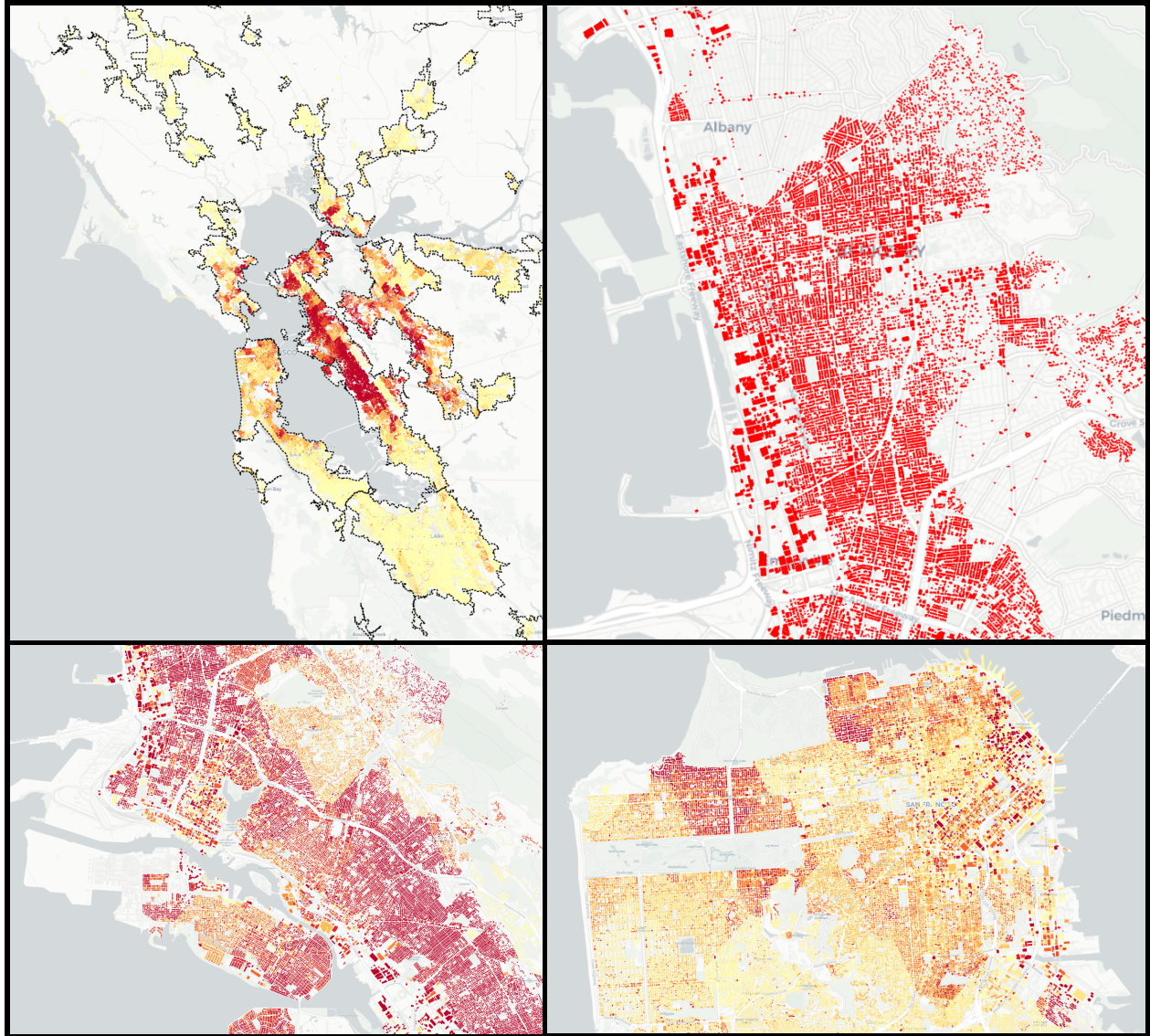# An Application Framework for Regional Earthquake Simulations

Version 1.0, June 2017

NHERI SimCenter

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

BIM	Building Information Model
CSV	Comma Separated Values
DL	Damage and Loss
EDP	Engineering Demand Parameter
ERF	Earthquake Rupture Forecast
GIS	Geographic Information System
GM	Ground Motion
HPC	High Performance Computer
IM	Intensity Measure
JSON	JavaScript Object Notation
MDOF	Multi-Degree of Freedom
PEER	Pacific Earthquake Engineering Research Center
SAM	Structural Analysis Model
SHA	Seismic Hazard Analysis
UQ	Uncertainty Quantification

Introduction

An application framework for creating scientific workflows to study the effects of earthquakes on a community at the regional scale was developed at the NHERI SimCenter. Two workflow applications were developed using this framework, one for running the application at the regional scale utilizing the resources of a high performance computer (HPC) and the other for testing the running of the application on a smaller scale on a local personal computer. Both applications take the same input file that describes the workflow applications and their specific data. The purpose of this document is to describe the framework and the applications developed with the purpose of detailing the workflow application inputs, the inputs for the individual applications in the workflows themselves, and how the user can extend the workflow applications by introducing new applications that meet the interfaces or new datasets. In addition, this document reports the assumptions and limitations, if any, of the current release of the workflow applications. To demonstrate the applications, a testbed example of a 7.0 magnitude earthquake on the Hayward fault will be presented. The results will be compared with the HayWired Scenario study [1].

In computer programming, an application framework is a collection of software for building applications in a specific domain. The framework defines the interfaces between the components of the software, provides example applications that can be developed using the provided software, and defines a clear set of interfaces such that the software can be extended to build other applications (Figure 1). On the other hand, a scientific workflow is the automation of a process in which information is passed from one application to the next to carry out actions according to a set of defined interfaces.

Figure 1: A Schematic Diagram Showing Interfaces Between Software Components Defined by an Application Framework

The next few sections will detail the application framework defined by the SimCenter for regional simulation workflows and will outline the interfaces between the different applications involved in the workflow. The interfaces defined as part of this framework, should facilitate replacing one or more application of the workflow with another that uses a different implementation or a different set of data. This release of the framework contains a basic set of console applications that can be used to carry out regional simulations of an earthquake scenario to estimate damages and losses experienced by buildings across a region (Figure 2). The framework can be used by researcher in the natural hazards field. This document explains the applications provided and how they can be used together to study regional earthquake effects. The workflow applications are under continuous development, thus contributions and

extensions from the hazard mitigation research community are highly encouraged. While, the current workflow considers an earthquake scenario, future developments will include other types of hazards (e.g. Hurricanes, Strong winds, Tsunamis...etc.).

The following is summary of the capabilities that are provided by the current workflow release:

1. Using a user-defined buildings dataset for a region of interest.
2. Using a user-defined earthquake scenario.
3. Customized mapping between building occupancy and structural types.
4. Modular design that allows users to include their own implementations, applications and/or data.
5. Ability to control input uncertainties, either by defining new sources of uncertainties or changing the parameters of the random variables included in the current base workflow provided in this release.
6. User customizable normative quantities for buildings damage and loss estimation.
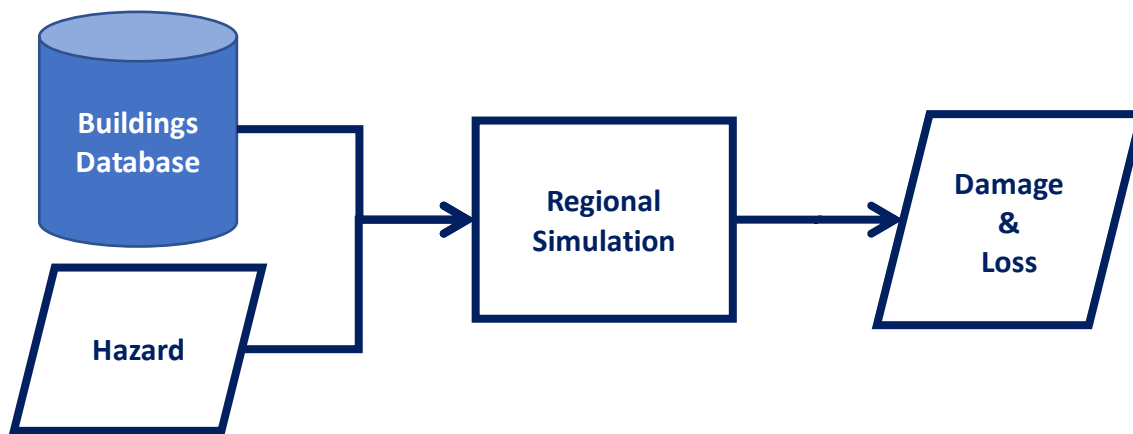
Figure 2: Regional Simulation

As an example, Table 1 compares two different workflow configurations that are provided with the current release of the framework, the first configuration uses ground motions provided by a 3D wave propagation simulation, while the second uses a user-defined earthquake scenario input that relies on seismic hazard analysis (SHA).

Table 1: Example Workflow Configurations Using Different Hazard Input Implementations

Workflow	Configuration

Workflow 1	```
	{
	    "Name": "Workflow 1",
	    "Author": "fmk",
	    "WorkflowType": "Regional Simulation",
	    "buildingFile":"buildings.json",
	    "Applications": {
	       "Buildings": {...}
	       },
	       "Events": [
	           {
	``

	``json
	                    "EventClassification": "Earthquake",
	                    "EventApplication": "LLNL-SW4",
	                    "ApplicationData": {
	                        "pathSW4results": "../NHERI/Hayward7.0/",
	                       "filenameHFmeta":"../createEVENT/HFmeta"
	                    }
	                }
	            ],
	            "Modeling": {...},
	            "EDP": {...},
	            "Simulation": {...},
	            "UQ-Simulation": {...},
	            "Damage&Loss": {...}
	            }
	        }
	    }
	``

Workflow 2	```json
	{
	    "Name": "Workflow 2",
	    "Author": "fmk",
	    "WorkflowType": "Regional Simulation",
	    "buildingFile":"buildings.json",
	    "Applications": {
	        "Buildings": {...},
	        "Events": [
	            {
	                "EventClassification": "Earthquake",
	                "EventApplication": "SHA-GM",
	                "ApplicationData": {
	                    "scenarioConfig": "./Hayward7.25.json"
	                }
	            }
	        ],
	        "Modeling": {...},
	        "EDP": {...},
	        "Simulation": {...},
	        "UQ-Simulation": {...},
	        "Damage&Loss": {...}
	    }
	}
	``

It has to be noted that the workflow is intended to be used on a compute cluster or high performance computer (HPC) to distribute the computations required to carry out a regional simulation. However, it is also possible to run the workflow locally for a small set of buildings.

Workflow Applications Overview

The regional simulation workflow is a set of applications that together can estimate building damage and loss based on BIM and hazard information. The workflow goes through the following five main processes to obtain damage and loss estimates for each building:

1. Obtain the Building Information Model (BIM).
2. Obtain the hazard event input on the building.
3. Produce a Structural Analysis Model (SAM) of the building.
4. Define and calculate the Engineering Demand Parameters (EDP) for the building model.
5. Estimate the damage and loss.

In addition to these main processes, the following five intermediate files are defined and are used in the workflow as intermediate interfaces between the processes:

1. BIM file: A file containing general information about the building.
2. EVENT file: A file containing information about the hazard events input for the building.
3. SAM file: A description of the structural analysis model including geometry, structural and material properties.
4. EDP file: a file containing the definition and values of the engineering demand parameters.
5. DL file: a file containing the obtained damage and loss estimates for a building.
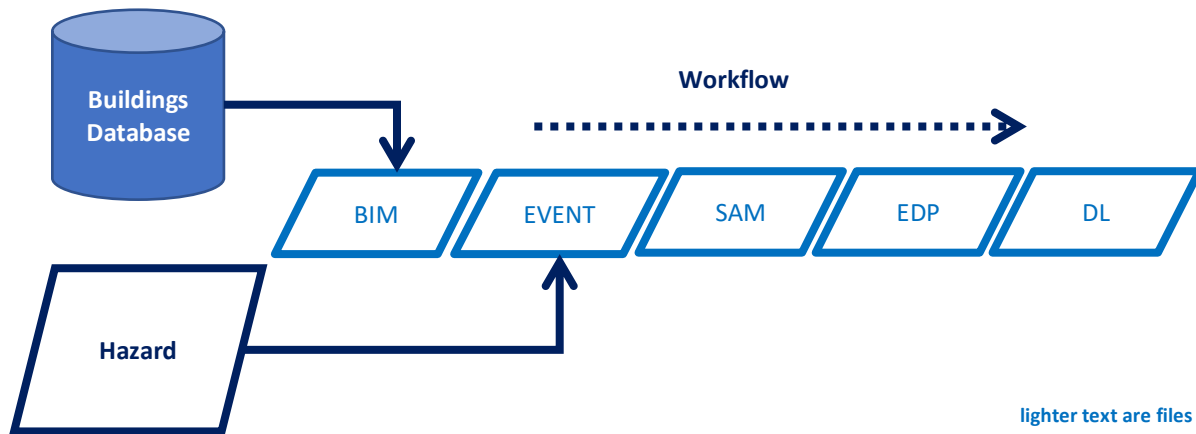
Figure 3: Regional Simulation Workflow Overview – Interface files

Applications Workflow

lighter text are files
darker text are applications
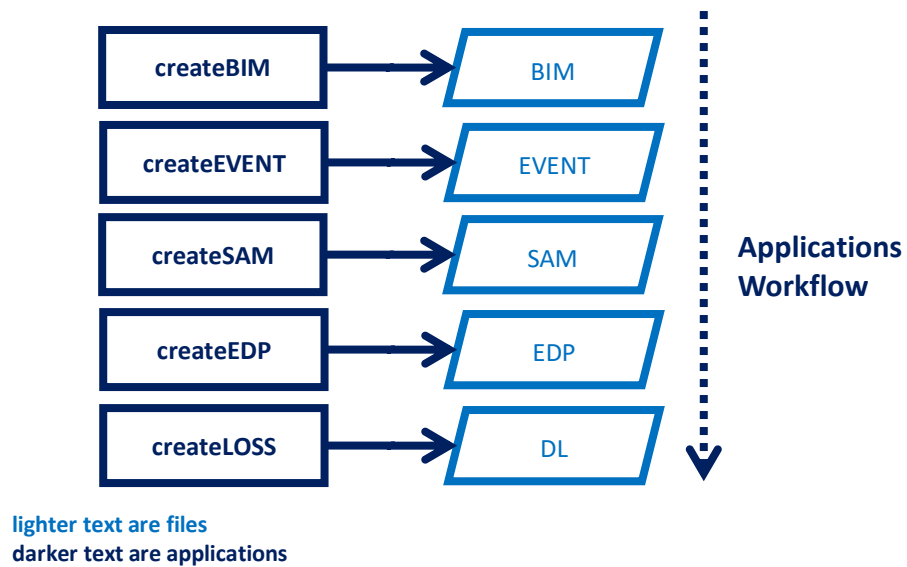
Figure 4: Main Workflow Applications Abstraction (Not showing inputs)

In this framework, the JSON format is adopted for all the intermediate files defined above. Based on the previous 5 main processes, the main workflow applications can be categorized based on their purpose as follows:

createBIM

These are applications that can obtain a Building Information Model (BIM) from a database of buildings (Figure 22). Some example implementation of these applications that are provided with the current release of the workflow are:

- **UrbanSimDatabase**: This application generates BIM models given UrbanSim buildings and parcels csv files.

