S. (2013, October 29). Hurricane Sandy showed that our rebuilding efforts need reform. *Al Jazeera America*. Retrieved from http://america.aljazeera.com/articles/2013/10/29/hurricane-sandy-disasterreliefreconstructionfunding.html

Coleman, J. S. (1961). Analysis of Social Structures and Simulation of Social Processes With Electronic Computers. *Educational and Psychological Measurement*, *21*(1), 203–218. https://doi.org/10.1177/001316446102100121

Comerio, M. C. (1995). *Northridge Housing Losses: A Study for the California Governor's Office of Emergency Services*. Center for Environmental Design Research, University of California at Berkeley.

Comerio, M. C. (1997). Housing Issues After Disasters. *Journal of Contingencies and Crisis Management*, *5*(3), 166–178. https://doi.org/10.1111/1468-5973.00052

Cushman, T. (2015). FEMA Official Admits Fraud in Sandy Flood Cases | JLC Online | Insurance, Legal Issues. Retrieved March 8, 2017, from http://www.jlconline.com/coastal-contractor-news/fema-official-admits-fraud-in-sandy-flood-cases_o

Dandoulaki, M., Panoutsopoulou, M., & Ioannides, K. (1998). An overview of post-earthquake building inspection practices in Greece and the introduction of a rapid building ... Greece and the introduction of a rapid building usability evaluation, (May).

Dwyer, C., & Horney, J. (2014). Validating Indicators of Disaster Recovery with Qualitative Research. *PLoS Currents*, (April). https://doi.org/10.1371/currents.dis.ec60859ff436919e096d51ef7d50736f

*Earthquake Insurance*. (2016). Retrieved from http://www.naic.org/cipr_topics/topic_earthquake_insurance.htm

Erdik, M., Sesetyan, K., Demircioglu, M. B., Hancilar, U., Zulfikar, C., & Erdik, M. (2011). Rapid Earthquake Loss Assessment After Damaging Earthquakes. *Soil Dynamics and Earthquake Engineerin*, *31*(2), 247–266. Retrieved from http://s3.amazonaws.com/academia.edu.documents/11523608/Erdik_et_al_EQEng_in_Europe.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1489019356&Signature=Tq%2B6pZ1eotCdyjT9h5z7BQgnB7Y%3D&response-content-disposition=inline%3B filename%3DEarthquake_Enginee

FEMA. (2015). What to Expect from a FEMA Inspection. Retrieved June 7, 2016, from http://www.fema.gov/news-release/2015/06/12/what-expect-fema-inspection

Fugate, C. (2013). *ONE YEAR LATER: EXAMINING THE ONGOING RECOVERY FROM HURRICANE SANDY*. Retrieved from https://www.fema.gov/media-library-data/1387208691195-dda79568fbb827633b3e095d61fa9f7a/11-6-2013_ONE YEAR LATER_EXAMINING THE ONGOING RECOVERY FROM HURRICANE SANDY.pdf

Gable, C. (2011, April). Pacific County regroups as Weyerhaeuser shrinks holdings. Retrieved June 12, 2014, from http://www.dailyastorian.com/special_sections/fate_of_our_forests/pacific-county-regroups-as-weyerhaeuser-shrinks-holdings/article_3da4ea90-5c7d-11e0-a3c9-001cc4c03286.html

Gharaati, M. (2006). an Overview of the Reconstruction Program After the Earthquake of Bam , Iran, 1–14.

Gregg, C., & Lofton, L. (2011). *Joplin, MO, Tornado Lessons Learned Study THE RESPONSE TO THE 2011 JOPLIN, MISSOURI, TORNADO LESSONS LEARNED STUDY*.

Hayashi, H. (2007). Long-term Recovery from Recent Disasters in Japan and the United States, 413–418.

Higuchi, Y., Inui, T., Hosoi, T., Takabe, I., & Kawakami, A. (2012). The impact of the Great East Japan Earthquake on the labor market: need to resolve the employment mismatch in the disaster-stricken areas. *Japan Labor Review*, 4–21. Retrieved from http://www.jil.go.jp/english/JLR/documents/2012/JLR36_higuchi.pdf

*Homeowners Insurance: Understanding, Attitudes and Shopping Practices*. (2017). Retrieved from http://www.iii.org/sites/default/files/docs/pdf/pulse-wp-020217-final.pdf

Huggel, C., Raissig, A., Rohrer, M., Romero, G., Diaz, A., & Salzmann, N. (2015). How useful and reliable are disaster databases in the context of climate and global change? A comparative case study analysis in Peru. *Nat. Hazards Earth Syst. Sci*, *15*, 475–485. https://doi.org/10.5194/nhess-15-475-2015

Johnson, L. a, & Olshansky, R. B. (2013). The Road to Recovery: Governing Post-Disaster Reconstruction. *Land Lines*, *25*, 14–21.

Karnon, J., Stahl, J., Brennan, A., Caro, J. J., Mar, J., & Möller, J. (2012). Modeling using Discrete Event Simulation : A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force-4 Background to The Task Force. *Value in Health*, *15*, 821–827. https://doi.org/10.1016/j.jval.2012.04.013

Kumar, S., Diaz, R., Behr, J. G., & Toba, A. L. (2015). Modeling the effects of labor on housing reconstruction: A system perspective. *International Journal of Disaster Risk Reduction*, *12*, 154–162. https://doi.org/10.1016/j.ijdrr.2015.01.001

Lindell, M. K. (2013). Recovery and reconstruction after disaster. In *Encyclopedia of Natural Hazards* (pp. 812–824). Springer.

Luna, R., Balakrishnan, N., & Dagli, C. H. (2011). Postearthquake recovery of a water distribution system: discrete event simulation using colored petri nets. *Journal of Infrastructure Systems*, *17*(1), 25–34.

Melnyk, S. A., Rodrigues, A., & Ragatz, G. L. (2009). Using simulation to investigate supply chain disruptions. In *Supply Chain Risk* (pp. 103–122). Springer.

Miles, S. B., & Chang, S. E. (2006). Modeling community recovery from earthquakes. *Earthquake Spectra*, *22*(2), 439–458.

Miles, S. B., & Chang, S. E. (2011). ResilUS: A Community Based Disaster Resilience Model Scott B. Miles and Stephanie E. Chang. *Cartography and Geographic Information Science*, *38*(1), 5–21.

Mummey, R. (2013). Lessons From Joplin's Tornado Recovery Effort. Retrieved March 1, 2017, from http://news.nationalgeographic.com/news/2013/05/pictures/130522-tornado-recovery-pictures-photos-joplin-missouri-moore-oklahoma/

Munnell, A. H., Soto, M., & Aubry, J.-P. (2007). DO PEOPLE PLAN TO TAP THEIR HOME EQUITY IN RETIREMENT? *Center for Retirement Research 2*, *7*(7).

Nejat. (2011). *Modeling Dynamics of Post Disaster Recovery*. Texas A&M University.

Nejat, A., & Damnjanovic, I. (2012). Modeling Dynamics of Post-Disaster Recovery. In *Construction Research Congress 2012@ sConstruction Challenges in a Flat World* (pp. 2200–2210).

Parsons, T., Trehu, A. M., Luetgert, J. H., Miller, K., Kilbride, F., Wells, R. E., … Christensen, N. I. (1998). A new view into the Cascadia subduction zone and volcanic arc: Implications for earthquake hazards along the Washington margin. *Geology*, *26*(3), 199–202.

Paul, B. K., & Stimers, M. (2015). Safety Measures After The 2011 Joplin, Missori, Tornado. *Geographical Review*, *105*(2), 199–215. Retrieved from https://www.researchgate.net/profile/Mitchel_Stimers/publication/273836056_Safety_Measures_after_the_2011_Joplin_Missouri_Tornado/links/552d7e880cf21acb09217681.pdf

Podger, O. (2013). Adapting {Professional} {Practices} for {Post}- {Disaster} {Reconstruction}. *PM World Journal*, *II*(Iv), 1–22.

Porpora, T. (2014, October 28). Many still await help, wrapped in Hurricane Sandy web of bureaucracy. *SILive*. Retrieved from http://www.silive.com/news/index.ssf/2014/10/born_out_of_hurricane_sandy_ar.html

Power, D. J. (2007). A brief history of decision support systems. *DSSResources. COM, World Wide Web, http://DSSResources. COM/history/dsshistory. Html, Version*, *4*.

Power, D. J., & Sharda, R. (2007). Model-driven decision support systems: Concepts and research directions. *Decision Support Systems*, *43*, 1044–1061. https://doi.org/10.1016/j.dss.2005.05.030

Provost, L. (2012). *Roles , responsibilities , and funding of public entities after the Canterbury earthquakes*.

Quarantelli, E. L. (1982). General and particular observations on sheltering and housing in American disasters. *Disasters*, *6*(4), 277–281. https://doi.org/10.1111/j.1467-7717.1982.tb00550.x

Red Cross Awards Nearly $6 Million in Grants for Sandy Recovery [Press Release]. (2013). Retrieved from http://www.redcross.org/news/press-release/Red-Cross-Awards-Nearly-6-million-in-Grants-for-Sandy-Recovery

Sargent, R. G. (2011). VERIFICATION AND VALIDATION OF SIMULATION MODELS. In *Proceedings of the 2011 Winter Simulation Conference* (pp. 183–198). Retrieved from http://www.informs-sim.org/wsc11papers/016.pdf

Scherfke, S., & Lünsdorf, O. (2015). Simpy 3.0.8 Documentation. Retrieved July 28, 2015, from https://simpy.readthedocs.org

Schmitt, A. J., & Quantifying, M. S. (2009). Quantifying supply chain disruption risk using Monte Carlo and discrete-event simulation. In *Proceedings of the 2009 Winter Simulation Conference*.

Schwanhausser. (1994, June 24). The Battle Over Quake Insurance Fear of the Big One has State Insurers Running for Cover. *San Jose Mercury News*, p. 1G. Retrieved from http://www.lexisnexis.com.offcampus.lib.washington.edu/hottopics/lnacademic/

Singh, B., & Wilkinson, S. (2008). Post-disaster Resource Availability following a Wellington Earthquake: Aggregates, Concrete and Cement. *I-Rec 2008 Building Resilience: Achieving Effective Post-Disaster Reconstruction*, *33*, 3–14.

Smith, D. (1995, October 9). Quake Repairs Fall Behind in Poor, Old Neighborhoods; Recovery: U.S. loan policies slow effort. In some areas, less than half the affected buildings have gotten permits. *Los Angeles Times*, p. 1.

