Human-Centered Recovery Modeling for Analytic-Deliberative Seismic Resilience Planning

Scott Miles
milessb@uw.edu

National Science Foundation
CMMI IMEE Grant #1560939
<table>
<thead>
<tr>
<th>SPUR Resilient City</th>
<th>Current</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail businesses</td>
<td>1095</td>
<td>120</td>
</tr>
<tr>
<td>Offices and workplaces</td>
<td>1095</td>
<td>120</td>
</tr>
<tr>
<td>Housing</td>
<td>1095</td>
<td>120</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resilient Washington State</th>
<th>Current</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finance and banking</td>
<td>7-90</td>
<td>7-90</td>
</tr>
<tr>
<td>Commercial facilities</td>
<td>30-90</td>
<td>7-90</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>30-90</td>
<td>30-90</td>
</tr>
<tr>
<td>Housing: single family</td>
<td>7-30</td>
<td>1-3</td>
</tr>
<tr>
<td>Housing: multi-family</td>
<td>30-90</td>
<td>7-30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Oregon Resilience Plan</th>
<th>Current</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finance and banking</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Retail businesses</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Housing</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Assess planning/policy aims and convene process

Identify key sectors, components, infrastructures

Define resilience goals and objectives

Identify driving forces; rank their importance and uncertainty to prioritize scenario axes
Gather data using agreed upon methods and experts; model scenarios → Evaluate findings in the context of aims, goals, objectives, modeling limitations → Translate modeling findings and make recommendations

Select 2-3 scenario axes; flesh out the scenarios constraints → Model impacts and recovery under 4-8 different scenarios → Evaluate alternative resilience policies across scenarios
Discrete Event Simulation of Disaster Recovery

https://github.com/milessb/DESaster
A discrete event disaster recovery simulation built on top of the Simpy discrete event simulation Python library.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>disaster</td>
<td></td>
</tr>
<tr>
<td>recovery</td>
<td></td>
</tr>
<tr>
<td>simulation</td>
<td></td>
</tr>
<tr>
<td>discrete-event</td>
<td></td>
</tr>
<tr>
<td>university-of-washington</td>
<td></td>
</tr>
</tbody>
</table>

- **229 commits**
- **4 branches**
- **2 releases**
- **2 contributors**
- **GPL-3.0**

<table>
<thead>
<tr>
<th>Author</th>
<th>Message</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>milessb</td>
<td>Deleted default scale from technical.RepairProgram()</td>
<td>4 days ago</td>
</tr>
<tr>
<td>.ipynb_checkpoints</td>
<td>Made recovery_funds a container, implemented prior_residences/propert...</td>
<td>a year ago</td>
</tr>
<tr>
<td>desaster</td>
<td>Deleted default scale from technical.RepairProgram()</td>
<td>4 days ago</td>
</tr>
<tr>
<td>images</td>
<td>Fixed type on README</td>
<td>2 months ago</td>
</tr>
<tr>
<td>inputs</td>
<td>Changed io.py and input_data_template.xlsx so that column names are l...</td>
<td>a month ago</td>
</tr>
<tr>
<td>scenarios</td>
<td>Updated Google Map API in visualize.py (Bokeh GMap call)</td>
<td>a month ago</td>
</tr>
<tr>
<td>.gitignore</td>
<td>ignore temp xls files</td>
<td>a year ago</td>
</tr>
<tr>
<td>LICENSE</td>
<td>Created license</td>
<td>3 months ago</td>
</tr>
</tbody>
</table>
Modular and extensible design for scenario development

Sequential Housing Assistance Policy

- Permit
  - Request Building Official
  - Request Contractor
  - Repair/Rebuild

- Damage
  - Request Inspector

- Inspection
  - Request Adjuster
  - Request ‘Engineer’

- Insurance
  - Request Processor

- SBA Loan
  - Request Processor

- FEMA IA

DEsaster
For example...
Gather data using agreed upon methods and experts; model scenarios

Evaluate findings in the context of aims, goals, objectives, modeling limitations

Translate modeling findings and make recommendations

Select 2-3 scenario axes; flesh out the scenarios constraints

Model impacts and recovery under 4-8 different scenarios

Evaluate alternative resilience policies across scenarios
DESa ter 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**
- Stories
- FoliumMap
- Dashboard
>>> def car(env):
...     while True:
...         print('Start parking at %d' % env.now)
...         parking_duration = 5
...         yield env.timeout(parking_duration)
...         print('Start driving at %d' % env.now)
...         trip_duration = 2
...         yield env.timeout(trip_duration)

>>> import simpy
>>> env = simpy.Environment()
>>> env.process(car(env))
<Process(car) object at 0x...>
>>> env.run(until=15)
Start parking at 0
Start driving at 5
Start parking at 7
Start driving at 12
Start parking at 14
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.**
```python
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_sag, 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))

        # 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))
```

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
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))  #

        # 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 {}. '.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()
### Input data requirements for entities....