- **GenericBimDatabase**: This application generates BIM models given a generic database in csv format with a minimal set of information about each building.

UrbanSimDatabase
GemericBimDatabase

lighter text are files
darker text are applications
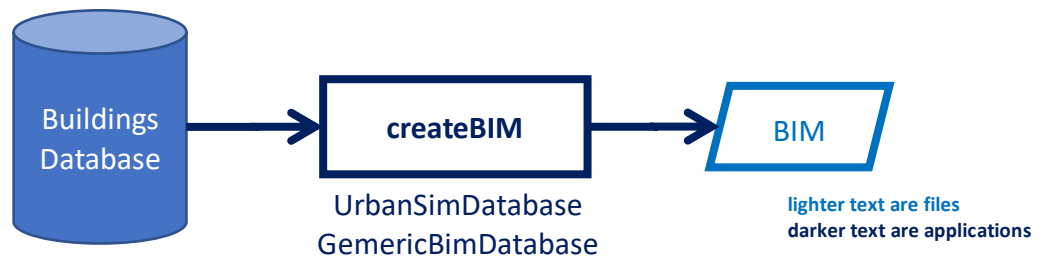
Figure 5: createBIM application

createEVENT

These are applications that can generate a hazard event input (e.g. ground motion), given BIM (Figure 6). The following implementations of createEVENT are provided with the current workflow release:

- **LLNL-SW4**: This application will read the building location from the BIM model and will output the ground motion for simulating building response. The application uses a set of precomputed ground motions for SF Bay Area earthquake event (Magnitude 7.0 earthquake at the Hayward fault) simulated using SW4 [2].

- **SHA-GM.py**: This application combines seismic hazard analysis (SHA) and ground motion record selection/scaling to provide ground motions for structural analysis based on a user-specified earthquake rupture, ground motion prediction equation and ground motion records database.

```
BIM  →  createEVENT  →  EVENT
           LLNL_SW4
           SHA-GM.py
                          lighter text are files
                          darker text are applications
```

Figure 6: createEVENT application

createSAM

These are application that can create a structural analysis model (SAM) given BIM and hazard event inputs (Figure 7).

- **MDOF_LU**: This application uses data from Hazus to generate a MDOF non-linear shear building model based on the structure type, area and year-built parameters from BIM.

- **AI-M-3** [Under Development]: This is an application that is currently under development and uses artificial intelligence to obtain SAM from BIM.

```
BIM  →  createSAM  →  SAM
          MDOF_LU
                        lighter text are files
                        darker text are applications
```

Figure 7: createSAM application

createEDP

These are applications that generate a list of the required Engineering Demand Parameters (EDPs) to be obtained from structural analysis (Figure 8).

▪ **StandardEarthquakeEDP**: This application uses the SAM to define the EDPs needed for damage and loss calculations due to an earthquake event.
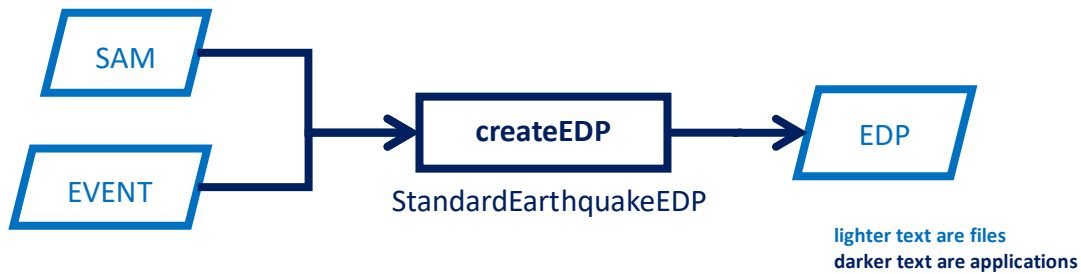
Figure 8: createEDP application

performSIMULATION

These are applications to perform pre-processing of SAM, structural analysis and post-processing to obtain EDP values (Figure 10).

▪ **OpenSeesSimulation.sh**: This application will pre-process the SAM file to create an OpenSees model, run the model and post-process OpenSees outputs to update the EDP values.
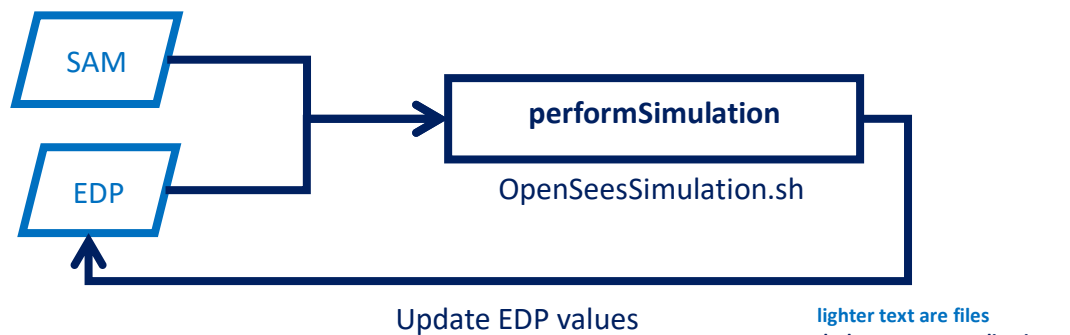
Figure 9: performSimulation application

createLOSS

These are applications that can determine damage and loss experienced by a building given BIM and EDPs (Figure 10).

▪ **FEMA_P58_LU**: Damage and loss assessment tool that uses FEMA-P58 methodology. This tool was developed by Prof. Xinzheng Lu's research group at Tsinghua University [?].

▪ **OpenPerform** [Under development]: An efficient object-oriented damage and loss assessment tool using FEMA-P58 methodology is currently under development in the SimCenter.
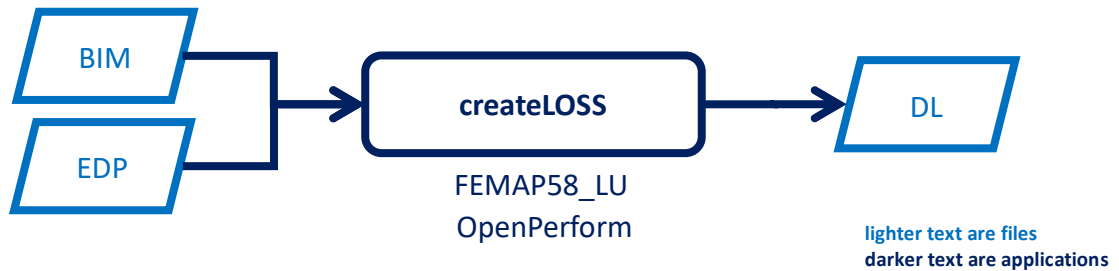
Figure 10: createLOSS application

performUQ

These are applications or scripts that handles the uncertainties considered by the different workflow applications (Figure 12).

▪ **Dakota-FEM-Simulation.sh**: This implementation of performUQ relies on DAKOTA for handling uncertainties.

Workflow

These are scripts that can be used to configure and run the regional simulation workflow both locally or on HPC using Pegasus (Figure 11).

▪ **RegionalEarthquakeSimulation.py**: This python script can be used to run the workflow applications for a range of buildings in a region given an earthquake scenario. The workflow applications to run can be configured by input files in JSON format. One input file (WorkflowApplications.json) is used to define the available applications that can be used by the workflow, and serves as applications registry and another file (Worfklow.json) is used to configure applications and the data they use for a single workflow run.
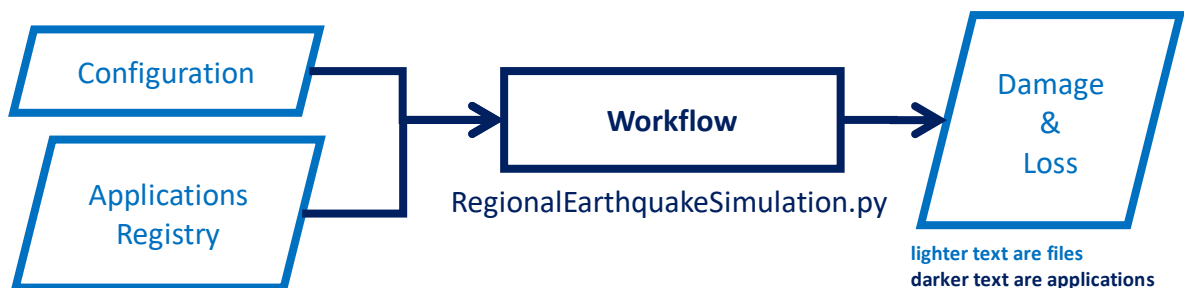
Figure 11: Workflow Application
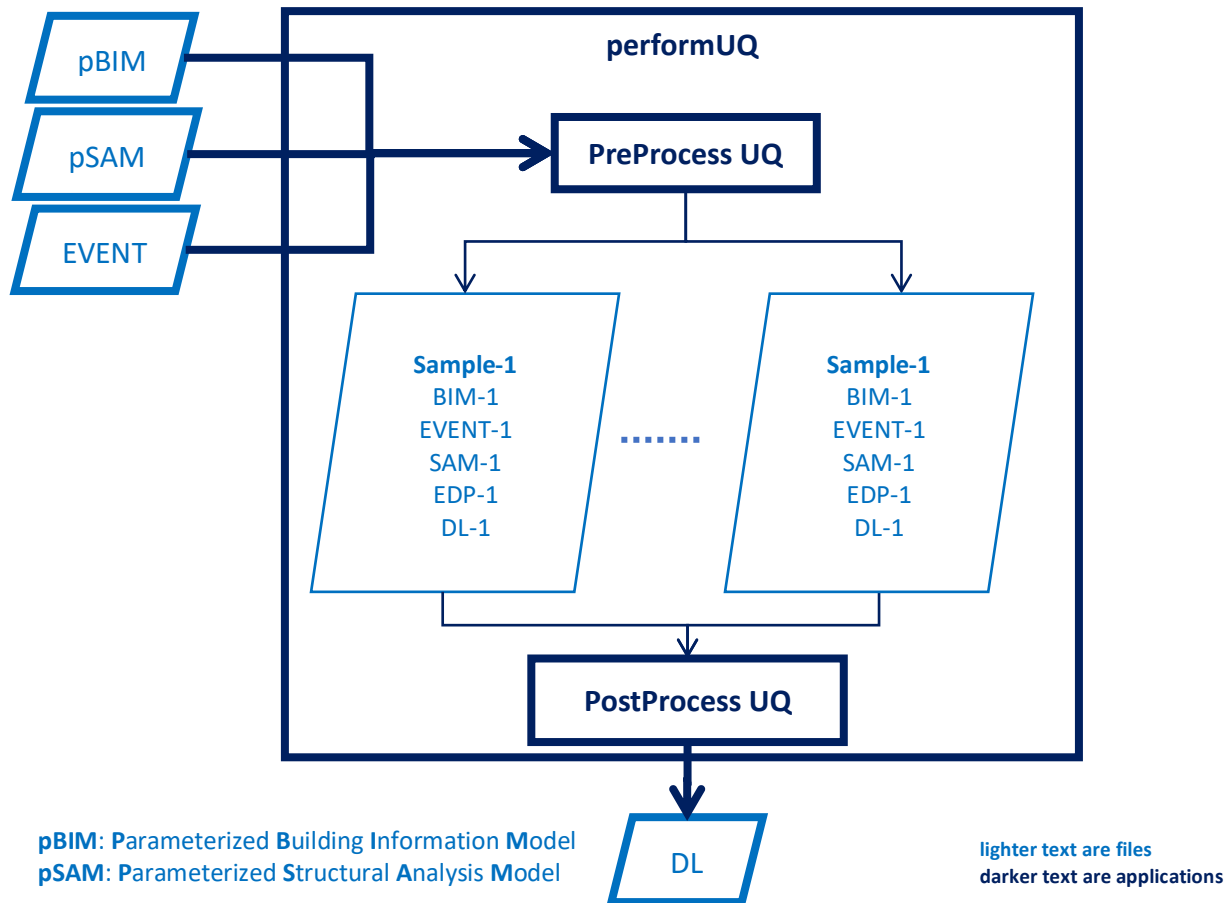
Figure 12: performUQ applications

Implementations

In this section, the different implementations provided by the SimCenter in this release for each of the workflow applications will be presented. Assumptions and limitations of each implementation will be discussed, and proposed improvements for future versions will be summarized as well. In addition, examples will be given to show how each application can be used.

createBIM Implementations

A createBIM application is a tool that reads building information from a database and creates a BIM file in JSON format. The application should be able to create BIM files for a single building or a range of buildings in a given region. Within the source code directory, the createBIM implementations are available in the createBIM directory.