Smith, G. B. (2013). Sandy victims fighting for FEMA payouts - NY Daily News. Retrieved March 2, 2017, from http://www.nydailynews.com/new-york/hurricane-sandy/sandy-1-year-fema-article-1.1497143

Snediker, D. E., Murray, A. T., & Matisziw, T. C. (2008). Decision support for network disruption mitigation. *Decision Support Systems*, *44*, 954–969.

Srbljinovic, A., & Škunca, O. (2003). An Introduction to Agent Based Modelling and Simulation of Social Processes. *Interdisciplinary Description of Complex Systems : INDECS*, *1*(1–2), 1–8. Retrieved from http://hrcak.srce.hr/76893

Stafford, R. T. (2013). The Robert T. Stafford Disaster Relief and Emergency Assistance Act, (April), 1–79. https://doi.org/10.1016/B978-1-85617-547-0.00014-2

Stevenson, J. R., Emrich, C. T., Mitchell, J. T., & Cutter, S. L. (2010). Using Building Permits to Monitor Disaster Recovery: A Spatio-Temporal Case Study of Coastal Mississippi Following Hurricane Katrina. *Cartography and Geographic Information Science*, *37*(1), 57–68. https://doi.org/10.1559/152304010790588052

Sullivan, K., & Uccellini, L. (2013). *Service Assessment: Hurricane/Post-Tropical Cyclone Sandy October 22-29, 2012*. Retrieved from http://www.weather.gov/media/publications/assessments/Sandy13.pdf

Tabucchi, T., Davidson, R., & Brink, S. (2010). Simulation of post-earthquake water supply system restoration. *Civil Engineering and Environmental Systems*, *27*(4), 263–279. https://doi.org/10.1080/10286600902862615

Venturato, A., Arcas, D., & Utku, K. (2007). *Modeling Tsunami Inundation from a Cascadia Subduction Zone Earthquake for Long Beach and Ocean Shores Washington*. Retrieved from http://www.pmel.noaa.gov/pubs/PDF/vent2949/vent2949.pdf

Vigeland, T. (2012). From Joplin Tornado Victims, Advice on Disaster-Proofing Finances. Retrieved February 20, 2017, from http://www.nytimes.com/2012/11/14/your-money/from-joplin-tornado-victims-advice-on-disaster-proofing-finances.html

Vleming, J. (2012). *Pacific County Profile*. Retrieved from https://fortress.wa.gov/esd/employmentdata/reports-publications/regional-reports/county-profiles/pacific-county-profile

Weston, L. P. (2017). Rethinking Your Stance on Earthquake Coverage - LA Times. *Los Angeles Times*. Retrieved from http://www.latimes.com/la-homeauto-story1-story.html

Wisner, B., Blaikie, P., Cannon, T., & Davis, I. (2004). At risk. *Natural Hazards, People's Vulnerability and Disasters*, *2*.

Wood, N., & Soulard, C. (2008). *Variations in community exposure and sensitivity to tsunami hazards on the open-ocean and Strait of Juan de Fuca coasts of Washington*.

Wu, J. Y. (2004). *A comparative study of housing reconstruction after two major earthquakes: The 1994 Northridge earthquake in the United States and the 1999 Chi-Chi earthquake in Taiwan*. Texas A&M University.

Wu, J. Y., & Lindell, M. K. (2004). Housing reconstruction after two major earthquakes: the 1994 Northridge earthquake in the United States and the 1999 Chi-Chi earthquake in Taiwan. *Disasters*, *28*(1), 63–81. https://doi.org/10.1111/j.0361-3666.2004.00243.x

Xu, N., Guikema, S. D., Davidson, R. A., Nozick, L. K., & Zehra, C. (2007). Optimizing scheduling of post-earthquake electric power restoration tasks. *Earthquake Engineering and Structural Dynamics*, (August 2006), 265–284. https://doi.org/10.1002/eqe

Zagier, A. S., Clark, C., & Lieb, D. A. (2012). From rubble to rebuilding: A year in Joplin, Mo. *The San Diego Union-Tribune*.

## Appendix A: Literature Typology Development Scratchpad

| **Phenomena** | **Notes** | **Type Schema** | **Hazard/Event [Time Interval]** | **Reference** |
|---|---|---|---|---|
| Sense of Place | Article suggests the "sense of place" is tied to home ownership, which is unusually high in the 9th ward | Movement | Katrina | |
| Previous Experience | previous experience with a similar event increases risk awareness, which may effect recovery | Index? | | http://www.nat-hazards-earth-syst-sci-discuss.net/2/6397/2014/nhessd-2- |

| | | | | |
|---|---|---|---|---|
| | | | | 6397-2014.pdf |
| Risk Awareness | | Index? | | http://www.nat-hazards-earth-syst-sci-discuss.net/2/6397/2014/nhessd-2-6397-2014.pdf |
| Risk Culture | regionally | Index? | | http://www.nat-hazards-earth-syst-sci-discuss.net/2/6397/2014/nhessd-2-6397-2014.pdf |
| Social Safety Nets | "necessary for disaster recovery" | Index? | theory | http://onlinelibrary.wiley.com/enhanced/doi/10.1111/1540-6237.8402002/ |
| Socio-Economic status | Little evidence that statuses change long-term after disaster | ? | Theory | |
| Income | This can affect access to loans, make aid more expensive | Amount | Katrina | Masozera 2007 |
| Demographic Inequalities | Gender, age (life cycle(Quarantelli 1999), race/ethnicity | Index? | theory | http://onlinelibrary.wiley.com/enhanced/doi/10.1111/1540- |

| | | | | |
|---|---|---|---|---|
| | | | | 6237.84020 02/ |
| Loss of Lifeline Services | you either have it or you don't, but there is a time associated. How does that time affect recovery outcomes? | Discrete Binary Duration | | Tierney, K. J., & Nigg, J. M. (1995). Business vulnerability to disaster-related lifeline disruption. |
| Temporary Business Relocation | | Discrete Event with Movement | | (Durkin, 1984), |
| Pre-Registering for Reconstruction | | Binary (possibly with amount for availability of services) | | (Wu and Lindell 2004)? |
| Relocation, planning | plan in place, with input from community increases recovery successs | Binary, Movement | Multiple | Maly E. 2014 |
| Presidential Disaster Declaration | unlock funding | Binary | Multiple | Disaster Recovery and Hazard Mitigation: Bridging the Intergovernmental Gap Claire B. Rubin and Daniel G. Barbee Public Administration Review, Vol. 45, Special Issue: Emergency Management: A Challenge for Public Administration (Jan., |

| | | | | |
|---|---|---|---|---|
| | | | | 1985), pp. 57-63 |
| Areal Size of Event | if surrounding areas were hit, less help can come in (community vs Regional vs National level events) | Spatial (maybe how many localities are affected? Or if it's a regional vs community disaster) | | |
| Intensity of Event | shaking, inundation, windspeed? | Index? | equakes | Dahlhamer Tierney 1996 |
| Business Size | smaller your business the more you're affected (longer recovery times, less access to credit, less mitigation strategies) | Negative Index? | | Dahlhamer Tierney 1996: Predicting Business Disaster Recovery outcomes following |
| Business Type | Construction, hospitality types of business do well, even if they were struggling before (Zhang etal). "If the general economic climate is poor for particular business sectors, disaster assistance is not likely to reverse those effects" (Dahlhamer and Tierney 1996 (Winners and Losers)) | Nominal / Categorical | | |

| | | | | |
|---|---|---|---|---|
| Risk Dispersion | Kind of similar to type. If its part of a chain, or a branch of a larger parent company | Categorical | | |
| Own/Lease | | Binary | | |
| Financial Condition | Article suggests this is a significant predictor, but this may be shaped by perception (survival is "winning" in the eyes of struggling businesses) | | | |
| Business Age | | | | |
| Business Interruption | inactivity, so called higher-order losses | Duration | | |
| Government Recovery Plan in place | | | | |
| Loss Containment | Insurance(Masozera says negative corr between poverty and flood insurance in katrina), financial aid. (Dahlhamer found aid recipients worse off, although this is because recipients of aid had to be really bad off before to receive it) | Index (No duration yet in literature), | | Dahlhamer Tierney 1996: Predicting Business Disaster Recovery outcomes following |
| Access to Transportation | car/truck/van for private use, not public transit | Binary or Index | Katrina | |
| Insurance Payouts | NEED ARTICLE ON INSURANCE PAYOUTS AND AFFECTS ON RECOV. | Discrete Payout with Duration | | |
| Beauracracy | stagnation and management issues (pace) (FEMA, SBA) | Duration (Processing applications, doling funds)fa | | |

| | | | | |
|---|---|---|---|---|
| Home Loan Applications | Within a year of katrina only 1/3rd of apps were reviewed, 82% of which were denied. This is related to "Loss Containment" | Binary, Amount | Katrina | http://www.nytimes.com/2005/12/15/national/nationalspecial/15loans.html?pagewanted=all&_r=0 |
| Post-event wages | Pay livable wages may affect recovery | ? | | |
| | | | | |
| | | | | |
| Inaccessability to business facility led to major or entire loss in business sales | 51% of respondents | Access / Topographic | 2010 Pakistan Floods [6 month study] | Asgary, A., Anjum, M. I., & Azimi, N. (2012). Disaster recovery and business continuity after the 2010 flood in Pakistan: Case of small businesses. *International Journal of Disaster Risk Reduction*, *2*, 46-56. |
| half of business owners had to borrow money to stay in business | | Debt | | |
| Dependency on water and transportation had significant impacts on recovery times | | Service Access | | |
| in developing countries, NGOs and the Gvn't focus more on households than businesses | | **** | | |
| Businesses with more personal savings had faster recovery | | Savings / Debt | | |
| "Social values and family bindings are the key survival factors in the emergency phase" | | Social Capital | | |

| | | | | |
|---|---|---|---|---|
| "...90% of the small businesses re-opened in the first six months after [event]… but only 22% of those re-opened businesses were performing at same or better than pre-flood level" | | **** | | |
| | | | | |
| During the recovery phase of emergencies, seniors are also less likely to receive financial aid or be considered candidates for post-disaster loan programmes | | inequality of services | | Powell, S., Plouffe, L., & Gorr, P. (2009). When ageing and disasters collide: lessons from 16 international case studies. *Radiation protection dosimetry*, *134*(3-4), 202-206. |
| New housing built for seniors following the earthquake in Kobe were also poorly adapted to their needs | | inequality of services | | |
| Lack of access to financial aid | | Access | | |
| not included in job recovery programs and retraining | | inequality of services | | |
| benefit application forms and processes that are difficult to understand or inaccessible | | transparency | | |
| Negative impacts were also mitigated by including seniors in recovery planning | | inclusion | | |

