OQ

OPENQUAKE
calculate share explore

USER
INSTRUCTIONS

HAZARD
SCIENCE

RISK
SCIENCE

OPENQUAKE ENGINE
USER INSTRUCTION MANUAL

Version 1.9.0

Hands-on-instructions on the different types of calculations you can carry out with the OpenQuake Engine software

GEM
GLOBAL EARTHQUAKE MODEL
working together to assess risk

OpenQuake: calculate, share, explore

The OpenQuake-engine
User Instruction Manual

Authors

Marco Pagani[1], Vitor Silva[1], Graeme Weatherill[1], Anirudh Rao[1], Michele Simionato[1]

[1] GEM Foundation
via Ferrata, 1
20133 Pavia
Italy

Email address for all the authors:
<name.surname>@globalquakemodel.org

Contents

Preface

The goal of this manual is to provide a comprehensive and transparent description of the features of the OpenQuake-engine v1.8. This manual is designed to be readable by someone with basic understanding of Probabilistic Seismic Hazard and Risk Analysis, but no previous knowledge of the OpenQuake-engine is assumed.

The OpenQuake-engine is an effort promoted and actively developed by the Global Earthquake Model, a public-private partnership initiated by the Global Science Forum of the Organisation for Economic Co-operation and Development (OECD)[1].

The OpenQuake-engine is the result of an effort carried out jointly by the Information Technology and Scientific teams working at the Global Earthquake Model (GEM) Secretariat. It is freely distributed under an Affero GPL license (http://www.gnu.org/licenses/agpl-3.0.html).

---

[1]A short description of the process promoted by OECD is available here:
http://www.oecd.org/science/sci-tech/theglobalearthquakemodelgem.htm

Part I

Introduction

1. OpenQuake-engine Background

OpenQuake-engine is the seismic hazard and risk calculation software developed by the Global Earthquake Model. By following current standards in software developments like test-driven development and continuous integration, the OpenQuake-engine aims at becoming an open, and community-driven tool for seismic hazard and risk analysis.

The source code of the OpenQuake-engine is available on a public web-based repository at the following address:

http://github.com/gem/oq-engine.

1.1 Running the OpenQuake-engine

An OpenQuake-engine (oq-engine) analysis is launched from the command line of a terminal.

A schematic list of the options that can be used for the execution of the oq-engine can be obtained with the following command:

```
user@ubuntu:~\$ oq-engine --help
```

The result is the following:

```
usage: oq-engine [-h]
                 [--version]
                 [--log-file LOG_FILE]
                 [--log-level {debug,info,progress,warn,error,critical}]
                 [--no-distribute]
                 [--run CONFIG_FILE] [--yes]
                 [--config-file CONFIG_FILE]
                 [--make-html-report YYYY-MM-DD|today]
                 [--upgrade-db]
                 [--version-db]
                 [--what-if-I-upgrade]
                 [--run-hazard CONFIG_FILE]
```

```
                    [--list-hazard-calculations]
                    [--delete-calculation CALCULATION_ID]
                    [--delete-uncompleted-calculations]
                    [--run-risk CONFIG_FILE]
                    [--hazard-output-id HAZARD_OUTPUT_ID]
                    [--hazard-calculation-id HAZARD_CALCULATION_ID]
                    [--list-risk-calculations]
                    [--list-outputs CALCULATION_ID]
                    [--show-view CALCULATION_ID VIEW_NAME]
                    [--exports EXPORT_FORMATS]
                    [--export-output OUTPUT_ID TARGET_DIR]
                    [--export-outputs CALCULATION_ID TARGET_DIR]
                    [--export-stats CALCULATION_ID TARGET_DIR OUTPUT_TYPE]


OpenQuake Seismic Hazard and Risk Analysis Engine

optional arguments:
  -h, --help             show this help message and exit

General:
  --version              Display version information
  --log-file LOG_FILE, -L LOG_FILE
                         Location to store log messages; if not specified, log
                         messages will be printed to the console (to stderr)
  --log-level            {debug,info,progress,warn,error,critical}
                         -l {debug,info,progress,warn,error,critical}
                         Defaults to "info"
  --no-distribute, --nd
                         Disable calculation task distribution and run the
                         computation in a single process. This is intended for
                         use in debugging and profiling.
  --run CONFIG_FILE      Run a job with the specified configuration file; you
                         can also pass a comma-separated pair of files for
                         hazard and risk
  --yes, -y              Automatically answer "yes" when asked to confirm an
                         action
  --config-file CONFIG_FILE
                         Custom openquake.cfg file, to override default
                         configurations
  --make-html-report YYYY-MM-DD|today, -r YYYY-MM-DD|today
                         Build an HTML report of the computation at the given
                         date

Database:
  --upgrade-db           Upgrade the openquake database
  --version-db           Show the current version of the openquake database
  --what-if-I-upgrade    Show what will happen to the openquake database if you
                         upgrade
```

```
Hazard:
  --run-hazard CONFIG_FILE, --rh CONFIG_FILE
                        Run a hazard job with the specified config file
  --list-hazard-calculations, --lhc
                        List hazard calculation information
  --delete-calculation CALCULATION_ID, --dc CALCULATION_ID
                        Delete a calculation and all associated outputs
  --delete-uncompleted-calculations, --duc
                        Delete all the uncompleted calculations


Risk:
  --run-risk CONFIG_FILE, --rr CONFIG_FILE
                        Run a risk job with the specified config file
  --hazard-output-id HAZARD_OUTPUT_ID, --ho HAZARD_OUTPUT_ID
                        Use the desired hazard output as input for the risk
                        job
  --hazard-calculation-id HAZARD_CALCULATION_ID, --hc HAZARD_CALCULATION_ID
                        Use the desired hazard job as input for the risk job
  --list-risk-calculations, --lrc
                        List risk calculation information


Export:
  --list-outputs CALCULATION_ID, --lo CALCULATION_ID
                        List outputs for the specified calculation
  --show-view CALCULATION_ID VIEW_NAME, --sv CALCULATION_ID VIEW_NAME
                        Show a view of the specified calculation
  --exports EXPORT_FORMATS
                        Comma-separated string specifing the export formats,
                        in order of priority
  --export-output OUTPUT_ID TARGET_DIR, --eo OUTPUT_ID TARGET_DIR
                        Export the desired output to the specified directory
  --export-outputs, --eos CALCULATION_ID TARGET_DIR
                        Export all the calculation outputs to the specified
                        directory
  --export-stats, --es CALCULATION_ID TARGET_DIR OUTPUT_TYPE
                        Export the statistical outputs to the specified
                        directory
```

1.2 Concurrent Computing with OpenQuake

The OpenQuake-engine supports concurrent computing on both standalone computers and computer clusters.

The OpenQuake-engine works by splitting a computation into a number of tasks which are then processed in parallel. The user has the ability to control the splitting procedure, at least to a certain extent, by setting the parameter concurrent_tasks in the job.ini file. The OpenQuake-engine will try to produce a number of tasks close to concurrent_tasks:

it could be more, it could be less. The details of the algorithm used can change depending on the release of the engine and this is why they are not documented here. Our suggestion is to provide a value:

`concurrent_tasks` = 2 * number of (hyperthread) cores in the workers

or more, if the computation has memory issues. With more tasks, less memory is used, but more data is transferred and the computation becomes slower.

If `concurrent_tasks` is set to zero, the parallelization is disabled and the job is executed by using a single core. This is useful when debugging errors.

Part II

Hazard

2. Introduction to the Hazard Module

The hazard component of the OpenQuake-engine builds on top of the OpenQuake hazard library (oq-hazardlib), a Python-based library containing tools for PSHA calculations.

The web repository of this library is available at the following address: http://github.com/gem/oq-hazardlib.

In this section we briefly illustrate the main properties of the hazard component of the OpenQuake-engine. In particular, we will describe the main typologies of sources supported and the main calculation workflows available.

2.1 Source typologies

An OpenQuake-engine seismic source input model contains a list of sources belonging to a finite set of possible typologies. Each source type is defined by a set of parameters - called source data - which are used to specify the source geometry and the properties of seismicity occurrence.

Currently the OpenQuake-engine supports the following source types:

- Sources for modelling distributed seismicity:
    - Point source - The elemental source type used to model distributed seismicity. Grid and area sources (described below) are different containers of point sources.
    - Area source - So far, the most frequently adopted source type in national and regional PSHA models.
    - Grid source - A replacement for area sources admitting spatially variable seismicity occurrence properties.
- Fault sources with floating ruptures:
    - Simple fault source - The simplest fault model in the OpenQuake-engine. This source is habitually used to describe shallow seismogenic faults.

- – Complex fault source - Often used to model subduction interface sources with a complex geometry.
- Fault sources with ruptures always covering the entire fault surface:
  - – Characteristic fault source - A typology of source where ruptures always fill the entire fault surface.

The OpenQuake-engine contains some basic assumptions for the definition of these source typologies:

- In the case of area and fault sources, the seismicity is homogeneously distributed over the source;
- Seismicity temporal occurrence follows a Poissonian model.

The above sets of sources may be referred to as "parametric" sources, that is to say that the generation of the Earthquake rupture forecast is done by the OpenQuake engine based on the parameters of the sources set by the user. In some cases, particularly if the user wishes for the temporal occurrence model to be non-Poissonian (such as the lognormal or Brownian Passage Time models) a different type of behaviour is needed. For this OpenQuake-engine supports a Non-parametric source in which the Earthquake rupture forecast is provided explicitly by the user as a set of ruptures and their corresponding probabilities of occurrence.

2.1.1 Source typologies for modelling distributed seismicity

2.1.1.1 Point sources
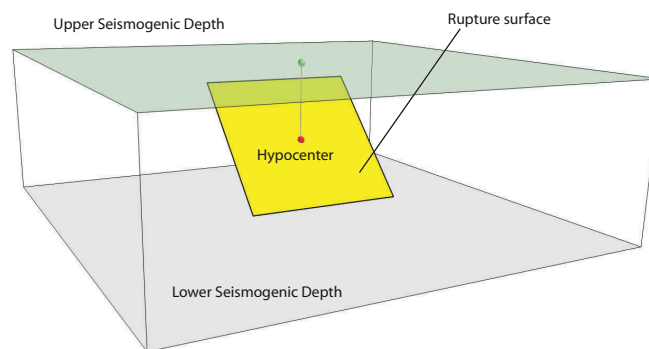
Figure 2.1 – *Single rupture*

The point source is the elemental source type adopted in the OpenQuake-engine for modelling distributed seismicity. The OpenQuake-engine always performs calculations considering finite ruptures, even in the case of point sources.

These are the basic assumptions used to generate ruptures with point sources:

- Ruptures have a rectangular shape
- Rupture hypocenter is located in the middle of the rupture
- Ruptures are limited at the top and at the bottom by two planes parallel to the topographic surface and placed at two characteristic depths named upper and lower

seismogenic depths, respectively (see Figure 2.1)

Source data

For the definition of a point source the following parameters are required (Figure 2.1 shows some of the parameters described below, together with an example of the surface of a generated rupture):

- The coordinates of the point (i.e. longitude and latitude) [decimal degrees]
- The upper and lower seismogenic depths [km]
- One magnitude-frequency distribution
- One magnitude-scaling relationship
- The rupture aspect ratio
- A distribution of nodal planes i.e. one (or several) instances of the following set of parameters:
    - strike [degrees]
    - dip [degrees]
    - rake [degrees]
- A magnitude independent depth distribution of hypocenters [km].

Figure 2.2 shows ruptures generated by a point source for a range of magnitudes. Each rupture is centered on the single hypocentral position admitted by this point source. Ruptures are created by conserving the area computed using the specified magnitude-area scaling relatioship and the corresponding value of magnitude.
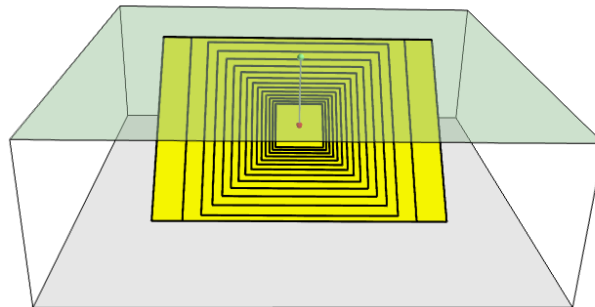
Figure 2.2 – *Point source with multiple ruptures. Note the change in the aspect ratio once the rupture width fills the entire seismogenic layer.*

Below we provide the excerpt of an .xml file used to describe the properties of a point source:

```
1  <pointSource id="1" name="point" tectonicRegion="Stable Continental Crust">
2      <pointGeometry>
3          <gml:Point>
4              <gml:pos>-122.0 38.0</gml:pos>
5          </gml:Point>
```

```
 6            <upperSeismoDepth>0.0</upperSeismoDepth>
 7            <lowerSeismoDepth>10.0</lowerSeismoDepth>
 8        </pointGeometry>
 9        <magScaleRel>WC1994</magScaleRel>
10        <ruptAspectRatio>0.5</ruptAspectRatio>
11        <truncGutenbergRichterMFD aValue="-3.5" bValue="1.0" minMag="5.0"
12                        maxMag="6.5" />
13        <nodalPlaneDist>
14            <nodalPlane probability="0.3" strike="0.0" dip="90.0" rake="0.0" />
15            <nodalPlane probability="0.7" strike="90.0" dip="45.0" rake="90.0" />
16        </nodalPlaneDist>
17        <hypoDepthDist>
18            <hypoDepth probability="0.5" depth="4.0" />
19            <hypoDepth probability="0.5" depth="8.0" />
20        </hypoDepthDist>
21  </pointSource>
```

The red part shows the the parameters used to describe the geometry of the point source, the blue part is the description of the magnitude-frequency distribution, the green text shows the nodal plane distribution and the text in magenta illustrates the hypocentral depth distribution.

The text in black describes the parameters needed to generate the ruptures such as the magnitude-scaling relationship and the aspect ratio.

Note that in this example, ruptures occur on two possible nodal planes and two hypocentral depths. Figure 2.3 shows the ruptures generated by the point source specified above.
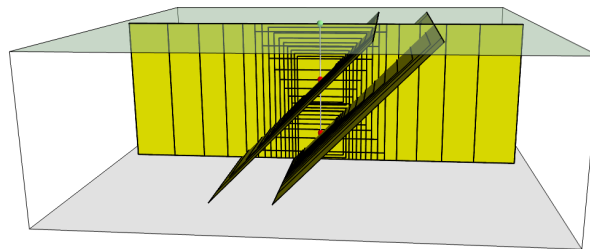
Figure 2.3 – *Ruptures produced by the source created using the information in the example .xml file described at page 24.*

2.1.1.2 Grid sources

A grid source is simply a collection of point sources distributed over a regular grid (usually equally spaced in longitude and latitude).

In probabilistic seismic hazard analysis a grid source can be considered a model alternative to area sources, since they both model distributed seismicity.

Grid sources are generally used to reproduce more faithfully the spatial pattern of seismicity depicted by the earthquakes occurred in the past; in some models (e.g. Petersen et al. (2008)) only events of low and intermediate magnitudes are considered.

Grid sources are generally computed using seismicity smoothing algorithms (Frankel, 1995; Woo, 1996, amongst many others).

The use of smoothing algorithms to produce grid sources brings some advantages compared to area sources, since (1) it removes most of the unavoidable degree of subjectivity due to the definition of the geometries of the area sources and (2) it produces a spatial pattern of seismicity that is usually closer to what observed in the reality. Nevertheless, in many cases smoothing algorithms require an a-priori definition of some setup parameters that expose the calculation to a certain degree of partiality.

Grid sources are modeled in oq-engine simply as a set of point sources; in other words, a grid source is just a long list of point sources specified as described in the previous section (see page 22).

2.1.1.3 Area sources

Area sources are usually adopted to describe the seismicity occurring over wide areas where the identification and characterization - i.e. the unambiguous definition of position, geometry and seismicity occurrence parameters - of single fault structures is difficult.

From a computation standpoint, area sources are comparable to grid sources since they are both represented in the engine by a list of point sources.

The oq-engine using the source data parameters (see below) creates an equally spaced in distance grid of point sources where each point has the same seismicity occurrence properties (i.e. rate of events generated).

Below we provide a brief description of the parameters necessary to completely describe an area source.

Source data
- A polygon defining the external border of the area (i.e. a list of Longitude-Latitude [degrees] tuples) The current version of the OQ-engine doesn't support the definition of internal borders.
- The upper and lower seismogenic depths [km]
- One glsmfd
- One glsmsr
- The rupture aspect ratio
- A distribution of nodal planes i.e. one (or several) instances of the following set of parameters
    - strike [degrees]
    - dip [degrees]
    - rake [degrees]

- A magnitude independent depth distribution of hypocenters [km].

Below we provide the exerpt of an .xml file used to describe the properties of an area source:

```
1  <areaSource id="1" name="Quito" tectonicRegion="Active Shallow Crust">
2      <areaGeometry>
3          <gml:Polygon>
4              <gml:exterior>
5                  <gml:LinearRing>
6                      <gml:posList>
7                      -122.5 37.5
8                      -121.5 37.5
9                      -121.5 38.5
10                     -122.5 38.5
11                     </gml:posList>
12                     </gml:LinearRing>
13              </gml:exterior>
14          </gml:Polygon>
15          <upperSeismoDepth>0.0</upperSeismoDepth>
16          <lowerSeismoDepth>10.0</lowerSeismoDepth>
17          </areaGeometry>
18      <magScaleRel>PeerMSR</magScaleRel>
19      <ruptAspectRatio>1.5</ruptAspectRatio>
20      <incrementalMFD minMag="6.55" binWidth="0.1">
21          <occurRates>0.0010614989 8.8291627E-4 7.3437777E-4 6.108288E-4
22              5.080653E-4</occurRates>
23      </incrementalMFD>
24      <nodalPlaneDist>
25          <nodalPlane probability="0.3" strike="0.0" dip="90.0" rake="0.0"/>
26          <nodalPlane probability="0.7" strike="90.0" dip="45.0" rake="90.0"/>
27      </nodalPlaneDist>
28      <hypoDepthDist>
29          <hypoDepth probability="0.5" depth="4.0" />
30          <hypoDepth probability="0.5" depth="8.0" />
31      </hypoDepthDist>
32  </areaSource>
```

The red text describes the parameters used to describe the geometry of the area source; the blue part is the description of the magnitude-frequency distribution; the green text displays the nodal plane distribution; and the text in magenta illustrates the hypocentral depth distribution.

The text in gray describes the parameters required to generate the ruptures such as the magnitude-scaling relationship and the aspect ratio.

The ruptures generated by the area source described in the example above are controlled by two nodal planes and have hypocenters at localized at two distinct depths.

2.1.2 Fault sources with floating ruptures

Fault sources in the oq-engine are classified according to the method adopted to distribute ruptures over the fault surface. Two options are currently supported:

- With the first option, ruptures with a surface lower than the whole fault surface are floated so as to cover as much as possible homogeneously the fault surface. This model is compatible with all the supported magnitude-frequency distributions.
- With the second option, ruptures always fill the entire fault surface. This model is compatible with magnitude-frequency distributions similar to a characteristic model (à la Schwartz and Coppersmith, 1984).

In this subsection we discuss the different fault source types that support floating ruptures. In the next subsection we will illustrate the fault typology available to model a characteristic rupturing behaviour.

2.1.2.1 Simple faults

Simple Faults are the most common source type used to model shallow faults; the "simple" adjective relates to the geometry description of the source which is obtained by projecting the fault trace (i.e. a polyline) along a characteristic dip direction.

The parameters used to create an instance of this source type are described in the following paragraph.

Source data

- A fault trace (usually a polyline). It is a list of longitude-latitude tuples [degrees]
- A frequency-magnitude distribution
- A magnitude-scaling relationship
- A representative value of the dip angle (specified following the Aki-Richards convention; see Aki and Richards (2002)) [degrees]
- Rake angle (specified following the Aki-Richards convention; see Aki and Richards (2002)) [degrees]
- Upper and lower depth values limiting the seismogenic interval [km]

For near-fault probabilistic seismic hazard analysis, two additional parameters are needed for characterising seismic sources:

- A hypocentre list. It is a list of the possible hypocentral positions, and the corresponding weights, e.g., alongStrike="0.25" downDip="0.25" weight="0.25". Each hypocentral position is defined in relative terms using as a reference the upper left corner of the rupture and by specifying the fraction of rupture length and rupture width.
- A slip list. It is a list of the possible rupture slip directions [degrees], and their corresponding weights. The angle describing each slip direction is measured counter-clockwise using the fault strike direction as reference.

In near-fault PSHA calculations, the hypocentre list and the slip list are mandatory. The weights in each list must always sum to one. The available GMPE which currently supports the

near-fault directivity PSHA calculation in OQ- engine is the ChiouYoungs2014NearFaultEffect GMPE developed by Chiou and Youngs (2014) (associated with an `Active Shallow Crust` tectonic region type).