An example BIM file obtained by a createBIM application will contain the following information:

```
{
    "RandomVariables": [
        {
            "mean": 3,
            "distribution": "normal",
            "stdDev": 0.16667000000000001,
            "name": "height",
            "value": "RV.height"
        }
    ],
    "GI": {
        "area": 288.58525817400027,
        "name": "7",
        "structType": "W1",
        "numStory": 1,
        "yearBuilt": 2008,
        "occupancy": "Residential",
        "height": "RV.height",
        "location": {
            "latitude": 37.462235730000003,
            "longitude": -121.9172232
        },
        "replacementCost": 640891.526331,
        "replacementTime": 180
    }
}
```

Figure 13: BIM file example

In this BIM JSON file, a random variable section is included where the height of the building is defined as a normally distributed random variable with mean 3.0 m and standard deviation 1/6 m. It has to be noted that SI Units are used in this file. The BIM file also contains a JSON element with the name `GI`, which stands for general information. In this section, relevant building information is provided. In this case, the area is provided in m², the number of stories, the year this building was built, its occupancy type, the location of the building, its replacement cost and replacement time.

UrbanSimDatabase

In collaboration with UrbanSIM, the SimCenter is providing a buildings dataset for SF Bay Area (1.8 million buildings). Sample files for 100 building are distributed with the repository. UrbanSim database includes two files:

▪ **Buildings File**: This is a comma-separated values (CSV) file that contains building information such as area, number of stories, year built, building occupancy and parcel id. Areas in this file are in square feet and is the total area of the building (i.e. all floors). Both the building id and the parcel id in this file are integer values.

- **Parcels File**: This CSV file provides information about the parcels containing the buildings, including the geographic location (latitude and longitude). The parcel id defined in this file is an integer.

Figure 14: UrbanSimDatabase Application

The application reads the building information from the buildings file and uses the parcel id to obtain the parcel location from the parcel file. In UrbanSim buildings file, the occupancy is represented using an integer, building occupancy and replacement costs are mapped to the occupancy id according to Table 2. The replacement time using this application is always assumed to be 180 days. It is also important to point out that for buildings whose are is missing or invalid (less than or equal to zero), the area is set to the average area of buildings.

Table 2: Mapping building type IDs in UrbanSimDatabase to building occupancy and replacement costs

ID	OCCUPANCY	REPLACEMENT COST (PER UNIT AREA IN SQFT)	CONTENTS FACTOR
1-3,12	Residential	137.5452	0.5
4,14	Office	131.8863	1.0
5	Hotel	137.271225	0.5
6	School	142.134265	1.25
7	Industrial	97.5247	1.5
8	Industrial	85.9586	1.5
9	Industrial	104.033475	1.5
10,11,13	Retail	105.33705	1.0
DEFAULT	Residential	137.5452	0.5

This application also map the building occupancy to structure types based on mapping shown in. Unless the building is mapped to a single structure type, the structure type is considered random, in which case all the mapped structure types are equally likely.

Table 3: Mapping building occupancy in UrbanSimDatabase to structure types

YEAR BUILT	STORIES	OCCUPANCY	MAPPED STRUCTURE TYPES
OLDER THAN 1900	ANY	ANY	Timber (W1) and Masonry (RM1, RM2, URM)
NEWER THAN 1900	1-3	Residential	Timber (W1)
		Commercial	Timber, Steel, Concrete and Reinforced Masonary (W1, S1, S2, C1, C2, C3, RM1, RM2)
		Industrial	Steel, Concrete and Reinforced Masonary (S1, S2, C1, C2, C3, RM1, RM2)
		Other	Timber (W1)
	4-7	Residential & Commercial	Timber, Steel, Concrete (W1, S1, S2, C1, C2)
		Industrial	Steel, Concrete (S1, S2, C1, C2)
		Other	Timber, Steel, Concrete (W1, S1, S2, C1, C2)
	More than 7		Steel and Concrete (S1, S2, C1, C2)

The mapping from building occupancy to structure types can be configured using a JSON config file (BIM Config). BIM Config can also be used to change he replacement costs and default area for buildings with missing or invalid area. An example configuration is shown in Figure 15.

```json
{
    "StructTypes": {…},
    "StructTypeIdMappings":
    [{"YearRange": {
                "Min": 1900
        },
            "StoryRangeMaps": [{
                "StoryRange": {
                    "Min": 1,
                    "Max": 3
                },
                "BuildingIdMaps": [{
                        "BuildingIds": [1, 2, 3, 12],
                        "StructureTypeId": 1,
                        "Description": "Residential"
                }, {

                        "BuildingIds": [4, 5, 6, 10, 11, 13, 14],
                        "StructureTypeId": {
                            "Type": "Random_Discrete_Uniform",
                            "Values": [1, 2, 3, 4, 5, 6, 7, 8]
                        },
                        "Description": "Commercial"
                }, {
                        "BuildingIds": [7, 8, 9],
                        "StructureTypeId": {
                            "Type": "Random_Discrete_Uniform",
                            "Values": [2, 3, 4, 5, 6, 7, 8]
                        },
                        "Description": "Commercial"
                }
            ],
…

}
```

Figure 15: BIM Config file example

GenericBimDatabase

This is a generic BIM application that only requires the minimal set of information needed for each building. The application uses as its input a single CSV file (Figure 16) that contains building Id, area, number of stories, year built, occupancy and location. Sample GenericDatabase.csv file is distributed with this repository. It is important to note that the Id is any valid string (comma is not allowed) and the area in in square feet and is the total area of the building (i.e. all floors).

Figure 16: Generic BIM Database Application

Except for the simplified format of inputs (Figure 17), the implementation of this application is similar to UrbanSimDatabase, that includes mapping occupancy, replacement costs, structure types, ...etc.


```
ID,Area,Stories,Year Built,Type ID,Latitude,Longitude
1,2029.42425042,1,1945,1,37.98999094,-122.5986709
2,2029.42425042,1,1965,1,37.97476531,-122.6056875
3,1568.0,1,1964,1,37.34473211,-122.0014662
```

Figure 17: Example Generic BIM Database File in CSV format

Examples

To use UrbanSimDatabase to generate the first 100 BIM files from UrbanSim database sample, you can use the following command in the createBIM directory:

```
./UrbanSimDatabase  buildings.json  -Min  1  -Max  100  -buildingsFile
buildingsSample.csv -parcelsFile parcelsSample.csv -getRV
```

GenericBIMDatabase application can be called similarly using this command:

```
./GenericBimDatabase  buildings.json  -Min  1  -Max  100  -buildingsFile
GenericBimDatabase.csv -getRV
```

The result of invoking these commands will be a buildings.json file listing all the BIM models that were generated, and 100 BIM files starting with 1-BIM.json, 2-BIM.json up to 100-BIM.json. Each of these files will include the building information, in addition to random variables definition. The random variables will include the height of the building and the structure type, if applicable.

createEVENT Implementations

This application takes as its input the BIM file and outputs the hazard event input for the given building. For an earthquake event, the output will be ground motion at the building's location. The current release of the workflow provides the following two implementations of createEVENT.

LLNL_SW4

This implementation of createEVENT uses the ground motions produced by SW4 [2] simulation and provides it as an input for the analysis of buildings. SW4 is a software developed at Lawrence Livermore

National Laboratory. With the help of the Computational Infrastructure for Geodynamics (CIG) research group at LLNL, the SimCenter is providing the ground motions data for a M7.0 earthquake scenario with a rupture along Hayward fault and hypocenter near Richmond, CA [3]. The LLNL_SW4 application uses the ground motions to create the event data for buildings in the SF Bay Area. Ground motions are computed over a grid of sites as shown in Figure 18. LLNL_SW4 uses k-dimensional tree method implemented by nanoflann C++ library to find the nearest site from the grid to a given building, then the ground motion from that site is used as the event for the building.
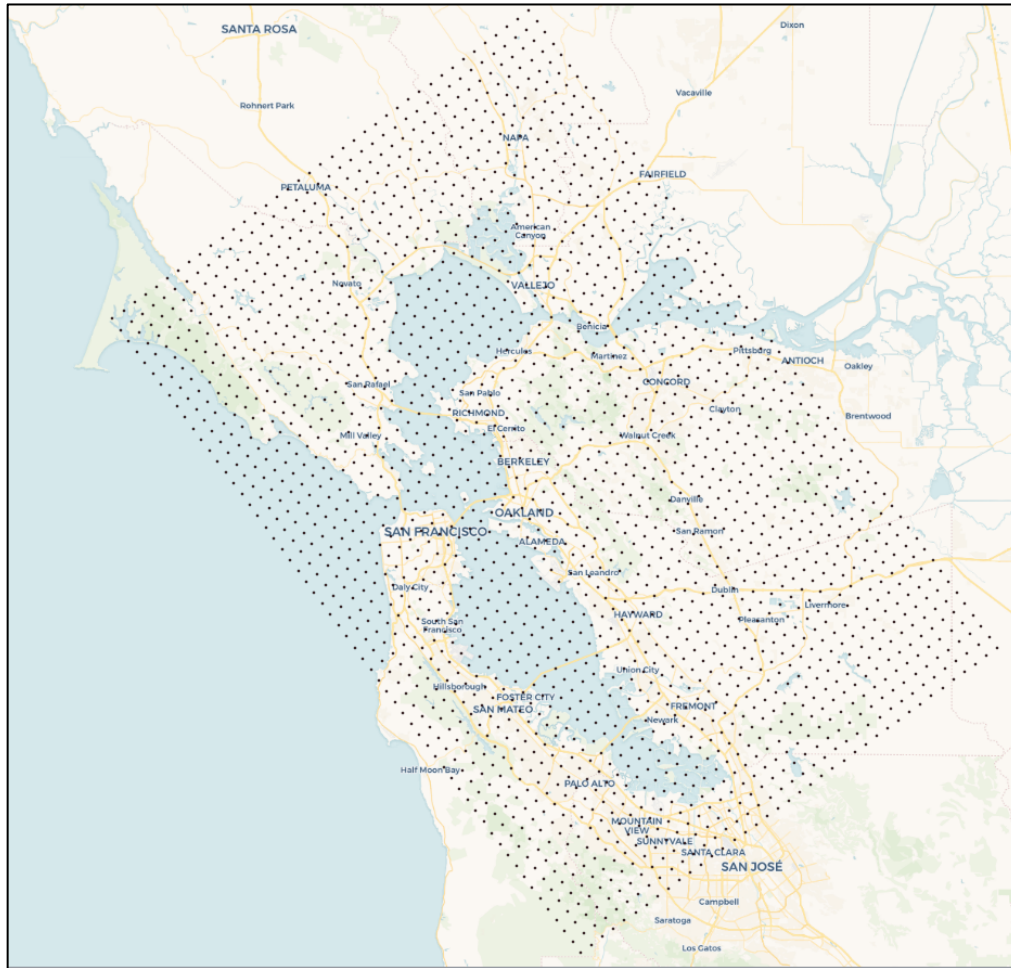
Figure 18: Sites in SF Bay Area with ground motions computed by SW4 for an earthquake scenario in the east bay

SHA-GM.py

This implementation of createEVENT combines seismic hazard analysis with record selection/scaling to provide ground motions over a grid of sites that can be used as event inputs for buildings (Figure 19). Similar to LLNL_SW4, this application uses k-dimensional tree to search for the nearest neighbor amongst the grid sites with ground motions (Figure 20). This application makes use of the ground motion tools described in the following sections to obtain simulated spectra and select ground motions from a database at each grid site. Data provided with the workflow includes samples for records database from PEER NGA-West2 database [4]. Yet, the same tools can be adapted to select records from a user provided database. In the following sections, each of the applications used to obtain ground motions at grid sites is explained.
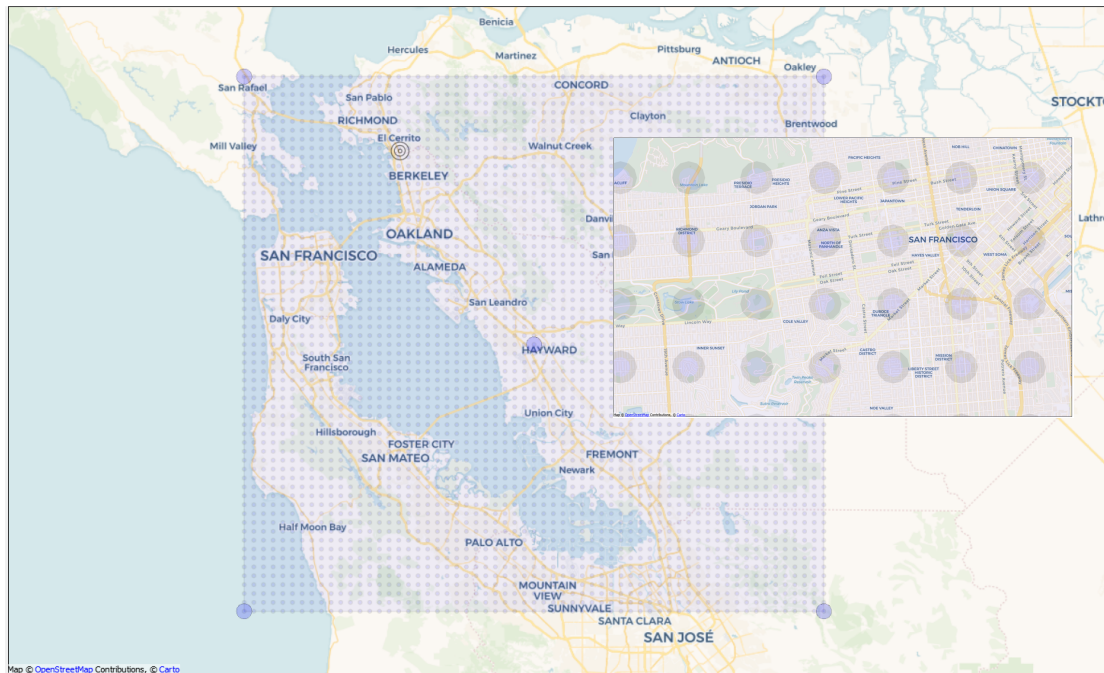
Figure 19: Sites in SF Bay Area with ground motions computed using the SimCenter ground motion applications for an earthquake scenario in the east bay
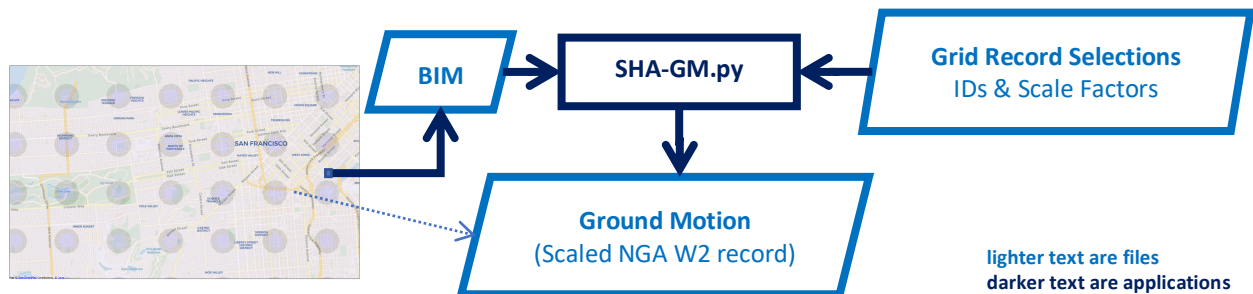
Figure 20: SHA-GM implementation, using scaled and selected record at nearest site from a site grid

Ground Motion Applications

The ground motion applications are a suite of console applications that are developed by the SimCenter for characterizing and representing the ground motions due to an earthquake for both site-specific and regional simulations. Initially, these applications will provide ground motion inputs to be used by the regional earthquake simulation, given an earthquake scenario. However, these are standalone tools that can also be used for different research applications.