| | | | | |
|---|---|---|---|---|
| pre-existing socio-economic conditions play a significant role in the ability for particular economic classes to respond immediately to the disaster and to cope with the aftermath of Hurricane Katrina | | capacity to respond | | Masozera, M., Bailey, M., & Kerchner, C. (2007). Distribution of impacts of natural disasters across income groups: A case study of New Orleans.*Ecological Economics*, *63*(2), 299-306. |
| "Another factor that influences the recovery process is the ease with which certain groups are able to negotiate bureaucratic systems." | middle and high income are more comfortable than low | transparency | | |
| "To support their long-term economic recovery, lower-income groups must have access to reconstruction jobs, investment funds, and housing in safe locations, and more importantly support from FEMA and the SBA" | | Access | | |
| | | | | |
| "All else being equal, larger firms were more likely to be recovered than smaller ones" | | Business Size | Northridge / Loma Prieta | Webb, G. R., Tierney, K. J., & Dahlhamer, J. M. (2000). |

| | | | | |
|---|---|---|---|---|
| "What happens to an individual business organization depends importantly on how neighborhoods, critical infrastructural systems, and communities are affected by a disaster." | | Impact of Event | Various | Businesses and disasters: Empirical patterns and unanswered questions. *Natural Hazards Review*, *1*(2), 83-90. |
| "Experience in previous disasters is unrelated to recovery outcomes" | | | | |
| "For many businesses, then, recovery assistance brings additional indebtedness and draws down savings." | | Savings / Debt | | |
| "there is some evidence to suggest that individual business fates may well be more dependent on larger economic trends than on disaster-related factors." | | Economic trends | | |
| | | | | |
| Wind speed, water height, and coastal storm surges are primary causes of economic damage of hurricanes | | Impact of Event | | Caffey, R., & Diop, H. (2006) LOUISIANA'S RESPONSE TO |

| | | | | |
|---|---|---|---|---|
| price of fuel, economic trends mean less fisherman can afford to return to fishing | | Commodity prices / Access / Economic Trends | Katrina | HURRICANES KATRINA AND RITA . LLFET 2006 Portsmouth Proceedings |
| "How long capacity will remain at this new equilibrium, however, will be determined by a number of factors, including docside prices, fuel costs, post-storm fisheries abundance, and the speed and specifics through which federal disaster funding is ultimately disbursed." | | Commodity prices / Access / Economic Trends | | |
| | | | | |
| | | | | |
| MIGRATION | Did somebody migrate or not? | Depends on many factors (listed below) | | |
| disadvantaged population, housing damage, and more densely built environments led to higher levels of outmigration [and] exhibited geographic clustering | | Demographics / Migration | | Myers, C. A., Slack, T., & Singelmann, J. (2008). Social vulnerability and migration in the wake of disaster: the case of Hurricanes Katrina and Rita.*Population and Environment*, *29*(6), 271-291. |
| the decision to return to the place of origin may become a more individualistic cost–benefit analysis as time progresses. | | Individual Decision over Time | | |

| | | | | |
|---|---|---|---|---|
| Furthermore, over-dependence on a single economic sector (versus a more diverse economy) increases vulnerability, because if the sector is destroyed, so is the local ability to maintain a livelihood (Gramling and Freudenburg 1990; Freudenburg 1992) | | Business Type | | |
| In cases where economies and jobs are devastated by disasters, outmigration is the rational economic response for workers and their families. | | Individual Decision over Time | | |
| | | | | |
| adaptive vs. forced migration | | Choice | | Krishnamurthy, P. K. (2012). Disaster-induced migration: Assessing the impact of extreme weather events on livelihoods. *Environmental Hazards*, *11* (2), 96-111. |
| "The article reviews recent discussions on the relationships between extreme weather events and migration (both voluntary and forced) and suggests that, if adequately planned, relocation strategies can be an effective adaptation strategy" | | Choice with Warning | | |

| | | | | |
|---|---|---|---|---|
| Some disasters might coerce displacement but the decision to move away from the place of origin is based on an individual cost−benefit analysis (Myers et al., 2008) | | Choice over Time | | |
| | | | | |
| "By providing a bundle of club goods that allowed the members in this community to create what amounts to a “second homeland” and the coordination capacity to withstand a physically and politically inhospitable environment, the MQVN church has facilitated a level of social cooperation that has been difficult for many other communities to achieve" | | Social capital / unity | katrina | Chamlee-Wright, E., & Storr, V. H. (2009). Club goods and post-disaster community return. Rationality and Society, 21(4), 429-458. |
| socially embedded resources (in this case club goods) can play in long term recovery of an entire community | | Social capital / unity | | |
| | | | | |

| | | | | |
|---|---|---|---|---|
| Results indicate that the economic recovery of the environment-dependent fisheries sector lagged behind the recovery of the general economy. This is caused by several factors such as decreased demand for fisheries products due to perception of environmental damage. | | Economic Trends | katrina | de Ruiter, M. C. (2011). *POST-DISASTER COMMUNITY RECOVERY* (Doctoral dissertation, University of British Columbia). |
| | | | | |
| We find that black residents returned to the city at a much slower pace than white residents even after controlling for socioeconomic status and demographic characteristics. However, the racial disparity disappears after controlling for housing damage. We conclude that blacks tended to live in areas that experienced greater flooding and hence suffered more severe housing damage which, in turn, led to their delayed return to the city. | | Impact of Event / Inequality | katrina | Fussell, E., Sastry, N., & VanLandingham, M. (2010). Race, socioeconomic status, and return migration to New Orleans after Hurricane Katrina. *Population and environment*, *31*(1-3), 20-42. |

| | | | | |
|---|---|---|---|---|
| "Social networks, which are based on family, friendship, work, and place of residence, affect migration decisions by demonstrating the feasibility of a move, by providing information and resources that increase the expected benefits, and by reducing the costs and uncertainty associated with a move (Massey 1990; Stark 1991; Taylor 1986)." | | Choice / Risk Reduction | | |
| | | | | |
| Promoting social capital either before or after a disaster is about local people having a sense of community encasing an attitude of trust and belonging and establishing a network of assistance and information. After the disaster, it is about having locals engaged in every aspect of the recovery process and creating a sense of resource equity | | Inclusion in Choice | rural "Recovery ville" (anonymiz ed) | Onstad, P. A., Danes, S. M., Hardman, A. M., Olson, P. D., Marczak, M. S., Heins, R. K., ... & Coffee, K. A. (2012). The road to recovery from a natural disaster: voices from the community. *Community Developmen* |

| | | | |
|---|---|---|---|
| pre-existing organizational networks and relationships enhance information dissemination and communication facilitating rapid mobilization of emergency and ongoing support services. | | Social Capital / networks | *t*, *43*(5), 566-580. |
| social capital in the form of communitycohesion and closeness enhanced resilience and was an asset in the recovery process | | social capital / cohesion | |
| During the flood, herbicides and pesticides from the fertilizerplant spilled into the neighboring mobile home park. The Environmental ProtectionAgency declared all mobile homes contaminated and no one was allowed to enter their property for some time. Most agreed that decisions were made too slowly for mobile home residents. The mobile homeowners who made a decision to leave Recoveryville | | Choice / Impact of event | |

| | | | | |
|---|---|---|---|---|
| remarked, ‘‘We felt like we were treated like second class citizens. Theothers were going back into their homes, checking things out and starting to cleanand we were kept out of ours.’’ | | | | |
| A lot of time, it was the older people, the more vulnerable people who didn’t want to get anything from FEMA. They were so shocked that they wouldn’t even fill out a FEMA report. So they took less. They deserve more, but they took less. | | capacity to respond | | |
| | | | | |
| Citizen participation (or perception of participation) correlated with happier citizenry post-disaster, lower levels of political upheaval, and higher perceived levels of recovery | | inclusion | East / grand fork floods | Kweit, M. G., & Kweit, R. W. (2004). Citizen participation and citizen evaluation in disaster recovery. |

| | | | | |
|---|---|---|---|---|
| This research suggests that the belief that citizens had an effect on decisions and that the cities attempted to get citizens involved had a substantial effect on the overall evaluation of the success of recovery. | | inclusion | | *The American Review of Public Administration*, *34*(4), 354-373. |
| | | | | |
| "The more a family utilizes institutional aid sources, the more likely will that family recovery or reestablish housing equivalent to that lost in the disaster." | | Outside Capital Inflow | | Bolin, R. (1976). Family recovery from natural disaster: A preliminary model. *Mass Emergencies*, *1*(4), 267-277. |
| "The higher the socioeconomic status of a victim family, the more likely will that family reestablish housing equivalent to that lost in the disaster." | | Economic Status | | |
| "The more severe the impact of a disaster on a family, the less likely will that family rely solely on extended kin for recovery aid." | | Impact of Event / social capital | | |
| "The later a victim family is in the life cycle, the less likely will it utilize kin-based aid for recovery." | | Demographics (age) / access | | |

| | | | | |
|---|---|---|---|---|
| "Families that rely solely on aid from extended kin groups are less likely to reestablish housing equivalent to that lost in the disaster." | | Social Capital / Outside Capital Inflow | | |
| "The higher a victim family's socio-economic status, the more likely will that family recover in a perceptual and emotional sense from the disaster" | | Economic Status | | |
| "The more a victim family utilizes aid from extended kin, the more likely will that family recover from the disaster in a perceptual and emotional sense." | | Social Capital / Wellbeing? | | |
| | | | | |
| "The development and use of a strategic approach to recovery reflects the local governments ability to act. Several of the communities studied rate very low in the categories of techinical ability, resources, and organizational flexibility and adaptiveness." | | Capacity to Respond | multiple | Rubin, C. B., Saperstein, M. D., & Barbee, D. G. (1985). Community recovery from a major natural disaster. |
| contributions to effective recovery include personal leadership, ability | | Capacity to Respond | multiple | |

| | | | | |
|---|---|---|---|---|
| to act, and knowledge of what to do. | | | | |
| "This substantial voluntary evacuation supports the findings of other researchers who maintain people tend to respond to a disaster warning based on their last major disaster experience." | | Experience(?) | Corpus Christi hurricane 1980 (allen) | |
| Intergovernmental relations between local/state/federal authorities. Cooperation at all levels seem to be most condusive to recovery (aid discemination) | | Cooperation | | |
| | | | | |
| Other things being equal, we found that organizations that were smaller, weaker, and under significant stress before the event were much more likely not to reopen their doors after the event. | | Agent Trends | | Alesch, D. J., Holly, J. N., Mittler, E., & Nagy, R. (2001). *Organizations at risk: What happens when small businesses and not-for-* |

| | | | | |
|---|---|---|---|---|
| Even strong firms can suffer badly from natural hazard events. Being out of business for any extended period of time can lead to a loss of market share. Even with business interruption and property and casualty insurance, it can be extremely difficult toregain market share | | Down Time | | *profits encounter natural disasters* (p. 105). Fairfax, VA: Public Entity Risk Institute. |
| Firms that operate in leased space with inadequate lease provisions concerning who repairs earthquake damage and how quickly it will be done will find themselves in trouble. InNorthridge, many business owners found themselves stuck in buildings that were not repaired for a long time by virtue of a lease that kept them from moving to another location where they could resume business | | Ownership of Property | | |
| We had to conclude that traditional structural precautions are necessary to reduce losses to life and | | | | |

| | | | | |
|---|---|---|---|---|
| property,but not sufficient to help businesses survive. | | | | |
| Our research suggests that only the weakest firms fail right after the disaster. Most firms that ultimately fail do so only after a desperatestruggle to recover. We found, too, that Small Business Administration loans are not an adequate answer. | | | | |
| We found that business losses go far beyond initial damage to the structure, equipment, and inventory. They include business interruption, lost income to employers and employees, and lost assets in the form of business equity. | | Impact of Event | | |
| businesses with more than one location were more likely to survive than those with a single location | | Diversification | | |

| | | | | |
|---|---|---|---|---|
| Both found that firms that survived and were recovered or recovering a year or two after the earthquake were those that were larger, had fewer of their eggs in one basket (did business in more than one location and/or had customers in unaffected locations), and were financially stronger. | | Diversification | | |