Below we provide two examples of simple fault source files. The first is an excerpt of an xml file used to describe the properties of a simple fault source and the second example shows the excerpt of an xml file used to describe the properties of a simple fault source that can be used to perform a PSHA calculation taking into account directivity effects.

```
1  <areaSource id="1" name="Quito" tectonicRegion="Active Shallow Crust">
2      <areaGeometry>
3          <gml:Polygon>
4              <gml:exterior>
5                  <gml:LinearRing>
6                      <gml:posList>
7                      -122.5 37.5
8                      -121.5 37.5
9                      -121.5 38.5
10                     -122.5 38.5
11                     </gml:posList>
12                 </gml:LinearRing>
13             </gml:exterior>
14         </gml:Polygon>
15         <upperSeismoDepth>0.0</upperSeismoDepth>
16         <lowerSeismoDepth>10.0</lowerSeismoDepth>
17     </areaGeometry>
18     <magScaleRel>PeerMSR</magScaleRel>
19     <ruptAspectRatio>1.5</ruptAspectRatio>
20     <incrementalMFD minMag="6.55" binWidth="0.1">
21         <occurRates>0.0010614989 8.8291627E-4 7.3437777E-4 6.108288E-4
22             5.080653E-4</occurRates>
23     </incrementalMFD>
24     <nodalPlaneDist>
25         <nodalPlane probability="0.3" strike="0.0" dip="90.0" rake="0.0"/>
26         <nodalPlane probability="0.7" strike="90.0" dip="45.0" rake="90.0"/>
27     </nodalPlaneDist>
28     <hypoDepthDist>
29         <hypoDepth probability="0.5" depth="4.0" />
30         <hypoDepth probability="0.5" depth="8.0" />
31     </hypoDepthDist>
32 </areaSource>
```

Below is an excerpt of a simple fault source xml file for near-fault directivity PSHA calculations:

```
1  <simpleFaultSource id="1" name="Mount Diablo Thrust"
2          tectonicRegion="Active Shallow Crust">
3      <simpleFaultGeometry>
```

```
 4          <gml:LineString>
 5              <gml:posList>
 6                  -121.82290 37.73010
 7                  -122.03880 37.87710
 8              </gml:posList>
 9          </gml:LineString>
10          <dip>45.0</dip>
11          <upperSeismoDepth>10.0</upperSeismoDepth>
12          <lowerSeismoDepth>20.0</lowerSeismoDepth>
13      </simpleFaultGeometry>
14      <magScaleRel>WC1994</magScaleRel>
15      <ruptAspectRatio>1.5</ruptAspectRatio>
16      <incrementalMFD minMag="5.0" binWidth="0.1">
17          <occurRates>0.0010614989 8.8291627E-4 7.3437777E-4 6.108288E-4
18                  5.080653E-4</occurRates>
19      </incrementalMFD>
20      <rake>30.0</rake>
21      <hypoList>
22          <hypo alongStrike="0.25" downDip="0.25" weight="0.25"/>
23          <hypo alongStrike="0.25" downDip="0.75" weight="0.25"/>
24          <hypo alongStrike="0.75" downDip="0.25" weight="0.25"/>
25          <hypo alongStrike="0.75" downDip="0.75" weight="0.25"/>
26      </hypoList>
27      <slipList>
28          <slip weight="0.333"> 0.0 </slip>
29          <slip weight="0.333"> 45.0 </slip>
30          <slip weight="0.334"> 90.0 </slip>
31      </slipList>
32  </simpleFaultSource>
```

As with the previous examples, the red text highlights the parameters used to specify the source geometry, the parameters in green describe the rupture mechanism, the text in blue describes the magnitude-frequency distribution and the gray text describes the rupture properties.

2.1.2.2 Complex faults

A complex fault differs from simple fault just by the way the geometry of the fault surface is defined and the fault surface is later created. The input parameters used to describe complex faults are, for the most part, the same used to describe the simple fault typology.

In case of complex faults the dip angle is not requested while the fault trace is substituted by two fault edges limiting at the top and bottom the fault surface. Additional curves lying over the fault surface can be specified to complement and refine the description of the fault surface geometry.

Usually, we use complex faults to model intraplate megathrust faults such as the big subduction structures active in the Pacific (Sumatra, South America, Japan) but this source

typology can be used also to create - for example - listric fault sources with a realistic
geometry.

```
1  <complexFaultSource id="1" name="Cascadia Megathrust"
2          tectonicRegion="Subduction Interface">
3      <complexFaultGeometry>
4          <faultTopEdge>
5              <gml:LineString>
6                  <gml:posList>
7                      -124.704   40.363   0.5493260E+01
8                      -124.977   41.214   0.4988560E+01
9                      -125.140   42.096   0.4897340E+01
10                 </gml:posList>
11             </gml:LineString>
12         </faultTopEdge>
13         <intermediateEdge>
14             <gml:LineString>
15                 <gml:posList>
16                     -124.704   40.363   0.5593260E+01
17                     -124.977   41.214   0.5088560E+01
18                     -125.140   42.096   0.4997340E+01
19                 </gml:posList>
20             </gml:LineString>
21         </intermediateEdge>
22         <intermediateEdge>
23             <gml:LineString>
24                 <gml:posList>
25                     -124.704   40.363   0.5693260E+01
26                     -124.977   41.214   0.5188560E+01
27                     -125.140   42.096   0.5097340E+01
28                 </gml:posList>
29             </gml:LineString>
30         </intermediateEdge>
31         <faultBottomEdge>
32             <gml:LineString>
33                 <gml:posList>
34                     -123.829   40.347   0.2038490E+02
35                     -124.137   41.218   0.1741390E+02
36                     -124.252   42.115   0.1752740E+02
37                 </gml:posList>
38             </gml:LineString>
39         </faultBottomEdge>
40     </complexFaultGeometry>
41     <magScaleRel>WC1994</magScaleRel>
42     <ruptAspectRatio>1.5</ruptAspectRatio>
43     <truncGutenbergRichterMFD aValue="-3.5" bValue="1.0" minMag="5.0"
44             maxMag="6.5" />
45     <rake>30.0</rake>
```

```
46   </complexFaultSource>
```

As with the previous examples, the red text highlights the parameters used to specify the source geometry, the parameters in green describe the rupture mechanism, the text in blue describes the magnitude-frequency distribution and the gray text describes the rupture properties.

2.1.3 Fault sources without floating ruptures

2.1.3.1 Characteristic faults

The characteristic fault source is a particular typology of fault created with the assumption that its ruptures will always cover the entire fault surface. As such, no floating is necessary on the surface. The characteristic fault may still take as input a magnitude frequency distribution. In this case, the fault surface can be represented either as a simple fault source surface or as a complex fault source surface or as a combination of rectangular ruptures as represented in Figure 2.4. Mutiple surfaces containing mixed geometry types are also supported.
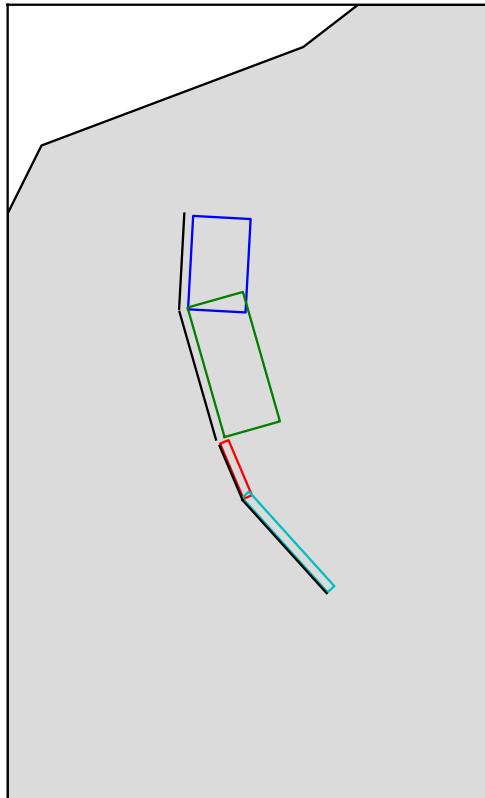
Figure 2.4 – *Geometry of a multi-segmented characteristic fault composed of four rectangular ruptures as modelled in OpenQuake.*

Source data

- The characteristic rupture surface is defined through one of the following options:
  - A list of rectangular ruptures ("planar surfaces")
  - A simple fault source geometry
  - A complex fault source geometry
- A frequency-magnitude distribution.
- Rake angle (specified following the Aki-Richards convention; see Aki and Richards (2002)).
- Upper and lower depth values limiting the seismogenic interval.

A comprehensive example enumerating the possible rupture surface configurations is shown below.

```
1    <characteristicFaultSource
2         id="5"
3         name="characteristic source, simple fault"
4         tectonicRegion="Volcanic">
5         <truncGutenbergRichterMFD aValue="-3.5" bValue="1.0"
6                                    minMag="5.0" maxMag="6.5" />
7         <rake>30.0</rake>
8         <surface>
9             <!-- Characteristic Fault with a simple fault surface -->
10            <simpleFaultGeometry>
11                <gml:LineString>
12                    <gml:posList>
13                        -121.82290 37.73010
14                        -122.03880 37.87710
15                    </gml:posList>
16                </gml:LineString>
17                <dip>45.0</dip>
18                <upperSeismoDepth>10.0</upperSeismoDepth>
19                <lowerSeismoDepth>20.0</lowerSeismoDepth>
20            </simpleFaultGeometry>
21        </surface>
22    </characteristicFaultSource>
23    <characteristicFaultSource
24         id="6"
25         name="characteristic source, complex fault"
26         tectonicRegion="Volcanic">
27         <incrementalMFD minMag="5.0" binWidth="0.1">
28             <occurRates>0.0010614989 8.8291627E-4 7.3437777E-4</occurRates>
29         </incrementalMFD>
30        <rake>60.0</rake>
31          <surface>
32             <!-- Characteristic source with a complex fault surface -->
33               <complexFaultGeometry>
34                   <faultTopEdge>
35                       <gml:LineString>
```

```
36                          <gml:posList>
37                              -124.704   40.363   0.5493260E+01
38                              -124.977   41.214   0.4988560E+01
39                              -125.140   42.096   0.4897340E+01
40                          </gml:posList>
41                      </gml:LineString>
42                  </faultTopEdge>
43                  <faultBottomEdge>
44                      <gml:LineString>
45                          <gml:posList>
46                              -123.829   40.347   0.2038490E+02
47                              -124.137   41.218   0.1741390E+02
48                              -124.252   42.115   0.1752740E+02
49                          </gml:posList>
50                      </gml:LineString>
51                  </faultBottomEdge>
52              </complexFaultGeometry>
53          </surface>
54      </characteristicFaultSource>
55      <characteristicFaultSource
56          id="7"
57          name="characteristic source, multi surface"
58          tectonicRegion="Volcanic">
59          <truncGutenbergRichterMFD aValue="-3.6" bValue="1.0"
60                                  minMag="5.2" maxMag="6.4" />
61          <rake>90.0</rake>
62          <surface>
63              <!-- Characteristic source with a collection of planar surfaces -->
64              <planarSurface strike="0.0" dip="90.0">
65                  <topLeft lon="-1.0" lat="1.0" depth="21.0" />
66                  <topRight lon="1.0" lat="1.0" depth="21.0" />
67                  <bottomLeft lon="-1.0" lat="-1.0" depth="59.0" />
68                  <bottomRight lon="1.0" lat="-1.0" depth="59.0" />
69              </planarSurface>
70              <planarSurface strike="20.0" dip="45.0">
71                  <topLeft lon="1.0" lat="1.0" depth="20.0" />
72                  <topRight lon="3.0" lat="1.0" depth="20.0" />
73                  <bottomLeft lon="1.0" lat="-1.0" depth="80.0" />
74                  <bottomRight lon="3.0" lat="-1.0" depth="80.0" />
75              </planarSurface>
76          </surface>
77      </characteristicFaultSource>
```

2.1.3.2 Non-Parametric Fault

The non-parametric fault typology requires that the user indicates the rupture properties (rupture surface, magnitude, rake and hypocentre) and the corresponding probabilities of the rupture. The probabilities are given as a list of floating point values that correspond to the probabilities of $0, 1, 2, \ldots\ldots N$ occurrences of the rupture within the specified investigation

time. Note that there is not, at present, any internal check to ensure that the investigation time to which the probabilities refer corresponds to that specified in the configuration file. As the surface of the rupture is set explicitly, no rupture floating occurs, and, as in the case of the characteristic fault source, the rupture surface can be defined as either a single planar rupture, a list of planar ruptures, a simple fault source geometry, a complex fault source geometry, or a combination of different geometries.

An comprehensive example enumerating the possible configurations is shown below:

```
1   <?xml version="1.0" encoding="utf-8"?>
2   <nrml
3   xmlns="http://openquake.org/xmlns/nrml/0.5"
4   xmlns:gml="http://www.opengis.net/gml"
5   >
6       <sourceModel name="Test Source Model">
7           <nonParametricSeismicSource id="1"
8           name="A Non Parametric Source" tectonicRegion="Some TRT">
9               <singlePlaneRupture probs_occur="0.544 0.456">
10                  <magnitude>8.3</magnitude>
11                  <rake>90.0</rake>
12                  <hypocenter depth="26.101" lat="40.726" lon="143.0"/>
13                  <planarSurface>
14                      <topLeft depth="9.0" lat="41.6" lon="143.1"/>
15                      <topRight depth="9.0" lat="40.2" lon="143.91"/>
16                      <bottomLeft depth="43.202" lat="41.252" lon="142.07"/>
17                      <bottomRight depth="43.202" lat="39.852" lon="142.91"/>
18                  </planarSurface>
19              </singlePlaneRupture>
20              <multiPlanesRupture probs_occur="0.9244 0.0756">
21                  <magnitude>6.9</magnitude>
22                  <rake>0.0</rake>
23                  <hypocenter depth="7.1423" lat="35.296" lon="139.31"/>
24                  <planarSurface>
25                      <topLeft depth="2.0" lat="35.363" lon="139.16"/>
26                      <topRight depth="2.0" lat="35.394" lon="138.99"/>
27                      <bottomLeft depth="14.728" lat="35.475" lon="139.19"/>
28                      <bottomRight depth="14.728" lat="35.505" lon="139.02"/>
29                  </planarSurface>
30                  <planarSurface>
31                      <topLeft depth="2.0" lat="35.169" lon="139.34"/>
32                      <topRight depth="2.0" lat="35.358" lon="139.17"/>
33                      <bottomLeft depth="12.285" lat="35.234" lon="139.45"/>
34                      <bottomRight depth="12.285" lat="35.423" lon="139.28"/>
35                  </planarSurface>
36              </multiPlanesRupture>
37              <simpleFaultRupture probs_occur="0.157 0.843">
38                  <magnitude>7.8</magnitude>
39                  <rake>90.0</rake>
40                  <hypocenter depth="22.341" lat="43.624" lon="147.94"/>
```

```
41                    <simpleFaultGeometry>
42                        <gml:LineString>
43                            <gml:posList>
44                                147.96 43.202
45                                148.38 43.438
46                                148.51 43.507
47                                148.68 43.603
48                                148.76 43.640
49                            </gml:posList>
50                        </gml:LineString>
51                        <dip>30.0</dip>
52                        <upperSeismoDepth>14.5</upperSeismoDepth>
53                        <lowerSeismoDepth>35.5</lowerSeismoDepth>
54                    </simpleFaultGeometry>
55                </simpleFaultRupture>
56                <complexFaultRupture probs_occur="0.157 0.843">
57                    <magnitude>7.8</magnitude>
58                    <rake>90.0</rake>
59                    <hypocenter depth="22.341" lat="43.624" lon="147.94"/>
60                    <complexFaultGeometry>
61                        <faultTopEdge>
62                            <gml:LineString>
63                                <gml:posList>
64                                    148.76 43.64 5.0
65                                    148.68 43.603 5.0
66                                    148.51 43.507 5.0
67                                    148.38 43.438 5.0
68                                    147.96 43.202 5.0
69                                </gml:posList>
70                            </gml:LineString>
71                        </faultTopEdge>
72                        <faultBottomEdge>
73                            <gml:LineString>
74                                <gml:posList>
75                                    147.92 44.002 35.5
76                                    147.81 43.946 35.5
77                                    147.71 43.897 35.5
78                                    147.5 43.803 35.5
79                                    147.36 43.727 35.5
80                                </gml:posList>
81                            </gml:LineString>
82                        </faultBottomEdge>
83                    </complexFaultGeometry>
84                </complexFaultRupture>
85            </nonParametricSeismicSource>
86        </sourceModel>
87  </nrml>
```

2.2   Magnitude-frequency distributions

The magnitude-frequency distributions currently supported by the oq-engine are the following:

A discrete incremental magnitude-frequency distribution   It is the simplest distribution supported. It is defined by the minimum value of magnitude (representing the mid point of the first bin) and the bin width. The distribution itself is simply a sequence of floats describing the annual number of events for different bins. The maximum magnitude admitted by this magnitude-frequency distribution is just the sum of the minimum magnitude and the product of the bin width by the number annual rate values.

Below we provide an example of the xml that should be incorporated in a seismic source description in order to define this Magnitude-Frequency Distribution (MFD). An additional example for this distribution can be also found at page 28.

```
<incrementalMFD minMag="5.05" binWidth="0.1">
    <occurRates>0.15 0.08 0.05 0.03 0.015</occurRates>
</incrementalMFD>
```

The magnitude-frequency distribution obtained with the above parameters is represented in Figure 2.5.
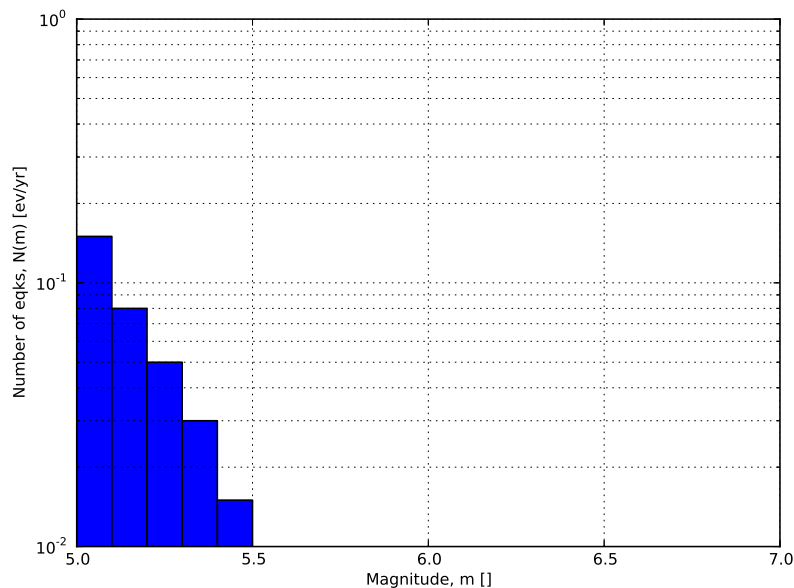
Figure 2.5 – *Example of an incremental magnitude-frequency distribution.*

A double truncated Gutenberg-Richter distribution   This distribution is described by means of a minimum minMag and maximum magnitude maxMag and by the a and b values of the Gutenberg-Richter relationship.