In the context of the regional simulation workflow, the applications will be used to provide seismic event input for time history analysis of structural models given an earthquake scenario (e.g. a magnitude 7.0 earthquake on the Hayward fault). The three tools perform deterministic seismic hazard analysis, intensity

measures(IM) simulation, and record selection/scaling. Each tool is described in the following sections. Features and use cases for each application are provided below as well.

EQHazard

EQHazard is a hazard analysis application written in Java and is a wrapper around OpenSHA [5]. This application can be used to obtain ground motion intensity measures(IM), such as peak ground acceleration (PGA) or spectral accelerations (SA), at a specific site or a grid of sites, assuming a scenario earthquake rupture occurring (deterministic hazard analysis). The application allows computing the intensity measure using different ground motion prediction equations (GMPE) provided by OpenSHA. This application takes the following as its input (Figure 21):

1. **Earthquake Rupture**: either a point source with magnitude, or a rupture defined in one of the Earthquake Rupture Forecasts(ERF) provided by OpenSHA.
2. **Site**: a single location or a grid of locations.
3. **GMPE**: the attenuation relationship used to estimate the IM mean and standard deviation.

The outputs of the tool are:

1. **IM**: Mean and standard deviation of the intensity measures at the sites specified
2. **Site Data**: data available for the site that was obtained during the analysis such as Vs30, Depth 1.0 km/s and Depth 2.5km/s
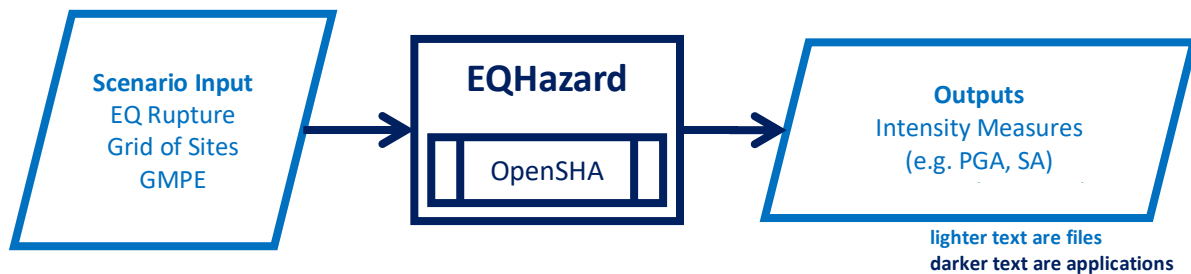
Figure 21: EQHazard Application

At its core this tool uses OpenSHA, which provide the implementation of the GMPEs and data for the ERFs, where an EQ scenario can be selected. In addition, Site data is available in OpenSHA through different providers.

The following GMPE implementations provided by OpenSHA can be used:

1. Abrahamson, Silva & Kamai (2014)
2. Boore, Stewart, Seyhan & Atkinson (2014)
3. Campbell & Bozorgnia (2014)
4. Chiou & Youngs (2014)
5. Idriss (2014)
6. Campbell & Bozorgnia (2008)
7. Boore & Atkinson (2008)
8. Abrahamson & Silva (2008)

9. Chiou & Youngs (2008)
10. Boore & Atkinson (2006)
11. Chiou & Youngs (2006)
12. Campbell & Bozorgnia (2006)
13. Campbell and Bozorgnia (2003)
14. Field (2000)
15. Campbell (1997) w/ erratum (2000) changes
16. Abrahamson & Silva (1997)
17. Boore, Joyner & Fumal (1997)

An earthquake rupture can be defined as a point source, using magnitude, location, dip and rake. In addition, the application allows using one of the earthquake fault rupture (ERF) models available through OpenSHA to specify an earthquake source and rupture. The available ERFs include:

1. WGCEP (2007) UCERF2 - Single Branch
2. USGS/CGS 2002 Adj. Cal. ERF
3. WGCEP UCERF 1.0 (2005)
4. GEM1 CEUS ERF
5. Mean UCERF3
6. Mean UCERF3 FM3.1
7. Mean UCERF3 FM3.2

Additionally, the site providers accessible through OpenSHA are used to obtain site data as show in Table 4.

Table 4: Site Data Providers in OpenSHA

Site Data Provider	Site Data Types
CGS/Wills VS30 Map (2015)	Vs30 and Vs30 measurement type
Global Vs30 from Topographic Slope (Wald & Allen 2008)	Vs30 and Vs30 measurement type
SCEC Community Velocity Model Version 4, Iteration 26, Basin Depth	Depth to Vs = 2.5 km/sec and Depth to Vs = 1.0 km/sec
SCEC CCA, Iteration 6, Basin Depth	Depth to Vs = 2.5 km/sec and Depth to Vs = 1.0 km/sec
SCEC Community Velocity Model Version 4 Basin Depth	Depth to Vs = 2.5 km/sec and Depth to Vs = 1.0 km/sec
SCEC/Harvard Community Velocity Model Version 11.9.0 Basin Depth	Depth to Vs = 2.5 km/sec and Depth to Vs = 1.0 km/sec
SCEC CCA, Iteration 6, Basin Depth	Depth to Vs = 2.5 km/sec and Depth to Vs = 1.0 km/sec
USGS Bay Area Velocity Model Release 8.3.0	Depth to Vs = 2.5 km/sec and Depth to Vs = 1.0 km/sec
SCEC Community Velocity Model Version 4, Iteration 26	Vs30 and Vs30 measurement type
SCEC Community Velocity Model Version 2 Basin Depth	Depth to Vs = 2.5 km/sec
CGS/Wills Preliminary Site Classification Map (2000)	Wills Site Class

It has to be noted that this is an ordered list of the providers according to preference, which is the default setting in OpenSHA library. If no site data is available, then the GMPE default values are used.

The inputs for the EQScenario tool is in json format, and the outputs can be in json format or comma separated values(csv) format. Optionally, outputs in GeoJson format can be provided as well to facilitate visualization in GIS application (Figure 22).
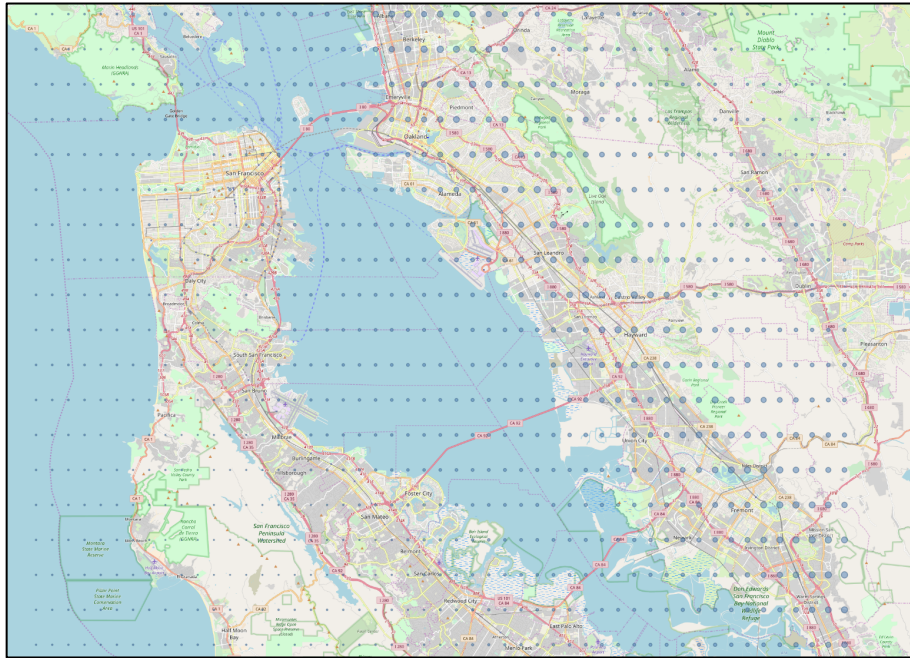

Figure 22: Sample output of EQScenario, visualized using QGIS, showing the Mean PGA over a grid in the San Francisco Bay area, given a magnitude 7.25 earthquake occurring in Hayward fault

SimulateIM

This application will simulate ground motion IMs given the mean and standard deviation. For simulation of geographically distributed systems, spatial correlation can be established among the simulated IMs for different sites across a region, using the method outlined in [6].

One application of this tool is to obtain simulations of spectral accelerations at different sites across a region to use it for ground motion selection and scaling. The inputs for this tool, such as mean and standard deviations of SA can be obtained using EQScenario tool. For the regional simulations, the tool will be used to obtain a single simulation of the spectral accelerations at each site, to be used to select a single ground motion record (Figure 23). The current implementation of SimulateIM only includes spatial correlation and does not include IMs correlation.

SelectRecord

This application can be used to select and scale ground motion records from a database of existing records (e.g. NGA-West2). The application will take as an input a target spectrum, a database of records, an error metric and a short-listing criterion. The output would be either a single record or a suite of records.

Initial version will use NGA-West flat file as a database to carry out selection/scaling. Later versions may include more record sources and automatic retrieval of the records. For the regional earthquake simulation, this application will select a single record to match a single simulated spectrum (Figure 23).

The output of this application is the IDs of the selected records, scaling factors and error values for each site.
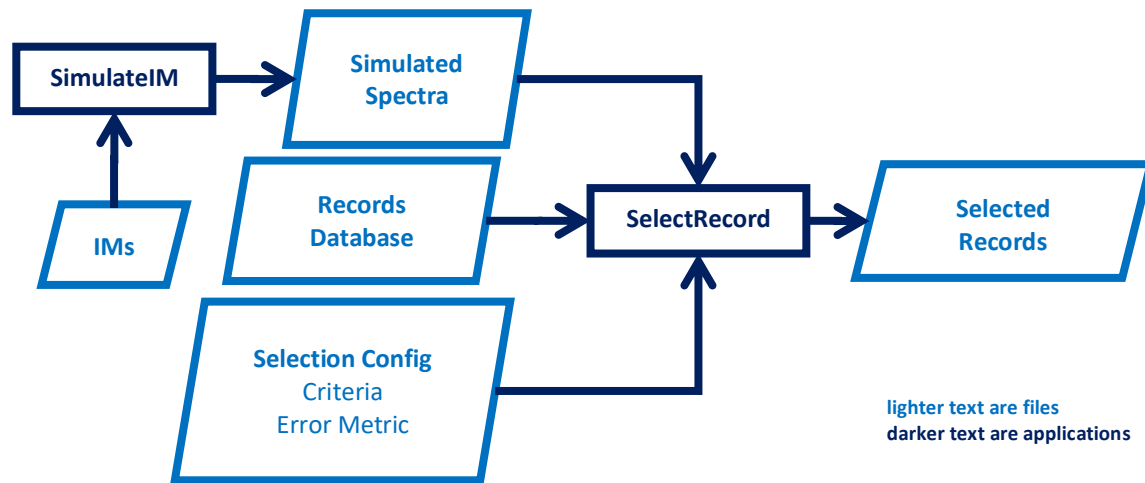
Figure 23: SimulateIM and SelectRecord Applications

Optionally, the application can produce outputs in GeoJson format to facilitate visualization in GIS applications. The following figure shows the IDs of selected records using QGIS plotted for a grid over the SF bay area.
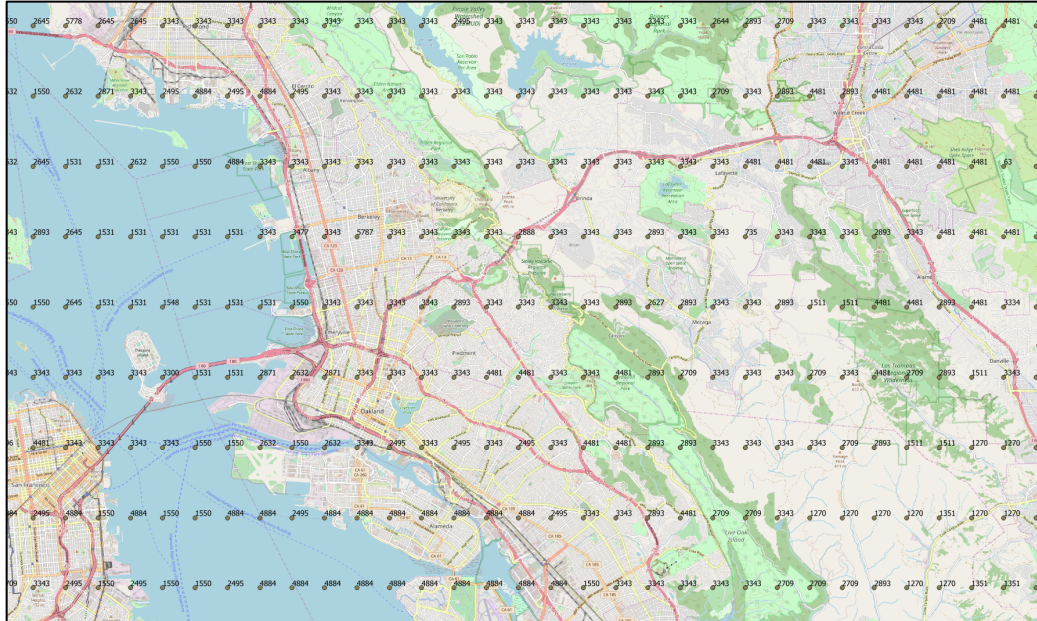
Figure 24: GeoJson output visualized in QGIS, showing record sequence numbers selected from NGA-West2 database using SelectGM

createSAM Implementations

A createSAM application reads a BIM file and produces a SAM file. SAM files includes the geometry and structural properties. Geometric information includes the nodal coordinates and elements connectivity.