## Appendix B: Code

The code appearing in this section is partitioned by module. The final code block is a “master” or “controlling” block used to assemble the pieces of the code together and run scenario simulations. Comments are indicated with a pound (#) symbol for single line comments or encased in triple quotations (“””comment”””) for multiline comments

### Capitals.py

```
# -*- coding: utf-8 -*-
"""
Module of classes that represent different types of capitals used by DESaster
entities.

DESaster capitals are basically fancy wrappers of SimPy resources, containers,
stores.

Classes:
HumanCapital(object)
FinancialCapital(object)
BuiltCapital(object)
Building(BuiltCapital)
```

```
Residence(Building)

Functions:
setHousingStock(simulation, stock_df)

@author: Derek Huling, Scott Miles
"""
from simpy import Resource, Container, FilterStore
from desaster.config import structural_damage_ratios
from desaster.config import acceleration_damage_ratios
from desaster.config import drift_damage_ratios
import random
class HumanCapital(object):
    """Define class for a collection of SimPy resources that represent different
types of
    human resources used by entities during recovery processes.
    """
    def __init__(self, simulation, human_capital_dict):
        """Initiate class based on current SimPy environment and human capital
        dictionary.

        Keyword Arguments:
        simulation -- Pointer to SimPy simulation environment.
        human_capital_dict -- Dictionary of all required human capital types
                              (as dict keys) with associated quantities
        """

        # Define a SimPy resource for each type of human capital.
        # Set initial quantity of each resource equal to the value specified
        # in the dictionary for the respective capital type.

        # Initial number of available inspectors
        self.inspectors = Resource(simulation, human_capital_dict['Inspectors'])
        # Initial number of available insurance claim adjusters
        self.insurance_adjusters = Resource(simulation,
                                   human_capital_dict['Insurance Adjusters'])
        # Initial number of available FEMA processors
        self.fema_processors = Resource(simulation,
                                       human_capital_dict['FEMA Processors'])
        # Initial number of available permit processors
        self.permit_processors = Resource(simulation,
                                       human_capital_dict['Permit Processors'])
        # Initial number of available contractors
        self.contractors = Resource(simulation,
                                   human_capital_dict['Contractors'])
        # Initial number of available loan processors
        self.loan_processors = Resource(simulation,
                                       human_capital_dict['Loan Processors'])
        # Initial number of available engineers
        self.engineers = Resource(simulation, human_capital_dict['Engineers'])

class FinancialCapital(object):
    """Define class for a collection of SimPy containers that represent different
types of
    financial resources used by entities during recovery processes.
    """
    def __init__(self, simulation, financial_capital_dict):
        """Initiate class based on current SimPy environment and financial
        capital dictionary.

        Keyword Arguments:
```

```
        simulation -- Pointer to SimPy simulation environment.
        financial_capital_dict -- Dictionary of all required financial capital
                                types (as dict keys) with associated quantities
        """
        # Initial $ amount of overall FEMA aid available to the
        # recovering area.
        self.fema_aid = Container(simulation,
                                    init=financial_capital_dict['FEMA Aid'])
        # Initial $ amount of overall construction resources available to
        # the recovering area.
        self.building_materials = Container(simulation,
                                    init=financial_capital_dict['Building
Materials'])

class BuiltCapital(object):
    """Define top-level class for representing the attributes and methods
    of types of built capital.

    """
    def __init__(self, simulation, asset):
        """Run initial methods for defining built capital attributes.

        Keyword Arguments:
        simulation -- Pointer to SimPy simulation environment.
        asset -- A dataframe row with required built capital attributes.
        """
        self.setYearBuilt(asset)
        self.setValue(asset)
        self.setDamageState(asset)
        self.setInspection(asset)
    def setYearBuilt(self, asset):
        try:
            self.age = asset['Year Built']  # Year asset was built
        except KeyError as e:
            self.age = random.randint(1900,2000)
    def setValue(self, asset):
        self.value = asset['Value']  # Value of the asset in $
    def setDamageState(self, asset):
        self.damage_state = asset['Damage State']  # HAZUS damage state
    def setInspection(self, asset):
        self.inspected = False  # Whether the asset has been inspected

class Building(BuiltCapital):
    """Define class that inherits from BuiltCapital() for representing the
    attributes and methods of types of buildings.
    """
    def __init__(self, simulation, building):
        """Run initial methods for defining building attributes.

        Keyword Arguments:
        simulation -- Pointer to SimPy simulation environment.
        building -- A dataframe row with required building attributes.
        """
        #since we're overriding the base class init, we need to call it
        #to maintain its attributes, unless we're explicitely changing
        #the structure
        BuiltCapital.__init__(self, simulation, building)

        self.setAddress(building)
        self.setOccupancy(building)
        self.setDamageValue(building)
```

```
        self.setCoordinates(building)
        self.setBuildingArea(building)

    def setAddress(self, building):
        try: #if address isn't in dataframe, we'll just set it to none
            self.address = building['Address']  # Address of building
        except KeyError as e:
            self.address = None

    def setCoordinates(self, building):
        try: #if lat/long aren't in data, we'll set to none
            self.latitude = building['Latitude']
            self.longitude = building['Longitude']
        except KeyError as e:
            self.latitude = None
            self.longitude = None


    def setOccupancy(self, building):
        self.occupancy = building['Occupancy']  # Occupancy type of building
    def setBuildingArea(self, building):
        self.area = building['Area']  # Floor area of building
    def setDamageValue(self, building):
        """Calculate damage value for building based on occupancy type and
        HAZUS damage state.

        Function uses three lookup tables (Table 15.2, 15.3, 15.4) from the HAZUS-
MH earthquake model
        technical manual for structural damage, acceleration related damage,
        and for drift related damage, respectively. Estimated damage value for
        each type of damage is summed for total damage value.
        http://www.fema.gov/media-library/assets/documents/24609

        Keyword Arguments:
        structural_damage_ratios -- dataframe set in config.py
        acceleration_damage_ratios -- dataframe set in config.py
        drift_damage_ratios -- dataframe set in config.py
        """
        struct_repair_ratio =
structural_damage_ratios.ix[building['Occupancy']][building['Damage State']] /
100.0
        accel_repair_ratio =
acceleration_damage_ratios.ix[building['Occupancy']][building['Damage State']] /
100.0
        drift_repair_ratio =
drift_damage_ratios.ix[building['Occupancy']][building['Damage State']] / 100.0
        self.damage_value = building['Value']*(struct_repair_ratio +
                                               accel_repair_ratio +
                                               drift_repair_ratio)

class Residence(Building):
    """Define class that inherits from Building() for representing the
    attributes and methods of types of residences.
    """
    def __init__(self, simulation, residence):
        """Run initial methods for defining residence attributes.

        Keyword Arguments:
        simulation -- Pointer to SimPy simulation environment.
        residence -- A dataframe row with required residence attributes.
        """
```

```
        Building.__init__(self, simulation, residence)

        self.setOccupancy(residence) #overriding base method for verification
        self.setBedrooms(residence)
        self.setBathrooms(residence)
        self.id = residence["ID Number"]
    def setOccupancy(self, residence):
        # Verify that residence dataframe has expected occupancy types
        if residence['Occupancy'] in ('Single Family Dwelling',
                            'Multi Family Dwelling', 'Mobile Home', 'Condo'):
            self.occupancy = residence['Occupancy']
        else:
            raise AttributeError(residence['Occupancy'])

    def setBedrooms(self, residence):
        try:
            self.bedrooms = residence['Bedrooms']  # Number of bedrooms in
residence
        except KeyError as e:
            self.bedrooms = random.randint(2, 5)

    def setBathrooms(self, residence):
        try:
            self.bathrooms = residence['Bathrooms']  # Number of bathrooms in
residence
        except KeyError as e:
            self.bathrooms = random.randint(1, self.bedrooms) #won't have more
bath than BRs

def importHousingStock(simulation, stock_df):
    """Define, populate and return a SimPy FilterStore with Residence() objects to
    represent a vacant housing stock.

    Keyword Arguments:
    simulation -- Pointer to SimPy simulation environment.
    stock_df -- Dataframe with required attributes for each vacant home in
                the stock.
    """
    stock_fs = FilterStore(simulation)

    for i in stock_df.index:
        stock_fs.put(Residence(simulation, stock_df.loc[i]))

    return stock_fs

def reloadBuildingMaterial(simulation, building_material, amount=2000000):
    yield building_material.put(amount)
```