The syntax of the xml used to describe this magnitude-frequency distribution is rather compact as demonstrated in the following example:

```
<truncGutenbergRichterMFD aValue="5.0" bValue="1.0" minMag="5.0"
        maxMag="6.0"/>
```

Figure 2.6 shows the magnitude-frequency distribution obtained using the parameters of the considered example.
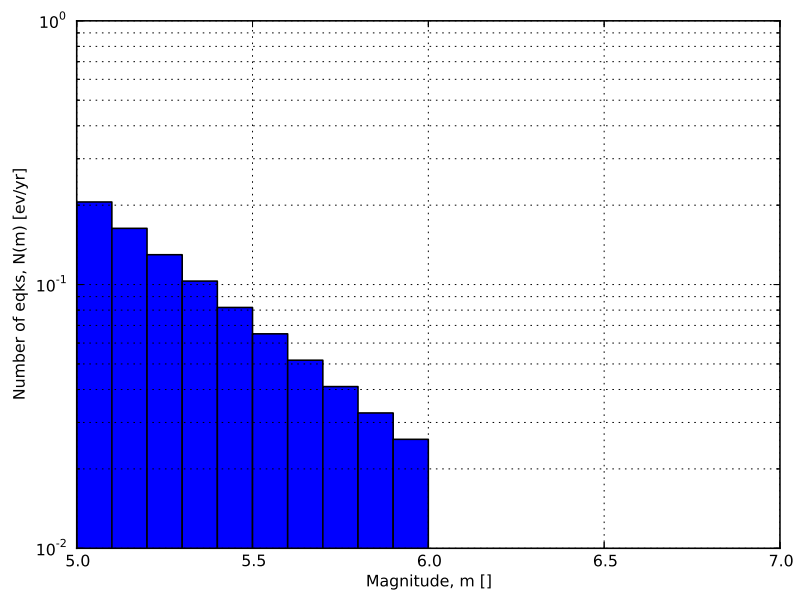
Figure 2.6 – *Example of a double truncated Gutenberg-Richter magnitude-frequency distribution.*

Hybrid Characteristic earthquake model (à la Youngs and Coppersmith, 1985) The hybrid characteristic earthquake model, presented by Youngs and Coppersmith, 1985, distributes seismic moment proportionally between a characteristic model (for larger magnitudes) and an exponential model. The rate of events is dependent on the magnitude of the characteristic earthquake, the b-value and the total moment rate of the system (Figure 2.7). However, the total moment rate may be defined in one of two ways. If the total moment rate of the source is known, as may be the case for a fault surface with known area and slip rate, then the distribution can be defined from the total moment rate (in N-m) of the source directly. Alternatively, the distribution can be defined from the rate of earthquakes in the characteristic bin, which may be preferable if the distribution is determined from observed seismicity behaviour. The option to define the distribution according to the total moment rate is input as:

```
    <YoungsCoppersmithMFD minmag="5.0" bValue="1.0" binWidth="0.1"
        characteristicMag="7.0" totalMomentRate="1.05E19"/>
```

whereas the option to define the distribution from the rate of the characteristic events is given as:

```
<YoungsCoppersmithMFD minmag="5.0" bValue="1.0" binWidth="0.1"
    characteristicMag="7.0" characteristicRate="0.005"/>
```

Note that in this distribution the width of the magnitude bin must be defined explicitly in the model.
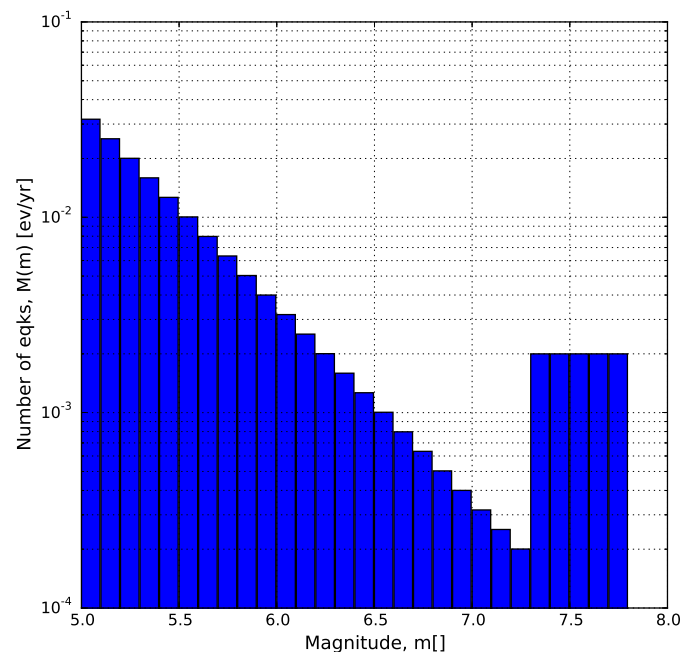
Figure 2.7 – *Youngs and Coppersmith, 1985 magnitude-frequency distribution.*

"Arbitrary" Magnitude Frequency Distribution The arbitrary magnitude frequency distribution is another non-parametric form of MFD, in which the rates are defined explicitly. Here, the magnitude frequency distribution is defined by a list of magnitudes and their corresponding rates of occurrence. There is no bin-width as the rates correspond exactly to the specific magnitude. Unlike the evenly discretised MFD, there is no requirement that the magnitudes be equally spaced. This distribution (illustrated in Figure 2.8) can be input as:

```
<arbitraryMFD>
    <occurRates>0.12 0.036 0.067 0.2</occurRates>
    <magnitudes>8.1 8.47 8.68 9.02</magnitude>
</arbitraryMFD>
```
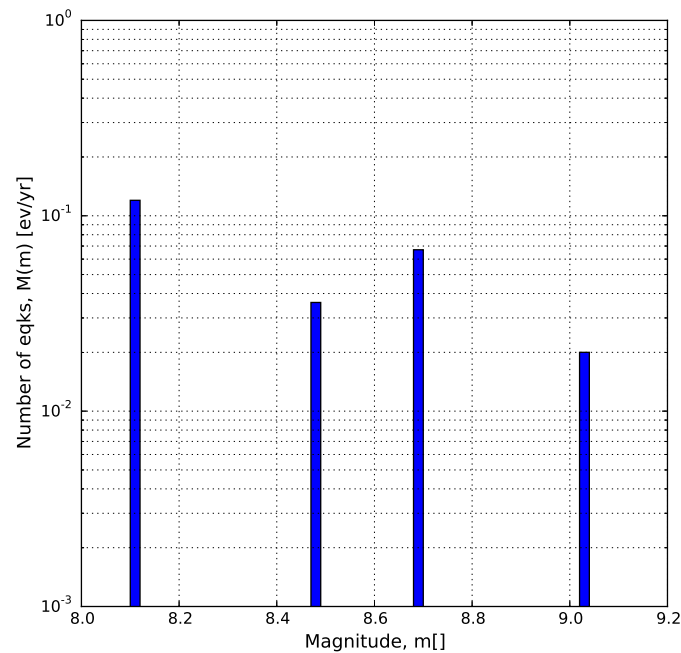
Figure 2.8 – *"Arbitrary" magnitude-frequency distribution.*

2.3 Magnitude-scaling relationships

We provide below a list of the magnitude-area scaling relationships implemented in the oq-hazardlib:

2.3.1 Relationships for shallow earthquakes in active tectonic regions

- Wells and Coppersmith, 1994 - One of the most well known magnitude scaling relationships, based on a global database of historical earthquake ruptures. The implemented relationship is the one linking magnitude to rupture area, and is called with the keyword `WC1994`

2.3.2 Magnitude-scaling relationships for subduction earthquakes

- Strasser et al., 2010 - Defines several magnitude scaling relationships for interface and in-slab earthquakes. Only the magnitude to rupture-area scaling relationships are implemented here, and are called with the keywords `StrasserInterface` and `StrasserIntraslab` respectively.

2.3.3 Magnitude-scaling relationships stable continental regions

- EPRI, 2011 - Defines a single magnitude to rupture-area scaling relationship for use in the central and eastern United States: $Area = 10.0^{M_W - 4.336}$. It is called with the keyword `CEUS2011`

2.3.4 Miscellaneous Magnitude-Scaling Relationships

- `PeerMSR` defines a simple magnitude scaling relation used as part of the Pacific Earthquake Engineering Research Center verification of probabilistic seismic hazard analysis programs: $Area = 10.0^{M_w - 4.0}$.

- `PointMSR` approximates a 'point' source by returning an infinitesimally small area for all magnitudes. Should only be used for distributed seismicity sources and not for fault sources.

2.4 Calculation workflows

The hazard component of the OpenQuake-engine can compute seismic hazard using various approaches. Three types of analysis are currently supported:

- *Classical Probabilistic Seismic Hazard Analysis (PSHA)*, allowing calculation of hazard curves and hazard maps following the classical integration procedure (Cornell, 1968, McGuire (1976)) as formulated by Field et al., 2003.

- *Event-Based Probabilistic Seismic Hazard Analysis*, allowing calculation of ground-motion fields from stochastic event sets. Traditional results - such as hazard curves - can be obtained by post- processing the set of computed ground-motion fields.

- *Scenario Based Seismic Hazard Analysis (SSHA)*, allowing the calculation of ground motion fields from a single earthquake rupture scenario taking into account ground-motion aleatory variability.

Each workflow has a modular structure, so that intermediate results can be exported and analyzed. Each calculator can be extended independently of the others so that additional calculation options and methodologies can be easily introduced, without affecting the overall calculation workflow.

2.4.1 Classical Probabilistic Seismic Hazard Analysis

Input data for the classical Probabilistic Seismic Hazard Analysis (PSHA) consist of a PSHA input model provided together with calculation settings.

The main calculators used to perform this analysis are the following:

1. *Logic Tree Processor*

   The Logic Tree Processor (LTP) takes as an input the PSHA Input Model and creates a Seismic Source Model. The LTP uses the information in the Initial Seismic Source Models and the Seismic Source Logic Tree to create a Seismic Source Input Model (i.e. a model describing geometry and activity rates of each source without any epistemic uncertainty).

   Following a procedure similar to the one just described the Logic Tree Processor creates a Ground Motion model (i.e. a data structure that associates to each tectonic region considered in the calculation a Ground Motion Prediction Equation (GMPE)).

2. *Earthquake Rupture Forecast Calculator*

The produced Seismic Source Input Model becomes an input information for the Earthquake Rupture Forecast (ERF) calculator which creates a list earthquake ruptures admitted by the source model, each one characterized by a probability of occurrence over a specified time span.

3. *Classical PSHA Calculator*

   The classical PSHA calculator uses the ERF and the Ground Motion model to compute hazard curves on each site specified in the calculation settings.

2.4.2 Event-Based Probabilistic Seismic Hazard Analysis

Input data for the Event-Based PSHA - as in the case of the Classical PSHA calculator - consists of a PSHA Input Model and a set of calculation settings.

   The main calculators used to perform this analysis are:

1. *Logic Tree Processor*

   The Logic Tree Processor works in the same way described in the description of the Classical PSHA workflow (see Section 2.4.1 at page 40).

2. *Earthquake Rupture Forecast Calculator*

   The Earthquake Rupture Forecast Calculator was already introduced in the description of the PSHA workflow (see Section 2.4.1 at page 40).

3. *Stochastic Event Set Calculator*

   The Stochastic Event Set Calculator generates a collection of stochastic event sets by sampling the ruptures contained in the ERF according to their probability of occurrence. A Stochastic Event Set (SES) thus represents a potential realisation of the seismicity (i.e. a list of ruptures) produced by the set of seismic sources considered in the analysis over the time span fixed for the calculation of hazard.

4. *Ground Motion Field Calculator*

   The Ground Motion Field Calculator computes for each event contained in a Stochastic Event Set a realization of the geographic distribution of the shaking by taking into account the aleatory uncertainties in the ground- motion model. Eventually, the Ground Motion Field calculator can consider the spatial correlation of the ground-motion during the generation of the Ground Motion Field (GMF).

5. *Event-based PSHA Calculator*

   The event-based PSHA calculator takes a (large) set of ground-motion fields representative of the possible shaking scenarios that the investigated area can experience over a (long) time span and for each site computes the corresponding hazard curve.

   This procedure is computationally intensive and is not recommended for investigating the hazard over large areas.

2.4.3 Scenario based Seismic Hazard Analysis

In case of SSHA, the input data consist of a single earthquake rupture model and one or more ground-motion models. Using the Ground Motion Field Calculator, multiple realizations of

ground shaking can be computed, each realization sampling the aleatory uncertainties in the ground-motion model.

The main calculators used to perform this analysis are:

1. *Ground Motion Field Calculator*

    The Ground Motion Field Calculator was already introduced during the description of the event based PSHA workflow (see Section 2.4.2 at page 41).

3. Using the Hazard Module

This Chapter summarises the structure of the information necessary to define a PSHA input model to be used with the OpenQuake-engine.

Input data for probabilistic based seismic hazard analysis (Classical, Event based, Disaggregation, and UHS) are organised into:

- A general configuration file.
- A file describing the Seismic Source System, that is the set of initial source models and associated epistemic uncertainties needed to model the seismic activity in the region of interest.
- A file describing the Ground Motion System, that is the set of ground motion prediction equations, per tectonic region type, needed to model the ground motion shaking in the region of interest.

Figure 3.1 summarises the structure of a PSHA input model for the OpenQuake-engine and the relationships between the different files.

3.1 Defining Logic Trees

The main components of a logic tree structure in the OpenQuake-engine are the following:

branch : the simplest component of a logic tree structure. A branch represent a possible interpretation of a value assignment for a specific type of uncertainty. It is fully described by the tuple (parameter or model, weight).

branching set : it is a key component in the logic tree structure used by the oq-engine. It groups a set of branches i.e. alternative interpretations of a parameter or a model. Each branching set is defined by:

- An ID
- An uncertainty type (for a comprehensive list of the types of uncertainty currently supported see page 46)
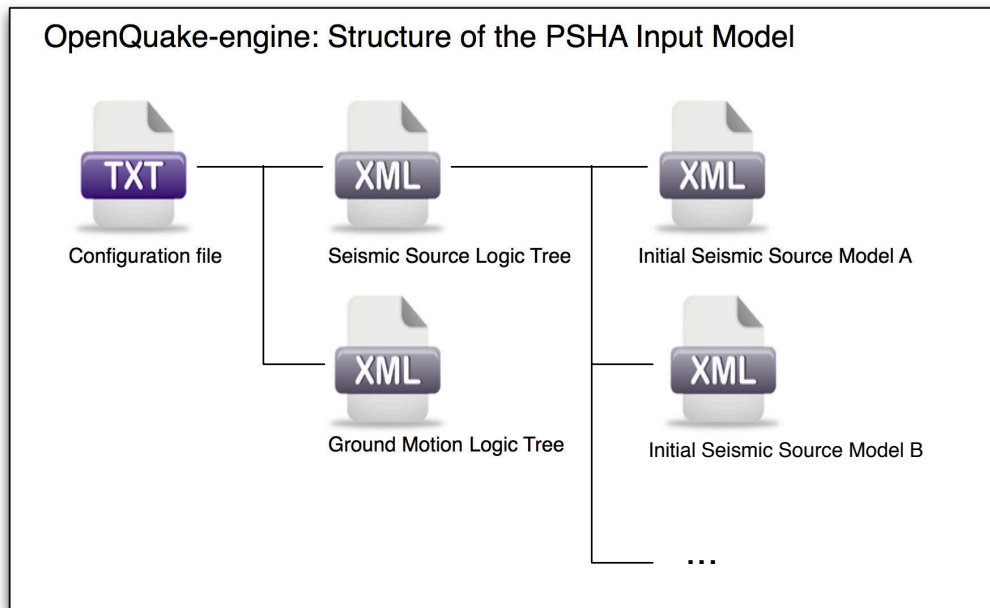
Figure 3.1 – *PSHA Input Model structure*

> • One or more branches

This set of uncertainties can be applied to the whole initial seismic source input model or just to a subset of seismic sources. The sum of the weights/probabilities assigned to the set of branches always correspond to one.

branching level : it is the largest container. It is not used in modelling uncertainty, but it is useful in maintaining a logic and an order in the structure of the tree.

Below we provide a simple schema illustrating the skeleton of xml file containing the desciption of a logic tree:

```
<logicTreeBranchingLevel branchingLevelID=ID>
    <logicTreeBranchSet branchSetID=ID
            uncertaintyType=TYPE>
        <logicTreeBranch>
            <uncertaintyModel>VALUE</uncertaintyModel>
            <uncertaintyWeight>WEIGHT</uncertaintyWeight>
        </logicTreeBranch>
    </logicTreeBranchSet>
</logicTreeBranchingLevel>
```

As it appears from this example, the structure of a logic tree is a set of nested elements.

A schematic representation of the elemental components of a logic tree structure is provided in Figure 3.2. A branching level identifies the position where branching occurs while a branch set identifies a collection of branches (i.e. individual branches) whose weights sum to 1.
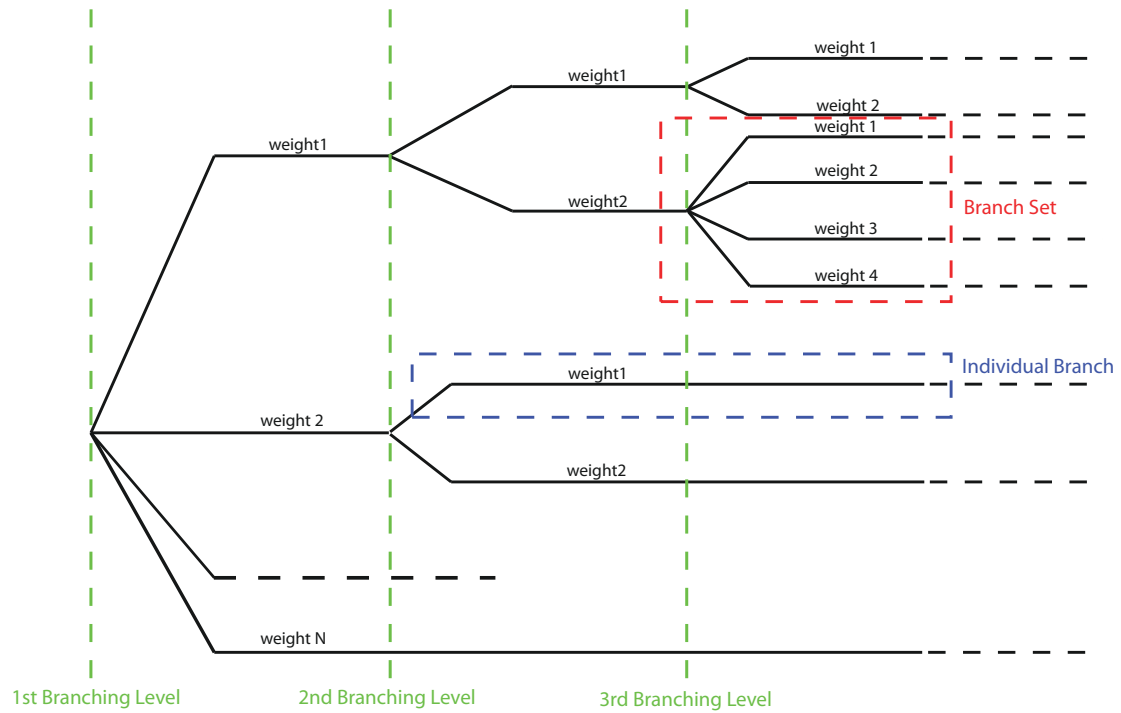
Figure 3.2 – *Generic Logic Tree structure as described in terms of branching levels, branch sets, and individual branches.*

3.1.1 Logic trees as described in the nrml schema

In the NRML schema, a logic tree structure is defined through the `logicTree` element:

```
<logicTree logicTreeID="ID">
...
</logicTree>
```

A `logicTree` contains as a sequence of `logicTreeBranchingLevel` elements. The position in the sequence of a `logicTreeBranchingLevel` specifies the level of the tree where it is located. That is, the first `logicTreeBranchingLevel` element in the sequence represents the first level in the tree, the second element the second level in the tree, and so on.

```
<logicTree logicTreeID="ID">
        <logicTreeBranchingLevel branchingLevelID="ID_1">
                ...
        </logicTreeBranchingLevel>
        <logicTreeBranchingLevel branchingLevelID="ID_2">
                ...
        </logicTreeBranchingLevel>
        ....
```

```
        <logicTreeBranchingLevel branchingLevelID="ID_N">
                ...
        </logicTreeBranchingLevel>
</logicTree>
```

There are no restrictions on the number of tree levels that can be defined.

A `logicTreeBranchingLevel` is defined as a sequence of `logicTreeBranchSet` elements where each `logicTreeBranchSet` defines a particular epistemic uncertainty inside a branching level.

A branch set has two required attributes: `branchSetID` and `uncertaintyType`. The latter defines the type of epistemic uncertainty this branch set is describing.

```
<logicTree logicTreeID="ID">
...
        <logicTreeBranchingLevel branchingLevelID="ID_#">
                <logicTreeBranchSet branchSetID="ID_1"
                        uncertaintyType="UNCERTAINTY_TYPE">
                        ...
                </logicTreeBranchSet>
                <logicTreeBranchSet branchSetID="ID_2"
                        uncertaintyType="UNCERTAINTY_TYPE">
                        ...
                </logicTreeBranchSet>
                ...
                <logicTreeBranchSet branchSetID="ID_N"
                        uncertaintyType="UNCERTAINTY_TYPE">
                        ...
                </logicTreeBranchSet>
        </logicTreeBranchingLevel>
...
</logicTree>
```

Possible values for the `uncertaintyType` attribute are:
- `gmpeModel`: indicates epistemic uncertainties on ground motion prediction equations
- `sourceModel`: indicates epistemic uncertainties on source models
- `maxMagGRRelative`: indicates relative (i.e. increments) epistemic uncertainties to be added (or subtracted, depending on the sign of the increment) to the Gutenberg-Richter maximum magnitude value.
- `bGRRelative`: indicates relative epistemic uncertainties to be applied to the Gutenberg-Richter b value.

- `abGRAbsolute`: indicates absolute (i.e. values used to replace original values) epistemic uncertainties on the Gutenberg-Richter a and b values.
- `maxMagGRAbsolute`: indicates (absolute) epistemic uncertainties on the Gutenberg-Richter maximum magnitude.
- `incrementalMFDAbsolute`: indicates (absolute) epistemic uncertainties on the incremental magnitude frequency distribution (i.e. alternative rates and/or minimum magnitude) of a specific source (can only be applied to individual sources)
- `simpleFaultGeometryAbsolute`: indicates alternative representations of the simple fault geometry for an individual simple fault source
- `simpleFaultDipRelative`: indicates a relative increase or decrease in fault dip for one or more simple fault sources
- `simpleFaultDipAbsolute`: indicates alternative values of fault dip for one or more simple fault sources
- `complexFaultGeometryAbsolute`: indicates alternative representations of complex fault geometry for an individual complex fault source
- `characteristicFaultGeometryAbsolute`: indicates alternative representations of the characteristic fault geometry for an individual characteristic fault source

There are no restrictions on the number of branch sets that can be defined inside a branching level.