Structural properties include material properties such as stiffness and damping. One implementation of createSAM is provided with this release of the workflow and was developed by [?].

MDOF_LU

This application creates a MDOF non-linear shear building model given the BIM by incorporating Hazus [7] data. Readers interested in the detailed method used in this application should refer to [8], the rest of this section provides a summary of the method to obtain the MDOF.

Some of the assumptions used in this application:

1. All floors have the same height, mass and initial stiffness.
2. The building has the same properties in both X and Y directions.
3. The mass per unit area is assumed to be 1000 kg/m².
4. Hazus low level code categorization is not used, instead all buildings older than 1941 are categorized as pre-code.
5. Stiffness and damping factors are treated as random variables with a mean value of 1.0 and a standard deviation of 0.1.
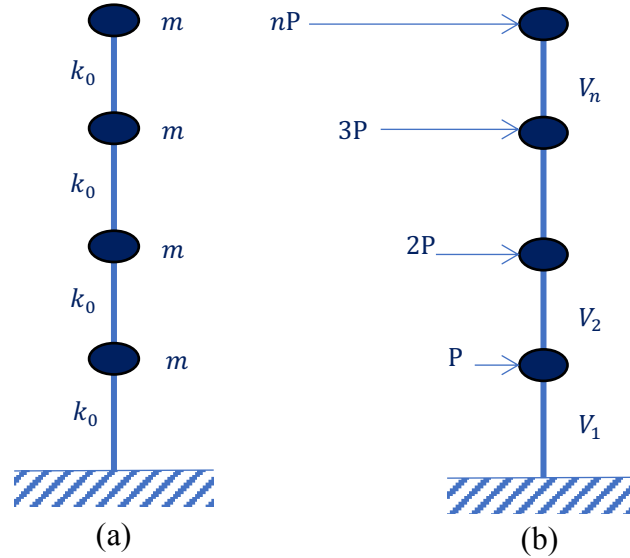
Figure 25: Assumption in the shear building model used by MDOF_LU, (a) masses and initial stiffness values, (b) lateral forces and story shear yield strength

Based on the previous assumptions, the initial stiffness of each building floor can then be obtained using (1).

$$k_0 = \lambda \frac{4\pi^2 m}{T_1} \tag{1}$$

Where m is the floor mass in kg, T_1 is the fundamental period in seconds and is obtained from Hazus technical manual table 5.5, and λ is given by,

$$\lambda = \frac{\{\phi_1\}^T[I]\{\phi_1\}}{\{\phi_1\}^T[A]\{\phi_1\}} \tag{2}$$

And where $\{\phi_1\}$ is the first mode shape of the building model and $[A]$ is a matrix of the form:

$$[A] = \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \tag{3}$$

In this case, λ can also be approximated based on the number of floors n using:

$$\lambda = 0.4053 * n^2 + 0.405 * n + 0.1869 \tag{4}$$

The story yield shear strength is then obtained using the following equation:

$$V_i = V_1 * \left(1 - \frac{i(i-1)}{n(n+1)}\right) \quad and \quad V_1 = A_y * \alpha_1 * w_{\text{Total}} \tag{5}$$

Where V_1 is the yield capacity of the building (base shear), A_y is the yield spectral acceleration of the building (Hazus technical manual Table 5.7 [7]), α_1 is the fraction of building weight effective in push-over mode (Hazus technical manual Table 5.5 [7]), i is an index indicating the floor, w_{Total} is the total weight of the building and n is the number of floors.

A hysteretic nonlinear material model is then defined for each floor with a story-force relationship (Figure 26) defined using the initial stiffness k_0, and an ultimate over-strength ratio α ratio (Hazus technical manual Table 5.5), a hardening ratio η and a degradation factor κ as defined in Hazus technical manual [7] to model degraded unloading stiffness based on ductility.
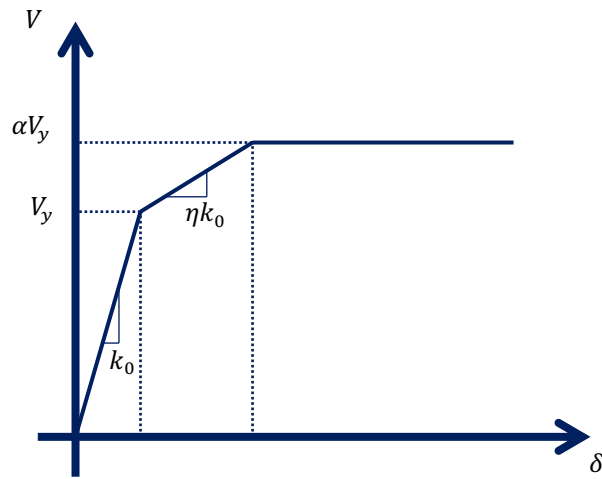

Figure 26: Story force displacement relationship

The hardening ratio η is obtained using the formula:

$$\eta = \frac{A_u - A_y}{D_u - D_y} * \frac{D_y}{A_y} \qquad (6)$$

Where D_y yield displacement capacity of the building, D_u yield displacement capacity of the building, A_y yield spectral acceleration of the building, A_u ultimate spectral acceleration of the building (values provided in Hazus technical manual Table 5.7 [7]).

createEDP Implementations

A createEDP application reads the BIM file, the SAM file and the events file and creates a file that lists the required EDPs for damage and loss calculations. An implementation of createEDP is provided with this release of the workflow that defines the standard EDPs used for loss estimation of a building subjected to an earthquake.

StandardEarthquakeEDP

The StandardEarthquakeEDP application reads the events file and checks if a seismic event is included in the simulation. If a seismic event is included, it then reads the BIM file and uses the number of floors to define the maximum inter-story drifts, the floors absolute accelerations and the residual displacement as the EDPs at corresponding nodes as defined in the SAM file.

performSimulation Implementations

performSimulation applications use the SAM file to create a simulation model (Preprocessor), run the model simulation and process the model outputs to read and update the EDP values defined in the EDP file (PostProcessor). This release of the workflow has an implementation of performSimulation that uses OpenSees [9] as its simulation engine.

OpenSeesSimulation.sh

This script implements the simulation step using OpenSees [9], which is done in three step. First, an OpenSees preprocessor reads the SAM file and creates an input Tcl file to be used as OpenSees input. Second, OpenSees is called to run the model Tcl file. Finally, an OpenSees postprocessor will read the outputs and update the EDP values in the EDP files.
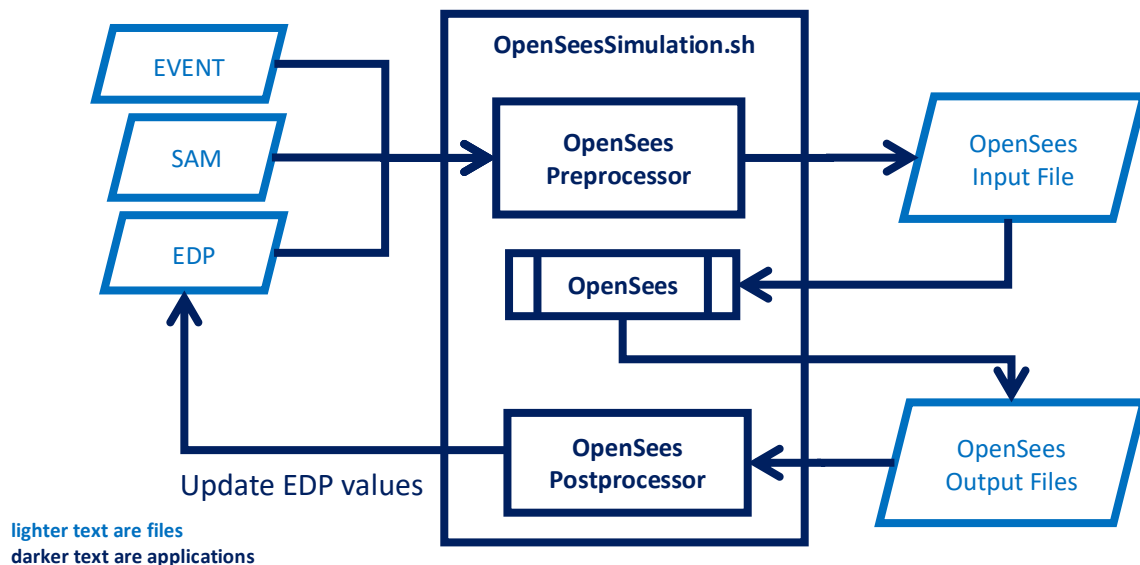
createLOSS Implementations

A createLoss application uses the data in the BIM and EDP file to estimate the building damage and losses. The current implementation distributed with this release of the workflow is FEMAP58_LU [8] and is summarized in the next section.

FEMAP58_LU

This application implements FEMA P58 methodology [10] for building damage and loss assessment. The following are summary of the assumptions used in this application:

- The loss assessment procedure uses 6 structural components (Table 5) and 33 non-structural components () are used in loss assessment.
- Random samples (1000 realizations by default) are generated for the building assigning random values to the quantity of each element or component.
- Uncertainty in the non-structural quantities is based on normative quantities provided by FEMA P-58 .
- Uncertainty in the structural components quantities is based on statistical analysis of buildings in Tsinghua campus in China.
- ATC fragility curves are used to asses the damage.

Table 5: Structural components considered in FEMAP58_LU

ID	DESCRIPTION
B1041.041a	ACI 318 OMF with weak joints and column flexural response
B1044.091	Slender Concrete Wall
B1035.001	Post-Northridge RBS connection with welded web; beam one side of column only
B1035.011	Post-Northridge RBS connection with welded web; beams both sides of column
B1052.011	Special reinforced masonry walls with fully grouted cells, shear dominated
B1071.001	Light framed wood walls with structural panel sheathing

Table 6: Non-Structural components considered in FEMAP58_LU (not all components are listed)

ID	Description
B2022.001	Curtain Walls
B3011.011	Concrete tile roof
C1011.001c	Wall Partition
C2011.001b	Prefabricated steel stair
C3027.001	Raised Access Floor
C3032.001a	Suspended Ceiling
C3034.001	Independent Pendant Lighting
D1014.011	Traction Elevator

D2021.011a	Cold Water Piping
D2022.011a	Hot Water Piping
D3031.021a	Cooling Tower - Capacity: < 100 Ton
D3041.011a	HVAC Galvanized Sheet Metal Ducting
D5092.031a	Diesel generator
F1012.001	Storage racks

User's Guide

This section documents how to obtain the data needed to run the workflow and describes how to configure and run individual applications and the whole workflow for a set of buildings subjected to an earthquake. In addition, the process of adding new or modified applications is explained. Visualization of the results of regional simulations using QGIS is also summarized and presented.

Data

Some data files required to run the workflow are not included in this repository and are distributed using the SimCenter Box account. If you cannot access the data, please contact the SimCenter using Slack or by email.

- **BIM Datasets**: The workflow is distributed with a sample UrbanSim data to run a sample number of buildings, e.g. sample buildings.csv and parcels.csv files for UrbanSimDatabase with 100 buildings. Data for SF Bay Area is available only for the GenericBimDatabase application and can be obtained from the SimCenter Box account.

- **Ground motion**: files for the LLNL Hayward 7.0 scenario are available through the SimCenter Box account.

Running individual applications

In this section, steps to run all the workflow applications individually for one building are listed. This demonstrates how each application can be executed and the command line arguments needed for each application.

1. Creating BIM for the first building in the database can be achieved by changing the working directory to the createBIM directory and running the command:

```
./GenericBimDatabase buildings.json –Min 1 -Max 1 –buildingsFile
GenericBimDatabase.csv -getRV
```

This will generate a BIM for the first building (1-BIM.json). Note that the model has a random variable defined for the height with a mean of 3.0m.

2. Open the 1-BIM file and replace "RV.height" with 3.0. This assigns the mean value to the building height.

3. Change the working directory to createEVENT and run the command:

```
./LLNL_SW4 -filenameBIM ../createBIM/1-BIM.json -filenameEVENT
1-EVENT.json -filenameHFmeta HFmeta -pathSW4results
./Hayward7.0/ -getRV
```

This will create the 1-EVENT.json file. The previous command assumed the Hayward 7.0 ground motion files mentioned in the data section is inside the createEVENT directory in a sub-directory called Hayward7.0, if the ground motion files are in different path, the command line argument will need to be adjusted accordingly.

4. Change the working directory to createSAM and run the command:

```
./MDOF-LU -filenameBIM ../createBIM/1-BIM.json -filenameEVENT
../createEVENT/1-EVENT.json -filenameSAM 1-SAM.json -hazusData
./data/HazusData.txt -getRV
```

This should create a 1-SAM.json file that only has. In this case, the createSAM application cannot create the structural and material properties without values assigned to the random variables for the stiffness and damping factors. Open the 1-SAM.json file and replace both "RV.kFactor" and "RV.dampFactor" with 1.0, which is the mean value for both random varriable.

5. After assigning values for the stiffness and damping factors, now you can run the MDOF-LU application again using the command:

```
./MDOF-LU -filenameBIM ../createBIM/1-BIM.json -filenameEVENT
../createEVENT/1-EVENT.json -filenameSAM 1-SAM.json -hazusData
./data/HazusData.txt
```

This is the same command as in the previous step without the -getRV argument. In this case, the application will use the provided values for the stiffness and damping factors to calculate the structural and material properties of the SAM.