---------------------------------------------Config.py-------------------------------------------------

```
# -*- coding: utf-8 -*-
"""
Module for defining variables for a suite of DESaster paramaters.

@author: Derek Huling, Scott Miles
"""

#configs

import random
```

```
import pandas as pd

# Excel workbook with lookup tables from HAZUS-MH earthquake model technical
# manual. (http://www.fema.gov/media-library/assets/documents/24609)
hazus_parameters_file = "../inputs/hazus_parameters.xlsx"

random.seed(69)

# Parameters for defining a normal distribution for representing the duration
# required to inspect structures from the time of a hazard event.
inspection_mean = 1.0
inspection_std = 0
inspection_time = abs(random.gauss(inspection_mean, inspection_std))

# Parameters for defining a normal distribution for representing the duration
# required to process an insurance claim from time claim is submitted.
adjuster_mean = 15.0
adjuster_std = 0.0
adjuster_time = abs(random.gauss(adjuster_mean, adjuster_std))

# Parameters for defining a normal distribution for representing the duration
# required to process an FEMA aid request from time request is submitted.
fema_process_mean = 20.0
fema_process_std = 0.0
fema_process_time = abs(random.gauss(fema_process_mean, fema_process_std))

# Parameters for defining a normal distribution for representing the duration
# required to conduct engineering assessment from time assessment is requested.
engineering_mean = 25.0
engineering_std = 0.0
engineering_assessment_time = abs(random.gauss(engineering_mean, engineering_std))

# Parameters for defining a normal distribution for representing the duration
# required to process a loan application from time application is submitted.
loan_process_mean = 30.0
loan_process_std = 0.0
loan_process_time = abs(random.gauss(loan_process_mean, loan_process_std))

# Parameters for defining a normal distribution for representing the duration
# required to process building permit request from time permit is requested.
permit_process_mean = 35.0
permit_process_std = 0.0
permit_process_time = abs(random.gauss(permit_process_mean, permit_process_std))

# % of damage value related to building materials (vs. labor and profit)
materials_cost_pct = 1.0

# Building repair time lookup table from HAZUS-MH earthquake model technical
# manual Table 15.9 (http://www.fema.gov/media-library/assets/documents/24609)
building_repair_times = pd.read_excel(hazus_parameters_file,
                            sheetname='Repair times',
                            index_col='Occupancy')

# Structural damage value ratio lookup table from HAZUS-MH earthquake model technical
# manual Table 15.2 (http://www.fema.gov/media-library/assets/documents/24609)
structural_damage_ratios = pd.read_excel(hazus_parameters_file,
                                sheetname='Struct. Repair Cost % of value',
                                index_col='Occupancy')

# Acceleration damage value ratio lookup table from HAZUS-MH earthquake model technical
# manual Table 15.3 (http://www.fema.gov/media-library/assets/documents/24609)
acceleration_damage_ratios = pd.read_excel(hazus_parameters_file,
                            sheetname='Accel non-struc repair cost',
                            index_col='Occupancy')

# Drift damage value ratio lookup table from HAZUS-MH earthquake model technical
# manual Table 15.4 (http://www.fema.gov/media-library/assets/documents/24609)
drift_damage_ratios = pd.read_excel(hazus_parameters_file,
```

```
                        sheetname='Deflect non-struc repair cost',
                        index_col='Occupancy')
```

--------------------------------------------entities.py--------------------------------------------

```
# -*- coding: utf-8 -*-
"""
Module of classes for implementing DESaster entities, such as households and
businesses.

Classes:
Household(object)

@author: Derek Huling, Scott Miles
"""
# Import Residence() class in order to assign households a residence.
from desaster.capitals import Residence

class Household(object):
    """Define a Household() class to represent a group of persons that reside
    together as a single analysis unit with attributes and methods.
    """
    def __init__(self, simulation, household_df, write_story = False):
        """Define household inputs and outputs attributes.
        Initiate household's story list string.

        simulation -- Pointer to SimPy simulation environment.
        household_df -- Dataframe row w/ household input attributes.
        write_story -- Boolean indicating whether to track a households story.
        """
         # Household simulation inputs
        self.household = household_df  # Dataframe w/ household input attributes
        self.name = household_df['Name']   # Name associated with household
        self.savings = household_df['Savings']  # Amount of household savings in $
        self.insurance = household_df['Insurance']  # Hazard-specific insurance coverage in
$
        self.residence = Residence(simulation, household_df)  # Pointer to household's
Residence() object

        # Household simulation outputs
        self.story = []  # The story of events for each household
        self.inspection_put = 0.0  # Time put request in for house inspection
        self.inspection_get = 0.0  # Time get  house inspection
        self.claim_put = 0.0  # Time put request in for insurance settlement
        self.claim_get = 0.0  # Time get insurance claim settled
        self.claim_payout = 0.0  # Amount of insurance claim payout
        self.assistance_put = 0.0  # Time put request in for FEMA assistance
        self.assistance_get = 0.0  # Time get FEMA assistance
        self.assistance_request = 0.0  # Amount of money requested from FEMA
        self.assistance_payout = 0.0  # Amount of assistance provided by FEMA
        self.money_to_rebuild = self.savings  # Total funds available to household to
rebuild house
        self.home_put = 0.0  # Time put request in for house rebuild
        self.home_get = 0.0  # Time get house rebuild completed
        self.loan_put = 0.0  # Time put request for loan
        self.loan_get = 0.0  # Time get requested loan
        self.loan_amount = 0.0  # Amount of loan received
        self.permit_put = 0.0  # Time put request for building permit
        self.permit_get = 0.0  # Time get requested building permit
        self.home_search_start = 0.0  # Time started searching for a new home
        self.home_search_stop = 0.0  # Time found a new home
        self.money_search_start = 0.0  # Time that household started search for money
        self.money_search_stop = 0.0  # Time that household found rebuild money
        self.gave_up_money_search = False  # Whether household gave up search for money
        self.gave_up_home_search = False  # Whether household gave up search for home

        # Initial method calls
```

```
        self.setStory(write_story)  # Start stories with non-disaster attributes

    def setStory(self, write_story):
        """Initiate the household's story based on input attributes.

        Keyword Arguments:
        write_story -- Boolean indicating whether to track a households story.
        """
        if write_story == True:
            # Set story with non-disaster attributes.
            self.story.append(
            '{0} lives in a {1} bedroom {2} at {3} worth ${4:,.0f}. '.format(self.name,
                                                                self.residence.bedrooms,
                                                                self.residence.occupancy,
                                                                self.residence.address,
                                                                self.residence.value
                                                                )
            )

    def story_to_text(self):
        """Join list of story strings into a single story string."""
        return ''.join(self.story)

def importHouseholds(simulation, households_df, write_story = False):
    """Return list of entities.Household() objects from dataframe containing
    data describing households.

    Keyword Arguments:
    simulation -- Pointer to SimPy simulation environment.
    household_df -- Dataframe row w/ household input attributes.
    write_story -- Boolean indicating whether to track a households story.
    """

    households = []

    # Population the simulation with households from the households dataframe
    for i in households_df.index:
        households.append(Household(simulation, households_df.iloc[i], write_story))

    return households
```

---------------------------------------------rebuild.py---------------------------------------------

```
# -*- coding: utf-8 -*-
"""
Module of functions for rebuilding/repairing individual homes and entire
building stocks. Eventually functions for non-residential buildings can be added.

Functions:
home(simulation, human_capital, financial_capital, household, write_story = True,
callbacks = None)
stock(simulation, structure_stock, fix_probability, human_capital)

@author: Scott Miles
"""
from desaster.config import building_repair_times, materials_cost_pct
from simpy import Interrupt
import random

def home(simulation, human_capital, financial_capital, household, write_story = True,
callbacks = None):
    """A process to rebuild a household's residence based on available contractors and
    building materials.

    Keyword Arguments:
    household -- A single entities.Household() object.
    human_capital -- A capitals.HumanCapital() object.
```

```
    financial_capital -- A capitals.FinancialCapital() object.
    write_story -- Boolean indicating whether to track a households story.

    Returns or Attribute Changes:
    household.story -- Process outcomes appended to story.
    household.home_put -- Record time money search starts
    household.home_get -- Record time money search stops
    household.residence.damage_state -- Set to 'None' if successful.
    household.residence.damage_value = Set to $0.0 if successful.
    """
    # Use exception handling in case process is interrupted by another process.
    try:

        # If household has enough money & there is enough available construction
        # materials in the region, then rebuild.
        if (household.money_to_rebuild >= household.residence.damage_value and
        household.residence.damage_value <= financial_capital.building_materials.level):
            # Record time put in request for home rebuild.
            household.home_put = simulation.now

            # Put in request for contractors to repair home.
            contractors_request = human_capital.contractors.request()
            yield contractors_request

            # Get the rebuild time for the household from config.py
            # which imports the HAZUS repair time look up table.
            # Rebuild time is based on occupancy type and damage state.
            rebuild_time =
building_repair_times.ix[household.residence.occupancy][household.residence.damage_state]

            # Obtain necessary construction materials from regional inventory.
            # materials_cost_pct is % of damage value related to building materials
            # (vs. labor and profit)
            yield financial_capital.building_materials.get(household.residence.damage_value
* materials_cost_pct)

            # Yield timeout equivalent to rebuild time.
            yield simulation.timeout(rebuild_time)

            # Release contractors.
            human_capital.contractors.release(contractors_request)

            # After successful rebuild, set damage to None & $0.
            household.residence.damage_state = 'None'
            household.residence.damage_value = 0.0

            # Record time when household gets home.
            household.home_get = simulation.now

            # If True, write outcome of successful rebuild to story.
            if write_story == True:
                household.story.append(
                    '{0}\'s home was repaired {1:,.0f} days after the event, taking {2:.0f}
days to repair. '.format(
                        household.name,
                        household.home_get,
                        household.home_get - household.home_put
                    )
                )

        # Deal with case that insufficient construction materials are available.
        if household.residence.damage_value > financial_capital.building_materials.level:
            # If true, write outcome of the process to their story
            if write_story == True:
                household.story.append(
                'There were insufficient construction materials available in the area for
{0} to rebuild. '
                .format(household.name)
```

```
                )

            return

        # Deal with case that household does not have enough money to rebuild.
        if household.money_to_rebuild < household.residence.damage_value:
            # If true, write outcome of the process to their story
            if write_story == True:
                household.story.append(
                    '{0} was unable to get enough money to rebuild. '.format(
                    household.name))

            return

    # Handle any interrupt thrown by another process
    except Interrupt as i:
        # If true, write outcome of the process to their story
        if write_story == True:
            household.story.append(
                    '{0} gave up {1:.0f} days into the home rebuilding process. '.format(
                    household.name, i.cause))

    if callbacks is not None:
        yield simulation.process(callbacks)

    else:
        pass

def stock(simulation, structure_stock, fix_probability):
    """Process to rebuild a part or an entire building stock (FilterStore) based
    on available contractors and specified proportion/probability.

    Keyword Arguments:
    structure_stock -- A SimPy FilterStore that contains one or more
        capitals.BuiltCapital(), capitals.Building(), or capitals.Residence()
        objects that represent vacant structures for purchase.
    fix_probability -- A value to set approximate percentage of number of structures
        in the stock to rebuild.

    Attribute Changes:
    put_structure.damage_state -- Changed to 'None' for selected structures.
    put_structure.damage_value = Changed to $0.0 for selected structures.
    """
    random.seed(15)

    structures_list = []  # Empty list to temporarily place FilterStore objects.

    # Remove all structures from the FilterStore; put in a list for processing.
    while len(structure_stock.items) > 0:
        get_structure = yield structure_stock.get(lambda getStructure:
                                                  getStructure.value >= 0.0
                                          )
        structures_list.append(get_structure)

    num_fixed = 0  # Counter
    # Iterate through structures, do processing, put back into the FilterStore
    for put_structure in structures_list:
        # Select inspected structures that have Moderate or Complete damage
        if (put_structure.inspected == True
        and (put_structure.damage_state == 'Moderate'
        or put_structure.damage_state == 'Complete')
        ):
            # Compare uniform random to prob to estimate percentage to fix.
            # Then set damage to None and $0. Put back in FilterStore.
            if random.uniform(0, 1.0) <= fix_probability:
                put_structure.damage_state = 'None'
                put_structure.damage_value = 0.0
                structure_stock.put(put_structure)
```

```
                num_fixed += 1
            else:
                # Put back in FilterStore if chosen not to be fixed.
                structure_stock.put(put_structure)

        else:
            # Put all other structures back in FilterStore.
            structure_stock.put(put_structure)

    print('{0} homes in the vacant building stock were fixed on day
{1:,.0f}.'.format(num_fixed, simulation.now))
```