A `branchSet` is defined as a sequence of `logicTreeBranch` elements, each specified by an `uncertaintyModel` element (a string identifying an uncertainty model; the content of the string varies with the `uncertaintyType` attribute value of the branchSet element) and the `uncertaintyWeight` element (specifying the probability/weight associated to the uncertaintyModel):

```
<logicTree logicTreeID="ID">
...
        <logicTreeBranchingLevel branchingLevelID="ID_#">
                ...
                <logicTreeBranchSet branchSetID="ID_#"
                                uncertaintyType="UNCERTAINTY_TYPE">
                        <logicTreeBranch branchID="ID_1">
                                <uncertaintyModel>
                                UNCERTAINTY_MODEL
                                </uncertaintyModel>
                                <uncertaintyWeight>
                                UNCERTAINTY_WEIGHT
                                </uncertaintyWeight>
                        </logicTreeBranch>
                                ...
```

```
                        <logicTreeBranch branchID="ID_N">
                                <uncertaintyModel>
                                UNCERTAINTY_MODEL
                                </uncertaintyModel>
                                <uncertaintyWeight>
                                UNCERTAINTY_WEIGHT
                                </uncertaintyWeight>
                        </logicTreeBranch>
                </logicTreeBranchSet>
                ...
        </logicTreeBranchingLevel>
...
</logicTree>
```

Depending on the `uncertaintyType` the content of the `<uncertaintyModel>` element changes:

- if `uncertaintyType="gmpeModel"`, the uncertainty model contains the name of a ground motion prediction equation (a list of available GMPEs are given in Appendix **??**), e.g.:

```
<uncertaintyModel>GMPE_NAME</uncertaintyModel>
```

- if `uncertaintyType="sourceModel"`, the uncertainty model contains the paths to a source model file, e.g.:

```
<uncertaintyModel>SOURCE_MODEL_FILE_PATH</uncertaintyModel>
```

- if `uncertaintyType="maxMagGRRelative"`, the uncertainty model contains the increment to be added (or subtracted, depending on the sign) to the Gutenberg-Richter maximum magnitude:

```
<uncertaintyModel>MAX_MAGNITUDE_INCREMENT</uncertaintyModel>
```

- if `uncertaintyType="bGRRelative"`, the uncertainty model contains the increment to be added (or subtracted, depending on the sign) to the Gutenberg-Richter b value:

```
<uncertaintyModel>B_VALUE_INCREMENT</uncertaintyModel>
```

- if `uncertaintyType="abGRAbsolute"`, the uncertainty model must contain one a and b pair:

```
<uncertaintyModel>A_VALUE B_VALUE</uncertaintyModel>
```

- if uncertaintyType="maxMagGRAbsolute", the uncertainty model must contain
  one Gutenberg-Richter maximum magnitude value:

```
<uncertaintyModel>MAX_MAGNITUDE</uncertaintyModel>
```

- if uncertaintyType="incrementalMFDAbsolute", the uncertainty model must
  contain an instance of the incremental MFD node:

```
<uncertaintyModel>
    <incrementalMFD
        minMag="MIN MAGNITUDE"
        binWidth="BIN WIDTH">
        <occurRates>RATE_1 RATE_2 ... RATE_N</occurRates>
    </incrementalMFD>
</uncertaintyModel>
```

- if uncertaintyType="simpleFaultGeometryAbsolute" then the uncertainty model
  must contain a *valid* instance of the simpleFaultGeometry node as described in
  section 2.1.2.1
- if uncertaintyType="simpleFaultDipRelative" then the uncertainty model
  must specify the number of degrees to increase (positive) or decrease (negative) the
  fault dip. Note that if this increase results in an adjusted fault dip greater than 90° or
  less than 0° an error will occur.

```
<uncertaintyModel>DIP_INCREMENT</uncertaintyModel>
```

- if uncertaintyType="simpleFaultDipAbsolute" then the uncertainty model
  must specify the dip angle (in degrees)

```
<uncertaintyModel>DIP</uncertaintyModel>
```

- if uncertaintyType="complexFaultGeometryAbsolute" then the uncertainty
  model must contain a *valid* instance of the complexFaultGeometry source node as
  described in section 2.1.2.2
- if uncertaintyType="characteristicFaultGeometryAbsolute" then the un-
  certainty model must contain a *valid* instance of the characteristicFaultGeometry
  source node, as described in section 2.1.3.1

There are no restrictions on the number of logicTreeBranch elements that can be
defined in a logicTreeBranchSet, as long as the uncertainty weights sum to 1.0.

The logicTreeBranchSet element offers also a number of optional attributes allowing
for complex tree definitions:

- applyToBranches: specifies to which logicTreeBranch elements (one or more),
  in the previous branching level, the branch set is linked to. The linking is established

by defining the IDs of the branches to link to:

```
            applyToBranches="branchID1 branchID2 .... branchIDN"
```

The default is the keyword ALL, which means that a branch set is by default linked to all branches in the previous branching level. By specifying one or more branches to which the branch set links to, non- symmetric logic trees can be defined.

- `applyToSources`: specifies to which source in a source model the uncertainty applies to. Sources are specified in terms of their IDs:

```
            applyToSources="srcID1 srcID2 .... srcIDN"
```

- `applyToSourceType`: specifies to which source type the uncertainty applies to. Only one source typology can be defined (`area`, `point`, `simpleFault`, `complexFault`), e.g.:

```
            applyToSources="area"
```

- `applyToTectonicRegionType`: specifies to which tectonic region type the uncertainty applies to. Only one tectonic region type can be defined (`Active Shallow Crust`, `Stable Shallow Crust`, `Subduction Interface`, `Subduction IntraSlab`, `Volcanic`), e.g.:

```
            applyToTectonicRegionType="Active Shallow Crust"
```

3.2 The Seismic Source System

The Seismic Source System contains the model (or the models) describing position, geometry and activity of seismic sources of engineering importance for a set of sites as well as the possible epistemic uncertainties to be incorporated into the calculation of seismic hazard.

3.2.1 The Seismic Source Logic Tree

The structure of the Seismic Source Logic Tree consists of at least one branching level. This branching level is the one used to define the initial seismic source input model (or a number of initial seismic source models, see Figure 3.1).

The example provided below shows the simplest Seismic Source Logic Tree structure that can be defined in a PSHA input model for oq-engine. It's a logic tree with just one branching level containing one branch set with one branch used to define the initial seismic source model (its weight will be equal to one).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <nrml xmlns:gml="http://www.opengis.net/gml"
3        xmlns="http://openquake.org/xmlns/nrml/0.5">
```

```
4        <logicTree logicTreeID="lt1">
5            <logicTreeBranchingLevel branchingLevelID="bl1">
6                <logicTreeBranchSet uncertaintyType="sourceModel"
7                                    branchSetID="bs1">
8                    <logicTreeBranch branchID="b1">
9                        <uncertaintyModel>seismic_source_model.xml
10                       </uncertaintyModel>
11                       <uncertaintyWeight>1.0</uncertaintyWeight>
12                   </logicTreeBranch>
13               </logicTreeBranchSet>
14           </logicTreeBranchingLevel>
15       </logicTree>
16   </nrml>
```

The optional branching levels will contain rules that modify parameters of the sources in the initial seismic source model.

For example, if the epistemic uncertainties to be considered are source geometry and maximum magnitude, the modeller can create a logic tree structure with three initial seismic source models (each one exploring a different definition of the geometry of sources) and one branching level accounting for the epistemic uncertainty on the maximum magnitude.

Below we provide an example of such logic tree structure:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <nrml xmlns:gml="http://www.opengis.net/gml"
3        xmlns="http://openquake.org/xmlns/nrml/0.5">
4      <logicTree logicTreeID="lt1">
5
6          <logicTreeBranchingLevel branchingLevelID="bl1">
7              <logicTreeBranchSet uncertaintyType="sourceModel"
8                                  branchSetID="bs1">
9                  <logicTreeBranch branchID="b1">
10                     <uncertaintyModel>seismic_source_model_A.xml
11                     </uncertaintyModel>
12                     <uncertaintyWeight>0.2</uncertaintyWeight>
13                 </logicTreeBranch>
14                 <logicTreeBranch branchID="b2">
15                     <uncertaintyModel>seismic_source_model_B.xml
16                     </uncertaintyModel>
17                     <uncertaintyWeight>0.3</uncertaintyWeight>
18                 </logicTreeBranch>
19                 <logicTreeBranch branchID="b3">
20                     <uncertaintyModel>seismic_source_model_C.xml
21                     </uncertaintyModel>
22                     <uncertaintyWeight>0.5</uncertaintyWeight>
```

```
23                    </logicTreeBranch>
24                </logicTreeBranchSet>
25            </logicTreeBranchingLevel>
26
27            <logicTreeBranchingLevel branchingLevelID="bl2">
28                <logicTreeBranchSet branchSetID="bs21"
29                        uncertaintyType="maxMagGRRelative">
30                    <logicTreeBranch branchID="b211">
31                        <uncertaintyModel>+0.0</uncertaintyModel>
32                        <uncertaintyWeight>0.6</uncertaintyWeight>
33                    </logicTreeBranch>
34                    <logicTreeBranch branchID="b212">
35                        <uncertaintyModel>+0.5</uncertaintyModel>
36                        <uncertaintyWeight>0.4</uncertaintyWeight>
37                    </logicTreeBranch>
38                </logicTreeBranchSet>
39            </logicTreeBranchingLevel>
40
41        </logicTree>
42  </nrml>
```

Note that the uncertainty on the maximum magnitude is specified in terms of relative increments with respect to the initial maximum magnitude defined for each source in the initial seismic source models.

3.2.2 The Seismic Source Model

The structure of the xml file representing the seismic source model corresponds to a list of sources, each one modelled using one out of the five typologies currently supported. Below we provide a schematic example of a seismic source model:

```
<sourceModel gml:id="ID">
        ...
        <areaSource gml:id="SOURCE_ID">
                <gml:name>SOURCE_NAME</gml:name>
                <tectonicRegion>TECT_REGION_TYPE</tectonicRegion>
                ...
        </areaSource>
        ...
        <pointSource gml:id="SOURCE_ID">
                <gml:name>SOURCE_NAME</gml:name>
                <tectonicRegion>TECT_REGION_TYPE</tectonicRegion>
                ...
        </pointSource>
        ...
```

```
        <simpleFaultSource gml:id="SOURCE_ID">
                <gml:name>SOURCE_NAME</gml:name>
                <tectonicRegion>TECT_REGION_TYPE</tectonicRegion>
                ...
        </simpleFaultSource>
        ...
        <complexFaultSource gml:id="SOURCE_ID">
                <gml:name>SOURCE_NAME</gml:name>
                <tectonicRegion>TECT_REGION_TYPE</tectonicRegion>
                ...
        </complexFaultSource>
        ...
</sourceModel>
```

3.3 The Ground Motion System

The Ground Motion System defines the models and the possible epistemic uncertainties related to ground motion modelling to be incorporated into the calculation.

3.3.1 The Ground Motion Logic Tree

The structure of the ground-motion logic tree consists of a list of ground motion prediction equations for each tectonic region used to characterise the sources in the PSHA input model.

The example below shows a simple ground-motion logic tree. This logic tree assumes that all the sources in the PSHA input model belong to "Active Shallow Crust" and uses for calculation the Chiou and Youngs (2008) GMPE.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <nrml xmlns:gml="http://www.opengis.net/gml"
3        xmlns="http://openquake.org/xmlns/nrml/0.5">
4      <logicTree logicTreeID='lt1'>
5          <logicTreeBranchingLevel branchingLevelID="bl1">
6              <logicTreeBranchSet uncertaintyType="gmpeModel"
7                      branchSetID="bs1"
8                      applyToTectonicRegionType="Active Shallow Crust">
9
10             <logicTreeBranch branchID="b1">
11                 <uncertaintyModel>
12                 ChiouYoungs2008
13                 </uncertaintyModel>
14                 <uncertaintyWeight>1.0</uncertaintyWeight>
15             </logicTreeBranch>
16
17         </logicTreeBranchSet>
```

```
18          </logicTreeBranchingLevel>
19       </logicTree>
20   </nrml>
```

3.4   Configuration file

The configuration file is the primary file controlling both the definition of the input model as well as parameters governing the calculation. We illustrate in the following different examples of the configuration file addressing different typologies of seismic hazard calculation.

3.4.1   Classical PSHA

In the following we describe the overall structure and the most typical parameters of a configuration file to be used for the computation of a seismic hazard map using a classical PSHA methodology.

Calculation type and model info

```
1   [general]
2   description = A demo OpenQuake-engine .ini file for classical PSHA
3   calculation_mode = classical
4   random_seed = 1024
```

In this section the user specifies the following parameters:

- `description`: a parameter that can be used to designate the model
- `calculation_mode`: it is used to set the kind of calculation. In this case it corresponds to `classical`. Alternative options for the calculation_mode are described later in this manual.
- `random_seed`: is used to control the random generator so that when Monte Carlo procedures are used calculations are replicable (if the same `random_seed` is used you get exactly the same results).

Geometry of the area (or the sites) where hazard is computed

This section is used to specify where the hazard will be computed. Two options are available:

The first ooption is to define a polygon (usually a rectangle) and a distance (in km) to be used to discretize the polygon area. The polygon is defined by a list of longitude-latitude tuples.

An example is provided below:

```
5   [geometry]
6   region = 10.0 43.0, 12.0 43.0, 12.0 46.0, 10.0 46.0
7   region_grid_spacing = 10.0
```

The second option allows the definition of a number of sites where the hazard will be computed. An example is provided below:

```
5  [geometry]
6  sites = 10.0 43.0, 12.0 43.0, 12.0 46.0, 10.0 46.0
```

If the list of sites is too long the user can specify the name of a csv file as shown below:

```
5  [geometry]
6  sites_csv = <name_of_the_csv_file>
```

The format of the csv file containing the list of sites is a sequence of points (one per row) specified in terms of the longitude, latitude tuple. An example is provided below:

```
1  179.0,90.0
2  178.0,89.0
3  177.0,88.0
```

Logic tree sampling

The oq-engine provides two options for processing the whole logic tree structure. The first option uses Montecarlo sampling; the user in this case specifies a number of realizations.

In the second option all the possible realizations are created. Below we provide an example for the latter option. In this case we set the `number_of_logic_tree_samples` to 0. oq-engine will perform a complete enumeration of all the possible paths from the roots to the leaves of the logic tree structure.

```
9   [logic_tree]
10  number_of_logic_tree_samples = 0
```

If the seismic source logic tree and the ground motion logic tree do not contain epistemic uncertainties the engine will create a single PSHA input.

Generation of the earthquake rupture forecast

```
11  [erf]
12  rupture_mesh_spacing = 5
13  width_of_mfd_bin = 0.1
14  area_source_discretization = 10
```

This section of the configuration file is used to specify the level of discretization of the mesh representing faults, the grid used to delineate the area sources and, the magnitude-frequency distribution. Note that the smaller is the mesh spacing (or the bin width) the larger are (1) the precision in the calculation and (2) the computation demand.

Parameters describing site conditions

```
17  [site_params]
18  reference_vs30_type = measured
19  reference_vs30_value = 760.0
20  reference_depth_to_2pt5km_per_sec = 5.0
21  reference_depth_to_1pt0km_per_sec = 100.0
```

In this section the user specifies local soil conditions. The simplest solution is to define uniform site conditions (i.e. all the sites have the same characteristics).

Alternatively it is possible to define spatially variable soil properties in a separate file; the engine will then assign to each investigation location the values of the closest point used to specify site conditions.

```
17  [site_params]
18  site_model_file = site_model.xml
```

The file containing the site model has the following structure:

```
<?xml version="1.0" encoding="utf-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.4">
    <siteModel>
        <site lon="10.0" lat="40.0" vs30="800.0"
            vs30Type="inferred"
            z1pt0="19.367196734" z2pt5="0.588625072259" />
        <site lon="10.1" lat="40.0" vs30="800.0"
            vs30Type="inferred"
            z1pt0="19.367196734" z2pt5="0.588625072259" />
        <site lon="10.2" lat="40.0" vs30="800.0"
            vs30Type="inferred"
            z1pt0="19.367196734" z2pt5="0.588625072259" />
        <site lon="10.3" lat="40.0" vs30="800.0"
            vs30Type="inferred"
            z1pt0="19.367196734" z2pt5="0.588625072259" />
        <site lon="10.4" lat="40.0" vs30="800.0"
            vs30Type="inferred"
            z1pt0="19.367196734" z2pt5="0.588625072259" />
        ...
    </siteModel>
</nrml>
```

If the closest available site with soil conditions is at a distance greater than 5 km from the investigation location, a warning is generated.

Calculation configuration

```
22  [calculation]
23  source_model_logic_tree_file = source_model_logic_tree.xml
24  gsim_logic_tree_file = gmpe_logic_tree.xml
25  investigation_time = 50.0
26  intensity_measure_types_and_levels = {"PGA": [0.005, ..., 2.13]}
27  truncation_level = 3
28  maximum_distance = 200.0
```

This section of the oq-engine configuration file specifies the parameters that are relevant for the calculation of hazard. These include the names of the two files containing the Seismic Source System and the Ground Motion System, the duration of the time window used to compute the hazard, the ground motion intensity measure types and levels for which the probability of exceedence will be computed, the level of truncation of the Gaussian distribution of the logarithm of ground motion used in the calculation of hazard and the maximum integration distance (i.e. the distance within which sources will contribute to the computation of the hazard).

Output

```
32  [output]
33  export_dir = outputs/
34  # given the specified 'intensity_measure_types_and_levels'
35  mean_hazard_curves = true
36  quantile_hazard_curves = 0.1 0.5 0.9
37  uniform_hazard_spectra = false
38  individual_curves = false
39  poes = 0.1
```

The final section of the configuration file is the one that contains the parameters controlling the types of output to be produced. Providing an export directory will tell OpenQuake where to place the output files when the `--exports` flag is used when running the program. Setting `mean_hazard_curves` to true will result in a specific output containing the mean curves of the logic tree, likewise `quantile_hazard_curves` will produce separate files containing the quantile hazard curves at the quantiles listed (0.1, 0.5 and 0.9 in the example above, leave blank or omit if no quantiles are required). Setting `uniform_hazard_spectra` to true will output the uniform hazard spectra at the same probabilities of exceedence (poes) as those specified by the later option `poes`. The probabilities specified here correspond to the set investigation time.

By default, OpenQuake will export the results for each of the logic tree end branches. If the logic tree contains a large number of end branches the process of exporting the results from each end branch can add a significant amount of computation time - and result in a

large volume of disk spaced being used. If instead the user wishes only to export the hazard results for the mean and quantiles then the `individual_curves` option should be set to false.

3.4.2   Seismic hazard disaggregation

In this section we describe the structure of the configuration file to be used to complete a seismic hazard disaggregation. Since only a few parts of the standard configuration file need to be changed we can use the description given in Section 3.4.1 at page 54 as a reference and we emphasize herein major differences.

Calculation type and model info

```
[general]
description = A demo .ini file for PSHA disaggregation
calculation_mode = disaggregation
random_seed = 1024
```

The calculation mode parameter in this case is set as `disaggregation`.

Geometry of the area (or the sites) where hazard is computed

```
[geometry]
sites = 11.0 44.5
```

In the section it is necessary to specify the geographic coordinates of the site (or sites) where the disaggregation will be performed.

Disaggregation parameters

```
[disaggregation]
poes_disagg = 0.02, 0.1
mag_bin_width = 1.0
distance_bin_width = 25.0
coordinate_bin_width = 1.5
num_epsilon_bins = 3
```

With the disaggregation settings shown above we'll disaggregate the intensity measure levels with 10% and 2% probability of exceedance using the `investigation_time` and the intensity measure types defined in the "Calculation configuration" section of the OpenQuake configuration file (see page 56).

The parameters `mag_bin_width`, `distance_bin_width`, `coordinate_bin_width` control the level of discretization of the disaggregation matrix computed. `num_epsilon_bins` indicates the number of bins used to represent the contributions provided by different values of epsilon.

If the user is interested in a specific type of disaggregation, we suggest to use a very coarse gridding for the parameters that are not necessary. For example, if the user is interested in a magnitude-distance disaggregation, we suggest the use of very large value for the `coordinate_bin_width` and to set `num_epsilon_bins` equal to 1.

3.4.3 Event based PSHA

In the following we describe the sections of the configuration file that are required to complete event based PSHA calculations

Calculation type and model info

This part is almost identical to the corresponding one described in Section 3.4.1. Note the setting of the `calculation_mode` parameter which now corresponds to `event_based`.

```
[general]
description = A demo OpenQuake-engine .ini file for classical PSHA
calculation_mode = event_based
random_seed = 1024
```

Event based parameters

This section is used to specify the number of stochastic event sets to be generated for each logic tree realisation (each stochastic event set represents a potential realisation of seismicity during the `investigation_time` specified in the `calculation_configuration` part). Additionally, in this section the user can specify the spatial correlation model to be used for the generation of ground motion fields.

```
[event_based_params]
ses_per_logic_tree_path = 5
ground_motion_correlation_model = JB2009
ground_motion_correlation_params = {"vs30_clustering": True}
```

The acceptable flags for the parameter `vs30_clustering` are `False` and `True`, with a capital F and T respectively. 0 and 1 are also acceptable flags.

Output

This part substitutes the `Output` part described in the configuration file example described in the Section 3.4.1 at page 54.

