Discrete Event Simulation of Disaster Recovery
https://github.com/milessb/DESaster

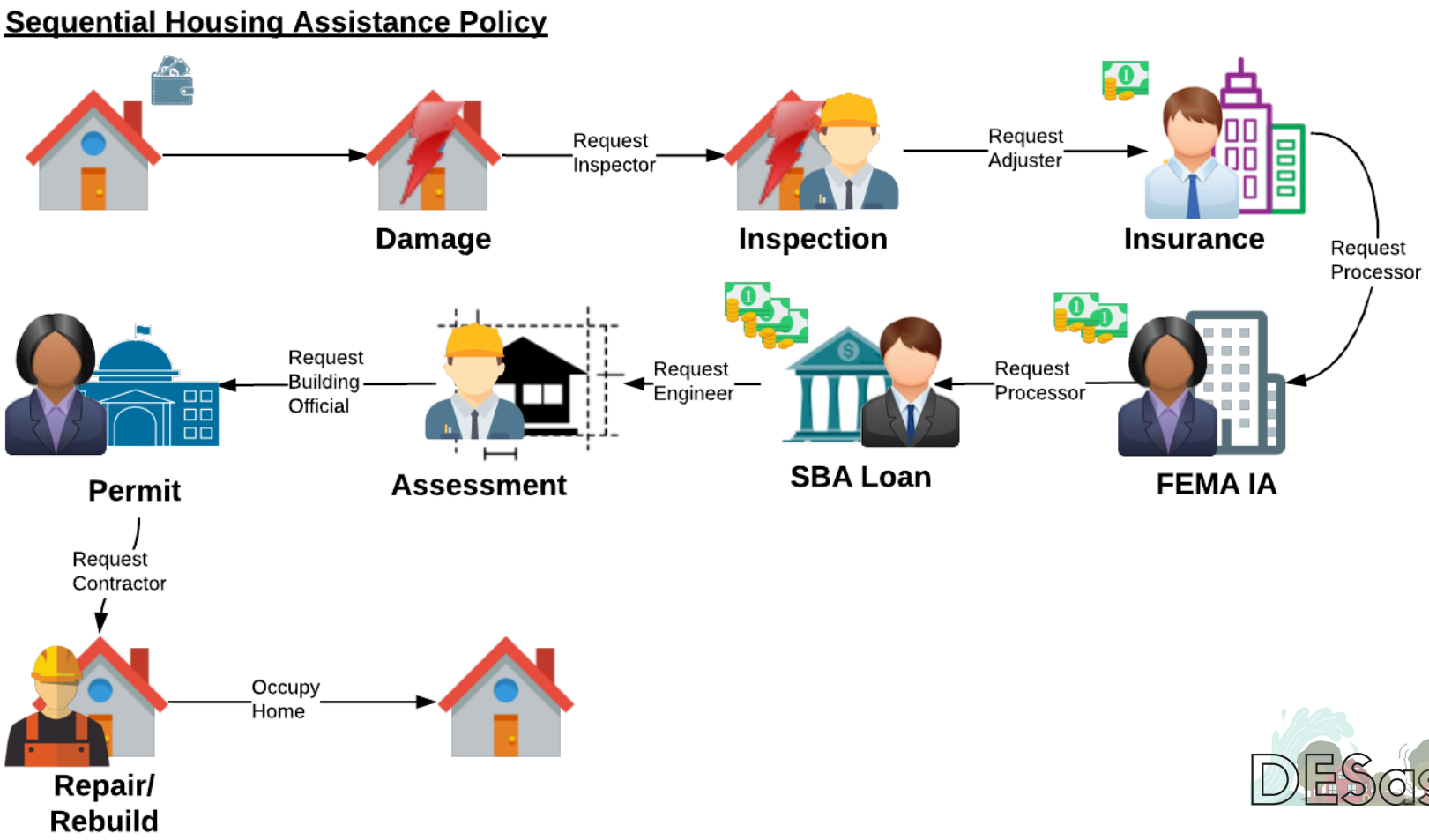
Scott Miles

Center for Collaborative Systems for Security, Safety, and Regional Resilience (CoSSaR)
milessb@uw.edu, www.resilscience.com

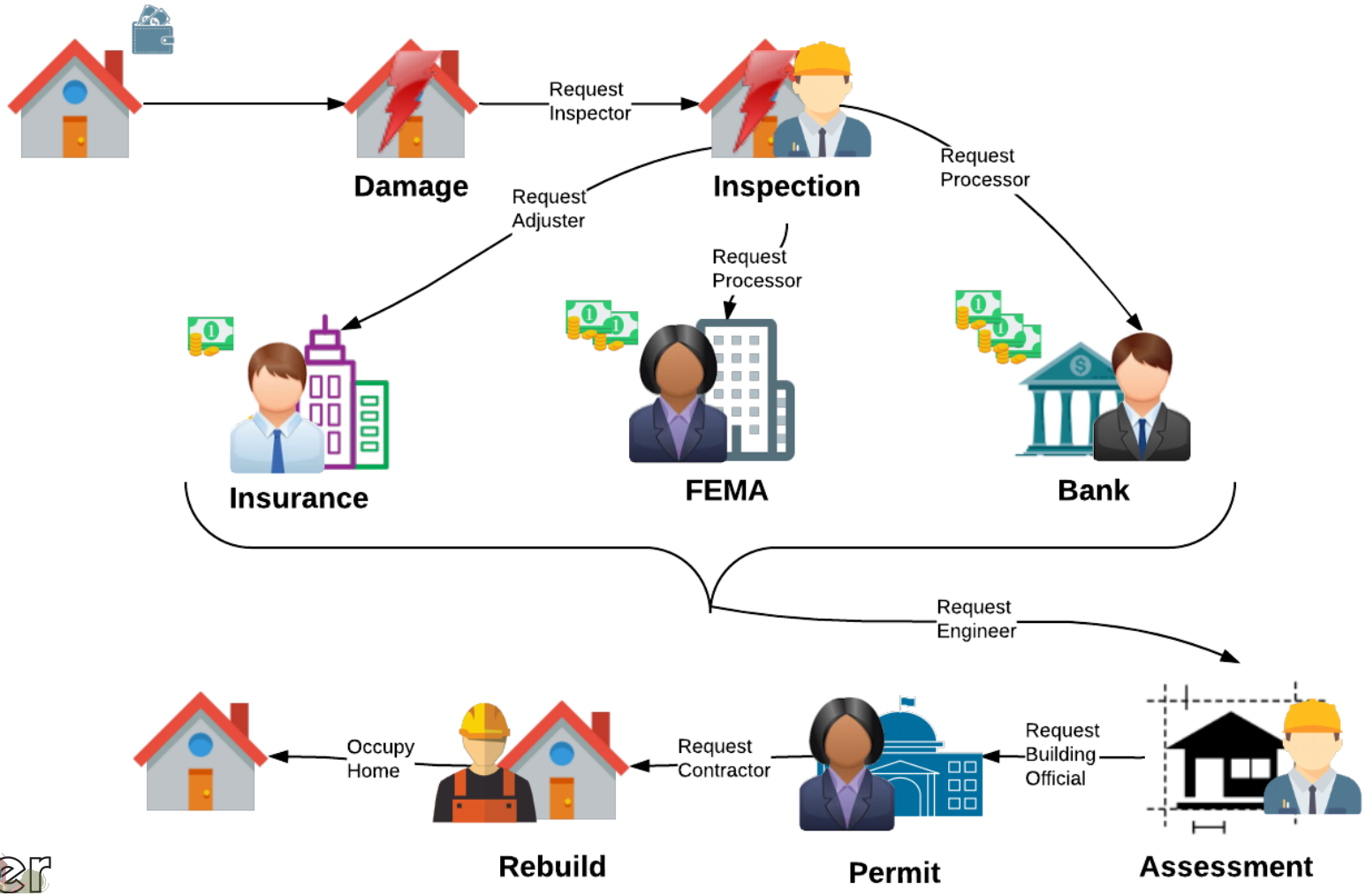
CENTER FOR COLLABORATIVE SYSTEMS FOR SECURITY, SAFETY & REGIONAL RESILIENCE

HUMAN CENTERED DESIGN & ENGINEERING
UNIVERSITY of WASHINGTON

Modular and extensible design. For example to configure different recovery funding assistance policies arrangements….