---------------------------------------------request.py-----------------------------------------------

```
# -*- coding: utf-8 -*-
"""
Module of various recovery processes that make complex requests to
resources.

Functions:
inspection(simulation, human_capital, structure, entity = None,
            write_story = False, callbacks = None)
insurance_claim(simulation, human_capital, entity, write_story = False,
                callbacks = None)
fema_assistance(simulation, human_capital, financial_capital, entity,
                write_story = False, callbacks = None)
engineering_assessment(simulation, human_capital, entity, write_story = False,
                        callbacks = None)
loan(simulation, human_capital, entity, write_story = False, callbacks = None)
permit(simulation, human_capital, entity, write_story = False,
            callbacks = None)

@author: Derek Huling, Scott Miles
"""
from simpy import Interrupt
from desaster.config import inspection_time, adjuster_time, fema_process_time
from desaster.config import engineering_assessment_time, loan_process_time
from desaster.config import permit_process_time

def inspection(simulation, human_capital, structure, entity = None,
    write_story = False, callbacks = None):
    """Define process for inspecting an entity's structure.

    Keyword Arguments:
    entity -- An entity object from the entity.py module, for example
                entities.Household(). Defaults to none so that can inspect
                structures that aren't associated with an entity.
                *** Currently, this function expects that the entity *is*
                an entities.Household() object because makes an assignment to
                Household().residence.inspected***
    simulation -- A simpy.Environment() object.
    callbacks -- a generator function containing processes to start after the
                    completion of this process.
    human_capital -- A capitals.HumanCapital() object.
    structure -- A capitals.BuiltCapital() object or an object of a BuiltCapital()
                sub-class, such as capitals.Residence()
    write_story -- Boolean indicating whether to track a households story.

    Returns or Attribute Changes:
    entity.inspection_put -- Record time of inspection request
    entity.inspection_get -- Record time of inspection completion
    entity.residence.inspected -- Whether inspection was successful.
    entity.story -- Summary of process outcome as string.
    """

    # Only record inspection request time if structure associated with an entity.
```

```
    if entity != None:
        # Put in request for an inspector (shared resource)
        entity.inspection_put = simulation.now

    # Request inspectors
    inspectors_request = human_capital.inspectors.request()
    yield inspectors_request

    # Yield timeout equivalent to time from hazard event to end of inspection.
    yield simulation.timeout(inspection_time)

    # Set attribute of structure to indicate its been inspected.
    structure.inspected = True

    # Release inspectors now that inspection is complete.
    human_capital.inspectors.release(inspectors_request)

    # Only record inspection time and write story if structure associated with
    # an entity.
    if entity != None:
        entity.inspection_get = simulation.now

        #If true, write process outcome to story
        if write_story == True:

            entity.story.append(
                                "{1}'s house was inspected {0:.0f} days after the event and
suffered ${2:,.0f} of damage.".format(entity.inspection_get, entity.name,
entity.residence.damage_value))

    if callbacks is not None:
        yield simulation.process(callbacks)
    else:
        pass

def insurance_claim(simulation, human_capital, entity, write_story = False,
                    callbacks = None):
    """Define process for entity to submit an insurance claim.

    Keyword arguments:
    entity -- An entity object from the entity.py module, for example
              entities.Household().
    simulation -- A simpy.Environment() object.
    human_capital -- A capitals.HumanCapital() object.
    write_story -- Boolean indicating whether to track a households story.
    callbacks -- a generator function containing processes to start after the
                    completion of this process.

    Returns or attribute changes:
    entity.claim_put -- Record current simulation time at the time the entity
                        enters the adjuster queue
    entity.claim_payout -- Set claim payout equal to damage value amount.
    entity.claim_get -- Record simulation time when entity recieves payout
    entity.story -- Append natural language sentences to entities story.
    """
    # Exception handling in case interrupted by another process.
    try:
        # Ensure entity has insurance.
        if entity.insurance <= 0.0:
            return
        # Has insurance so submits a claim.
        else:
            # Record time that claim request is put.
            entity.claim_put = simulation.now

            #If true, write claim submission time to story.
            if write_story == True:
                entity.story.append(
```

```
                    '{0} submitted an insurance claim {1:.0f} days after the event.
'.format(
                        entity.name, entity.claim_put)
                    )

            # Submit request for insurance adjusters.
            request = human_capital.insurance_adjusters.request()
            yield request

            # Timeout process to simulate claims processing duration.
            yield simulation.timeout(adjuster_time)

            # Determine payout amount and add to entity's rebuild money.
            # Only payout amount equal to the damage, not the full coverage.
            if entity.residence.damage_value < entity.insurance:
                entity.claim_payout = entity.residence.damage_value
            else:
                entity.claim_payout = entity.insurance
            entity.money_to_rebuild += entity.claim_payout

            # Record when the time when household gets claim payout
            entity.claim_get = simulation.now

            # Release insurance adjusters so they can process other claims.
            human_capital.insurance_adjusters.release(request)

            #If true, write process outcome to story.
            if write_story == True:
                entity.story.append(
                    '{0} received a ${1:,.0f} insurance payout {2:.0f} days after the event.
'.format(
                        entity.name,
                        entity.claim_payout,
                        entity.claim_get
                        )
                    )
    # Handle any interrupt thrown by another process.
    except Interrupt as i:
        #If true, write that the process was interrupted to their story.
        if write_story == True:
            entity.story.append(
                    '{0} gave up during the insurance claim process after a {1} day search
for money. '.format(
                    entity.name, i.cause))

    if callbacks is not None:
        yield simulation.process(callbacks)
    else:
        pass

def fema_assistance(simulation, human_capital, financial_capital, entity,
                    write_story = False, callbacks = None):
    """Define process for entity to submit request for FEMA individual assistance.

    entity -- An entity object from the entity.py module, for example
                entities.Household().
    simulation -- A simpy.Environment() object.
    human_capital -- A capitals.HumanCapital() object.
    financial_capital -- A capitals.FinancialCapital() object.
    write_story -- Boolean indicating whether to track a households story.
    callbacks -- a generator function containing processes to start after the
                    completion of this process.

    Returns or Attribute Changes:
    entity.assistance_put -- Records sim time of fema processor request
    entity.assistance_get -- Records sim time of fema assistance reciept
    entity.assistance_request -- The amount of assistance requested.
    entity.assistance_payout -- Amount of FEMA aid given to the entity.
```

```
    """
    # Exception handling in case interrupted by another process.
    try:
        #Ensure that entity does not have enough money to rebuild already.
        if entity.money_to_rebuild >= entity.residence.damage_value:
            return
        # If does not have enough money to rebuild, submit request to FEMA.
        else:
            # Record time requests FEMA assistance.
            entity.assistance_put = simulation.now
            #If true, write FEMA request time to story.
            if write_story == True:
                entity.story.append(
                    '{0} submitted a request to FEMA {1:.0f} days after the event. '.format(
                        entity.name, entity.assistance_put
                        )
                    )
            # Request a FEMA processor to review aid application.
            request = human_capital.fema_processors.request()
            yield request

            # Yield timeout for duration necessary to process FEMA aid request.
            yield simulation.timeout(fema_process_time)

            # Release FEMA processors.
            human_capital.fema_processors.release(request)

            # Record time received FEMA assistance.
            entity.assistance_get = simulation.now

            # Must subtract any insurance payout from FEMA payout.
            entity.assistance_request = (entity.residence.damage_value
                                        - entity.claim_payout)

            # If requesting assistance, determine if FEMA has money left to
            # provide assistance.
            if entity.assistance_request <= financial_capital.fema_aid.level:
                # FEMA has enough money to fully pay requested amount.
                entity.assistance_payout = entity.assistance_request
                entity.money_to_rebuild += entity.assistance_payout

                # Subtract payout amount from the overall amount of assistance
                # FEMA has available to payout to all requests.
                yield financial_capital.fema_aid.get(entity.assistance_request)

                #If true, write process outcome to story.
                if write_story == True:
                    entity.story.append(
                        '{0} received ${1:,.0f} from FEMA {2:.0f} days after the event.
'.format(
                            entity.name,
                            entity.assistance_payout,
                            entity.assistance_get
                            )
                        )
            elif financial_capital.fema_aid.level > 0:
                # FEMA has money left but less than requested.
                # Set payout equal to remaining funds.
                entity.assistance_payout = financial_capital.fema_aid.level
                entity.money_to_rebuild += entity.assistance_payout

                # Subtract payout amount from the overall amount of assistance
                # FEMA has available to payout to all requests.
                yield financial_capital.fema_aid.get(financial_capital.fema_aid.level)

                #If true, write process outcome to story.
                if write_story == True:
                    entity.story.append(
```

```
                     '{0} requested ${1:,.0f} from FEMA but only received ${2:,.0f}, {3}
days after the event.. '
                     .format(
                                entity.name,
                                entity.assistance_request,
                                entity.assistance_payout,
                                entity.assistance_get
                            )
                        )
            else:
                # FEMA has no money left to make payout.
                entity.assistance_payout = 0.0

                #If true, write process outcome to story.
                if write_story == True:
                    entity.story.append(
                    '{0} received no money from FEMA because of inadequate funding. '
                    .format(entity.name)
                    )

    # Catch any interrupt from another process.
    except Interrupt as i:
        #If true, write process outcome to story.
        if write_story == True:
            entity.story.append(
                    '{0} gave up during the FEMA assistance process after a {1} day search
for money. '.format(
                        entity.name, i.cause)
                    )

    if callbacks is not None:
        yield simulation.process(callbacks)
    else:
        pass

def engineering_assessment(simulation, human_capital, entity, write_story = False,
                            callbacks = None):
    """Define process for entity to request an engineering assessment of their
    structure.

    Keyword Arguments:
    entity -- An entity object from the entity.py module, for example
                entities.Household().
    simulation -- A simpy.Environment() object.
    human_capital -- A capitals.HumanCapital() object.
    write_story -- Boolean indicating whether to track a households story.
    callbacks -- a generator function containing processes to start after the
                    completion of this process.

    Returns or Attribute Changes:
    entity.assessment_put -- Records sim time of assessment request
    entity.assistance_get -- Records sim time of assessment reciept
    """

    # Record time that assessment request put in.
    entity.assessment_put = simulation.now

    # Request an engineer.
    request = human_capital.engineers.request()
    yield request

    # Yield process timeout for duration necessary to assess entity's structure.
    yield simulation.timeout(engineering_assessment_time)

    # Release engineer so it can assess other structures.
    human_capital.engineers.release(request)

    # Record time when assessment complete.
```

```
    entity.assessment_get = simulation.now

    # If true, write the outcome of the process to story.
    if write_story == True:
        entity.story.append(
        '{0} received an engineering assessment {1:.0f} days after the event. '
        .format(entity.name, entity.assessment_get)
        )

    if callbacks is not None:
        yield simulation.process(callbacks)
    else:
        pass

def loan(simulation, human_capital, entity, write_story = False, callbacks = None):
    """Define process for entity to submit request for loan (e.g., from SBA).

    entity -- An entity object from the entity.py module, for example
                entities.Household().
    simulation -- A simpy.Environment() object.
    human_capital -- A capitals.HumanCapital() object.
    write_story -- Boolean indicating whether to track a households story.
    callbacks -- a generator function containing processes to start after the
                    completion of this process.

    Returns or Attribute Changes:
    entity.loan_put -- Records sim time of loan request
    entity.loan_get -- Records sim time of loan reciept
    entity.loan_amount -- The amount of loan requested.
    """
    # Exception handling in case interrupted by another process.
    try:
        # Ensure entity does not have enough money to rebuild.
        if entity.money_to_rebuild >= entity.residence.damage_value:
            return
        else:
            # Does not have enough money to rebuild.
            # Record time application submitted.
            entity.loan_put = simulation.now

            # If true, write loan request time to story.
            if write_story == True:

                entity.story.append(
                    '{0} submitted a loan application {1:.0f} days after the event.
'.format(
                        entity.name, entity.loan_put)
                    )

            # Request a loan processor.
            request = human_capital.loan_processors.request()
            yield request

            # Yield process timeout for duration needed to process loan request.
            yield simulation.timeout(loan_process_time)

            # Release loan processor so that they can process other loans.
            human_capital.loan_processors.release(request)

            # Record time loan is given.
            entity.loan_get = simulation.now

            # Subtract any insurance or FEMA payouts from damage value to
            # arrive at loan amount.
            entity.loan_amount = (
                                    entity.residence.damage_value
                                    - entity.claim_payout
                                    - entity.assistance_payout
```

```
                        )

            # Add loan amount to entity's money to rebuild.
            if entity.loan_amount > 0.0:
                entity.money_to_rebuild += entity.loan_amount

                #If true, write process outcome to story.
                if write_story == True:

                    entity.story.append(
                    "{0} received a loan for ${1:,.0f} {2:.0f} days after the event. "
                    .format(entity.name, entity.loan_amount, entity.loan_get))

    # Handle any interrupt from another process.
    except Interrupt as i:
        #If true, write interrupt outcome to story.
        if write_story == True:
            entity.story.append(
                    '{0} gave up during the loan approval process after a {1} day search for
money. '.format(
                    entity.name, i.cause))

    if callbacks is not None:
        yield simulation.process(callbacks)
    else:
        pass

def permit(simulation, human_capital, entity, write_story = False, callbacks = None):
    """Define process for entity to request an engineering assessment of their
    structure.

    Keyword Arguments:
    entity -- An entity object from the entity.py module, for example
                entities.Household().
    simulation -- A simpy.Environment() object.
    human_capital -- A capitals.HumanCapital() object.
    write_story -- Boolean indicating whether to track a households story.
    callbacks -- a generator function containing processes to start after the
                    completion of this process.

    Returns or Attribute Changes:
    entity.permit_put -- Records sim time of permit request
    entity.permit_get -- Records sim time of permit reciept
    """
    # Record time permit application submitted.
    entity.permit_put = simulation.now

    # Request permit processor / building official.
    request = human_capital.permit_processors.request()
    yield request

    # Yield process timeout equal to duration required to review permit request.
    yield simulation.timeout(permit_process_time)

    # Release permit process to allow them to review other requests.
    human_capital.permit_processors.release(request)

    # Record time that permit is granted.
    entity.permit_get = simulation.now

    #If true, write outcome of process to story.
    if write_story == True:
        entity.story.append(
        "{0} received permit approval {1:.0f} days after the event. "
        .format(entity.name, entity.permit_get)
        )

    if callbacks is not None:
```

```
        yield simulation.process(callbacks)
    else:
        pass
```