```
[output]
export_dir = /tmp/xxx
ground_motion_fields = true
# post-process ground motion fields into hazard curves,
# given the specified 'intensity_measure_types_and_levels'
hazard_curves_from_gmfs = true
mean_hazard_curves = true
```

```
quantile_hazard_curves = 0.15 0.5 0.85
poes = 0.1 0.2
```

The option `hazard_curves_from_gmfs` instructs the user to use the event-based ground motion values to provide hazard curves indicating the probabilities of exceeding the intensity measure levels set previously in the `intensity_measure_types_and_levels` option.

4. Hazard Calculations and Results

In this Chapter we provide a desciption of the main commands available for running hazard with the oq-engine and the file formats used to represent the results of the analyses.

A general introduction on the use of the OpenQuake-engine is provided in Section 1.1 at page 15. The reader is invited to consult this part before diving into the following sections.

4.1 Running OpenQuake-engine for hazard calculations

The execution of a hazard analysis using the OpenQuake-engine is straightforward. Below we provide an example of the simplest command that can be used to launch a hazard calculation. It consists in the invocation of `oq-engine` together with the `--run` or `--rh` (run-hazard) option, and the name of a configuration file (in the example below it corresponds to `job.ini`):

```
user@ubuntu:~$ oq-engine --run job.ini
```

The amount of information prompted during the execution of the analysis can be controlled through the `--log-level` flag as shown in the example below:

```
user@ubuntu:~$ oq-engine --rh job.ini --log-level debug
```

In this example we ask the engine to provide an extensive amount of information (usually not justified for a standard analysis). Alternative options are: `debug`, `info`, `progress`, `warn`, `error`, `critical`.

4.2   Exporting results from a hazard calculation

There are two alternative ways to get results from the OpenQuake-engine: directly through the calculation or by exporting them from the internal oq-engine database once a calculation is completed.

The first option is defined at the OpenQuake-engine invocation through the flag `-exports xml`, as shown in the example below:

```
user@ubuntu:~$ oq-engine --run job.ini --exports xml
```

This will export the results to the `results` directory specified in the `job.ini` file.

The second option allows the user to export the computed results or just a subset of them whenever they want. In order to obtain the list of results of the hazard calculations stored in the oq-engine database the user can utilize the `--lhc` command ('list hazard calculations') to list the hazard calculations:

```
user@ubuntu:~$ oq-engine --lhc
```

The execution of this command will produce a list similar to the one provided below (the numbers in red are the calculations IDs):

```
user@ubuntu:~$ oq-engine --lhc
job_id | latest_job_status | last_update | description
1 | failed | 2013-03-01 09:49:34 | Classical PSHA
2 | successful | 2013-03-01 09:49:56 | Classical PSHA
3 | failed | 2013-03-01 10:24:04 | Classical PSHA
4 | failed | 2013-03-01 10:28:16 | Classical PSHA
5 | failed | 2013-03-01 10:30:04 | Classical PSHA
6 | successful | 2013-03-01 10:31:53 | Classical PSHA
7 | failed | 2013-03-09 08:15:14 | Classical PSHA
8 | successful | 2013-03-09 08:18:04 | Classical PSHA
```

Subsequently the user can get the list of result stored for a specific hazard analysis by using the `--list-outputs`, or `--lo`, command, as in the example below (note that the number in blue emphasizes the result ID):

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
3 | datastore | hcurves
```

and finally extract an xml file for a specific hazard result:

```
user@ubuntu:~$ oq-engine --export-outputs <result_id> <output_folder>
```

4.3 Description of hazard outputs

The results generated by the OpenQuake-engine are fundamentally of two distinct typologies differentiated by the presence (or absence) of epistemic uncertainty in the PSHA input model.

When epistemic uncertainty is incorporated into the calculation, the OpenQuake-engine calculators (e.g. Classical PSHA, Event Based PSHA, Disaggregation, UHS) produce a set of results (i.e. hazard curves, ground motion fields, disaggregation matrices, UHS, for each logic-tree realisation) which reflects epistemic uncertainties introduced in the PSHA input model. The user can specify that only the mean results and desired quantiles are exported by using the optional configuration `individual_curves=False` in the `job.ini` file.

For each logic tree sample, results are computed and stored. Calculation of results statistics (mean, standard deviation, quantiles) are supported by all the calculators, with the exception of the disaggregation calculator.

4.3.1 Outputs from Classical PSHA

By default, the classical PSHA calculator computes and stores hazard curves for each logic tree sample considered.

When the PSHA input model doesn't contain epistemic uncertainties the results is a set of hazard curves (one for each investigated site). The command below illustrates how is possible to retrieve the group of hazard curves obtained for a calculation with a given identifier `<calc_id>` (see Section 4.2 for an explanation about how to obtain the list of calculations performed with their corresponding ID):

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
3 | datastore | hcurves
4  | datastore | realizations
```

To export from the database the outputs (in this case hazard curves) contained in one of the output identifies, one can do so with the following command:

```
user@ubuntu:~$ oq-engine --export-output <output_id> <output_directory>
```

Alternatively, if the user wishes to export all of the outputs associated with a particular calculation then they can use the `--export-outputs` with the corresponding calculation key:

```
user@ubuntu:~$ oq-engine --export-outputs <calc_id> <output_directory>
```

The exports will produce one or more nrml files containing the seismic hazard curves, as represented in the example in the inset below:

```xml
<?xml version='1.0' encoding='UTF-8'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
  <hazardCurves sourceModelTreePath="b1|b212"
      gsimTreePath="b2" IMT="PGA" investigationTime="50.0">
    <IMLs>0.005 0.007 0.0098 ... 1.09 1.52 2.13</IMLs>
    <hazardCurve>
      <gml:Point>
      <gml:pos>10.0 45.0</gml:pos>
      </gml:Point>
      <poEs>1.0 1.0 1.0 ... 0.000688359310522 0.0 0.0</poEs>
    </hazardCurve>
    ...
    <hazardCurve>
      <gml:Point>
      <gml:pos>lon lat</gml:pos>
      </gml:Point>
      <poEs>poe1 poe2 ... poeN</poEs>
    </hazardCurve>
  </hazardCurves>
</nrml>
```

Notwithstanding the intuitiveness of this file, let's have a brief overview of the information included.

The overall content of this file is a list of hazard curves, one for each investigated site, computed using a PSHA input model representing one possible realisation obtained using the complete logic tree structure.

The attributes of the `hazardCurves` element (see text in red) specify the path of the logic tree used to create the seismic source model (`sourceModelTreePath`) and the ground motion model (`gsimTreePath`) plus the intensity measure type and the investigation time used to compute the probability of exceedance.

The `IMLs` element (in green in the example) contains the values of shaking used by the engine to compute the probability of exceedance in the investigation time. For each site this file contains a `hazardCurve` element which has the coordinates (longitude and latitude in decimal degrees) of the site and the values of the probability of exceedance for all the intensity measure levels specified in the `IMLs` element.

If the hazard calculation is configured to produce results including seismic hazard maps and uniform hazard spectra, then the list of outputs would display the following:

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
3 | datastore | hcurves
```

```
4 | datastore | hmaps
5  | datastore | realizations
6 | datastore | uhs
```

The following inset shows a sample of the nrml file used to describe a hazard map:

```
<?xml version='1.0' encoding='UTF-8'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
  <hazardMap sourceModelTreePath="b1" gsimTreePath="b1"
        IMT="PGA" investigationTime="50.0" poE="0.1">
    <node lon="119.596690957" lat="21.5497682591" iml="0.204569990197"/>
    <node lon="119.596751048" lat="21.6397004197" iml="0.212391638188"/>
    <node lon="119.596811453" lat="21.7296325803" iml="0.221407505615"/>
    ...
  </hazardMap>
</nrml>
```

and a uniform hazard spectrum:

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
    <uniformHazardSpectra
      sourceModelTreePath="b1_b2_b4"
      gsimTreePath="b1_b2"
      investigationTime="50.0" poE="0.1"
    >
        <periods>0.0 0.025 0.1 0.2</periods>
        <uhs>
            <gml:Point>
                <gml:pos>0.0 0.0</gml:pos>
            </gml:Point>
            <IMLs>0.3 0.5 0.2 0.1</IMLs>
        </uhs>
        <uhs>
            <gml:Point>
                <gml:pos>0.0 1.0</gml:pos>
            </gml:Point>
            <IMLs>0.3 0.5 0.2 0.1</IMLs>
        </uhs>
    </uniformHazardSpectra>
</nrml>
```

4.3.2   Outputs from Hazard Disaggregation

The OpenQuake-engine output of a disaggregation analysis corresponds to the combination of a hazard curve and a multidimensional matrix containing the results of the disaggregation.

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
3 | datastore | disagg
4  | datastore | realizations
```

Running `--export-output` to export the disaggregation results will produce individual files for each IMT, probability of exceedence and logic tree realisation. In the following inset we show an example of the nrml file used to represent the different disaggregation matrices (highlighted in red) produced by oq-engine:

```
<?xml version='2.0' encoding='UTF-8'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
  <disaggMatrices sourceModelTreePath="b1" gsimTreePath="b1" IMT="PGA"
        investigationTime="50.0" lon="10.1" lat="40.1"
        magBinEdges="5.0, 6.0, 7.0, 8.0"
        distBinEdges="0.0, 25.0, 50.0, 75.0, 100.0"
        lonBinEdges="9.0, 10.5, 12.0"
        latBinEdges="39.0, 40.5"
        pdfBinEdges="-3.0, -1.0, 1.0, 3.0"
        tectonicRegionTypes="Active Shallow Crust">
    <disaggMatrix type="Mag" dims="3" poE="0.1"
            iml="0.033424622602">
      <prob index="0" value="0.987374744394"/>
      <prob index="1" value="0.704295394366"/>
      <prob index="2" value="0.0802318409498"/>
    </disaggMatrix>
    <disaggMatrix type="Dist" dims="4" poE="0.1"
            iml="0.033424622602">
      <prob index="0" value="0.700851969171"/>
      <prob index="1" value="0.936680387051"/>
      <prob index="2" value="0.761883595568"/>
      <prob index="3" value="0.238687565571"/>
    </disaggMatrix>
    <disaggMatrix type="TRT" dims="1" poE="0.1"
            iml="0.033424622602">
      <prob index="0" value="0.996566187011"/>
    </disaggMatrix>
    <disaggMatrix type="Mag,Dist" dims="3,4" poE="0.1"
            iml="0.033424622602">
```

```
        <prob index="2,3" value="0.0"/>
    </disaggMatrix>
    <disaggMatrix type="Mag,Dist,pdf" dims="3,4,3" poE="0.1"
            iml="0.033424622602">
        <prob index="0,0,0" value="0.0785857271425"/>
        ...
    </disaggMatrix>
    <disaggMatrix type="Lon,Lat" dims="2,1" poE="0.1"
            iml="0.033424622602">
        <prob index="0,0" value="0.996566187011"/>
        <prob index="1,0" value="0.0"/>
    </disaggMatrix>
    <disaggMatrix type="Mag,Lon,Lat" dims="3,2,1" poE="0.1"
            iml="0.033424622602">
        <prob index="0,0,0" value="0.987374744394"/>
        <prob index="0,1,0" value="0.0"/>
        <prob index="1,0,0" value="0.704295394366"/>
        <prob index="1,1,0" value="0.0"/>
        <prob index="2,0,0" value="0.0802318409498"/>
        <prob index="2,1,0" value="0.0"/>
    </disaggMatrix>
    <disaggMatrix type="Lon,Lat,TRT" dims="2,1,1" poE="0.1"
            iml="0.033424622602">
        <prob index="0,0,0" value="0.996566187011"/>
        <prob index="1,0,0" value="0.0"/>
    </disaggMatrix>
  </disaggMatrices>
</nrml>
```

4.3.3 Outputs from Event Based PSHA

The Event Based PSHA calculator computes and stores stochastic event sets and the corresponding ground motion fields.

This calculator can also produce hazard curves and hazard maps exactly in the same way as done using the Classical PSHA calculator.

The inset below shows an example of the list of results provided by the oq-engine at the end of an event-based PSHA calculation:

```
user@ubuntu:~$ oq-engine --lo <calc_id>
id | output_type | name
10 | datastore   | gmfs
11 | datastore   | hcurves
12 | datastore   | realizations
13 | datastore   | sescollection
```

This list in the inset above contains a set of stochastic events (in green) and their corresponding sets of ground motion fields (in red).

Exporting the outputs from the stochastic event set will produce, for each realisation, a nrml file containing a collection of stochastic event sets. Below is an example of the nrml file for a stochastic event set containing only 2 ruptures:

```xml
<?xml version='1.0' encoding='UTF-8'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
         xmlns="http://openquake.org/xmlns/nrml/0.5">
  <stochasticEventSetCollection sourceModelTreePath="b1">
    <stochasticEventSet id="12" investigationTime="50.0">
     <rupture id="col=00~ses=0001~src=10138~rup=0049-00"
         magnitude="4.55" strike="90.0" dip="90.0"
         rake="90.0" tectonicRegion="Active Shallow Crust">
       <planarSurface>
         <topLeft lon="12.233903801" lat="43.256198599"
                  depth="11.3933265259"/>
         <topRight lon="12.263958243" lat="43.2562025344"
                  depth="11.3933265259"/>
         <bottomLeft lon="12.233903801" lat="43.256198599"
                  depth="12.6066734741"/>
         <bottomRight lon="12.263958243" lat="43.2562025344"
                  depth="12.6066734741"/>
       </planarSurface>
     </rupture>
      <rupture id="col=00~ses=0001~src=12412~rup=0121-00"
             magnitude="4.65" strike="135.0" dip="90.0"
             rake="90.0" tectonicRegion="Active Shallow Crust">
        <planarSurface>
          <topLeft lon="11.45858812" lat="42.7429056814"
                  depth="11.3208667302"/>
          <topRight lon="11.4822820715" lat="42.7256333907"
                  depth="11.3208667302"/>
          <bottomLeft lon="11.45858812" lat="42.7429056814"
                  depth="12.6791332698"/>
          <bottomRight lon="11.4822820715" lat="42.7256333907"
                  depth="12.6791332698"/>
        </planarSurface>
      </rupture>
    </stochasticEventSet>
  </stochasticEventSetCollection>
</nrml>
```

The text in red shows the part which describes the id of the generated stochastic event

set and the investigation time covered.

The text in green emphasises the portion of the text used to describe a rupture. The information provided describes entirely the geometry of the rupture as well as its rupturing properties (e.g. rake, magnitude). The rupture ID is a string that represents each rupture uniquely (including the case where the same rupture is sampled multiple times). The exact format may be subject to change from time-to-time; however, the ID will usually contain information regarding the specific logic tree branch sample (realisation), the stochastic event set counter, the ID of the source from which the rupture was samples and a unique number indicating the rupture identifier within the earthquake rupture forecast of the specific source.

Exporting the outputs from the gmfs will produce an xml file for each realisation containing the corresponding ground motion fields. Below is an example of a gmf collection nrml file containing one ground motion field:

```
<?xml version='1.0' encoding='UTF-8'?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
  <gmfCollection sourceModelTreePath="b1" gsimTreePath="b1">
    <gmfSet investigationTime="50.0" stochasticEventSetId="12">
      <gmf IMT="PGA" ruptureId="col=00~ses=0001~src=10138~rup=0049-00">
        <node gmv="0.0105891230432" lon="11.1240023202"
            lat="43.5107462335"/>
        <node gmv="0.00905803920023" lon="11.1241875202"
            lat="43.6006783941"/>
        <node gmv="0.00637664420977" lon="11.1243735810"
            lat="43.6906105547"/>
        <node gmv="0.00476533134789" lon="11.1245605075"
            lat="43.7805427153"/>
        <node gmv="0.00452594698469" lon="11.1247483046"
            lat="43.8704748759"/>
        ...
        <node gmv="0.00017301076646" lon="11.3782630185"
            lat="44.5129482397"/>
      </gmf>
    </gmfSet>
  </gmfCollection>
</nrml>
```

5. Demonstrative Examples

A number of hazard calculation demos are provided with the oq-engine installation,showing different examples of input and configuration files, for different use cases.

This is the list of demos which illustrate how to use the oq-engine for various seismic hazard analysis:

- AreaSourceClassicalPSHA
- CharacteristicFaultSourceCase1ClassicalPSHA
- CharacteristicFaultSourceCase2ClassicalPSHA
- CharacteristicFaultSourceCase3ClassicalPSHA
- ComplexFaultSourceClassicalPSHA
- Disaggregation
- EventBasedPSHA
- LogicTreeCase1ClassicalPSHA
- LogicTreeCase2ClassicalPSHA
- LogicTreeCase3ClassicalPSHA
- PointSourceClassicalPSHA
- SimpleFaultSourceClassicalPSHA

5.1 Classical PSHA Demos

A number of demos have been designed to show how to perform a classical PSHA calculation using the different available source typologies and how to define non-trivial logic trees. It should be noted that the input files that will be illustrated are valid not only for a classical PSHA calculation but also for event based and disaggregation analysis.

All the classical PSHA demos illustrating the different source typologies (all demos but the ones about Logic Tree definition) share the same GSIM logic tree file, which for clarity is provided below.

Since this logic tree consideres only one tectonic region (i.e. `Active Shallow Crust`) all the seismic sources will belong be considered active shallow crust sources.

```xml
<?xml version="1.0" encoding="UTF-8"?>

<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
   <logicTree logicTreeID='lt1'>

       <logicTreeBranchingLevel branchingLevelID="bl1">
           <logicTreeBranchSet
               uncertaintyType="gmpeModel"
               applyToTectonicRegionType="Active Shallow Crust"
               branchSetID="bs1">

               <logicTreeBranch branchID="b1">
                   <uncertaintyModel>
                       BooreAtkinson2008
                   </uncertaintyModel>
                   <uncertaintyWeight>1.0</uncertaintyWeight>
               </logicTreeBranch>

           </logicTreeBranchSet>
       </logicTreeBranchingLevel>

   </logicTree>
</nrml>
```

5.1.1 Classical PSHA with different source typologies

This section discusses the following examples:

- AreaSourceClassicalPSHA
- CharacteristicFaultSourceCase1ClassicalPSHA
- CharacteristicFaultSourceCase2ClassicalPSHA
- CharacteristicFaultSourceCase3ClassicalPSHA
- ComplexFaultSourceClassicalPSHA
- PointSourceClassicalPSHA
- SimpleFaultSourceClassicalPSHA

The configuration file (see below) is defined to compute hazard curves for several intensity measure types (PGV, PGA and Spectral acceleration at different periods), hazard maps and uniform hazard spectra for different probabilities of exceedance:

```
[general]
description = ...
calculation_mode = classical
random_seed = 23

[geometry]
region = ...
region_grid_spacing = 5.0

[logic_tree]
number_of_logic_tree_samples = 0

[erf]
rupture_mesh_spacing = 2
width_of_mfd_bin = 0.1
area_source_discretization = 5.0

[site_params]
reference_vs30_type = measured
reference_vs30_value = 600.0
reference_depth_to_2pt5km_per_sec = 5.0
reference_depth_to_1pt0km_per_sec = 100.0

[calculation]
source_model_logic_tree_file = source_model_logic_tree.xml
gsim_logic_tree_file = gmpe_logic_tree.xml
investigation_time = 50.0
intensity_measure_types_and_levels ={
"PGV": [2, 4, 6 ,8, 10, ...],
"PGA": [0.005, 0.007, ...],
"SA(0.025)": [...],
"SA(0.05)": [...],
"SA(0.1)": [...],
"SA(0.2)": [...],
"SA(0.5)": [...],
"SA(1.0)": [...],
"SA(2.0)": [...]}
truncation_level = 3
maximum_distance = 200.0
```
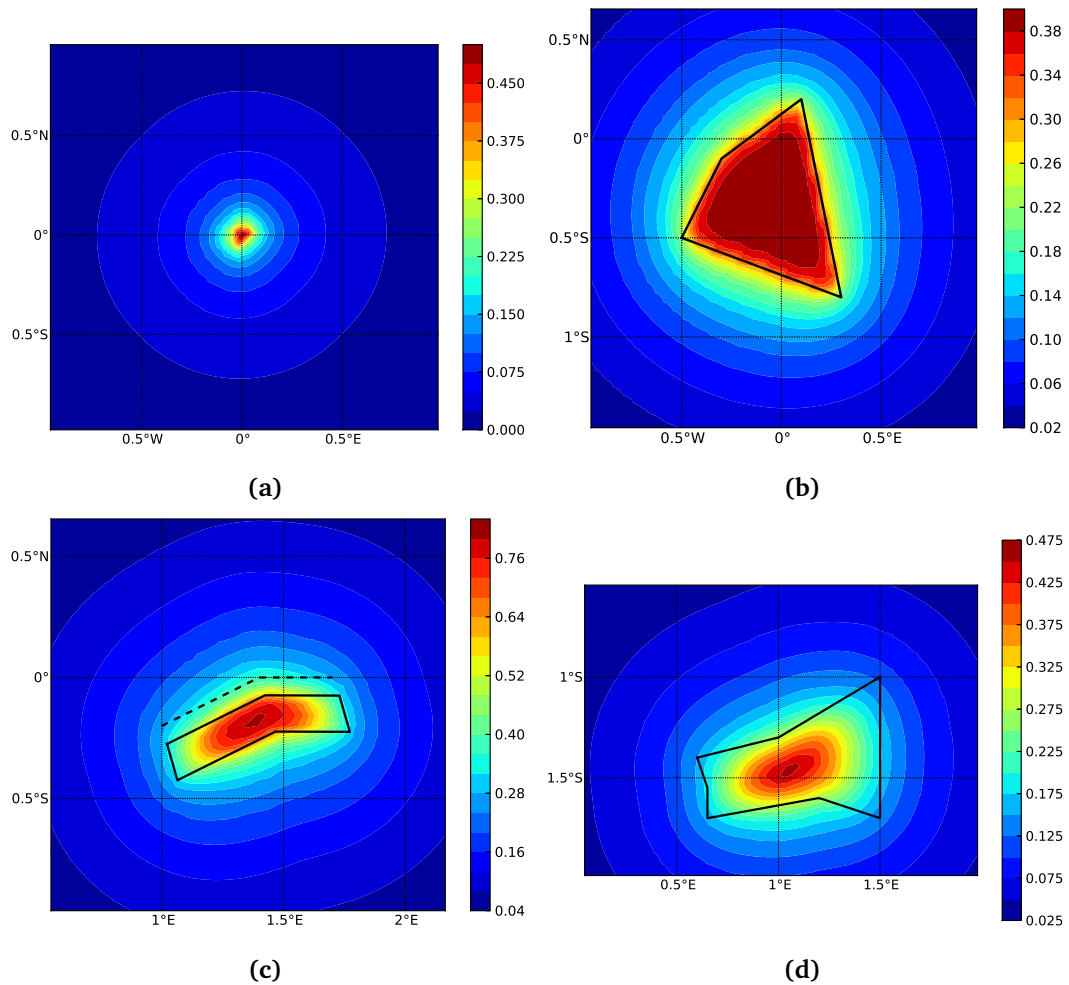
Figure 5.1 – *Hazard maps (for PGA, 10% in 50 years) as obtained from the different* oq-engine *source typologies. (a) Point Source. (b) Area source. The solid black line represents the area boundary. (c) Simple Fault Source. The dashed line represents the fault trace, while the solid line the fault surface projection. (d) Complex Fault Source. The solid line represent the fault surface projection (d)*

```
[output]
export_dir = ...
mean_hazard_curves = false
quantile_hazard_curves =
hazard_maps = true
uniform_hazard_spectra = true
poes = 0.1 0.02
```

Hazard maps for the different demos are shown in Figure 5.1 and Figure 5.2.