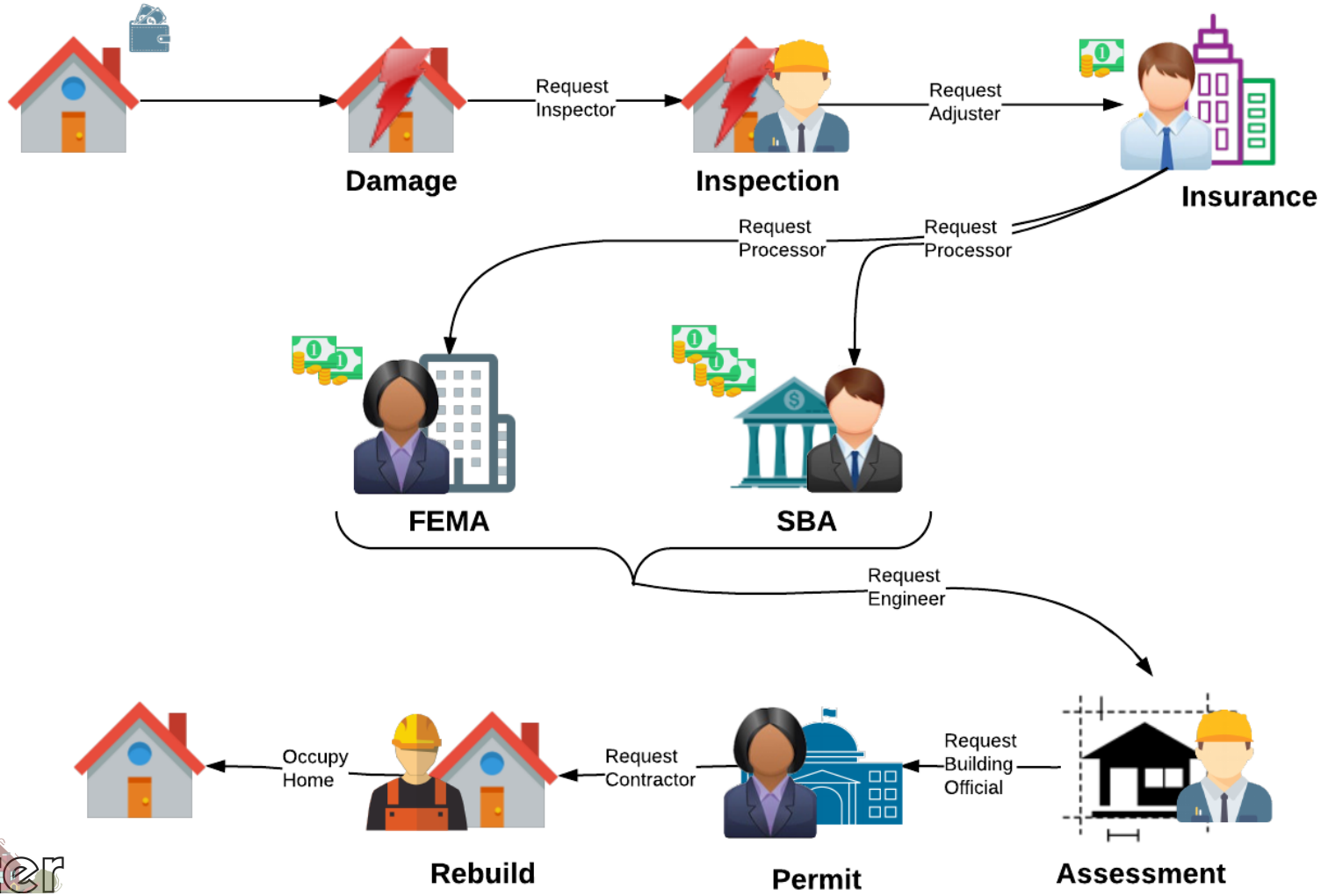


Parallel Housing Assistance Policy



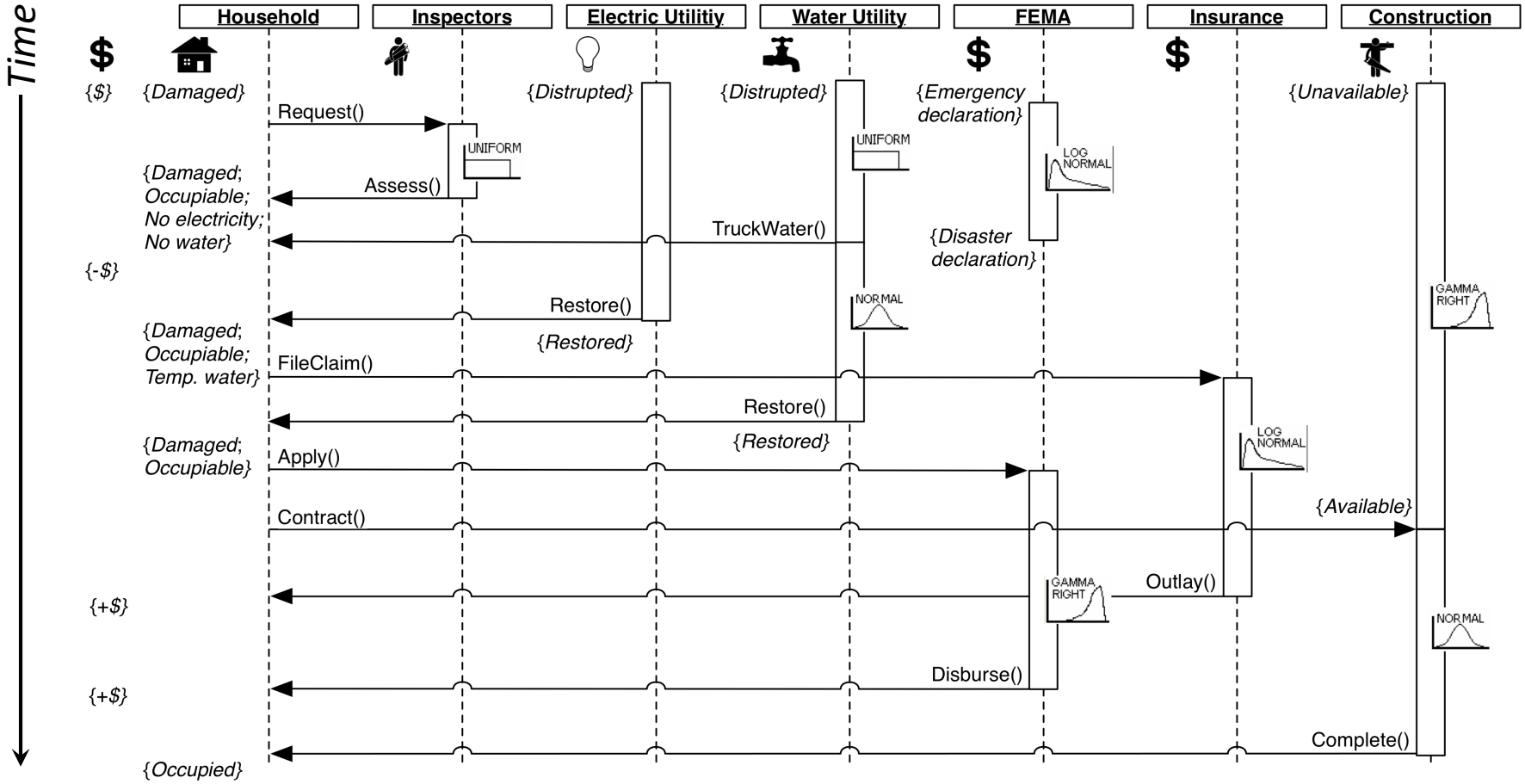
DESaster

Hybrid Housing Assistance Policy (Actual)