-----------------------------------------------search.py-----------------------------------------------

```
# -*- coding: utf-8 -*-
"""
Module of functions that implement complex searches for resources by simulated
entities.

Functions:
permanent_housing(simulation, entity, search_patience, housing_stock,
                    human_capital, write_story = False)

rebuild_money(simulation, human_capital, financial_capital, entity,
                    search_patience, write_story = False):

@author: Scott Miles
"""
from desaster import request

def permanent_housing(simulation, household, search_patience, housing_stock,
                        human_capital, write_story = False):
    """A process (generator) representing household search for permanent housing
    based on housing preferences, available housing stock, and patience finding
    a new home.

    Keyword Arguments:
    simulation -- Pointer to SimPy simulation environment.
    household -- A single entities.Household() object.
    search_patience -- The search duration in which the household is willing to wait
                        to find a new home. Does not include the process of
                        securing money.
    housing_stock -- A SimPy FilterStore that contains one or more
                    capitals.Residence() objects that represent vacant homes for
                    purchase.
    human_capital -- A capitals.HumanCapital() object.
    write_story -- Boolean indicating whether to track a households story.

    Returns or Attribute Changes:
    household.story -- Process outcomes appended to story.
    household.home_search_start -- Record time home search starts
    household.home_search_stop -- Record time home search stops
    household.residence -- Potentially assigned a new capitals.Residence() object.
    household.gave_up_home_search -- Set to True if search patience runs out.
    """
    # Record when housing search starts
    # Calculate the time that housing search patience ends
    # If write_story == True, write search start time to household's story
    household.home_search_start = simulation.now
    patience_end = household.home_search_start + search_patience
    if write_story == True:
        household.story.append(
            '{0} started searching for a {1} with a value under ${2:,.0f} {3:,.0f} days
after the event. '.format(
            household.name, household.residence.occupancy,
            household.residence.value, household.home_search_start)
            )

    # Define timeout process representing household's *remaining* search patience.
    # Return 'Gave up' if timeout process completes.
    find_search_patience = simulation.timeout(patience_end - simulation.now,
        value='Gave up')

    # Define a FilterStore get process to find a new home from the vacant
```

```
    # housing stock with similar attributes as current home.
    new_residence = housing_stock.get(lambda getResidence:
                    (
                        getResidence.damage_state == 'None'
                        or getResidence.damage_state == 'Slight'
                    )
                    and getResidence.occupancy == household.residence.occupancy
                    and getResidence.value < household.residence.value
                    and getResidence.inspected == True
                   )

    # Yield both the patience timeout and the housing stock FilterStore get.
    # Wait until one or the other process is completed.
    # Assign the process that is completed first to the variable.
    home_search_outcome = yield find_search_patience | new_residence

    # Exit the function if the patience timeout completes before a suitable
    # home is found in the housing stock.
    if home_search_outcome == {find_search_patience: 'Gave up'}:

        household.gave_up_home_search = True

        # If write_story == True, note in the story that the household gave up
        # the search.
        if write_story == True:
            household.story.append(
                'On day {0:,.0f}, after a {1:,.0f} day search, {2} gave up looking for a new
home in the local area. '.format(
                    simulation.now,
                    simulation.now - household.home_search_start,
                    household.name
                    )
                )

        return

    # If a new home is found before patience runs out place household's current
    # residence in vacant housing stock -- "sell" the house.
    yield housing_stock.put(household.residence)

    # Set the newly found residence as the household's residence.
    household.residence = home_search_outcome[new_residence]

    # Record the time that the housing search ends.
    household.home_search_stop = simulation.now

    # If write_story is True, then write results of successful home search to
    # household's story.
    if write_story == True:
        household.story.append(
            'On day {0:,.0f}, {1} received a {2} at {3} with a value of ${4:,.0f} and
${5:,.0f} of damage. '.format(
                household.home_search_stop,
                household.name, household.residence.occupancy,
                household.residence.address,
                household.residence.value,
                household.residence.damage_value
                )
            )

def rebuild_money(simulation, human_capital, financial_capital, entity,
                    search_patience, write_story = False):
    """A process (generator) representing entity search for money to rebuild or
    repair home based on requests for insurance and/or FEMA aid and/or loan.

    simulation -- Pointer to SimPy simulation environment.
    entity -- A single entities object, such as Household().
    search_patience -- The search duration in which the household is willing to
```

```
                        wait to find a new home. Does not include the process of
                        securing money.
    financial_capital -- A capitals.FinancialCapital() object.
    human_capital -- A capitals.HumanCapital() object.
    write_story -- Boolean indicating whether to track a households story.

    Returns or Attribute Changes:
    entity.story -- Process outcomes appended to story.
    entity.money_search_start -- Record time money search starts
    entity.money_search_stop -- Record time money search stops
    entity.gave_up_money_search -- Set to True if search patience runs out.
    entity.money_to_rebuild -- Technically changed (increased) by functions
                                called within.
    """

    # Record when money search starts
    # Calculate the time that money search patience ends
    entity.money_search_start = simulation.now
    patience_end = entity.money_search_start + search_patience

    # Return out of function if entity has enough money to rebuild and does not
    # have any insurance coverage.
    if (entity.money_to_rebuild >= entity.residence.damage_value
        and entity.insurance == 0.0):

        # If True, append search outcome to story.
        if write_story == True:
            entity.story.append(
                '{0} already had enough money to rebuild (1:,.0f) and did not seek
assistance. '.format(
                                    entity.name,
                                    entity.money_to_rebuild
                                    )
                                )
        return

    # If entity has insurance then yield an insurance claim request, the duration
    # of which is limited by entity's money search patience.
    if entity.insurance > 0.0:

        # Define a timeout process to represent search patience, with duration
        # equal to the *remaining* patience. Pass the value "Gave up" if the
        # process completes.
        find_search_patience = simulation.timeout(
                                                    patience_end - simulation.now,
                                                    value='Gave up'
                                                )

        # Define insurance claim request process. Pass data about available
        # insurance claim adjusters.
        try_insurance = simulation.process(
                                            request.insurance_claim(
                                                                    simulation,
                                                                    human_capital,
                                                                    entity,
                                                                    write_story
                                                                    )
                                            )

        # Yield both the patience timeout and the insurance claim request.
        # Pass result for the process that completes first.
        money_search_outcome = yield find_search_patience | try_insurance

        # If patience process completes first, interrupt the insurance claim
        # request and return out of function.
        if money_search_outcome == {find_search_patience: 'Gave up'}:
            entity.gave_up_money_search = True
            try_insurance.interrupt(simulation.now - entity.money_search_start)
```

```
        return

# If entity (still) does not have enough rebuild money then yield an FEMA aid
# request, the duration of which is limited by entity's money search patience.
if entity.money_to_rebuild < entity.residence.damage_value:

    # Define a timeout process to represent search patience, with duration
    # equal to the *remaining* patience. Pass the value "Gave up" if the
    # process completes.
    find_search_patience = simulation.timeout(
                                      patience_end - simulation.now,
                                      value='Gave up'
                                      )

    # Define FEMA aid request process. Pass data about available
    # FEMA processors.
    try_fema = simulation.process(
                                request.fema_assistance(
                                                        simulation,
                                                        human_capital,
                                                        financial_capital,
                                                        entity, write_story
                                                        )
                            )
    # Yield both the patience timeout and the FEMA aid request.
    # Pass result for the process that completes first.
    money_search_outcome = yield find_search_patience | try_fema

    # If patience process completes first, interrupt the FEMA aid
    # request and return out of function.
    if money_search_outcome == {find_search_patience: 'Gave up'}:
        entity.gave_up_money_search = True
        try_fema.interrupt(simulation.now - entity.money_search_start)
        return

# If entity (still) does not have enough rebuild money then yield a loan
# request, the duration of which is limited by entity's money search patience.
if entity.money_to_rebuild < entity.residence.damage_value:

    # Define a timeout process to represent search patience, with duration
    # equal to the *remaining* patience. Pass the value "Gave up" if the
    # process completes.
    find_search_patience = simulation.timeout(patience_end - simulation.now,
                        value='Gave up')

    # Define loan request process. Pass data about available
    # loan processors.
    try_loan = simulation.process(
                                request.loan(
                                            simulation,
                                            human_capital,
                                            entity,
                                            write_story
                                            )
                            )

    # Yield both the patience timeout and the loan request.
    # Pass result for the process that completes first.
    money_search_outcome = yield find_search_patience | try_loan

    # If patience process completes first, interrupt the loan
    # request and return out of function.
    if money_search_outcome == {find_search_patience: 'Gave up'}:
        entity.gave_up_money_search = True
        try_loan.interrupt(simulation.now - entity.money_search_start)
        return

# Record the time and duration when entity's search for money ends without
```

```
    # giving up.
    entity.money_search_stop = simulation.now
    search_duration = entity.money_search_stop - entity.money_search_start

    # If write_story is True, then append money search outcome to entity's story.
    if write_story == True:
        entity.story.append(
            'It took {0} {1:.0f} days to receive enough financial assistance and now has
${2:,.0f} to rebuild. '.format(
                    entity.name,
                    search_duration,
                    entity.money_to_rebuild
                    )
            )
```