6. Change the working directory to createEDP and run the command:

```
./StandardEarthquakeEDP -filenameBIM ../createBIM/1-BIM.json -
filenameEVENT ../createEVENT/1-EVENT.json -filenameSAM
../createSAM/1-SAM.json -filenameEDP 1-EDP.json -getRV
```

This should create a 1-EDP.json file that defines the EDPs that are needed to be obtained from the building model simulation.

7. Change the working directory to performSimulation and run the command:

```
./OpenSeesSimulation.sh -filenameBIM ../createBIM/1-BIM.json -
filenameSAM ../createSAM/1-SAM.json -filenameEVENT
../createEVENT/1-EVENT.json  -filenameEDP ../createEDP/1-
EDP.json -filenameEDP 1-SIM.json -getRV
```

This will create and run an openSees model for the building and will obtain are the required EDPs from the output and update their values in 1-EDP.json.

8. Finally, change the working directory to createLOSS and run the command:

```
./FEMAP58_LU -filenameBIM ../createBIM/1-BIM.json -filenameEDP
../createEDP/1-EDP.json -filenameLOSS 1-DL.json -
filenameSettings ./data/settings.ini -pathCurves
./data/ATCCurves/ -pathNormative ./data/normative/
```

This will create the 1-DL.json file, which contains the economic loss, downtime and red tag probabilities for the first building in the database.

Configuring and running a workflow

It is also possible to configure the regional simulation to run the workflow applications for a set of buildings. There are two configuration files involved in this process:

1- **Applications Registry**: This file serves as a registry where all the possible applications to be used by a workflow run are defined. In this registry, the paths to the applications are defined and each application is given a name as an identifier. An application configuration file (WorkflowApplciations.json) is distributed with the current release of the workflow that registers all the applications provided with the release.

2- **Workflow Configuration**: This file configures a single run of the workflow, setting the applications that will be used in this run and any data these applications will need.

Few examples are provided with this release of the workflow and are available in the Workflow sub-directory of the source. Table 7 provides a summary of the current example workflow configurations and the applications they are configured to use.

Table 7: Example workflow configurations

Example Workflow Configurations	Workflow 1 & 4	Workflow 2 & 3	Workflow 5
createBIM	UrbanSimDatabase	UrbanSimDatabase	GenericBimDatabase
createEVENT	LLNL_SW4	SHA-GM.py	LLNL_SW4
createSAM	MDOF-LU	MDOF-LU	MDOF-LU
createEDP	StandardEarthquakeEDP	StandardEarthquakeEDP	StandardEarthquakeEDP
performSimulation	OpenSeesSimulation.sh	OpenSeesSimulation.sh	OpenSeesSimulation.sh
createLOSS	FEMAP58_LU	FEMAP58_LU	FEMAP58_LU
performUQ	Dakota-FEM	Dakota-FEM	Dakota-FEM

For example, we can use the following command to run Workflow 5, assuming we changed the current working directory to the Workflow sub-directory of the source code:

```
python RegionalEarthquakeSimulation.py run Workflow5.json
WorkflowApplications.json
```

In this case, the workflow will run the configured applications (as in Table 7) and process the first 10 buildings as specified in Workflow5.json. It is then possible to edit the workflow5.json file and process a different set of buildings such as the first 50 buildings as follows:

```
{
   "Name": "Workflow 5",
   "Author": "fmk",
   "WorkflowType": "Regional Simulation",
   "buildingFile":"buildings.json",
   "Applications": {
      "Buildings": {
          "BuildingApplication": "GenericBimDatabase",
          "ApplicationData": {
                "Min": "1",
                "Max":"50",
                "buildingsFile":"../createBIM/GenericBimDatabase.csv"
            }
      },…
}
```

Notice that it is also possible to switch the buildings database by pointing the GenericBimDatabase application to a different buildings file. It is possible to check if a workflow.json configuration file is valid before run it, which will check all the paths specified in the configuration files to make sure they are correct. This can be checked using the command:

```
python RegionalEarthquakeSimulation.py check Workflow5.json
WorkflowApplications.json
```

After running the workflow, damage and loss files will be created for all the buildings (e.g. 1-DL.json, 2-DL.json,…..,N-DL.json, etc.). It is then possible to gather all these results into a single CSV file to facilitate visualization. The following command can be used to gather all the damage and loss output in a single file:

Adding new or modified applications

Adding new or modified applications to the workflow is possible as long as the formats of the intermediate files (BIM, EVENT, SAM, EDP and DL files) are the same. As explained in previous section, the workflow is developed in a modular way, where processes in the workflow can be performed using a console application or a script. This makes it possible to replace any of the current applications provided with the workflow with improved, modified or different implementations. For instance, it is possible to develop a new application that obtains BIM files for a particular region using a different source or a different buildings database. In this case, it is possible to use the new createBIM application by adding it to the application registry (WorkflowApplications.json) and configuring new workflows (e.g. Workflow.json) to use the new application.

Visualization in QGIS

This section demonstrated how to visualize the results of the regional simulation in QGIS [11]. After running the workflow, all the outputs can be gathered in a single CSV file as explained in the previous section. The following steps can be used to visualize all the results of the regional simulation on building footprints using QGIS running on a Windows operating system (steps may vary on other operating systems):

1. First, we need to obtain the building footprints shape file for the region being simulated. In this example the SF Bay Area is used. The shape file used herein file obtained from Microsoft's buildings footprints data, which are available in this [webpage](#) and are licensed under the Open Data Commons Open Database License ([ODBL](#)). The dataset includes footprints for 9 million buildings in Metro areas in 44 states across the US.

2. Add a basemap in QGIS. In this example, the TileServer(XYZ) tool (from the Browser Panel) is used with a basemap provided by Carto Db using the tile server url:

   ```
   https://tiles.basemaps.cartocdn.com/rastertiles/light_all/{z}/{x
   }/{y}.png
   ```

   It has to be noted that this basemap is free for non-commercial use. Users of the workflow and this document are referred for Carto Db [website](#) for the map license information. If this is done correctly, you should be able to see the basemap in QGIS as in Figure 28. Alternatively, the OpenLayers QGIS plugin can be used to provide a different basemap.
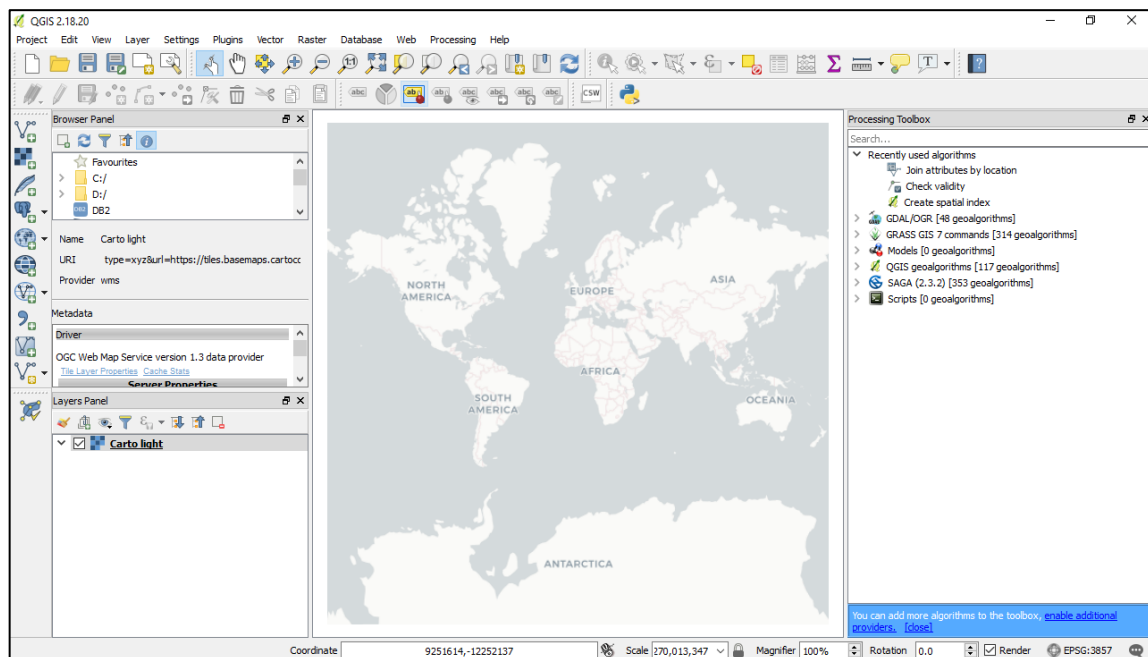

Figure 28: A basemap viewed in QGIS

3. Importing the building footprints in QGIS can be done by using the Layers → Add Layer → Add Vector Layer menu item or by using the keyboard shortcut Ctrl+Shift+V. This will Open the Add

Vector Layer Dialog as show in Figure 29. Now, click on the Browse button and select the shape file, In this case the shape file is called Bay_Area.shp. Figure 30 shows the imported buildings footprints.
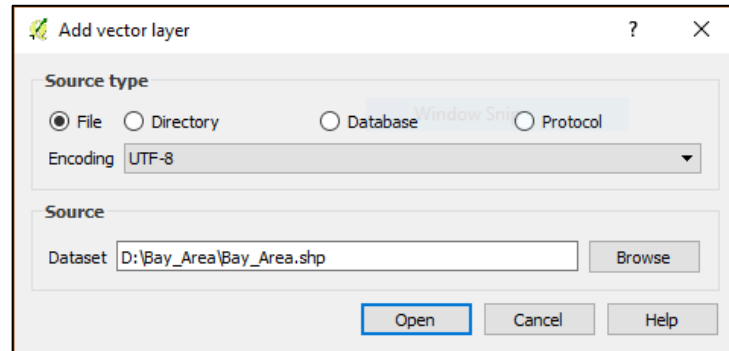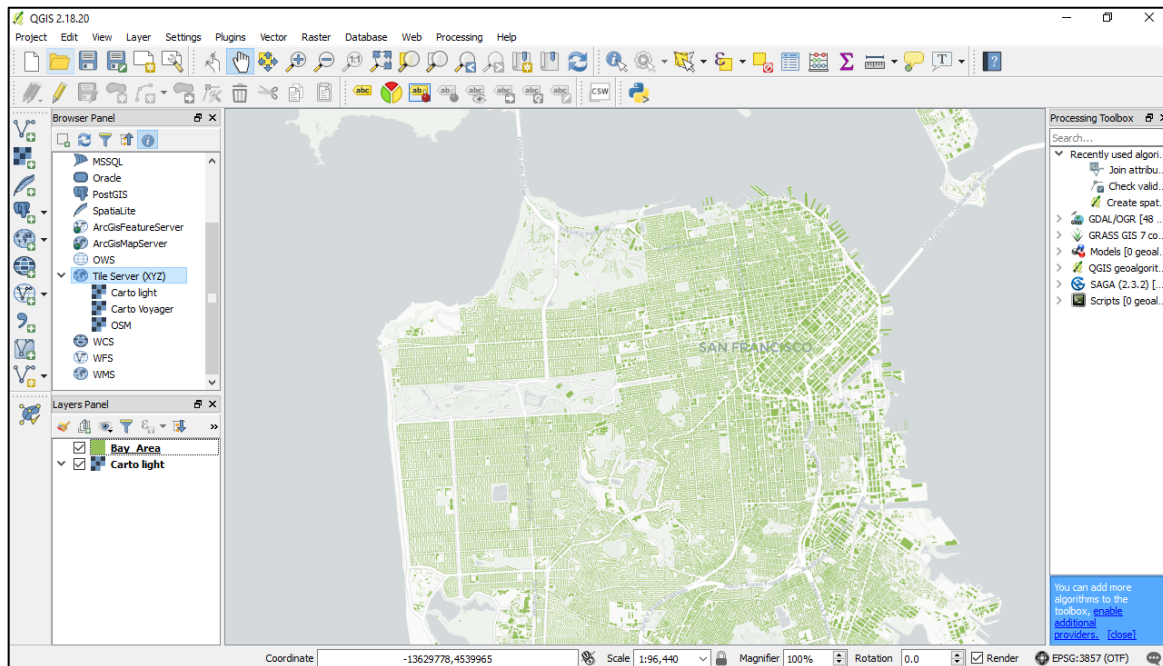

Figure 29: Add Vector Layer Dialog in QGIS

Figure 30: Building footprints displayed in QGIS

4. Now, we can import the results layer. In this example, results from a regional simulation of the SF Bay area for 1.8 million buildings will be imported. Use Layers → Add Layer → Add Delimited Text Layer menu item which will open the Create a Layer from a Delimited Text file Dialog shown in Figure 31. Notice that QGIS automatically detected the longitude and latitude columns from the results file as points coordinates. Now, if we click the OK button, all the results will be imported in a layer as point features as shown in Figure 32. Notice that selecting a point in the results layer shows all the features and attributes imported for this layer which include repair cost, down time, loss ratio as shown in the right panel in the figure.