DESaster

Implemented w/ SimPy discrete event simulation library;
Represents event durations & resources w/ probability distributions



DESaster Modules and Classes (class functions not listed)

Entities
- Household
- Owner
- OwnerHousehold
- RenterHousehold
- Landlord

Structures
- Building
- SingleFamilyResidential

Financial Recovery Programs
- HousingAssistanceFEMA
- RealPropertyLoanSBA
- OwnersInsurance

Technical Recovery Programs
- InspectionProgram
- EngineeringAssessment
- PermitProgram
- DemolitionProgram
- RepairProgram

Recovery Policies
- Insurance_IA_SBA_Parallel
- Insurance_IA_SBA_Sequential
- Insurance_FirstThen_IA_SBA_Parallel
- Insurance_SBA_Parallel
- Insurance_SBA_Sequential
- RepairVacantBuildings

Visualize
- FoliumMap
- Dashboard

Can use with Jupyter notebook for easy deployment and to facilitate interactive scenario exploration.

Import DESaster modules and classes which can then be arranged in custom ways to model different housing recovery scenarios.

jupyter desaster_application_template Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python [default]

Markdown

DESaster Application Template

v. Master for 10312017

Required Modules

In [1]:

```
# System modules
import sys, random, inspect
from datetime import datetime
from IPython.display import display
import sys, inspect

# Data analysis modules
import pandas as pd
import numpy as np
from scipy.stats import uniform, norm, beta, weibull_min, rv_discrete
import random
from ipywidgets import Dropdown


# SimPy modules
import simpy
from simpy import Resource, Container, FilterStore
from simpy.util import start_delayed

# Viz modules
import matplotlib.pyplot as plt
import seaborn as sns
import folium
from folium import plugins
import branca.colormap as cm
from folium.plugins import MarkerCluster
from folium import Map, FeatureGroup, Marker, LayerControl

### DESaster Modules
desaster_path = "/Users/geomando/Dropbox/github/DESaster"
sys.path.append(desaster_path)
import desaster
from desaster.io import *
from desaster.structures import *
from desaster.financial import *
from desaster.technical import *
from desaster.entities import *
from desaster.policies import *
from desaster.visualize import dashboard, folium_map
```

A custom master process for OwnerHouseholds (owner occupiers). Don't do anything if no damage suffered. If residence damage is "Complete", abandon home and look to buy a different one. Otherwise look for financial assistance for repairs. If money for repairs can't be found (patience runs out), look for a new home. If home search patience runs out, simply stop.

```
def owner_process(env, inspection_program, insurance_program, fema_program, loan_program,
                  assessment_program, permit_program, demolish_program, rebuild_program, search_stock, entity):

    money_patience = 200000  # days until give up the search for rebuild money
    home_patience = 15000  # days until give up the search for a new home

    # Do inspections after inspectors are mobilized
    yield env.timeout(start_delay_dist.rvs())
    yield env.process(inspection_program.process(entity.property, entity))

    # Process damaged properties
    if entity.property.damage_state == 'None':
        yield env.process(entity.occupy(duration = occupy_dist))
    else:

        # Homeowner search for financial assistance using an Insurance_SBA policy. Note two alternate versions
        # can be used: insurance_ia_sba_para, insurance_ia_sba_seq, and insurance_firstthen_ia_sba_para.
        # Paste in the desired policy approach below.
        yield env.process(insurance_firstthen_ia_sba_para.policy(insurance_program, fema_program,
                                                 loan_program, entity, money_patience))

        # If not enough money to repair home or home completely damaged, search for a new home to purchase.
        if (entity.recovery_funds.level < entity.property.damage_value or
                entity.property.damage_state == 'Complete'):

            yield env.process(entity.find_home(search_stock, find_home_dist, down_payment_pct = 0.10,
                                               search_patience = home_patience))

            if entity.gave_up_home_search == None:
                yield env.process(entity.occupy(duration = occupy_dist))

            return
        # Otherwise repair home.
        elif entity.recovery_funds.level >= entity.property.damage_value:

            yield env.process(assessment_program.process(entity.property, entity))
            yield env.process(permit_program.process(entity.property, entity))
            if entity.property.damage_state == 'Extensive' or entity.property.damage_state == 'Complete':
                yield env.process(demolish_program.process(entity.property, entity))
            yield env.process(rebuild_program.process(entity.property, entity))
            yield env.process(entity.occupy(duration = occupy_dist))
```

A custom master process for RenterHouseholds. For the most part it simply initiates a process for their landlords. If they are evicted by their landlords, the renter will look for a new home. If home search patience runs out, simply stop. Otherwise, occupy home after landlord repairs it.

```
def renter_process(env, inspection_program, landlord_insurance,
                   landlord_loan, assessment_program, permit_program, demolish_program, rebuild_program,
                   search_stock, entity):

    money_patience = 365  # days until give up the search for rebuild money
    home_patience = 550  # days until give up the search for a new home

    # Process damaged homes
    if entity.residence.damage_state == 'None':
        yield env.process(entity.occupy(duration = occupy_dist))

    else:
        # Process landlord property repairs
        yield env.process(landlord_process(env, inspection_program, landlord_insurance,
                                landlord_loan, assessment_program, permit_program, demolish_program,
                                rebuild_program, entity.landlord))

        # Check to see if renter has a residence, occupy if so.
        if entity.residence != None:
            yield env.process(entity.occupy(duration = occupy_dist))
        # Otherwise look for a new residence
        else:

            yield env.process(entity.find_home(search_stock, find_home_dist, search_patience = home_patience))

            if not entity.gave_up_home_search:
                    yield env.process(entity.occupy(duration = occupy_dist))

        # For printing and viz convenience, add the landlord's story to renter's story
        entity.story += entity.landlord.story
```