#### Owner Input Data

| Name            | Income | Owner Savin | Owner Insur. | Address       | Monthly Cost | Occupancy | Tenure | Bedrooms | Bathrooms | Area | Year Built | Value | Damage State | Listed | Longitude | Latitude | Owner Credit |
|-----------------|--------|-------------|--------------|---------------|--------------|------------|--------|----------|-----------|----------|-------|-----------|-------|-------------|--------|-----------|-----------|--------------|
| Alfred          | 60000  | 100         | 1 62 Th St   | 1,483 Mobile | Owner Occup. | 1          | 1      | 1        | 1         | 1100    | 1920  | 306900    | Complete | FALSE     | -90.296127 | 43.224344 | 700         |
| Bruce           | 100000 | 100         | 1 720 This Rd| 4,044 Single | Family Occup.| 4          | 3      | 2        | 3         | 3000    | 1920  | 837000    | Moderate | FALSE     | -90.295697 | 43.224219 | 700         |
| Selena          | 45000  | 100         | 0 1001 Other | 1,111 Single | Family Occup.| 2          | 1      | 1        | 1         | 750     | 1960  | 2092500   | Extensive | FALSE     | -90.296706 | 43.223984 | 700         |
| Fish            | 125000 | 100         | 0 26000 Out T | 2,696 Single | Family Occup.| 3          | 2      | 2        | 2         | 2000    | 2010  | 558000    | Slight   | FALSE     | -90.296642 | 43.223757 | 700         |
| Jerome          | 1000000| 10000000    | 1 100 New Ave | 1,440 Mobile | Home Occup. | 1          | 1      | 1        | 1         | 1100    | 1920  | 300000    | None     | TRUE      | -90.295054 | 43.224375 | 700         |
| Barbara         | 1000000| 10000000    | 1 101 New Ave | 3,865 Single | Family Occup.| 4          | 3      | 3        | 3         | 3000    | 1920  | 800000    | None     | TRUE      | -90.295439 | 43.224391 | 700         |
| Lucius          | 1000000| 10000000    | 1 102 New Ave | 1,440 Single | Family Occup.| 2          | 1      | 1        | 1         | 750     | 1960  | 300000    | None     | TRUE      | -90.295225 | 43.223953 | 700         |
| Dick            | 1000000| 10000000    | 1 103 New Ave | 2,416 Single | Family Occup.| 3          | 2      | 2        | 2         | 2000    | 2010  | 500000    | None     | TRUE      | -90.295225 | 43.223718 | 700         |