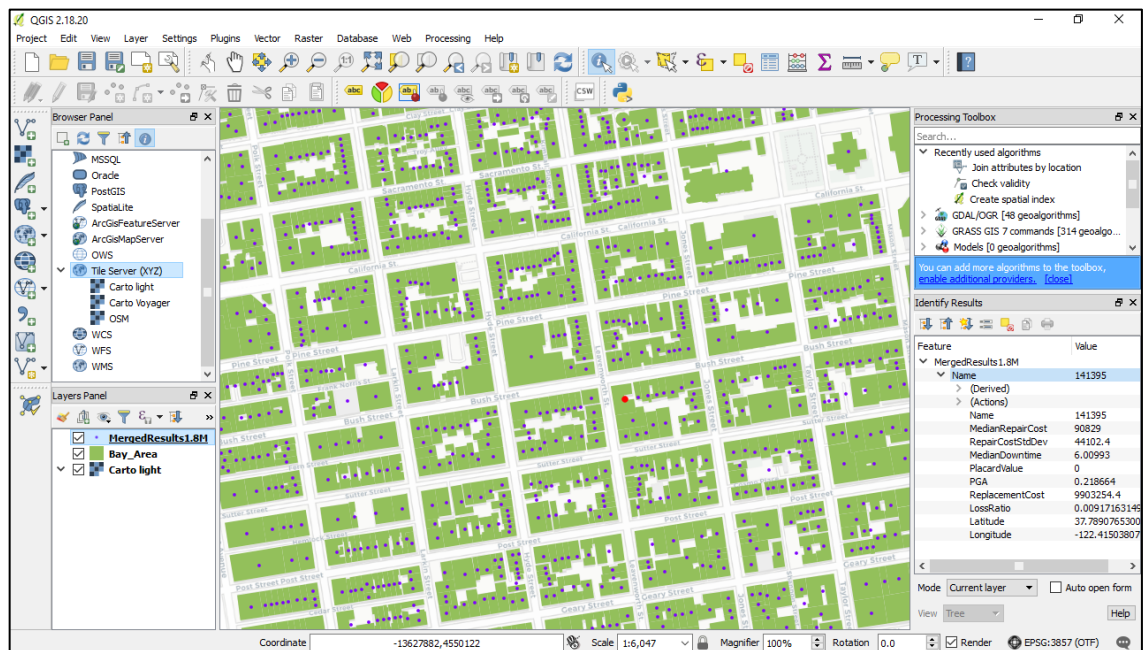Figure 31: Create a Layer from a Delimited Text file Dialog in QGIS

Figure 32: Results Imported as Point Features and Printed Over Building Footprints

5. Now we can perform a spatial join between the building footprints layer and the results layer to create a new joint layer that has building footprints polygons with all the results features imported at building locations attached to them. In this example we will use the join attributes by location tool by using Vector → Data Management Tools → Join Attributes by location menu item as shown in Figure 33. Depending on the number of buildings, this process may take a significant amount of time. After the spatial join is performed, a joined layer will be created and will show in the layers panel, this layer can then be saved in a shape file by right clicking on the newly created layer and clicking on save as. For a large number of buildings, the previous steps may benefit from using spatial indices, this is outside the scope of this document, however, interested readers are referred to QGIS website and documentation.
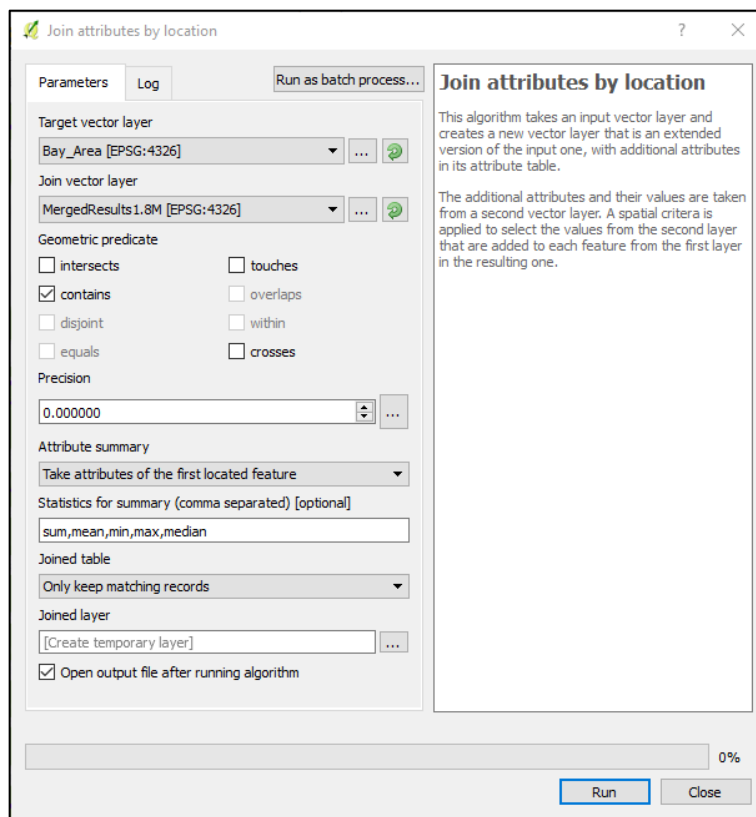

Figure 33: Join Attributes by Location Dialog

6. Finally, we can use the joined layer from the previous step to visualize any of the results on the building footprints. For color coded results visualization, we can change the fill and outline colors of building footprints to be based on an expression evaluated from the results. The layer properties dialog shown in can be opened by double clicking on the joined layer. Now we can select style from the left tab, then select the simple fill from the fill tree view. After that we can edit the fill and outline colors, by clicking on the data defined override icon to the right of the colors, then clicking edit. This will open the expression dialog where we can assign colors based on an expression calculated from the results. In this example we will try to show color coded loss ratios by using the expression `ramp_color('YlOrRd', (LossRatio) * 5)`. This

expression creates a color gradient based on the loss ratio that varies from yellow color for loss ratio value of 0.0 to red for a loss ratio value of 0.2 as shown in Figure 35 and Figure 36.
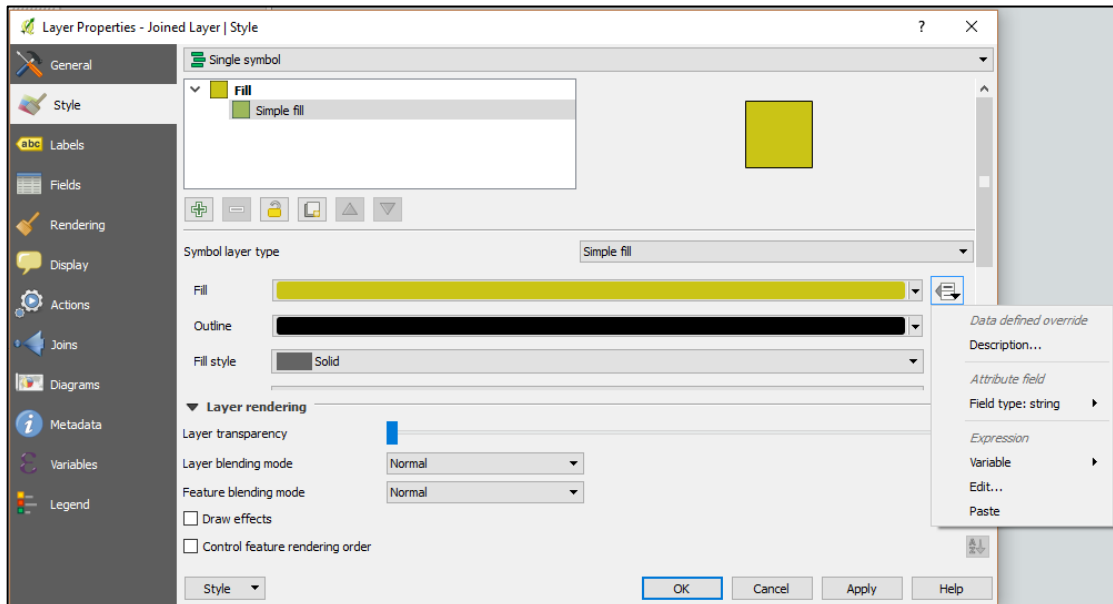
Figure 34: Join Attributes by Location Dialog
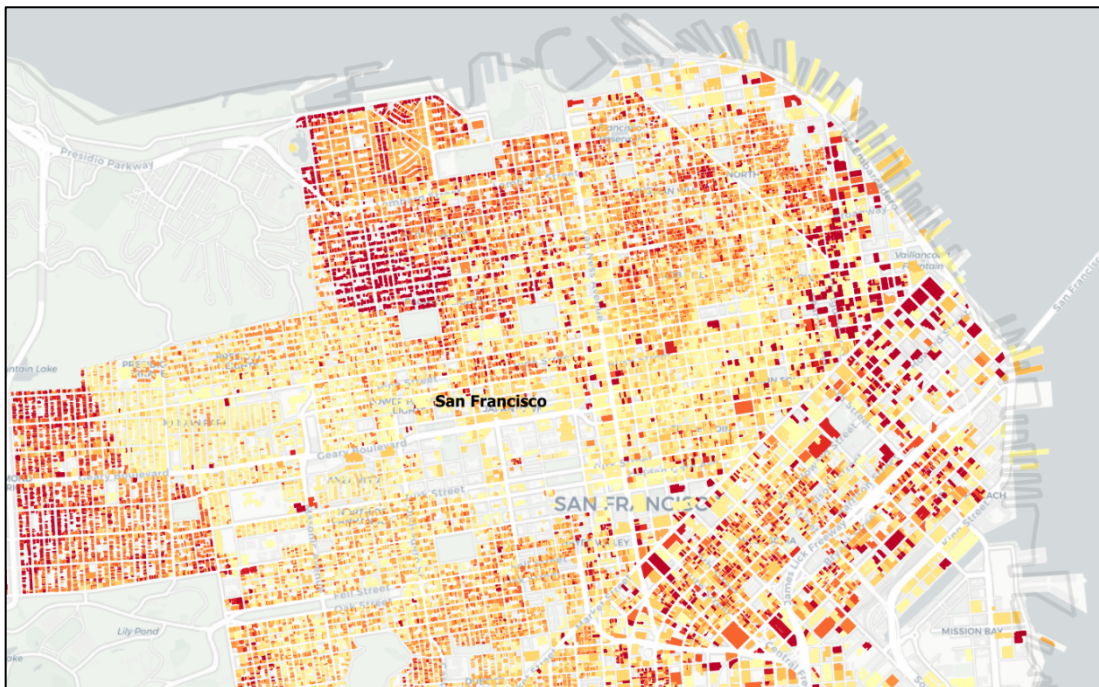
Figure 35: Visualization of Loss Ratios for Buildings in San Francisco

For visualization of the red tagged building as shown Figure 37, the expression can be used `color_rgba(255 *(PlacardValue), 0, 0, 255 * (PlacardValue))`. This expression

create the color using reg green blue and alpha components and will produce a red color for building with placard value equal to 1 and a transparent color for buildings with placard value of 0.
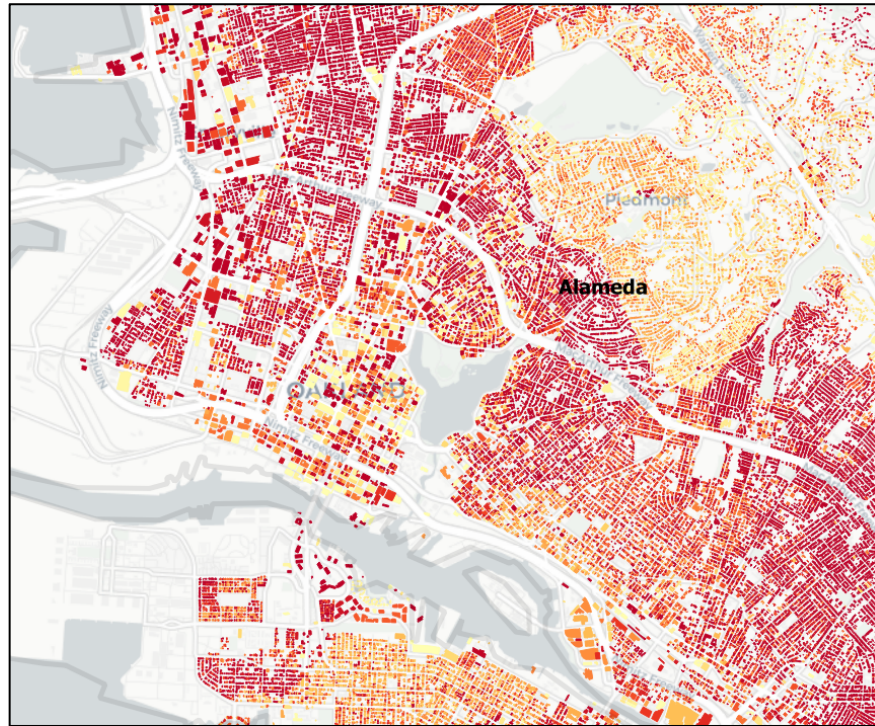
Figure 36: Visualization of Loss Ratios for Buildings in Alameda County
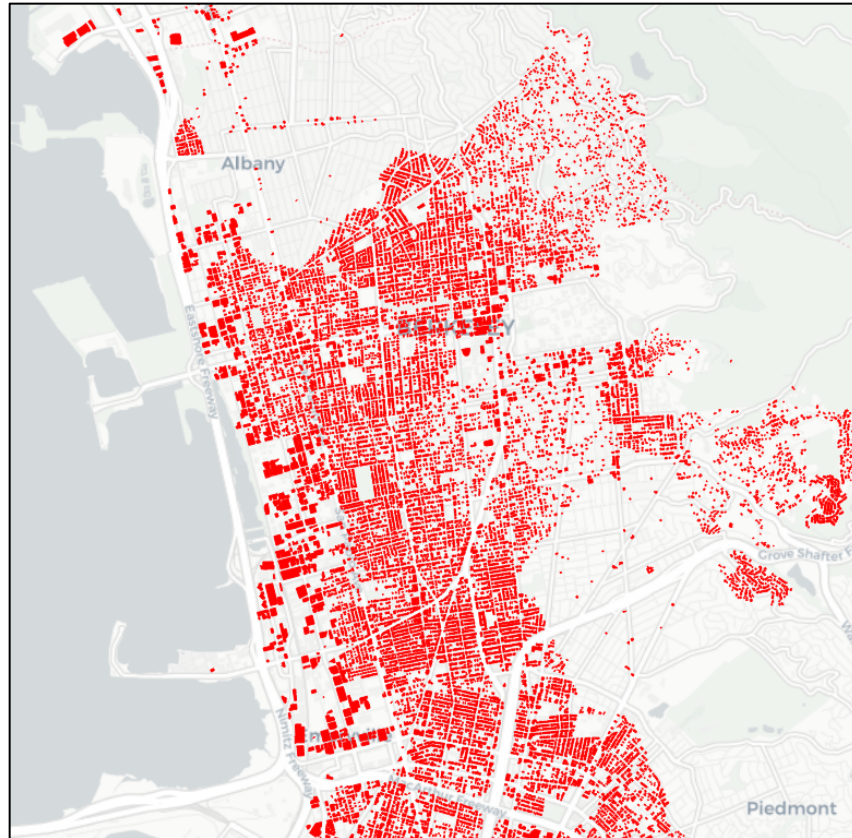
Figure 37: Red-Tagged Buildings in the East Bay

Limitations

The previous methods outlined in this section may not be capable of visualizing all the results for all the buildings in a region. The quality of the visualization depends on the accuracy and quality of the buildings footprints and the locations of the buildings. In many cases, accurate building locations may not be available, or parcels locations may be used instead. In cases where, the same parcel location is used for multiple buildings, it is possible the results will not be mapped to the buildings or will be mapped to a different building in the same parcel.