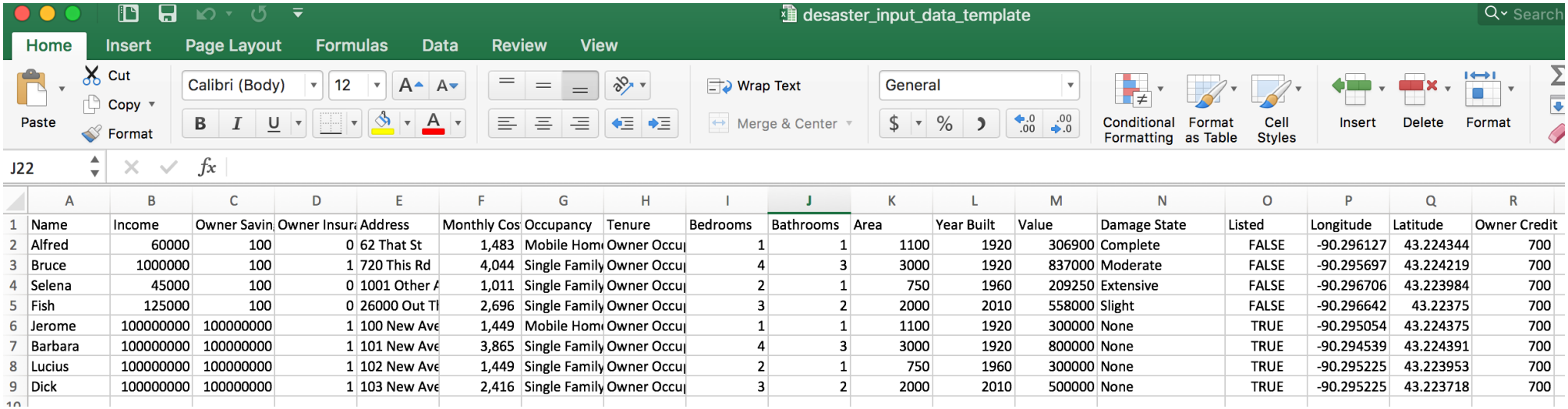
A custom master process for Landlords. Landlords are the owners of renters' residences and so are the ones to seek financial assistance for repairs.

```python
def landlord_process(env, inspection_program, insurance_program, loan_program,
                     assessment_program, permit_program, demolish_program, repair_program, entity):

    money_patience = 100000  # days until give up the search for repair money

    # Do inspection after inspectors are mobilized
    yield env.timeout(start_delay_dist.rvs())
    yield env.process(inspection_program.process(entity.property, entity))

    # Simulate damaged properties
    if entity.property.damage_state != 'None':

        # If is extensively/completely damaged, evict tenant. Eventually initiate temp/transition shelter etc.
        if entity.property.damage_state == 'Extensive' or entity.property.damage_state == 'Complete':
            entity.evict_tenant()

        # Landlord search for financial assistance using an Insurance_SBA policy. Note two alternate versions
        # can be used: insurance_sba_para or insurance_sba_seq. Paste in the desired policy approach below.
        yield env.process(insurance_sba_seq.policy(insurance_program, loan_program, entity, money_patience)) # Sequ

        # If landlord gives up looking for recovery funds, evict their tenant
        if entity.gave_up_funding_search != None:
            entity.evict_tenant()

            if entity.write_story:
                entity.story.append(
                '{0} decided not to repair their {1}. '.format(
                entity.name, entity.property.occupancy.lower()
                                                               )
                                    )
            return

        # If has enough recovery funds, repair; if not, evict tenant.
        if entity.recovery_funds.level >= entity.property.damage_value:
            yield env.process(assessment_program.process(entity.property, entity))
            yield env.process(permit_program.process(entity.property, entity))

            # Demolish property if > extensive damage
            if entity.property.damage_state == 'Extensive' or entity.property.damage_state == 'Complete':
                yield env.process(demolish_program.process(entity.property, entity))
            yield env.process(repair_program.process(entity.property, entity))
        else:
            if entity.tenant.residence != None:
                entity.evict_tenant()
```

Data requirements for…..

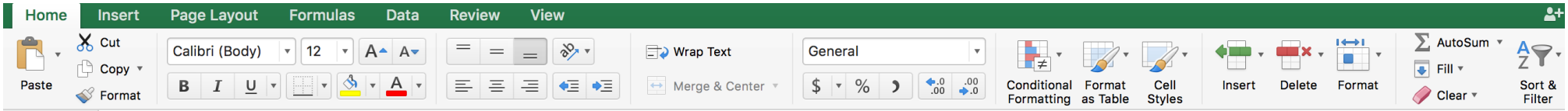
Resource and event duration parameters must also be defined, but typically done so interactively

Owners



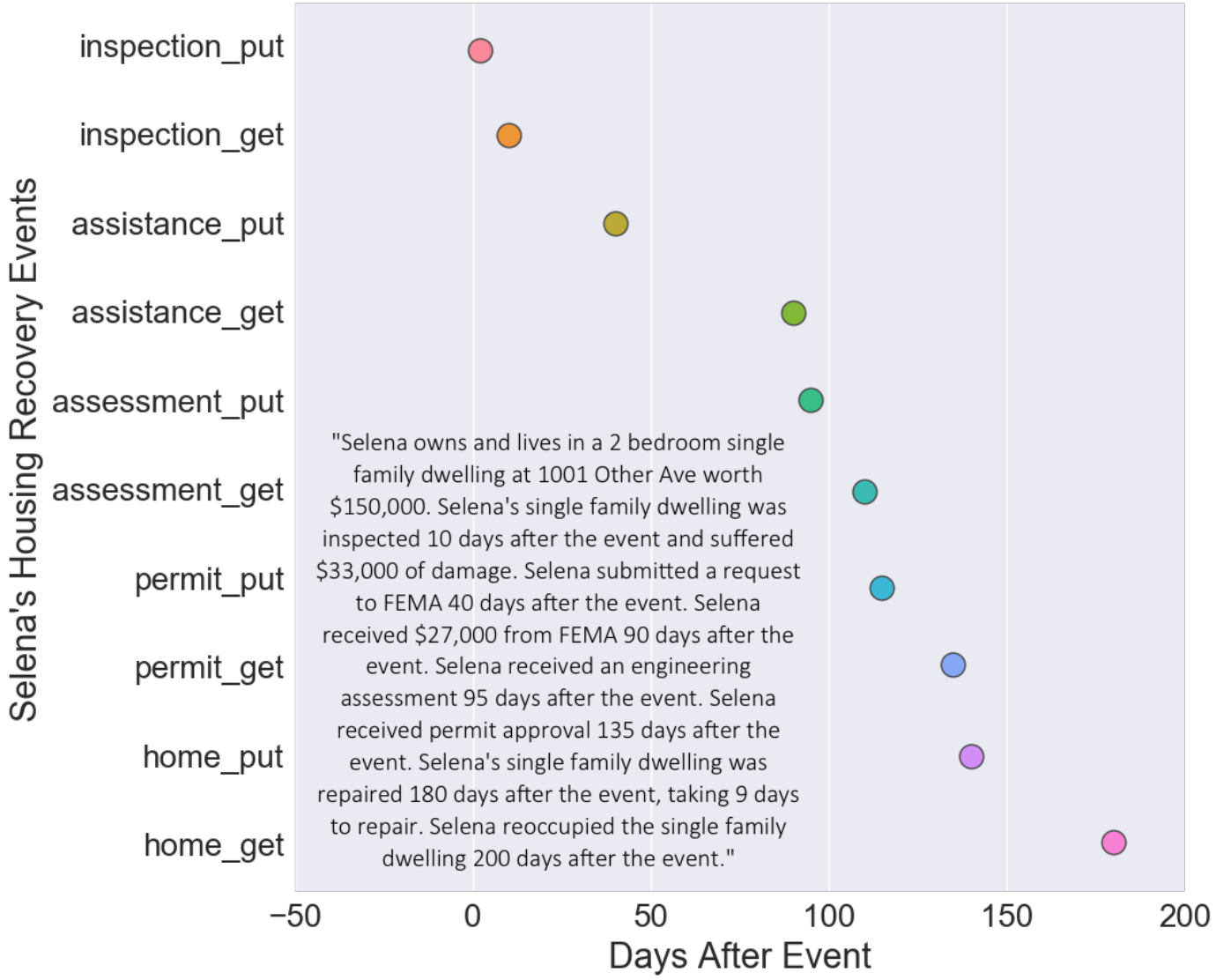
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Name	Income	Owner Savin	Owner Insur	Address	Monthly Cos	Occupancy	Tenure	Bedrooms	Bathrooms	Area	Year Built	Value	Damage State	Listed	Longitude	Latitude	Owner Credit
2	Alfred	60000	100	0	62 That St	1,483	Mobile Hom	Owner Occu	1	1	1100	1920	306900	Complete	FALSE	-90.296127	43.224344	700
3	Bruce	1000000	100	1	720 This Rd	4,044	Single Family	Owner Occu	4	3	3000	1920	837000	Moderate	FALSE	-90.295697	43.224219	700
4	Selena	45000	100	0	1001 Other A	1,011	Single Family	Owner Occu	2	1	750	1960	209250	Extensive	FALSE	-90.296706	43.223984	700
5	Fish	125000	100	0	26000 Out Th	2,696	Single Family	Owner Occu	3	2	2000	2010	558000	Slight	FALSE	-90.296642	43.22375	700
6	Jerome	100000000	100000000	1	100 New Ave	1,449	Mobile Hom	Owner Occu	1	1	1100	1920	300000	None	TRUE	-90.295054	43.224375	700
7	Barbara	100000000	100000000	1	101 New Ave	3,865	Single Family	Owner Occu	4	3	3000	1920	800000	None	TRUE	-90.294539	43.224391	700
8	Lucius	100000000	100000000	1	102 New Ave	1,449	Single Family	Owner Occu	2	1	750	1960	300000	None	TRUE	-90.295225	43.223953	700
9	Dick	100000000	100000000	1	103 New Ave	2,416	Single Family	Owner Occu	3	2	2000	2010	500000	None	TRUE	-90.295225	43.223718	700