(a)

(b)

(c)
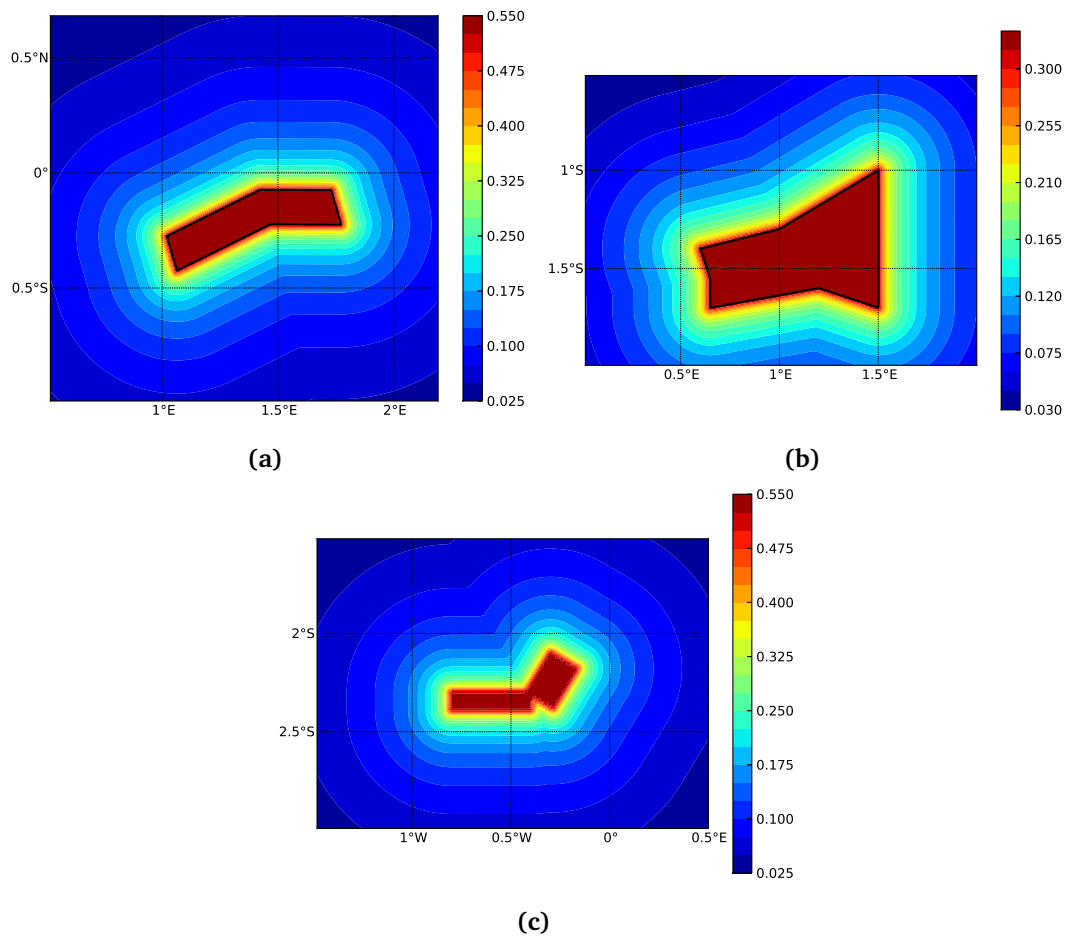
Figure 5.2 – *Hazard maps (for PGA, 10% in 50 years) as obtained from characteristic fault sources with simple fault geometry (e), complex fault geometry (f), and collection of planar surfaces (g)*

5.1.2   Classical PSHA with non trivial logic trees

Three demos are provided to illustrate how the logic tree formalism can be used to express epistemic uncertainties in seismic hazard analysis.

LogicTreeCase1ClassicalPSHA shows an example of logic tree defining two alternative source models, with sources belonging to two different tectonic region types, and with two alternative GMPEs for each tectonic region type. The source model logic tree is therefore defined in the following way:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
    <logicTree logicTreeID="lt1">

        <logicTreeBranchingLevel branchingLevelID="bl1">

            <logicTreeBranchSet uncertaintyType="sourceModel"
                            branchSetID="bs1">
                <logicTreeBranch branchID="b1">
                    <uncertaintyModel>
                      source_model_1.xml
                    </uncertaintyModel>
                    <uncertaintyWeight>0.5</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b2">
                    <uncertaintyModel>
                       source_model_2.xml
                    </uncertaintyModel>
                    <uncertaintyWeight>0.5</uncertaintyWeight>
                </logicTreeBranch>
            </logicTreeBranchSet>

        </logicTreeBranchingLevel>

    </logicTree>
</nrml>
```

The two source models are defined in two separate files: `source_model_1.xml` and `source_model_2.xml` each one associated to a corresponding weight (0.5 for both).

The GSIM logic tree file contains the following structure:

```xml
<?xml version="1.0" encoding="UTF-8"?>

<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
    <logicTree logicTreeID='lt1'>

        <logicTreeBranchingLevel branchingLevelID="bl1">
            <logicTreeBranchSet uncertaintyType="gmpeModel"
                applyToTectonicRegionType="Active Shallow Crust"
                branchSetID="bs1">
              <logicTreeBranch branchID="b11">
                 <uncertaintyModel>
                     BooreAtkinson2008
                 </uncertaintyModel>
                 <uncertaintyWeight>0.5</uncertaintyWeight>
              </logicTreeBranch>
              <logicTreeBranch branchID="b12">
                 <uncertaintyModel>
                     ChiouYoungs2008
                 </uncertaintyModel>
                 <uncertaintyWeight>0.5</uncertaintyWeight>
              </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>

        <logicTreeBranchingLevel branchingLevelID="bl2">
            <logicTreeBranchSet uncertaintyType="gmpeModel"
               applyToTectonicRegionType="Stable Continental Crust"
               branchSetID="bs2">
               <logicTreeBranch branchID="b21">
                 <uncertaintyModel>
                     ToroEtAl2002</uncertaintyModel>
                 <uncertaintyWeight>0.5</uncertaintyWeight>
                 </logicTreeBranch>
                 <logicTreeBranch branchID="b22">
                  <uncertaintyModel>
                      Campbell2003</uncertaintyModel>
                  <uncertaintyWeight>0.5</uncertaintyWeight>
                 </logicTreeBranch>
```

```
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>


    </logicTree>
</nrml>
```

The source model contains sources belonging to Active Shallow Crust and Stable Continental Crust, therefore the GSIM logic tree defines two branching levels, one for each considered tectonic region type. Moreover for each tectonic region a branch set with two GMPEs is defined: Boore and Atkinson 2008 and Chiou and Youngs 2008 for Active Shallow Crust and Toro et al. 2003 and Campbell 2003 for Stable Continental Crust. By processing the above logic tree files using the logic tree path enumeration mode (enabled by setting in the configuration file `number_of_logic_tree_samples = 0`) hazard results are computed for 8 logic tree paths (2 source models x 2 GMPEs for Active x 2 GMPEs for Stable).

LogicTreeCase2ClassicalPSHA defines a single source model consisting of only two sources (area and simple fault) belonging to different tectonic region types (Active Shallow Crust and Stable Continental Region) and both characterized by a truncated Gutenberg-Richter distribution. The logic tree defines uncertainties for G-R a and b values (three possible pairs for each source), maximum magnitude (three values for each source) and uncertainties on the GMPEs for each tectonic region type (two GMPE per region type).

To accommodate such a structure the GSIM logic tree is defined in the following way:

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
    <logicTree logicTreeID="lt1">


        <logicTreeBranchingLevel branchingLevelID="bl1">
            <logicTreeBranchSet uncertaintyType="sourceModel"
                                branchSetID="bs1">
                <logicTreeBranch branchID="b11">
                    <uncertaintyModel>
                     source_model.xml
                    </uncertaintyModel>
                    <uncertaintyWeight>1.0</uncertaintyWeight>
                </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>


        <logicTreeBranchingLevel branchingLevelID="bl2">
```

```
            <logicTreeBranchSet uncertaintyType="abGRAbsolute"
                                applyToSources="1"
                                branchSetID="bs21">
                <logicTreeBranch branchID="b21">
                    <uncertaintyModel>4.6 1.1</uncertaintyModel>
                    <uncertaintyWeight>0.333</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b22">
                    <uncertaintyModel>4.5 1.0</uncertaintyModel>
                    <uncertaintyWeight>0.333</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b23">
                    <uncertaintyModel>4.4 0.9</uncertaintyModel>
                    <uncertaintyWeight>0.334</uncertaintyWeight>
                </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>

        <logicTreeBranchingLevel branchingLevelID="bl3">
            <logicTreeBranchSet uncertaintyType="abGRAbsolute"
                                applyToSources="2"
                                branchSetID="bs31">
                <logicTreeBranch branchID="b31">
                    <uncertaintyModel>3.3 1.0</uncertaintyModel>
                    <uncertaintyWeight>0.333</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b32">
                    <uncertaintyModel>3.2 0.9</uncertaintyModel>
                    <uncertaintyWeight>0.333</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b33">
                    <uncertaintyModel>3.1 0.8</uncertaintyModel>
                    <uncertaintyWeight>0.334</uncertaintyWeight>
                </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>

        <logicTreeBranchingLevel branchingLevelID="bl4">
            <logicTreeBranchSet uncertaintyType="maxMagGRAbsolute"
```

```xml
                                    applyToSources="1"
                                    branchSetID="bs41">
                <logicTreeBranch branchID="b41">
                    <uncertaintyModel>7.0</uncertaintyModel>
                    <uncertaintyWeight>0.333</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b42">
                    <uncertaintyModel>7.3</uncertaintyModel>
                    <uncertaintyWeight>0.333</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b43">
                    <uncertaintyModel>7.6</uncertaintyModel>
                    <uncertaintyWeight>0.334</uncertaintyWeight>
                </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>

        <logicTreeBranchingLevel branchingLevelID="bl5">
            <logicTreeBranchSet uncertaintyType="maxMagGRAbsolute"
                                    applyToSources="2"
                                    branchSetID="bs51">
                <logicTreeBranch branchID="b51">
                    <uncertaintyModel>7.5</uncertaintyModel>
                    <uncertaintyWeight>0.333</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b52">
                    <uncertaintyModel>7.8</uncertaintyModel>
                    <uncertaintyWeight>0.333</uncertaintyWeight>
                </logicTreeBranch>
                <logicTreeBranch branchID="b53">
                    <uncertaintyModel>8.0</uncertaintyModel>
                    <uncertaintyWeight>0.334</uncertaintyWeight>
                </logicTreeBranch>
            </logicTreeBranchSet>
        </logicTreeBranchingLevel>

    </logicTree>
</nrml>
```
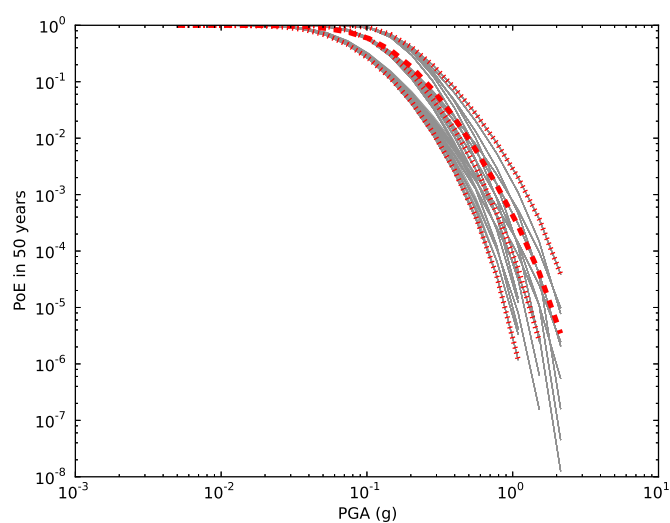
The first branching level defines the source model. For each source, two branching

(a)

Figure 5.3 – *Hazard curves as obtained from the LogicTreeCase2 demo. Solid gray lines represent individual hazard curves from the different logic tree path (a total of 324 curves). The red dashed line represents the mean hazard curve, while the red dotted lines depict the quantile levels (0.15, 0.5, 0.95).*

levels are created, one defining uncertainties on G-R a and b values (defined by setting uncertaintyType="abGRAbsolute") and G-R maximum magnitude (uncertaintyType="maxMagGRAbsol

It is important to notice that each branch set is applied to a specific source by defining the attribute `applyToSources`, followed by the source ID. The GSIM logic tree file is the same as used for LogicTreeCase1ClassicalPSHA. By setting in the configuration file `number_of_-logic_tree_samples = 0`, hazard results are obtained for 324 paths (1 source model x 3 (a, b) pairs for source 1 x 3 (a, b) pairs for source 2 x 3 max magnitude values for source 1 x 3 max magnitude values for source 2 x 2 GMPEs for Active Shallow Crust X 2 GMPEs for Stable Continental Crust), see Figure 5.3.

LogicTreeCase3ClassicalPSHA illustrates an example of logic tree defining relative uncertainties on G-R maximum magnitude and b value. A single source model is considered containing two sources belonging to different tectonic region types and both characterized by a G-R magnitude frequency distribution. The source model logic tree is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<nrml xmlns:gml="http://www.opengis.net/gml"
      xmlns="http://openquake.org/xmlns/nrml/0.5">
    <logicTree logicTreeID="lt1">

        <logicTreeBranchingLevel branchingLevelID="bl1">
            <logicTreeBranchSet uncertaintyType="sourceModel"
                                branchSetID="bs1">
```

```xml
            <logicTreeBranch branchID="b11">
                <uncertaintyModel>
                 source_model.xml
                </uncertaintyModel>
                <uncertaintyWeight>1.0</uncertaintyWeight>
            </logicTreeBranch>
        </logicTreeBranchSet>
    </logicTreeBranchingLevel>

    <logicTreeBranchingLevel branchingLevelID="bl2">
        <logicTreeBranchSet uncertaintyType="bGRRelative"
                            branchSetID="bs21">
            <logicTreeBranch branchID="b21">
                <uncertaintyModel>+0.1</uncertaintyModel>
                <uncertaintyWeight>0.333</uncertaintyWeight>
            </logicTreeBranch>
            <logicTreeBranch branchID="b22">
                <uncertaintyModel>0.0</uncertaintyModel>
                <uncertaintyWeight>0.333</uncertaintyWeight>
            </logicTreeBranch>
            <logicTreeBranch branchID="b23">
                <uncertaintyModel>-0.1</uncertaintyModel>
                <uncertaintyWeight>0.334</uncertaintyWeight>
            </logicTreeBranch>
        </logicTreeBranchSet>
    </logicTreeBranchingLevel>

    <logicTreeBranchingLevel branchingLevelID="bl3">
        <logicTreeBranchSet uncertaintyType="maxMagGRRelative"
                            branchSetID="bs31">
            <logicTreeBranch branchID="b31">
                <uncertaintyModel>0.0</uncertaintyModel>
                <uncertaintyWeight>0.333</uncertaintyWeight>
            </logicTreeBranch>
            <logicTreeBranch branchID="b32">
                <uncertaintyModel>+0.5</uncertaintyModel>
                <uncertaintyWeight>0.333</uncertaintyWeight>
            </logicTreeBranch>
            <logicTreeBranch branchID="b33">
```

```
                        <uncertaintyModel>+1.0</uncertaintyModel>
                        <uncertaintyWeight>0.334</uncertaintyWeight>
                    </logicTreeBranch>
                </logicTreeBranchSet>
            </logicTreeBranchingLevel>


        </logicTree>
</nrml>
```

After the first branching level defining the source model, two additional branching levels are defined, one defining relative uncertainties on b value (bGRRelative applied consistently to all sources in the source model) and the second uncertainties on maximum magnitude (maxMagGRRelative). Similar to the other cases, two GMPEs are considered for each tectonic region type and therefore the total number of logic tree path is 36 (1 source model x 3 b value increments x 3 maximum magnitude increments x 2 GMPE for Active x 2 GMPEs for Stable).

5.2 Hazard Disaggregation Demos

An example of disaggregation calculation is given considering a source model consisting of two sources (area and simple fault) belonging to two different tectonic region types.

The calculation is defined with the following configuration file:

```
[general]
description = ...
calculation_mode = disaggregation
random_seed = 23

[geometry]
sites = 0.5 -0.5

[logic_tree]
number_of_logic_tree_samples = 0

[erf]
rupture_mesh_spacing = 2
width_of_mfd_bin = 0.1
area_source_discretization = 5.0

[site_params]
```

```
reference_vs30_type = measured
reference_vs30_value = 600.0
reference_depth_to_2pt5km_per_sec = 5.0
reference_depth_to_1pt0km_per_sec = 100.0

[calculation]
source_model_logic_tree_file = source_model_logic_tree.xml
gsim_logic_tree_file = gmpe_logic_tree.xml
investigation_time = 50.0
intensity_measure_types_and_levels = "PGA": [...]
truncation_level = 3
maximum_distance = 200.0

[disaggregation]
poes_disagg = 0.1
mag_bin_width = 1.0
distance_bin_width = 10.0
coordinate_bin_width = 0.2
num_epsilon_bins = 3

[output]
export_dir = ...
```

Disaggregation matrices are computed for a single site (located between the two sources) for a ground motion value corresponding to a probability value equal to 0.1 (poes_disagg = 0.1). Magnitude values are classified in one magnitude unit bins (mag_bin_width = 1.0), distances in bins of 10 km (distance_bin_width = 10.0), coordinates in bins of 0.2 degrees (coordinate_bin_width = 0.2). 3 epsilons bins are considered (num_epsilon_bins = 3).

5.3 Event Based PSHA Demos

A demo showing an example of Event Based PSHA calculation is provided with the following configuration file:

```
[general]
description = Event Based PSHA using Area Source
calculation_mode = event_based
random_seed = 23
```

```
[geometry]
sites = 0.5 -0.5

[logic_tree]
number_of_logic_tree_samples = 0

[erf]
rupture_mesh_spacing = 2
width_of_mfd_bin = 0.1
area_source_discretization = 5.0

[site_params]
reference_vs30_type = measured
reference_vs30_value = 600.0
reference_depth_to_2pt5km_per_sec = 5.0
reference_depth_to_1pt0km_per_sec = 100.0

[calculation]
source_model_logic_tree_file = source_model_logic_tree.xml
gsim_logic_tree_file = gmpe_logic_tree.xml
investigation_time = 50.0
intensity_measure_types_and_levels = "PGA": [...]
truncation_level = 3
maximum_distance = 200.0

[event_based_params]
ses_per_logic_tree_path = 100
ground_motion_correlation_model =
ground_motion_correlation_params =