Developer's Guide

This section provides information about obtaining the source code, the dependencies needed by the workflow and building the workflow applications from source code.

Dependencies

The following dependencies are needed to build and run the workflow:

- **Jansson Library**: Many of the provided applications with the workflow require the Jansson library to be installed. Jansson is a free native C library for encoding, decoding and manipulating JSON data. It is licensed under the MIT license. For unix-like systems, it is assumed the library is installed in /usr/local/jansson. It is possible to install the library at a different location, but that would require changing the included Makefiles or using symbolic links to point to the install location.

- **C++11 compliant compiler**: Some applications in the createLOSS and performUQ folders use C++11 features, consequently they may need a newer C++11 compliant compiler.

- **OpenSees**: The workflow applications require an installation of OpenSees to carry out structural analysis using the finite element method.

- **DAKOTA**: The workflow applications require an installation of DAKOTA to handle and propagate the uncertainties defined in the input files for the workflow applications.

- **Python**: The workflow requires Python 2.7.

- **Perl**: The workflow makes use of a modified version of the `DPrePro` preprocessor Perl script distributed with DAKOTA, and thus requires Perl. Later releases may switch to the newer python version of the script as the Perl script is deprecated as of DAKOTA version 6.8.

- **nanoflann**: A C++11 library that performs nearest neighbor search using k-dimensional trees. nanoflann is a header-only library and is included with the source of the workflow. It is currently being used by LLNL_SW4 createEVENT application.

- **Simcenter-EQSS**: the ground motion tools developed at the SimCenter are needed, if using SHA-GM.py as a createEVENT application. To obtain and build these applications please refer to SimCenter-EQSS repository for documentation.

Building from source on Unix-like systems

Before building the workflow, the following dependencies will need to be installed:

1. GNU Compiler Collection (gcc & g++) version 4.8.1 or newer.
2. GNU Make.
3. Jansson Library.

The repository uses Makefiles to build the applications. To build all the workflow applications, use `make` or `make all` in the root directory. To build for debugging use `make debug` and to clean, use `make clean`. Individual applications can be built using make in their own subdirectories as well.

Building the Source Code on Ubuntu Linux

Before building the source code on Ubuntu Linux and running it, the dependencies can be installed using the following commands:

```
sudo apt-get update

sudo apt-get install gcc g++ python perl
```

After installing the dependencies, building the workflow applications can then be done similar to other unix systems, using `make all`. Running the workflow applications will require installing other dependencies like OpenSees and DAKOTA. Instructions to building these software applications from their source code on Ubuntu Linux are available on their respective websites.

Building the Source Code on Windows

Visual studio projects are under development. Currently the workflow can be built and run in windows using the Windows subsystem for Linux and building using the same commands for a Linux system.

Contributing to the Source Code

The regional earthquake simulation workflow is an open source project and is hosted on Github. Developers or researchers interested in contributing to the source can fork the Github repository from the SimCenter account and create a pull request from the fork. For information about creating a fork and creating pull requests, interested readers are referred to this webpages:

- GitHub working with forks web page
- GitHub create a pull request webpage
- GitHub pull request help webpage

Pegasus Workflow

The Pegasus workflow can be found in the Pegasus directory. Before submitting, please make sure all the applications are built successfully. The workflow is set up to execute in two different execution environments: local HTCondor pool, Stampede 2 or TACC Wrangler. The first is good for small to medium sized runs as turnaround time is usually pretty good. Large workflows should be mapped to TACC Wrangler. Submitting a workflow for local HTCondor pool execution can be done by executing:

```
./submit-workflow
```

Submitting to TACC Stampede2 can be done on the workflow.isi.edu host with the command:

```
./submit-workflow stampede2

./submit-workflow wrangler
```

Testbed and Validation

In this section, results from regional testbed simulation of SF Bay area are compared to results of the HayWired Scenario [1]. The HayWired Scenario is a study of a hypothetical earthquake scenario in SF Bay Area. The study is published by USGS in a series of scientific reports as part of a regional and statewide earthquake preparedness initiative in California. The scenario considers a main shock of magnitude 7.0 with a rupture along the Hayward fault and an epicenter in Oakland, CA. In addition, the effects of aftershocks, liquefaction, landslides and fires are included in the study as well. In the HayWired Scenario study, a physics-based ground motion model based on wave propagation and a kinematic rupture model was employed. Hazus-MH [7] was used for damage and loss assessment including 16 aftershocks of magnitude 5.0 or more, liquefaction and landslides. In this section, only the results from the main shock are compared to results from a simulation using SimCenter's regional earthquake testbed. The following is a summary of the HayWired Scenario study.

- The main shock is a M7.0 rupture along the Hayward fault with an epicenter in Oakland, CA.
- Approximately, 3.0 million buildings were included in the study in 17 counties in SF Bay Area.
- Using Enhanced Hazus building inventory data originally developed in 2005–2006.

- Included retrofitted building model for unreinforced masonry, non-ductile concrete and soft-story wood frame structures.
- Used enhanced mapping schemes to relate building occupancy to percent distribution among structure type and model building types.
- 22 mapping schemes were used, with a separate set of schemes for San Francisco and Alameda counties.

The results of the HayWired scenario results can be summarized as follows:

- 800 deaths and 16,000 non-fatal injuries are estimated to occur from the shaking hazard alone.
- 500,000 buildings would be impaired (collapsed, red-tagged or yellow-tagged), assuming these buildings are code-compliant (Designed using Importance Factor = 1.0).
- 8000 collapsed buildings, 101,000 red-tagged buildings and 386,000 yellow-tagged buildings.
- Building damage is $30.3 billion in 2005 dollars or $37.3 billion in 2016 dollars.
- Net buildings damage ratio is 2.91%.

On the other hand, the SimCenter regional testbed simulates a similar earthquake scenario and studies its damage and loss effects on the SF Bay are region. The study can be summarized as follows:

- A magnitude 7.0 rupture along the Hayward fault with a hypocenter in Richmond, CA was used in the study and was modeled using SW4 [3].
- 1.84 million buildings were included in the Simulation.
- Buildings information is based on data obtained from UrbanSim [12].
- Damage and Loss is computed using FEMA_P59_LU application [8].
- Structural analysis models are based on MDOF_LU application.

The results of this study can be summarized as follows:

- Red-tagged buildings are 141,459
- The total buildings damage for the region is $84.1 billion
- Net buildings damage ratio is 5.6%

Results Comparison

Table 8 compares the total results of both studies. It can be seen from both studies that the total buildings damage estimated is more than twice as much as the HayWired Scenario. In addition, the total replacement costs of all the buildings is significantly larger in the regional simulation performed by the SimCenter workflow simulation. When results are aggregated over counties, it is shown that a different in the building damage results is more pronounced in Dan Francisco and Alameda counties. Further detailed analysis is needed to compare the building inventories in both simulations. Figure 38 compares the loss ratios of both studies visualized for the SF Bay area, It is important to point out here that result from the SimCenter simulation are more fine-grained as they are visualized on building footprints, whereas results from Hazus analysis in the HayWired Scenario are aggregated over zip code areas.

Table 8: Comparison of total results of the two regional simulations

Result	HayWired Scenario	SimCenter Testbed
Number of Buildings	3 Million	1.84 Million
Red Tagged Buildings	101,000	141,459
Building Damage	$30.3 Billion	$84.1 Billion
Net Damage Ratio	2.91%	5.6%
Total Buildings Cost	$1.04 Trillion	$1.5 Trillion

Table 9: Simulation results aggregated over the SF Bay area counties. HS represents the HayWired scenario and SCT represents the SimCenter Testbed

County	Number of Buildings [Thousands]		Total Repair Cost [$ Billion]		Total Replacement Cost [$ Billion]	
	HS	SCT	HS*	SCT	HS	SCT
Santa Clara	495.3	421.6	4.6	3.4	183	415
Alameda	413.5	371	16.25	41	155	283
Contra Costa	321	300	5.4	17.0	103	201
San Mateo	219.8	180.6	1.5	5.0	84	153
San Francisco	173	150	1.6	12.0	100	186
Sonoma	169	153.7	0.03	0.03	50.9	92.2
Solano	120.8	127.1	0.16	2.1	34.8	75.5
Marin	93.2	89.4	0.28	2.8	36.1	59.3
Napa	45.1	47.3	0.03	0.16	14.6	34.3
Other**	987.7	0	0.38	0	280	0
Total	**3038.4**	**1840.7**	**30.3**	**84.1**	**1041**	**1498**

* Repair costs in this column are from shaking of the main shock only, and are in 2005 U.S. dollar value
** Counties that were not included in the SimCenter regional simulation (Sacramento, Merced, Monterey, San Benito, San Joaquin, Santa Cruz, Stanislaus and Yolo counties)
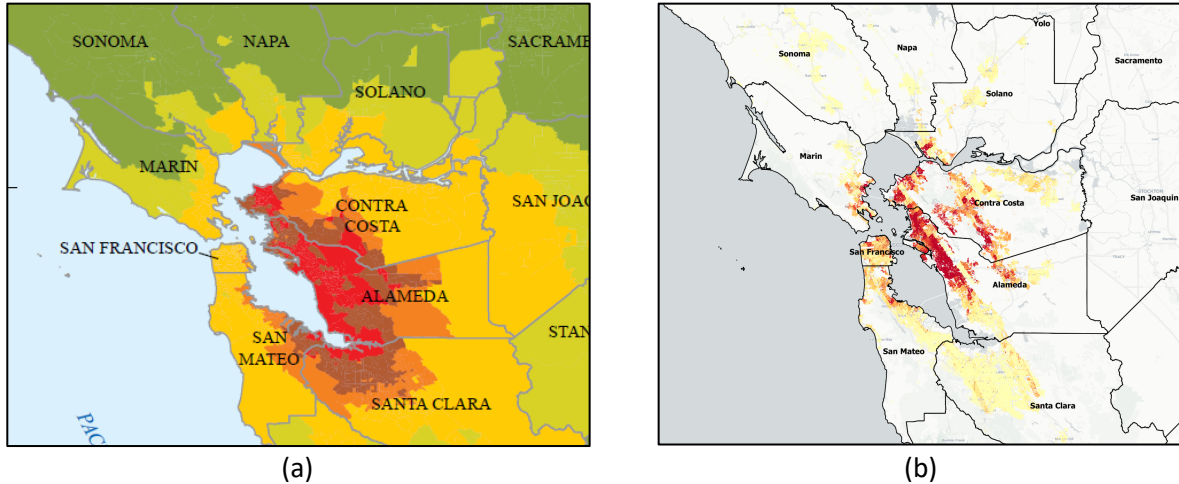
Figure 38: Building loss ratios visualization (a) HayWired Scenario (b) SimCenter Testbed

Disclaimer

NHERI SimCenter provides the source code of the regional earthquake simulation workflow as-is and makes no representations or warranties of any kind concerning the source code and the results obtained from them, whether expressed, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable.

Acknowledgements

References

[1] S. Detweiler and A. Wein, "The HayWired earthquake scenario—Engineering implications: U.S. Geological Survey Scientific Investigations Report 2017–5013–I–Q," U.S. Geological Survey, 2018.

[2]   N. Petersson and B. Sjogreen, "SW4, version 2.01 [software]," Computational Infrastructure of Geodynamics, 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1063644.

[3]   A. J. Rodgers, A. Pitarka, N. A. Petersson, B. Sjögreen and D. B. McCallen, "Broadband (0–4 Hz) ground motions for a magnitude 7.0 Hayward fault earthquake with three-dimensional structure and topography," *Geophysical Research Letters,* vol. 45, p. 739–747, 2018.

[4]   T. D. Ancheta, R. B. Darragh, J. P. Stewart, E. Seyhan, W. J. Silva, B. S. Chiou, K. E. Wooddell, R. W. Graves, A. R. Kottke, D. M. Boore, T. Kishida and J. L. Donahue, "PEER NGA-West2 Database," Pacific Earthquake Engineering Research Center (PEER), Berkeley, 2013/03.

[5]   E. H. Field, T. H. Jordan and C. A. Cornell, "OpenSHA: A Developing Community-Modeling Environment for Seismic Hazard Analysis," *Seismological Research Letters,* vol. 74, no. 74, pp. 406-419, 2003.

[6]   N. Jayaram and J. W. Baker, "Correlation model for spatially distributed," *Earthquake Engineering and Structural Dynamics,* vol. 38, pp. 1687-1708, 2009.

[7]   "Hazus®–MH 2.1, Earthquake Model Technical Manual," Federal Emergency Management Agency , Department of Homeland Security, Washington, D.C., 2017.

[8]   X. Zeng, Z. Xu and a. X. Lu, "Regional Seismic Damage Simulation of Buildings: A Case Study of the Tsinghua Campus in China," in *2015 International Workshop on Computing in Civil Engineering*, Austin, Texas, 2015.

[9]   F. McKenna, G. Fenves and M. Scott, "Open System for Earthquake Engineering Simulation," Pacific Earthquake Engineering Research Center (PEER), Berkeley, CA, 2000.

[10] FEMA, "Seismic performance assessment of buildings, FEMA P 58-1," Applied Technology Council, Redwood City, CA, 2012.

[11] QGIS Development Team (2018), "QGIS Geographic Information System," [Online]. Available: http://qgis.osgeo.org/. [Accessed 2018].

[12] P. Waddell, "UrbanSim: Modeling Urban Development for Land Use, Transportation, and Environmental Planning," *Journal of the American Planning Association,* vol. 68, no. 3, pp. 297-314, 2002.