Renters

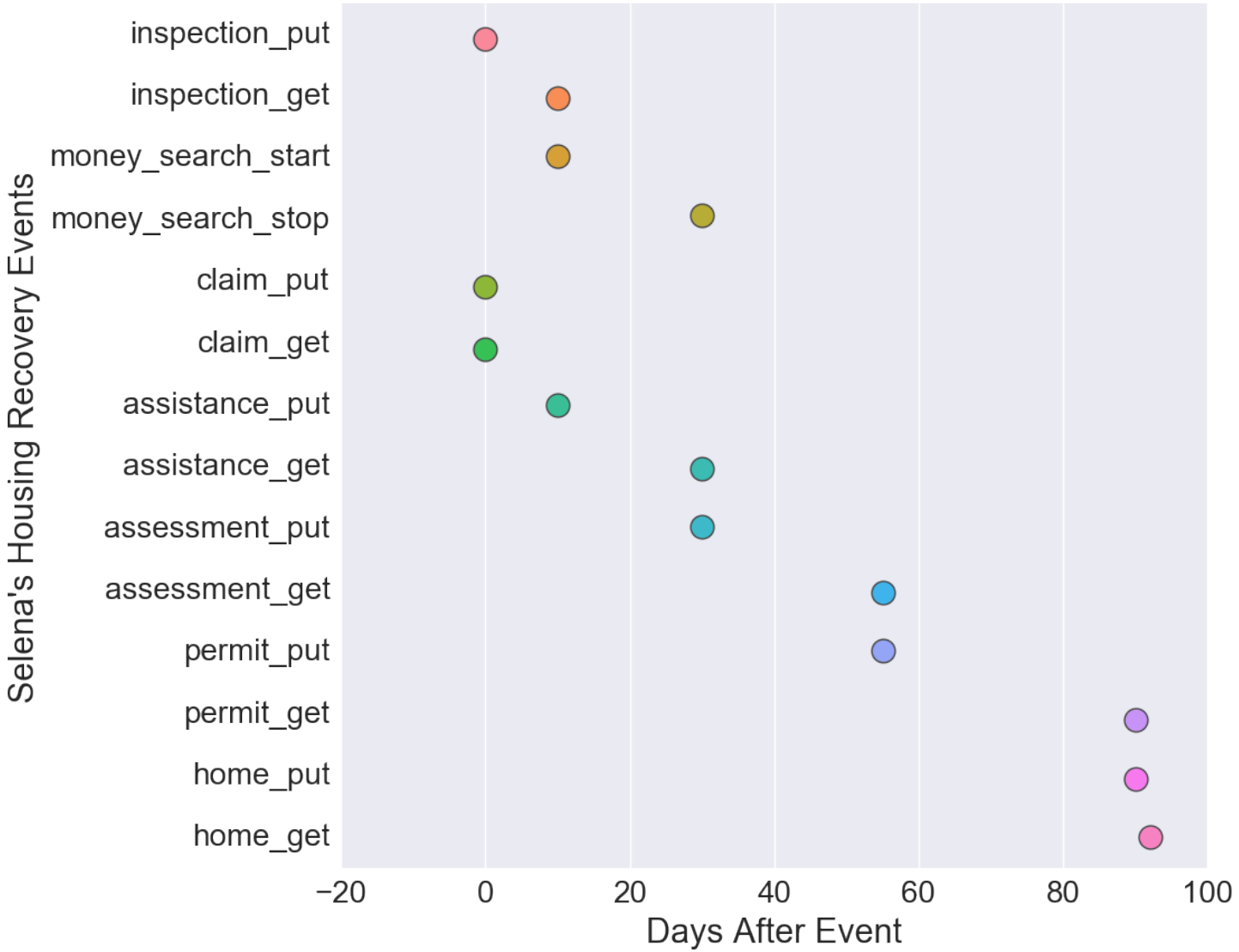


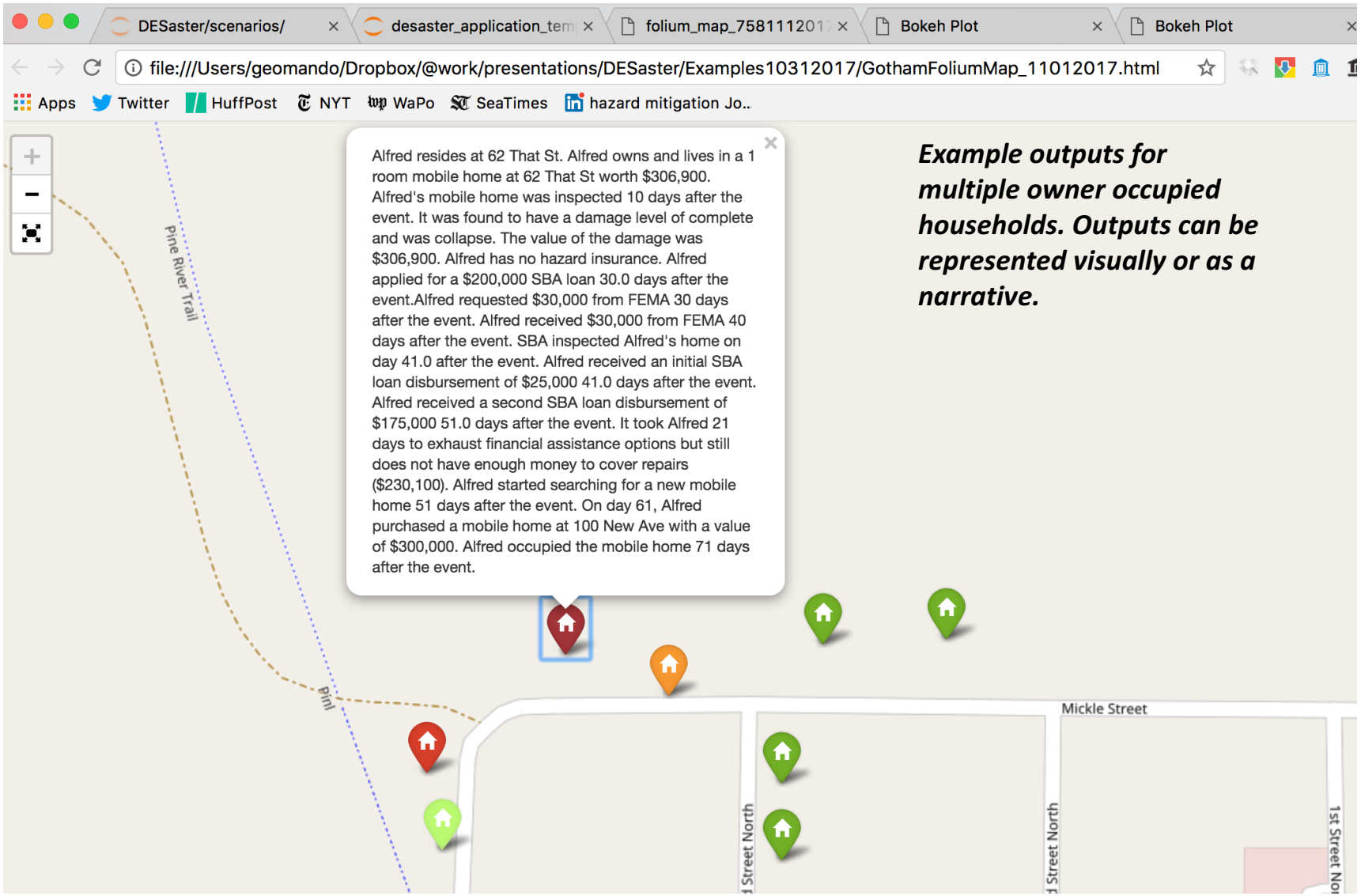
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Name	Address	Occupancy	Tenure	Income	Tenant Savin	Tenant Credi	Tenant Insur	Monthly Cos	Bedrooms	Bathrooms	Area	Year Built	Value	Damage State	Landlord	Owner Savin	Owner Credi	Owner Insur	Listed	Longitude	Latitude
2	Ivy	262 That St	Mobile Hom	Rental	119040	1000	700	0	2480	1	1	1000	1920	279000	Moderate	Alice	30000	700	0	FALSE	-90.29424	43.224015
3	Edward	4720 This Rd	Single Family	Rental	178560	1000	700	0	3720	3	2	1500	1920	418500	Slight	Julie	30000	700	0	FALSE	-90.29377	43.224062
4	Oswald	2301 Other A	Single Family	Rental	59520	1000	700	0	1240	0	1	500	1960	139500	Extensive	Gerry	30000	700	1	FALSE	-90.29329	43.224125
5	James	74000 Out Th	Single Family	Rental	238080	1000	700	0	4960	2	2	2000	2010	558000	Complete	Sally	30000	700	1	FALSE	-90.29306	43.223937
6	Butch	100 Old Ave	Mobile Hom	Rental	100000000	1000	700	0	2480	1	1	1000	1920	279000	None	Greg	100000000	700	1	TRUE	-90.29308	43.223671
7	Harvey	101 Old Ave	Single Family	Rental	100000000	1000	700	0	3720	3	2	1500	1920	418500	None	Allison	100000000	700	1	TRUE	-90.29274	43.223546
8	Lee	102 Old Ave	Single Family	Rental	100000000	1000	700	0	1240	0	1	500	1960	139500	None	Rachel	100000000	700	1	TRUE	-90.29619	43.223406
9	Carmine	103 Old Ave	Single Family	Rental	100000000	1000	700	0	4960	2	2	2000	2010	558000	None	Larry	100000000	700	1	TRUE	-90.29574	43.223359