#### Renter & Landlord Input Data

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Occupancy</th>
<th>Tenure</th>
<th>Income</th>
<th>Tenant Savin</th>
<th>Tenant Insur</th>
<th>Monthly Cost</th>
<th>Bathrooms</th>
<th>Area</th>
<th>Year Built</th>
<th>Value</th>
<th>Damage State</th>
<th>Landlord</th>
<th>Owner Savin</th>
<th>Owner Insur</th>
<th>Listed</th>
<th>Longitude</th>
<th>Latitude</th>
<th>Owner Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ivy</td>
<td>262 Th St</td>
<td>Mobile Home Rental</td>
<td>11904</td>
<td>100</td>
<td>700</td>
<td>0</td>
<td>2480</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1000</td>
<td>1920</td>
<td>Alice</td>
<td>30000</td>
<td>700</td>
<td>FALSE</td>
<td>-90.29424</td>
<td>43.22015</td>
<td></td>
</tr>
<tr>
<td>Edward</td>
<td>4720 Ths Rd</td>
<td>Single Family Rental</td>
<td>17850</td>
<td>100</td>
<td>700</td>
<td>0</td>
<td>3720</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1500</td>
<td>1920</td>
<td>Julie</td>
<td>30000</td>
<td>700</td>
<td>FALSE</td>
<td>-90.29377</td>
<td>43.220462</td>
<td></td>
</tr>
<tr>
<td>Oswald</td>
<td>2301 Other V Single Family Rental</td>
<td>59520</td>
<td>100</td>
<td>700</td>
<td>0</td>
<td>1240</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>500</td>
<td>1960</td>
<td>Gerry</td>
<td>30000</td>
<td>700</td>
<td>FALSE</td>
<td>-90.29329</td>
<td>43.224125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>James</td>
<td>74000 Out T Single Family Rental</td>
<td>238080</td>
<td>100</td>
<td>700</td>
<td>0</td>
<td>4960</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2000</td>
<td>2010</td>
<td>Sally</td>
<td>30000</td>
<td>700</td>
<td>FALSE</td>
<td>-90.29306</td>
<td>43.223537</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Butch</td>
<td>100 Old Ave</td>
<td>Mobile Home Rental</td>
<td>10000000</td>
<td>100</td>
<td>700</td>
<td>0</td>
<td>2480</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1000</td>
<td>1920</td>
<td>Greg</td>
<td>10000000</td>
<td>700</td>
<td>TRUE</td>
<td>-90.29308</td>
<td>43.223671</td>
<td></td>
</tr>
<tr>
<td>Harvey</td>
<td>101 Old Ave</td>
<td>Single Family Rental</td>
<td>10000000</td>
<td>100</td>
<td>700</td>
<td>0</td>
<td>3720</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1500</td>
<td>1920</td>
<td>Allison</td>
<td>10000000</td>
<td>700</td>
<td>TRUE</td>
<td>-90.29274</td>
<td>43.223546</td>
<td></td>
</tr>
<tr>
<td>Lee</td>
<td>102 Old Ave</td>
<td>Single Family Rental</td>
<td>10000000</td>
<td>100</td>
<td>700</td>
<td>0</td>
<td>1240</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>500</td>
<td>1960</td>
<td>Rachel</td>
<td>10000000</td>
<td>700</td>
<td>TRUE</td>
<td>-90.29269</td>
<td>43.223460</td>
<td></td>
</tr>
<tr>
<td>Carmine</td>
<td>103 Old Ave</td>
<td>Single Family Rental</td>
<td>10000000</td>
<td>100</td>
<td>700</td>
<td>0</td>
<td>4960</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2000</td>
<td>2010</td>
<td>Larry</td>
<td>10000000</td>
<td>700</td>
<td>TRUE</td>
<td>-90.29574</td>
<td>43.223359</td>
<td></td>
</tr>
</tbody>
</table>
For example...
Resource and event duration parameters must also be defined...

...three types of data sources are conceivable
Jaiswal et al. (2012)
Prototype sentences: Examples to find matches in the corpus

“People were unhappy with the quality of their home repairs.”
“The city's power system was mostly restored within hours.”

Text Corpora: Big text data sets compiled for 2010-2011 Canterbury earthquakes disaster. Additional disaster case studies possible.

Pre-processing:
Tokenization, sentence splitting, part-of-speech tagging, etc.

Prototype alignment:
Find high-scoring matches of prototypes from corpus.

Time expression and entity identification: Extract and ground times and entity mentions.


NSF #1541025
Example outputs for a single owner occupied household.

Outputs can be represented visually or as a narrative.

"Selena owns and lives in a 2 bedroom single family dwelling at 1001 Other Ave worth $150,000. Selena's single family dwelling was inspected 10 days after the event and suffered $33,000 of damage. Selena submitted a request to FEMA 40 days after the event. Selena received $27,000 from FEMA 90 days after the event. Selena received an engineering assessment 95 days after the event. Selena received permit approval 135 days after the event. Selena's single family dwelling was repaired 180 days after the event, taking 9 days to repair. Selena reoccupied the single family dwelling 200 days after the event."
Different scenarios may create different outputs (e.g., if human resource parameters are varied).
Example outputs for multiple owner occupied households. Outputs can be represented visually or as a narrative.

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.
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)
A discrete event disaster recovery simulation built on top of the Simpy discrete event simulation Python library.

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Latest commit</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>.ipynb_checkpoints</td>
<td>d934037</td>
<td>4 days ago</td>
</tr>
<tr>
<td>.gitignore</td>
<td></td>
<td>a year ago</td>
</tr>
<tr>
<td>LICENSE</td>
<td></td>
<td>3 months ago</td>
</tr>
<tr>
<td>images</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td>desaster</td>
<td></td>
<td>4 days ago</td>
</tr>
<tr>
<td>scenarios</td>
<td></td>
<td>a month ago</td>
</tr>
<tr>
<td>inputs</td>
<td></td>
<td>a month ago</td>
</tr>
<tr>
<td>.py</td>
<td></td>
<td>a year ago</td>
</tr>
<tr>
<td>technical.RepairProgram()</td>
<td></td>
<td></td>
</tr>
<tr>
<td>README</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.py</td>
<td></td>
<td></td>
</tr>
<tr>
<td>technical.RepairProgram()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

229 commits | 4 branches | 2 releases | 2 contributors | GPL-3.0

Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download

milessb Deleted default scale from technical.RepairProgram()
Human-Centered Recovery Modeling for Analytic-Deliberative Seismic Resilience Planning

Scott Miles
milessb@uw.edu

National Science Foundation
CMMI IMEE Grant #1560939