---------------------------------------------Master-----------------------------------------------

```
# coding: utf-8

# #                      -- DESASTER --
# Simulating household reconstruction with Discrete Event Simulation
#
# Requires python 3.4

# In[10]:

#stdlib and 3rd Party imports
import sys, datetime
import numpy as np
import pandas as pd
import simpy
from simpy.util import start_delayed

#add path to desaster module, later we'll install this into site-packages so we shouldn't
need to do this
sys.path.append("/home/ubuntu/seagrantsim/")
#import desaster files
from desaster import entities, capitals, request, io, movement, search, rebuild


# Here we're importing the modules and setting up the stuff

# ## Load input files for the scenario

# In[11]:

#scenario_file = '../inputs/scenario_test1.xlsx'
scenario_file = "../inputs/household_inputs.xlsx"

# Create Pandas dataframe of attribute data for all households to be modeled in the
simulation
# required column names, exactly as written: Name , Savings , Insurance
households_df = pd.read_excel(scenario_file, sheetname='households')

# Create Pandas dataframe of attribute data for all vacant homes (housing stock) to be
modeled in the simulation
housing_stock_df = pd.read_excel(scenario_file, sheetname='housing_stock')

# Set input data for all human capital types, as dict or Pandas Series
# .loc stuff is to convert the DataFrame to a Series ... data will function the same as a
dictionary as well
human_cap_data = pd.read_excel(scenario_file, sheetname='human_capital',
index_col=0).iloc[:,0]

# Set input data for all financial capital types, as dict or Pandas Series
```

```
financial_cap_data = pd.read_excel(scenario_file, sheetname='financial_capital',
index_col=0).iloc[:,0]


# ### Randomize households and reset index

# In[12]:

households_df = households_df.sample(frac=1).reset_index(drop=True)
#takes a random sample, frac is a fraction to sample (1 means take a 100% sample),
#reset index drops the old scrambled index and puts in a fresh ascending count
households_df.head()


# # Initiate Simulation

# In[13]:

simulation = simpy.Environment()


# In[14]:

write_story = True #do we want the story of each household?


# In[15]:

financial_capital = capitals.FinancialCapital(simulation, financial_cap_data) #resource

human_capital = capitals.HumanCapital(simulation, human_cap_data) #resource

households = entities.importHouseholds(simulation, households_df, write_story) #entity
object container

housing_stock = capitals.importHousingStock(simulation, housing_stock_df) #available housing


# ### Write a function that controls the flow for each household.

# In[16]:

def master_process(simulation, human_capital, financial_capital, entity, write_story):

    yield simulation.process(request.inspection(simulation, human_capital, entity.residence,
entity, write_story))

    # Specify the event sequence for households from the time of the hazard through the
decisions to relocate
    # or rebuild
    if entity.residence.damage_state != 'None':

        money_patience = 1000  # days until give up the search for rebuild money

        # Search for rebuild money
        yield simulation.process(search.rebuild_money(simulation, human_capital,
                                                financial_capital, entity,
                                                money_patience, write_story))

        if entity.gave_up_money_search == True:
                return

        # If home is completely damaged, search for a new home to purchase.
        if entity.residence.damage_state == 'Complete':

            home_patience = 550  # days until give up the search for a new home
```

```
            search_outcome = yield simulation.process(search.permanent_housing(simulation,
entity, home_patience, housing_stock, human_capital, write_story))

            if entity.gave_up_home_search == True:
                return

        if entity.residence.damage_state != 'None':
            yield simulation.process(request.engineering_assessment(simulation,
human_capital, entity, write_story))

            yield simulation.process(request.permit(simulation, human_capital, entity,
write_story))

            yield simulation.process(rebuild.home(simulation, human_capital,
financial_capital, entity, write_story))


# In[17]:

# Initiate a master process for each household to be modeled in the simulation
for i in range(len(households)):
    simulation.process(master_process(simulation, human_capital, financial_capital,
households[i], write_story))


# In[18]:

undamaged_housing = 0
for i in housing_stock.items:
    if i.damage_state == "None":
        undamaged_housing += 1

print (undamaged_housing)


# ## Rebuild the housing stock

# In[19]:

# Do inspections on all of the vacant homes in the housing stock
for home in housing_stock.items:
    simulation.process(request.inspection(simulation, human_capital, home))

# Schedule an event that randomly fixes moderately or completely damaged homes in the vacant
housing stock
# with probability = fix_probability
fix_probability = 1.0
fix_schedule = 100

start_delayed(simulation, rebuild.stock(simulation, housing_stock, fix_probability),
fix_schedule)


# ## Run the model

# In[20]:

#Reload building material at a preordained time
start_delayed(simulation, capitals.reloadBuildingMaterial(simulation,
financial_capital.building_materials, amount = 100000000), 100)

simulation.run()


# ## Outputs to verify model ran correctly

# In[21]:
```

```
num_undamaged = 0
num_rebuilt = 0
num_gave_up_money_search = 0
num_relocated = 0
num_gave_up_home_search = 0

for household in households:
    if household.money_search_start == 0.0: num_undamaged += 1
    if household.home_get > 0.0: num_rebuilt += 1
    if household.gave_up_money_search: num_gave_up_money_search += 1
    if household.home_search_stop > 0.0: num_relocated += 1
    if household.gave_up_home_search: num_gave_up_home_search += 1

print('{0} out of {1} households suffered no damage to their homes.\n'.format(num_undamaged,
len(households)),
      '{0} out of {1} households rebuilt or repaired their damaged
home.\n'.format(num_rebuilt, len(households)),
        '{0} out of {1} households gave up searching for
money.\n'.format(num_gave_up_money_search, len(households)),
        '{0} out of {1} households decided to find a new home.\n'.format(num_relocated,
len(households)),
        '{0} out of {1} households gave up searching for a
home.'.format(num_gave_up_home_search, len(households))
      )


# # MAKE A NEW DATAFRAME FOR EXPORT

# In[22]:

#fills the empty dataframe we made above for the output. incredibly badly written
a = list(vars(households[4]).keys()) #gets all potential column names
a.remove("household");a.remove("residence") #remove the stuff we don't want
a.append("latitude");a.append("longitude") #add stuff we do want
df = pd.DataFrame(columns=a)
iters = 0
att_itter = 0
new_column={}
log = []
for i in households: #loop through all entities
    i.latitude = i.household["Latitude"] #extracting lat and long from the residence object
    i.longitude = i.household["Longitude"]
    for att in a: #loop through the attributes in our list of column names we want
        try:
            new_column[att] = i.__getattribute__(att) #set the b dictionary
            #mydata[att]= i.__getattribute__(att)

        except ValueError:
            new_column[att] = 'NaN'
        except AttributeError as e:
            new_column[att] = 'NaN'
            log.append("Household {0} had an attr error, {1}".format(i.name, e))
        finally:
            att_itter += 1
    mydata=pd.DataFrame([new_column]) #this turns our newly made column into a database
where it can be combined with the df

    df = df.append(mydata, ignore_index=True)

    iters += 1


print(iters)
print(att_itter)


# In[24]:
```

```
df.head()
output_path = "../Outputs/Output{}.xlsx".format(str(datetime.date.today()))
df.to_excel(output_path)
```