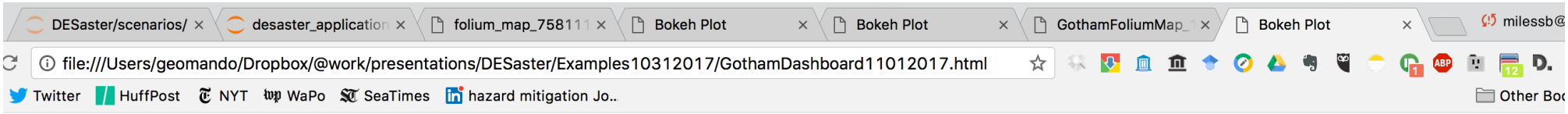
Example outputs for a single owner occupied household. Outputs can be represented visually or as a narrative.



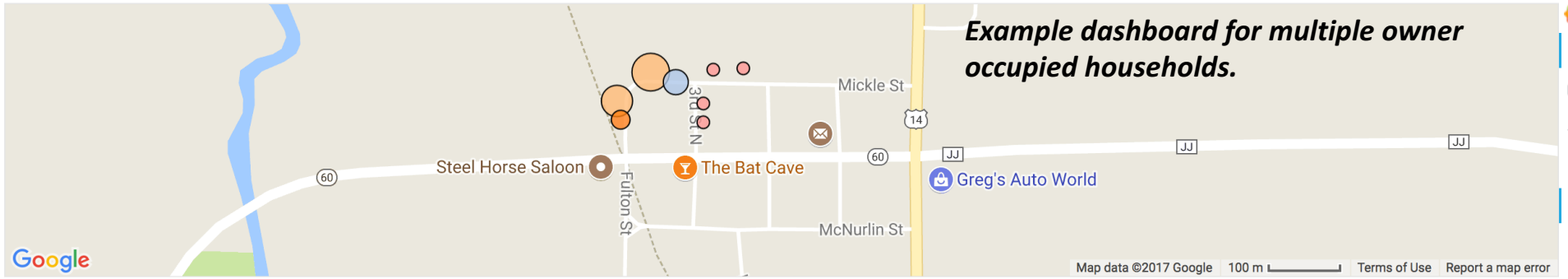
Different scenarios may create different outputs (e.g., if human resource parameters are varied).



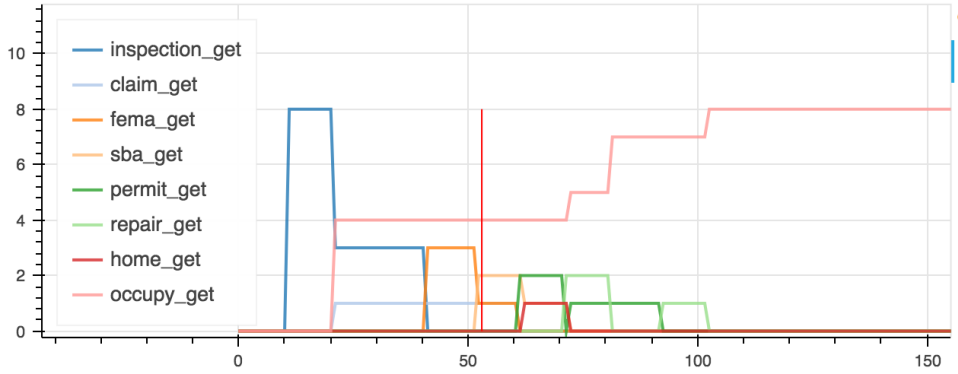

DESaster/scenarios/
desaster_application_tem
folium_map_7581112017
Bokeh Plot
Bokeh Plot
file:///Users/geomando/Dropbox/@work/presentations/DESaster/Examples10312017/GothamFoliumMap_11012017.html
Apps
Twitter
HuffPost
NYT
WaPo
SeaTimes
hazard mitigation Jo..
Alfred resides at 62 That St. Alfred owns and lives in a 1 room mobile home at 62 That St worth $306,900. Alfred's mobile home was inspected 10 days after the event. It was found to have a damage level of complete and was collapse. The value of the damage was $306,900. Alfred has no hazard insurance. Alfred applied for a $200,000 SBA loan 30.0 days after the event.Alfred requested $30,000 from FEMA 30 days after the event. Alfred received $30,000 from FEMA 40 days after the event. SBA inspected Alfred's home on day 41.0 after the event. Alfred received an initial SBA loan disbursement of $25,000 41.0 days after the event. Alfred received a second SBA loan disbursement of $175,000 51.0 days after the event. It took Alfred 21 days to exhaust financial assistance options but still does not have enough money to cover repairs ($230,100). Alfred started searching for a new mobile home 51 days after the event. On day 61, Alfred purchased a mobile home at 100 New Ave with a value of $300,000. Alfred occupied the mobile home 71 days after the event.
Example outputs for multiple owner occupied households. Outputs can be represented visually or as a narrative.
Pine River Trail
Pine
Mickle Street
Street North
Street North
1st Street



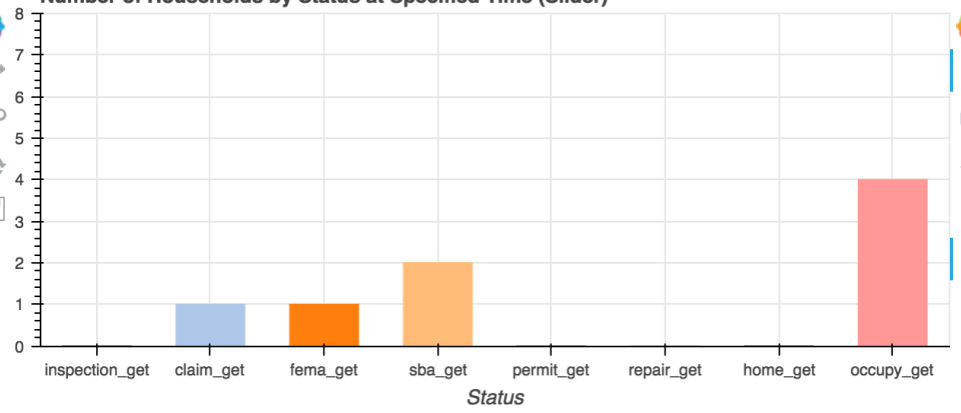
Locations of Modeled Homes (Marker Size Proportional to Original Damage State; Colors Defined Below)



Number of Households By Status vs. Simulated Time

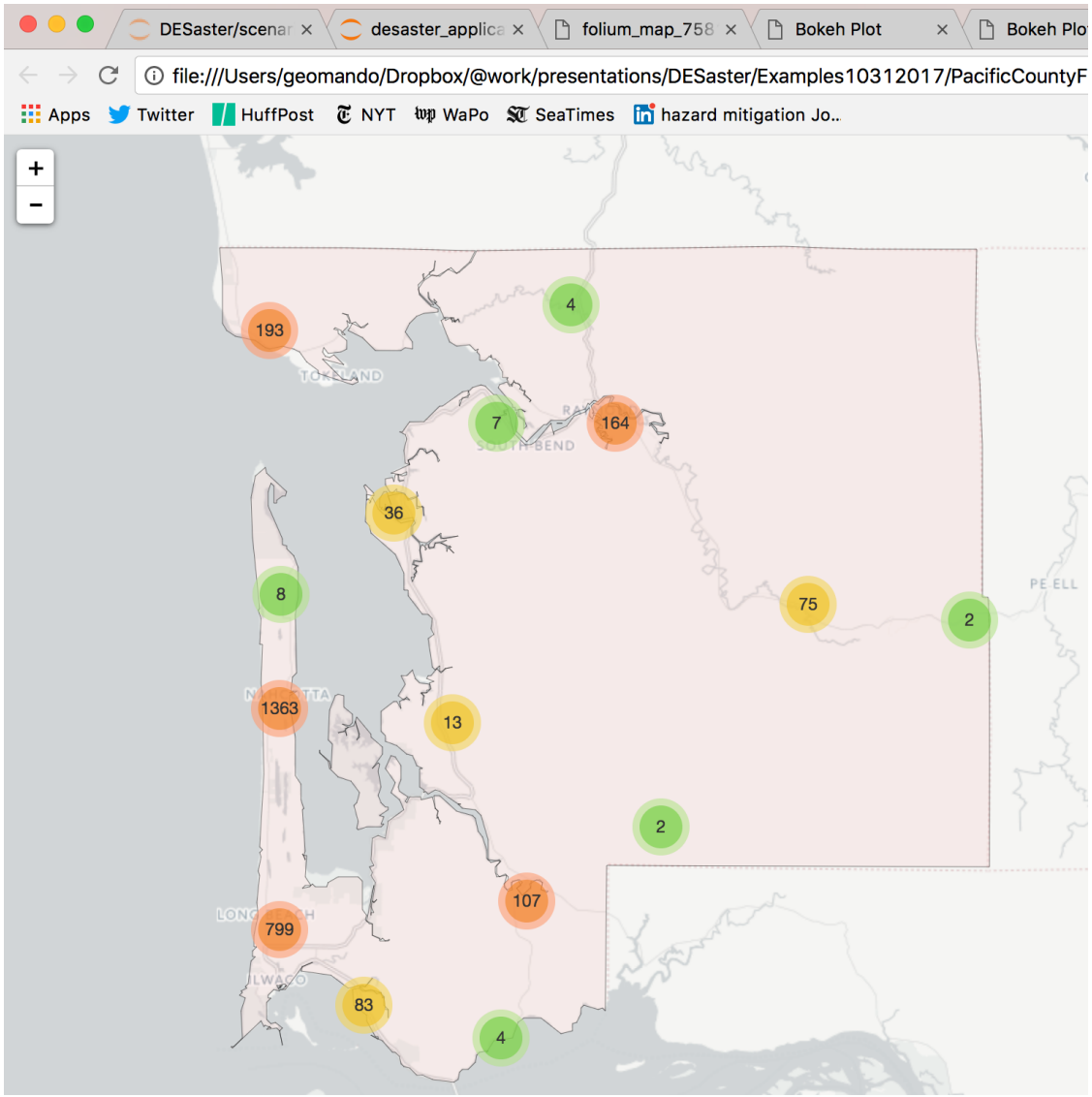


Number of Households by Status at Specified Time (Slider)

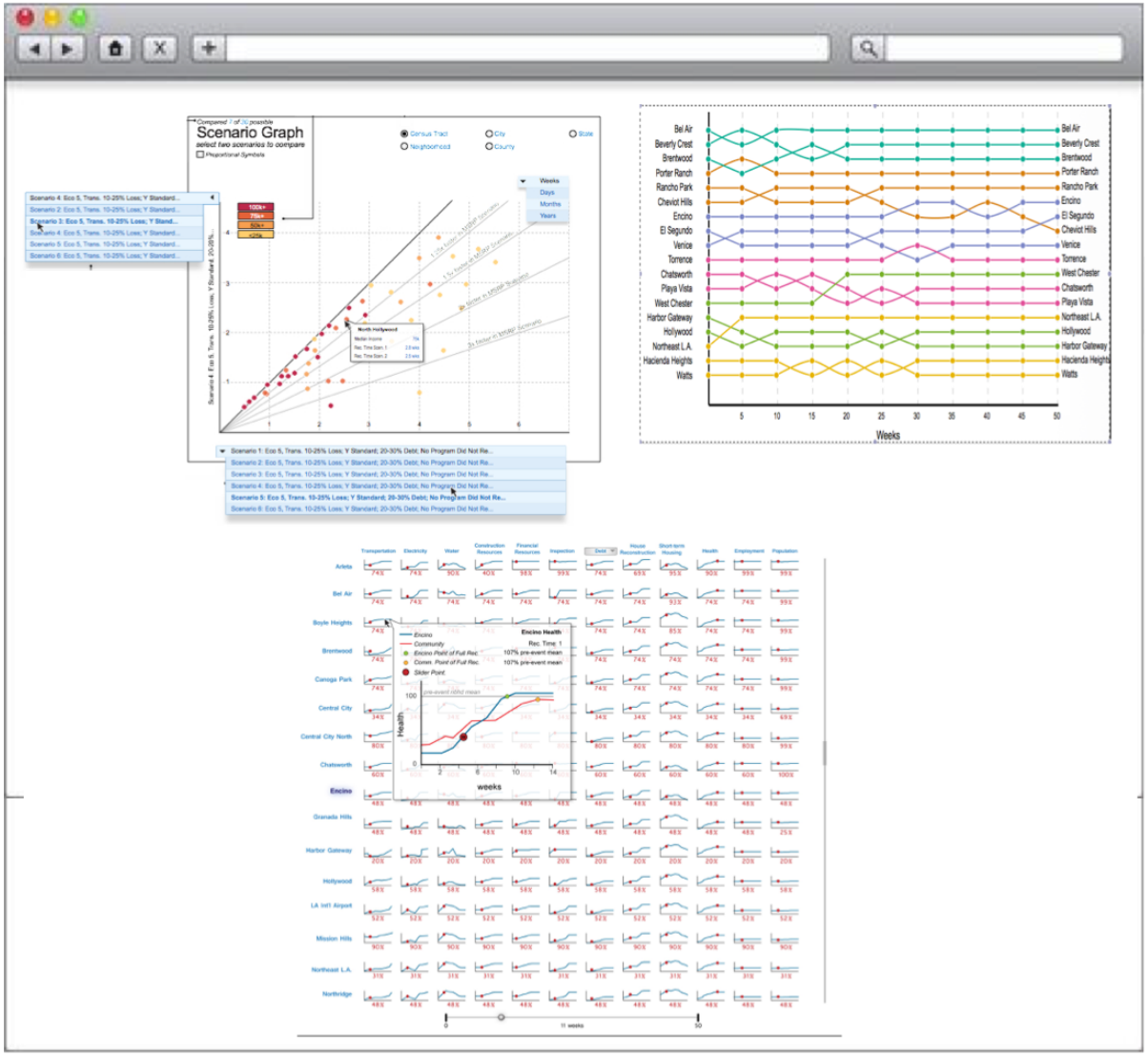


DAY: **53**

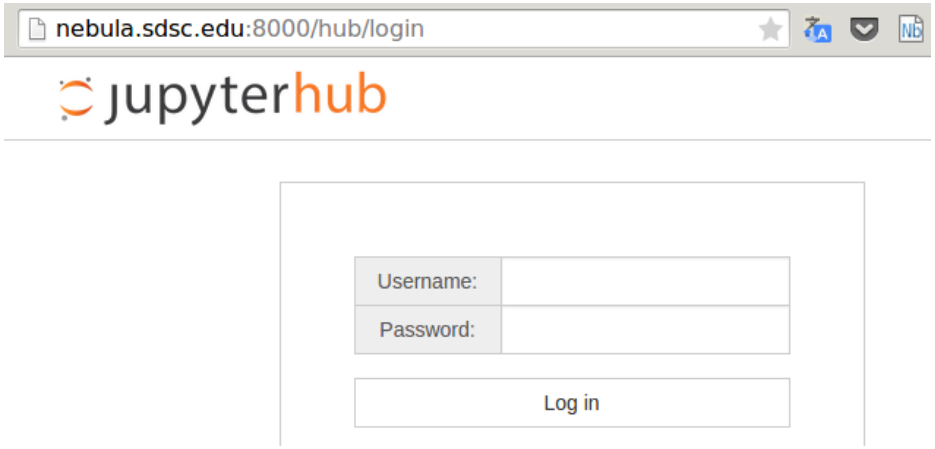
Development work has been done with synthetic data and data representing households in Pacific County, WA

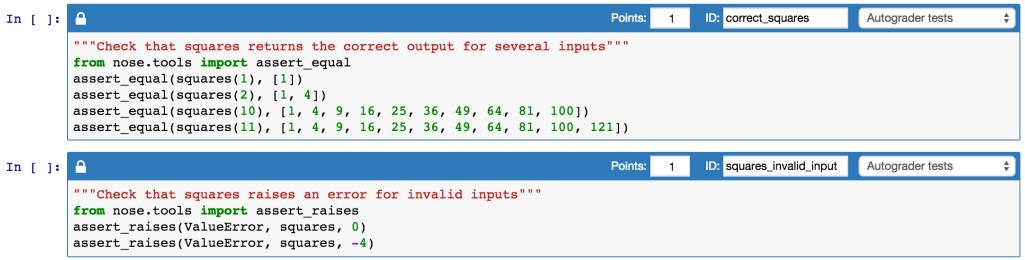


Much more geovisualization and graphical interface development work is needed to make DESaster useable in participatory and collaborative settings (e.g., workshops).

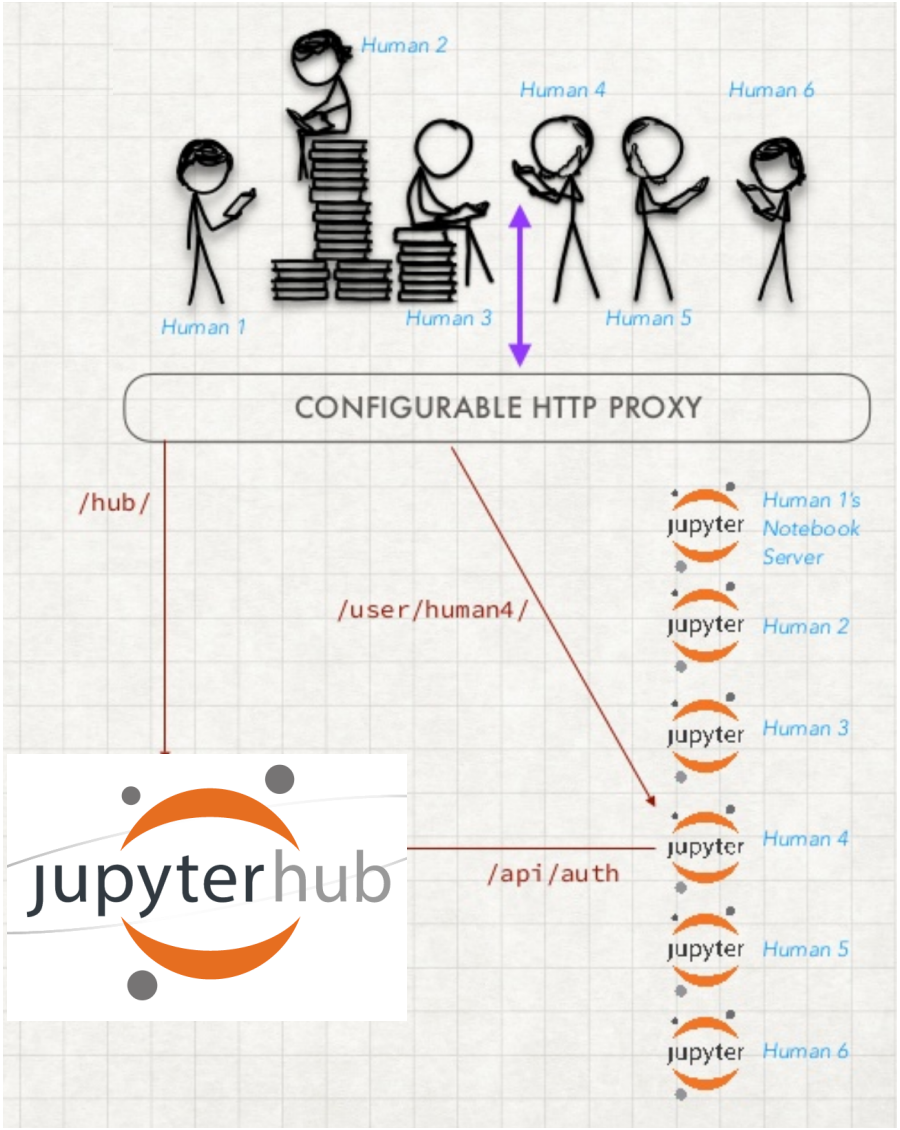


JupyterHub can be used to eventually permit parallel synchronous or asynchronous model building and scenario exploration (e.g., computer-supported workshop)





nbgrader



Discrete Event Simulation of Disaster Recovery
https://github.com/milessb/DESaster

Scott Miles

Center for Collaborative Systems for Security, Safety, and Regional Resilience (CoSSaR)
milessb@uw.edu, www.resilscience.com

CENTER FOR COLLABORATIVE SYSTEMS FOR SECURITY, SAFETY & REGIONAL RESILIENCE

HUMAN CENTERED DESIGN & ENGINEERING
UNIVERSITY of WASHINGTON