[output]
export_dir = ...
ground_motion_fields = true
hazard_curves_from_gmfs = true
mean_hazard_curves = false
quantile_hazard_curves =
hazard_maps = true
poes = 0.1
```

The source model consist of one source (area). 100 stochastic event sets are generated
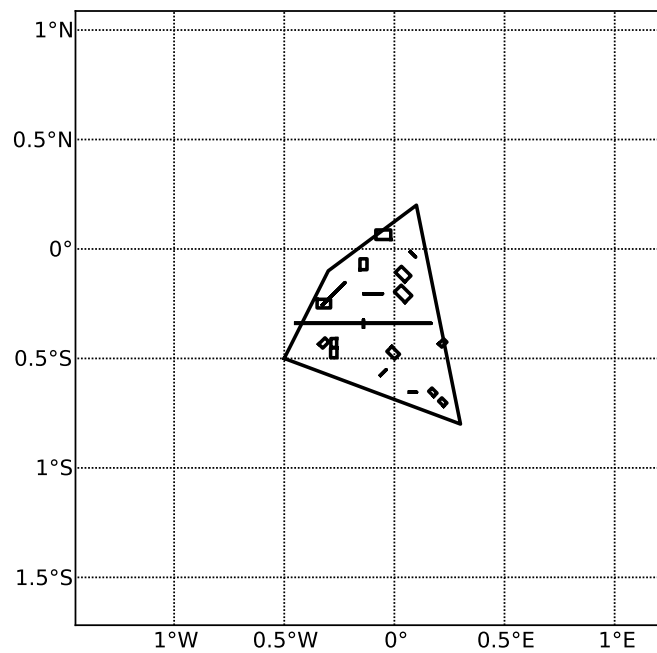
(a)

Figure 5.4 – *A stochastic event set generated with the event based PSHA demo. The area source defines a nodal plane distribution which distributes events among vertical and dipping (50 degrees) faults with equal weights. Vertical ruptures are then distributed equally in the range 0-180 degrees while the dipping ones in the range 0-360, both with a step of 45 degrees.*

(`ses_per_logic_tree_path = 100`) (an example can be seen in Figure 5.4). Ground motion fields are computed (`ground_motion_fields = true`, Figure 5.5) and also hazard curves from ground motion fields are extracted (`hazard_curves_from_gmfs = true`). The corresponding hazard maps for 0.1 probability are also calculated (`hazard_maps = true`)

(a)   **(b)**
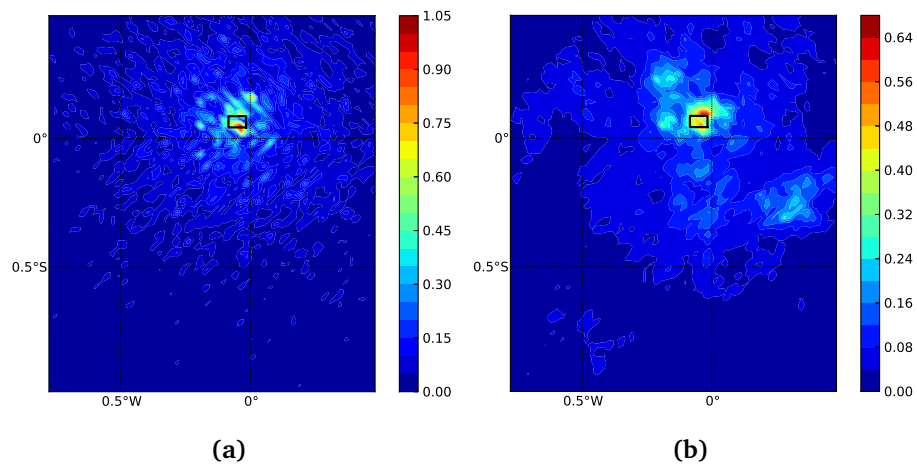
Figure 5.5 – *Ground motion fields (PGA) with no spatial correlations (a) and with spatial correlation (b)*

Part III

Risk

6. Introduction to the Risk Module

The seismic risk results are calculated using the OpenQuake risk library (oq-risklib), an open-source suite of tools for seismic risk assessment and loss estimation. This library is written in the Python programming language and available in the form of a "developers" release at the following location: http://github.com/gem/oq-risklib.

The risk component of the OpenQuake-engine can compute both scenario-based and probabilistic seismic damage and risk using various approaches. The following types of analysis are currently supported:

- **Scenario Damage Assessment**, for the calculation of damage distribution statistics for a portfolio of buildings from a single earthquake rupture scenario taking into account aleatory and epistemic ground-motion variability.

- **Scenario Risk Assessment**, for the calculation of individual asset and portfolio loss statistics due to a single earthquake rupture scenario taking into account aleatory and epistemic ground-motion variability. Correlation in the vulnerability of different assets of the same typology can also be taken into consideration.

- **Classical Probabilistic Seismic Damage Analysis**, for the calculation of damage state probabilities over a specified time period, and probabilistic collapse maps, starting from the hazard curves computed following the classical integration procedure (Cornell, 1968, McGuire (1976)) as formulated by Field et al., 2003.

- **Classical Probabilistic Seismic Risk Analysis**, for the calculation of loss curves and loss maps, starting from the hazard curves computed following the classical integration procedure (Cornell, 1968, McGuire (1976)) as formulated by Field et al., 2003.

- **Stochastic Event Based Probabilistic Seismic Risk Analysis**, for the calculation of event loss tables starting from stochastic event sets. Other results such as loss-exceedance curves, probabilistic loss maps, average annual losses, and insured loss statistics can be obtained by post-processing the event loss tables.

- *Retrofit Benefit-Cost Ratio Analysis*, which is useful in estimating the net-present value of the potential benefits of performing retrofitting for a portfolio of assets (in terms of decreased losses in seismic events), measured relative to the upfront cost of retrofitting.

Each calculation workflow has a modular structure, so that intermediate results can be saved and analyzed. Moreover, each calculator can be extended independently of the others so that additional calculation options and methodologies can be easily introduced, without affecting the overall calculation workflow. Each workflow is described in more detail in the following sections.

6.1 Scenario Damage Assessment

The scenario damage calculator computes damage distribution statistics for all assets in a given exposure model for a single specified earthquake rupture. Damage distribution statistics include the mean and standard deviation of damage fractions for different damage states. This calculator requires the definition of a finite rupture model, an exposure model and a fragility model; the main results are the damage distribution statistics per asset, aggregated damage distribution statistics per taxonomy, aggregated damage distribution statistics for the region, and collapse maps, which contain the spatial distribution of the number or area of collapsed buildings throughout the region of interest.

The earthquake rupture characteristics—i.e. the magnitude, hypocenter and fault geometry—are modelled as deterministic in the scenario calculators. Multiple realizations of different possible GMFs due to the single earthquake rupture are generated, taking into consideration both the inter-event variability of ground motions, and the intra-event residuals obtained from a spatial correlation model for ground motion residuals. The use of logic trees allows for the consideration of uncertainty in the choice of a ground motion model for the given tectonic region.

As an alternative to computing the GMFs with OpenQuake-engine, users can also provide their own sets of GMFs as input to the scenario damage calculator.

For each GMF realization, damage fractions (the fraction of buildings in each damage state) are estimated for every asset in the exposure model using the provided fragility model, and finally the damage distribution statistics (i.e., the mean damage fractions and standard deviation of damage fractions for all damage states) across all realizations are calculated. The calculator also provides aggregated damage distribution statistics for the portfolio, such as mean damage fractions and standard deviation of damage fractions for each taxonomy in the exposure model, and the mean damage fractions and standard deviation of damage fractions for the entire region of study.

The required input files required for running a scenario damage calculation and the resulting output files are depicted in Figure 6.1.

Starting with OpenQuake-engine v1.7, consequence model files can also be provided
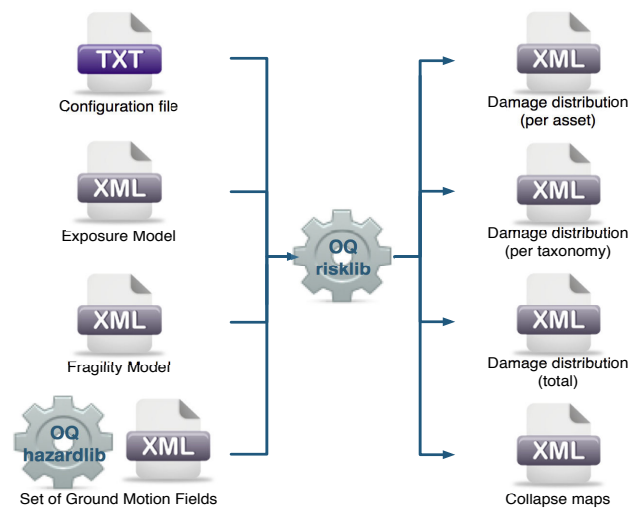
Figure 6.1 – *Scenario Damage Calculator input/output structure.*

as inputs for a scenario damage calculation in addition to fragility models files, in order to estimate consequences based on the calculated damage distribution. The user may provide one consequence model file corresponding to each loss type (amongst structural, nonstructural, contents, and business interruption) for which a fragility model file is provided. Whereas providing a fragility model file for at least one loss type is mandatory for running a Scenario Damage calculation, providing corresponding consequence model files is optional.

6.2 Scenario Risk Assessment

The scenario risk calculator computes loss statistics for all assets in a given exposure model for a single specified earthquake rupture. Loss statistics include the mean and standard deviation of ground-up losses and insured losses for each loss type considered in the analysis. Loss statistics can currently be computed for five different loss types using this calculator: structural losses, nonstructural losses, contents losses, downtime losses, and occupant fatalities. This calculator requires the definition of a finite rupture model, an exposure model and a vulnerability model for each loss type considered; the main results are the loss statistics per asset and mean loss maps.

The earthquake rupture characteristics—i.e. the magnitude, hypocenter and fault geometry—are modelled as deterministic in the scenario calculators. Multiple realizations of different possible GMFs due to the single earthquake rupture are generated, taking into consideration both the inter-event variability of ground motions, and the intra-event residuals obtained from a spatial correlation model for ground motion residuals. The use of logic trees allows for the consideration of uncertainty in the choice of a ground motion model for the given tectonic region.

As an alternative to computing the GMFs with OpenQuake, users can also provide their own sets of GMFs as input to the scenario risk calculator.

For each GMF realization, a loss ratio is sampled for every asset in the exposure model using the provided probabilistic vulnerability model taking into consideration the correlation model for vulnerability of different assets of a given taxonomy. Finally loss statistics, i.e., the mean loss and standard deviation of loss for both ground-up losses and insured losses across all realizations, are calculated for each asset. Mean loss maps are also generated by this calculator, describing the mean ground-up losses and mean insured losses caused by the scenario event for the different assets in the exposure model.

The required input files required for running a scenario risk calculation and the resulting output files are depicted in Figure 6.2.

Figure 6.2 – *Scenario Risk Calculator input/output structure.*

6.3 Classical Probabilistic Seismic Damage Analysis

The classical PSHA-based damage calculator integrates the fragility functions for an asset with the seismic hazard curve at the location of the asset, to give the expected damage distribution for the asset within a specified time period. The calculator requires the definition of an exposure model, a fragility model with fragility functions for each taxonomy represented in the exposure model, and hazard curves calculated in the region of interest. The main results of this calculator are the expected damage distribution for each asset, which describe the probability of the asset being in different damage states, and collapse maps for the region, which describe the probability of collapse for different assets in the portfolio over the specified time period. Damage distribution aggregated by taxonomy or of the total portfolio (considering all assets in the exposure model) can not be extracted using this calculator, as the spatial correlation of the ground motion residuals is not taken into consideration.

The hazard curves required for this calculator can be calculated by the OpenQuake-engine for all asset locations in the exposure model using the classical PSHA approach (Cornell, 1968; McGuire, 1976).

The required input files required for running a classical probabilistic damage calculation and the resulting output files are depicted in Figure 6.3.

Figure 6.3 – *Classical PSHA-based Damage Calculator input/output structure.*

6.4 Classical Probabilistic Seismic Risk Analysis

The classical PSHA-based risk calculator convolves through numerical integration, the probabilistic vulnerability functions for an asset with the seismic hazard curve at the location of the asset, to give the loss distribution for the asset within a specified time period. The calculator requires the definition of an exposure model, a vulnerability model for each loss type of interest with vulnerability functions for each taxonomy represented in the exposure model, and hazard curves calculated in the region of interest. Loss curves and loss maps can currently be calculated for five different loss types using this calculator: structural losses, nonstructural losses, contents losses, downtime losses, and occupant fatalities. The main results of this calculator are loss exceedance curves for each asset, which describe the probability of exceedance of different loss levels over the specified time period, and loss maps for the region, which describe the loss values that have a given probability of exceedance over the specified time

Unlike the probabilistic event-based risk calculator, an aggregate loss curve (considering all assets in the exposure model) can not be extracted using this calculator, as the correlation of the ground motion residuals and vulnerability uncertainty is not taken into consideration in this calculator.

The hazard curves required for this calculator can be calculated by the OpenQuake-engine for all asset locations in the exposure model using the classical PSHA approach (Cornell, 1968; McGuire, 1976). The use of logic- trees allows for the consideration of model uncertainty in the choice of a ground motion prediction equation for the different tectonic region types in the region. Unlike what was described in the previous calculator, a total

loss curve (considering all assets in the exposure model) can not be extracted using this calculator, as the correlation of the ground motion residuals and vulnerability uncertainty is not taken into consideration.

The required input files required for running a classical probabilistic risk calculation and the resulting output files are depicted in Figure 6.4.
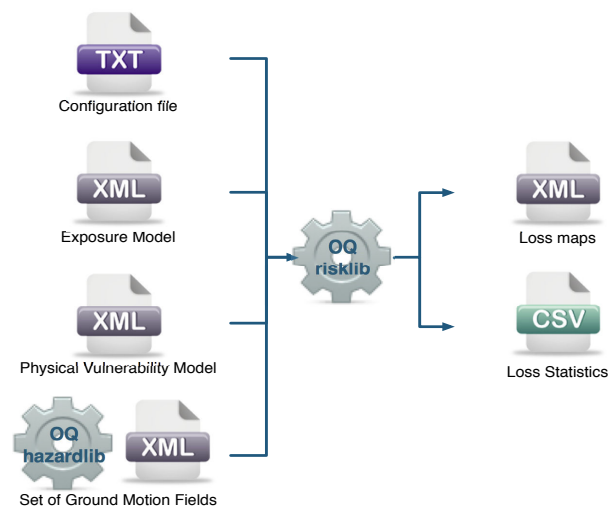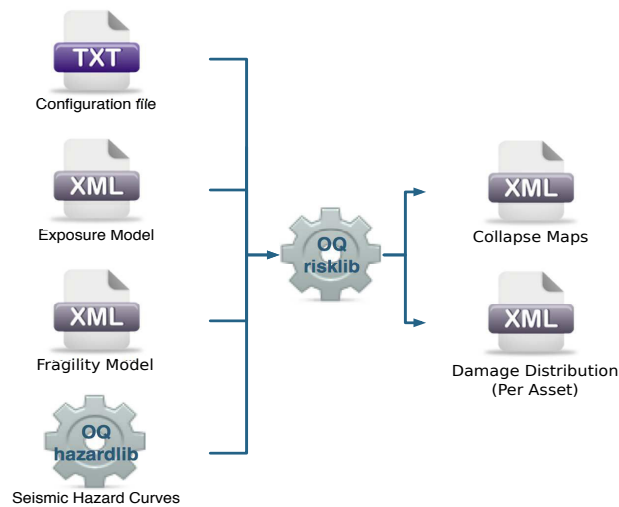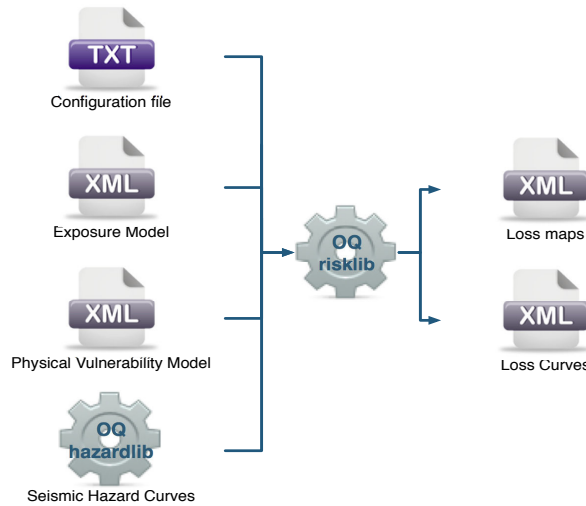

Figure 6.4 – *Classical PSHA-based Risk Calculator input/output structure.*

6.5   Stochastic Event Based Probabilistic Seismic Risk Analysis

This calculator employs an event-based Monte Carlo simulation approach to probabilistic risk assessment in order to estimate the loss distribution for individual assets and aggregated loss distribution for a spatially distributed portfolio of assets within a specified time period. The calculator requires the definition of an exposure model, a vulnerability model for each loss type of interest with vulnerability functions for each taxonomy represented in the exposure model, and a Stochastic Event Set (also known as a *synthetic catalog*) representative of the seismicity of the region over the specified time period. Loss curves and loss maps can currently be calculated for five different loss types using this calculator: structural losses, nonstructural losses, contents losses, downtime losses, and occupant fatalities.

The main results of this calculator are loss exceedance curves for each asset, which describe the probability of exceedance of different loss levels over the specified time period, and loss maps for the region, which describe the loss values that have a given probability of exceedance over the specified time period. Aggregate loss exceedance curves can be also be produced using this calculator; these describe the probability of exceedance of different loss levels for all assets in the portfolio. Finally, event loss tables can be produced using this calculator; these tables describe the total loss across the portfolio for each seismic event in the Stochastic Event Set (SES).

This calculator relies on the probabilistic event-based hazard calculator, which simulates

the seismicity of the chosen time period T by producing a SES. For each earthquake rupture generated by a seismic source, the number of occurrences in the given time span T is simulated by sampling the corresponding probability distribution as given by $P_{rup}(k|T)$. A SES is therefore a *sample* of the full population of earthquake ruptures as defined by a Seismic Source Model. Each earthquake rupture is present zero, one or more times, depending on its probability. Symbolically, we can define a SES as:

$$SES(T) = \left\{ k \times rup, \ k \sim P_{rup}(k|T) \ \forall \ rup \ in \ Src \ \forall \ Src \ in \ SSM \right\} \tag{6.1}$$

where k, the number of occurrences, is a random sample of $P_{rup}(k|T)$, and $k \times rup$ means that earthquake rupture rup is repeated k times in the SES.

For each earthquake rupture or event in the SESs, a spatially correlated GMF realisation is generated, taking into consideration both the inter-event variability of ground motions, and the intra-event residuals obtained from a spatial correlation model for ground motion residuals. The use of logic trees allows for the consideration of uncertainty in the choice of a Seismic Source Model, and in the choice of ground-motion models for the different tectonic regions.

For each GMF realization, a loss ratio is sampled for every asset in the exposure model using the provided probabilistic vulnerability model, taking into consideration the correlation model for vulnerability of different assets of a given taxonomy. Finally loss exceedance curves are computed for both ground-up losses and insured losses.

The required input files required for running a probabilistic stochastic event-based risk calculation and the resulting output files are depicted in Figure 6.5
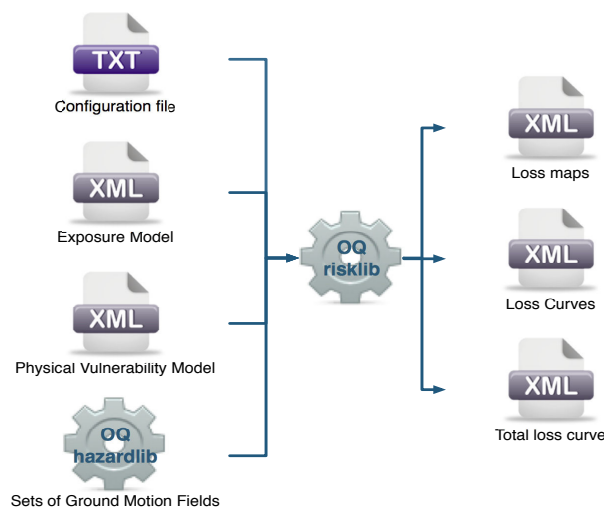
Figure 6.5 – *Probabilistic Event-based Risk Calculator input/output structure.*

6.6   **Retrofit Benefit-Cost Ratio Analysis**

This calculator represents a decision-support tool for deciding whether the employment of retrofitting measures to a collection of existing buildings is advantageous from an economical point of view. For this assessment, the expected losses considering the original and retrofitted configuration of the buildings are estimated, and the economic benefit due to the better seismic design is divided by the retrofitting cost, leading to the benefit/cost ratio. These loss curves are computed using the previously described Classical PSHA- based Risk calculator. The output of this calculator is a benefit/cost ratio for each asset, in which a ratio above one indicates that employing a retrofitting intervention is economically viable.

In Figure 6.6, the input/output structure for this calculator is depicted.
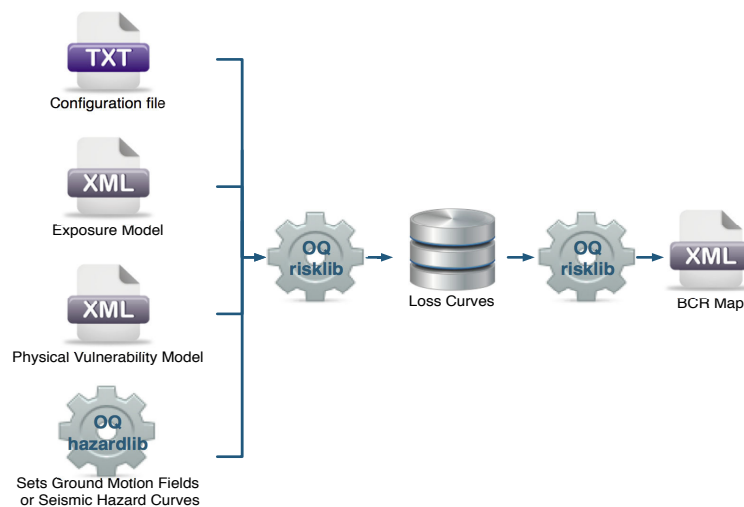
Figure 6.6 – *Retrofitting Benefit/Cost Ratio Calculator input/output structure.*

For further information regarding the theoretical background of the methodologies used for each calculator, users are referred to the OpenQuake- engine Book (Risk).

7. Risk Input Models

The following sections describe the basic inputs required for a risk calculation, including exposure models, fragility models, consequence models, and vulnerability models. In addition, each risk calculator also requires the appropriate hazard inputs computed in the region of interest. Hazard inputs include hazard curves for the classical probabilistic damage and risk calculators, GMF for the scenario damage and risk calculators, or SESs for the probabilistic event based calculators.

7.1 Exposure Models

All risk calculators in the OpenQuake-engine require an exposure model that needs to be provided in the Natural hazards' Risk Markup Language (NRML) schema, the use of which is illustrated through several examples in this section. The information included in an exposure model comprises a metadata section listing general information about the exposure, followed by a cost conversions section that describes how the different areas, costs, and occupancies for the assets will be specified, followed by data regarding each individual asset in the portfolio.

A simple exposure model comprising a single asset is shown in Listing 1.

Let us take a look at each of the sections in the above example file in turn. The first part of the file contains the metadata section:

```
5    <exposureModel id="exposure_example"
6                   category="buildings"
7                   taxonomySource="GEM_Building_Taxonomy_2.0">
8      <description>Exposure Model Example</description>
```

The information in the metadata section is common to all of the assets in the portfolio and needs to be incorporated at the beginning of every exposure model file. There are a number of parameters that compose the metadata section, which is intended to provide

```
1   <?xml version='1.0' encoding='UTF-8'?>
2   <nrml xmlns:gml="http://www.opengis.net/gml"
3         xmlns="http://openquake.org/xmlns/nrml/0.5">
4
5   <exposureModel id="exposure_example"
6                  category="buildings"
7                  taxonomySource="GEM_Building_Taxonomy_2.0">
8     <description>Exposure Model Example</description>
9
10    <conversions>
11      <costTypes>
12        <costType name="structural" type="per_area" unit="USD" />
13      </costTypes>
14      <area type="per_asset" unit="SQM" />
15    </conversions>
16
17    <assets>
18      <asset id="a1" taxonomy="Adobe" number="5" area="100" >
19        <location lon="-122.000" lat="38.113" />
20        <costs>
21          <cost type="structural" value="10000" />
22        </costs>
23        <occupancies>
24          <occupancy occupants="20" period="day" />
25        </occupancies>
26      </asset>
27    </assets>
28
29  </exposureModel>
30
31  </nrml>
```

Listing 1 – *Example exposure model comprising a single asset (Download example)*

general information regarding the assets within the exposure model. These parameters are described below:

- id: mandatory; a unique string used to identify the exposure model. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (–), and underscores (_), with a maximum of 100 characters.
- category: an optional string used to define the type of assets being stored (e.g: buildings, lifelines).
- taxonomySource: an optional attribute used to define the taxonomy being used to classify the assets.
- description: mandatory; a brief string (ASCII) with further information about the exposure model.

Next, let us look at the part of the file describing the area and cost conversions:

```
10    <conversions>
11      <costTypes>
12        <costType name="structural" type="per_area" unit="USD" />
13      </costTypes>
14      <area type="per_asset" unit="SQM" />
15    </conversions>
```

Notice that the costType element defines a name, a type, and a unit attribute.

The NRML schema for the exposure model allows the definition of a structural cost, a nonstructural components cost, a contents cost, and a business interruption or downtime cost for each asset in the portfolio. Thus, the valid values for the name attribute of the costType element are the following:

- structural: used to specify the structural replacement cost of assets
- nonstructural: used to specify the replacement cost for the nonstructural components of assets
- contents: used to specify the contents replacement cost
- business_interruption: used to specify the cost that will be incurred per unit time that a damaged asset remains closed following an earthquake

The exposure model shown in the example above defines only the structural values for the assets. However, multiple cost types can be defined for each asset in the same exposure model.

The unit attribute of the costType element is used for specifying the currency unit for the corresponding cost type. Note that the OpenQuake-engine itself is agnostic to the currency units; the unit is thus a descriptive attribute which is used by the OpenQuake-engine to annotate the results of a risk assessment. This attribute can be set to any valid Unicode string.

The type attribute of the costType element specifies whether the costs will be provided as an aggregated value for an asset, or per building or unit comprising an asset, or per unit area of an asset. The valid values for the type attribute of the costType element are the

following:

- `aggregated`: indicates that the replacement costs will be provided as an aggregated value for each asset
- `per_asset`: indicates that the replacement costs will be provided per structural unit comprising each asset
- `per_area`: indicates that the replacement costs will be provided per unit area for each asset

If the costs are to be specified `per_area` for any of the `costTypes`, the `area` element will also need to be defined in the conversions section. The `area` element defines a `type`, and a `unit` attribute.

The `unit` attribute of the `area` element is used for specifying the units for the area of an asset. The OpenQuake-engine itself is agnostic to the area units; the `unit` is thus a descriptive attribute which is used by the OpenQuake-engine to annotate the results of a risk assessment. This attribute can be set to any valid ASCII string.

The `type` attribute of the `area` element specifies whether the area will be provided as an aggregated value for an asset, or per building or unit comprising an asset. The valid values for the `type` attribute of the `area` element are the following:

- `aggregated`: indicates that the area will be provided as an aggregated value for each asset
- `per_asset`: indicates that the area will be provided per building or unit comprising each asset

The way the information about the characteristics of the assets in an exposure model are stored can vary strongly depending on how and why the data was compiled. As an example, if national census information is used to estimated the distribution of assets in a given region, it is likely that the number of buildings within a given geographical area will be used to define the dataset, and will be used for estimating the number of collapsed buildings for a scenario earthquake. On the other hand, if simplified methodologies based on proxy data such as population distribution are used to develop the exposure model, then it is likely that the built up area or economic cost of each building typology will be directly derived, and will be used for the estimation of economic losses.

Finally, let us look at the part of the file describing the set of assets in the portfolio to be used in seismic damage or risk calculations:

```
17    <assets>
18      <asset id="a1" taxonomy="Adobe" number="5" area="100" >
19        <location lon="-122.000" lat="38.113" />
20        <costs>
21          <cost type="structural" value="10000" />
22        </costs>
23        <occupancies>
24          <occupancy occupants="20" period="day" />
25        </occupancies>
26      </asset>
27    </assets>
```

Each asset definition involves specifiying a set of mandatory and optional attributes concerning the asset. The following set of attributes can be assigned to each asset based on the current schema for the exposure model:

- `id`: mandatory; a unique string used to identify the given asset, which is used by the OpenQuake-engine to relate each asset with its associated results. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_), with a maximum of 100 characters.

- `taxonomy`: mandatory; this string specifies the building typology of the given asset. The taxonomy strings can be user-defined, or based on an existing classification scheme such as the GEM Taxonomy, PAGER, or EMS-98.

- `number`: the number of individual structural units comprising a given asset. This attribute is mandatory for damage calculations. For risk calculations, this attribute must be defined if either the area or any of the costs are provided per structural unit comprising each asset.

- `area`: area of the asset, at a given location. As mentioned earlier, the area is a mandatory attribute only if any one of the costs for the asset is specified per unit area.

- `location`: mandatory; specifies the longitude (between -180° to 180°) and latitude (between -90° to 90 °) of the given asset, both specified in decimal degrees[1].

- `costs`: specifies a set of costs for the given asset. The replacement value for different cost types must be provided on separate lines within the `costs` element. As shown in the example above, each cost entry must define the `type` and the `value`. Currently supported valid options for the cost `type` are: `structural`, `nonstructural`, `contents`, and `business_interruption`.

- `occupancies`: mandatory only for probabilistic or scenario risk calculations that specify an `occupants_vulnerability_file`. Each entry within this element specifies the number of occupants for the asset for a particular period of the day. As shown in the example above, each occupancy entry must define the `period` and the `occupants`. Currently supported valid options for the `period` are: `day`, `transit`,

---

[1]Within the OpenQuake-engine, longitude and latitude coordinates are internally rounded to a precision of 5 digits after the decimal point.

and `night`. Currently, the number of `occupants` for an asset can only be provided as an aggregated value for the asset.

For the purposes of performing a retrofitting benefit/cost analysis, it is also necessary to define the retrofitting cost (`retrofitted`). The combination between the possible options in which these three attributes can be defined leads to four ways of storing the information about the assets. For each of these cases a brief explanation and example is provided in this section.

Example 1

This example illustrates an exposure model in which the aggregated cost (structural, non-structural, contents and business interruption) of the assets of each taxonomy for a set of locations is directly provided. Thus, in order to indicate how the various costs will be defined, the following information needs to be stored in the exposure model file, as shown in Listing 2.

```
 8    <description>
 9      Exposure model with aggregated replacement costs for each asset
10    </description>
11    <conversions>
12      <costTypes>
13        <costType name="structural" type="aggregated" unit="USD" />
14        <costType name="nonstructural" type="aggregated" unit="USD" />
15        <costType name="contents" type="aggregated" unit="USD" />
16        <costType name="business_interruption" type="aggregated" unit="USD/month"/>
17      </costTypes>
18    </conversions>
```

Listing 2 – *Example exposure model using aggregate costs: metadata definition (Download example)*

In this case, the cost `type` of each component as been defined as `aggregated`. Once the way in which each cost is going to be defined has been established, the values for each asset can be stored according to the format shown in Listing 3.

Each asset is uniquely identified by its `id`. Then, a pair of coordinates (latitude and longitude) for a `location` where the asset is assumed to exist is defined. Each asset must be classified according to a `taxonomy`, so that the OpenQuake-engine is capable of employing the appropriate vulnerability function or fragility function in the risk calculations. Finally, the cost values of each `type` are stored within the `costs` attribute. In this example, the aggregated value for all structural units (within a given asset) at each location is provided directly, so there is no need to define other attributes such as `number` or `area`. This mode of representing an exposure model is probably the simplest one.

Example 2

In the snippet shown in Listing 4, an exposure model containing the number of structural units and the associated costs per unit of each asset is presented.

```
19    <assets>
20      <asset id="a1" taxonomy="Adobe" >
21        <location lon="-122.000" lat="38.113" />
22        <costs>
23          <cost type="structural" value="20000" />
24          <cost type="nonstructural" value="30000" />
25          <cost type="contents" value="10000" />
26          <cost type="business_interruption" value="4000" />
27        </costs>
28      </asset>
29    </assets>
```

Listing 3 – *Example exposure model using aggregate costs: assets definition (Download example)*

```
8     <description>
9       Exposure model with replacement costs per building for each asset
10    </description>
11    <conversions>
12      <costTypes>
13        <costType name="structural" type="per_asset" unit="USD" />
14        <costType name="nonstructural" type="per_asset" unit="USD" />
15        <costType name="contents" type="per_asset" unit="USD" />
16        <costType name="business_interruption" type="per_asset" unit="USD/month"/>
17      </costTypes>
18    </conversions>
```

Listing 4 – *Example exposure model using costs per unit: metadata definition (Download example)*

For this case, the cost `type` has been set to `per_asset`. Then, the information from each asset can be stored following the format shown in Listing 5.

```
19    <assets>
20      <asset id="a1" number="2" taxonomy="Adobe" >
21        <location lon="-122.000" lat="38.113" />
22        <costs>
23          <cost type="structural" value="7500" />
24          <cost type="nonstructural" value="11250" />
25          <cost type="contents" value="3750" />
26          <cost type="business_interruption" value="1500" />
27        </costs>
28      </asset>
29    </assets>
```

Listing 5 – *Example exposure model using costs per unit: assets definition (Download example)*

In this example, the various costs for each asset is not provided directly, as in the previous example. In order to carry out the risk calculations in which the economic cost of each asset is provided, the OpenQuake-engine multiplies, for each asset, the number of units (buildings) by the "per asset" replacement cost. Note that in this case, there is no need to specify the attribute `area`.

Example 3

The example shown in Listing 6 comprises an exposure model containing the built up area of each asset, and the associated costs are provided per unit area.

```
8     <description>
9       Exposure model with replacement costs per unit area;
10      and areas provided as aggregated values for each asset
11    </description>
12    <conversions>
13      <area type="aggregated" unit="SQM" />
14      <costTypes>
15        <costType name="structural" type="per_area" unit="USD" />
16        <costType name="nonstructural" type="per_area" unit="USD" />
17        <costType name="contents" type="per_area" unit="USD" />
18        <costType name="business_interruption" type="per_area" unit="USD/month"/>
19      </costTypes>
20    </conversions>
```

Listing 6 – *Example exposure model using costs per unit area and aggregated areas: metadata definition (Download example)*

In order to compile an exposure model with this structure, the cost `type` should be

set to `per_area`. In addition, it is also necessary to specify if the `area` that is being store represents the aggregated area of number of units within an asset, or the average area of a single unit. In this particular case, the `area` that is being stored is the aggregated built up area per asset, and thus this attribute was set to `aggregated`. Listing 7 illustrates the definition of the assets for this example.

```
21   <assets>
22     <asset id="a1" area="1000" taxonomy="Adobe" >
23       <location lon="-122.000" lat="38.113" />
24       <costs>
25         <cost type="structural" value="5" />
26         <cost type="nonstructural" value="7.5" />
27         <cost type="contents" value="2.5" />
28         <cost type="business_interruption" value="1" />
29       </costs>
30     </asset>
31   </assets>
```

Listing 7 – *Example exposure model using costs per unit area and aggregated areas: assets definition* (*Download example*)

Once again, the OpenQuake-engine needs to carry out some calculations in order to compute the different costs per asset. In this case, this value is computed by multiplying the aggregated built up `area` of each asset by the associated cost per unit area. Notice that in this case, there is no need to specify the attribute `number`.

Example 4

This example demonstrates an exposure model that defines the number of structural units for each asset, the average built up area per structural unit and the associated costs per unit area. Listing 8 shows the metadata definition for an exposure model built in this manner.

Similarly to what was described in the previous example, the various costs `type` also need to be established as `per_area`, but the `type` of area is now defined as `per_asset`. Listing 9 illustrates the definition of the assets for this example.

In this example, the OpenQuake-engine will make use of all the parameters to estimate the various costs of each asset, by multiplying the number of structural units by its average built up area, and then by the respective cost per unit area.

Example 5

In this example, additional information will be included, which is required for other risk analysis besides loss estimation, such as the calculation of insured losses or benefit/cost analysis. For the calculation of insured losses, it is necessary to establish how the insurance limits and deductibles are going to be defined. Listing 10 illustrates the metadata section of an exposure model where the insurance limits and deductibles for structural components

```
 8      <description>
 9        Exposure model with replacement costs per unit area;
10        and areas provided per building for each asset
11      </description>
12      <conversions>
13        <area type="per_asset" unit="SQM" />
14        <costTypes>
15          <costType name="structural" type="per_area" unit="USD" />
16          <costType name="nonstructural" type="per_area" unit="USD" />
17          <costType name="contents" type="per_area" unit="USD" />
18          <costType name="business_interruption" type="per_area" unit="USD/month"/>
19        </costTypes>
20      </conversions>
```

Listing 8 – *Example exposure model using costs per unit area and areas per unit: metadata definition*
(*Download example*)

```
21      <assets>
22        <asset id="a1" number="3" area="400" taxonomy="Adobe" >
23          <location lon="-122.000" lat="38.113" />
24          <costs>
25            <cost type="structural" value="10" />
26            <cost type="nonstructural" value="15" />
27            <cost type="contents" value="5" />
28            <cost type="business_interruption" value="2" />
29          </costs>
30        </asset>
31      </assets>
```

Listing 9 – *Example exposure model using costs per unit area and areas per unit: assets definition*
(*Download example*)

will be defined relative to the structural replacement cost.

```
8    <description>
9      Exposure model with relative insurance limits and deductibles
10   </description>
11   <conversions>
12     <area type="per_asset" unit="SQM" />
13     <costTypes>
14       <costType name="structural" type="per_asset" unit="USD" />
15       <costType name="nonstructural" type="per_asset" unit="USD" />
16       <costType name="contents" type="per_asset" unit="USD" />
17       <costType name="business_interruption" type="per_asset" unit="USD/month" />
18     </costTypes>
19     <insuranceLimit isAbsolute="false"/>
20     <deductible isAbsolute="false"/>
21   </conversions>
```

Listing 10 – *Example exposure model using relative insurance limits and deductibles: metadata definition (Download example)*

In this example, both the insurance limit and the deductible are defined as a fraction of the replacement cost, by setting the attribute `isAbsolute` to `false`. Then, for each type of cost, the limit and deductible value can be stored for each asset, as illustrated in the snippet shown in Listing 11.

```
22   <assets>
23     <asset id="a1" number="1" area="100" taxonomy="Adobe" >
24       <location lon="-122.000" lat="38.113" />
25       <costs>
26         <cost type="structural" value="10000" deductible="0.1" insuranceLimit="0.8" />
27         <cost type="nonstructural" value="15000" />
28         <cost type="contents" value="5000" />
29         <cost type="business_interruption" value="2000" />
30       </costs>
31     </asset>
32   </assets>
```

Listing 11 – *Example exposure model using relative insurance limits and deductibles: assets definition (Download example)*

On the other hand, a user could define one or both of these parameters as absolute values, by setting the aforementioned attribute to `true`. This is shown in the example shown in Listing 12.

Moreover, in order to perform a benefit/cost assessment, it is also necessary to indicate the retrofitting cost. This parameter is handled in the same manner as the structural cost,

```xml
1   <?xml version='1.0' encoding='UTF-8'?>
2   <nrml xmlns:gml="http://www.opengis.net/gml"
3         xmlns="http://openquake.org/xmlns/nrml/0.4">
4
5   <exposureModel id="exposure_example"
6                   category="buildings"
7                   taxonomySource="GEM_Building_Taxonomy_2.0">
8     <description>
9       Exposure model with absolute insurance limits and deductibles
10    </description>
11    <conversions>
12      <area type="per_asset" unit="SQM" />
13      <costTypes>
14        <costType name="structural" type="per_area" unit="USD" />
15        <costType name="nonstructural" type="per_area" unit="USD" />
16        <costType name="contents" type="per_area" unit="USD" />
17        <costType name="business_interruption" type="per_area" unit="USD/month"/>
18      </costTypes>
19      <insuranceLimit isAbsolute="true"/>
20      <deductible isAbsolute="true"/>
21    </conversions>
22    <assets>
23      <asset id="a1" number="2" area="1000" taxonomy="Adobe" >
24        <location lon="-122.000" lat="38.113" />
25        <costs>
26          <cost type="structural" value="5" deductible="1" insuranceLimit="4"/>
27          <cost type="nonstructural" value="7.5" />
28          <cost type="contents" value="2.5" />
29          <cost type="business_interruption" value="1" />
30        </costs>
31      </asset>
32    </assets>
33  </exposureModel>
34
35  </nrml>
```

Listing 12 – *Example exposure model using absolute insurance limits and deductibles (Download example)*

and it should be stored according to the format shown in Listing 13.

```xml
1   <?xml version='1.0' encoding='UTF-8'?>
2   <nrml xmlns:gml="http://www.opengis.net/gml"
3         xmlns="http://openquake.org/xmlns/nrml/0.5">
4
5   <exposureModel id="exposure_example"
6                  category="buildings"
7                  taxonomySource="GEM_Building_Taxonomy_2.0">
8     <description>Exposure model illustrating retrofit costs</description>
9     <conversions>
10      <costTypes>
11        <costType name="structural" type="aggregated" unit="USD"
12                  retrofittedType="per_asset" retrofittedUnit="USD" />
13      </costTypes>
14    </conversions>
15    <assets>
16      <asset id="a1" taxonomy="Adobe" number="1" >
17        <location lon="-122.000" lat="38.113" />
18        <costs>
19          <cost type="structural" value="10000" retrofitted="2000" />
20        </costs>
21      </asset>
22    </assets>
23  </exposureModel>
24
25  </nrml>
```

Listing 13 – *Example exposure model specifying retrofit costs (Download example)*

Despite the fact that for the demonstration of how the insurance parameters and retrofitting cost can be stored the per building type of cost structure described in Example 1 was used, it is important to mention that any of the other cost storing approaches can also be employed (Examples 2–4).

Example 6

The OpenQuake-engine is also capable of estimating human losses, based on the number of occupants in an asset, at a certain time of the day. The example exposure model shown in Listing 14 illustrates how this parameter is defined for each asset. In addition, this example also serves the purpose of presenting an exposure model in which three cost types have been defined using three different options.

As previously mentioned, in this example only three costs are being stored, and each one follows a different approach. The `structural` cost is being defined as the aggregate replacement cost for all of the buildings comprising the asset (Example 1), the `nonstructural` value is defined as the replacement cost per unit area where the area is defined per building

comprising the asset (Example 4), and the `contents` and `business_interruption` values are provided per building comprising the asset (Example 2). The number of occupants at different times of the day are also provided as aggregated values for all of the buildings comprising the asset.

Reusing an exposure model in subsequent calculations

For calculations involving large exposure models, the OpenQuake-engine makes it possible to read the exposure model once and reuse the same exposure in subsequent calculations. The first time the computation involving the exposure model is run, the exposure will be read and saved in the datastore. Then, if you need to re-run the computation by changing some parameter external to the exposure, you can comment out the exposure from the job.ini file and give the command

```
user@ubuntu:~\$ oq-engine --run job.ini --hc <calc_id>
```

where calc_id is the ID of the previous calculation. The original exposure will be re-used. Note that if a hazard calculation uses an exposure model to specify the sites where the hazard is computed; then any other exposure model provided in a subsequent risk calculation which uses this hazard calculation will be ignored. In such cases, the risk results will be computed based on the exposure model previously specified in the hazard calculation.

A web-based tool to build an exposure model in the NRML schema are also under development, and can be found at the OpenQuake platform at the following address: https://platform.openquake.org/ript/.

7.2 Fragility Models

This section describes the schema currently used to store fragility models, which are required for the Scenario Damage Calculator and the Classical Probabilistic Seismic Damage Calculator. In order to perform probabilistic or scenario damage calculations, it is necessary to define a fragility function for each building typology present in the exposure model. A fragility model defines a set of fragility functions, describing the probability of exceeding a set of limit, or damage, states. The fragility functions can be defined using either a discrete or a continuous format, and the fragility model file can include a mix of both types of fragility functions.

For discrete fragility functions, sets of probabilities of exceedance (one set per limit state) are defined for a list of intensity measure levels, as illustrated in Figure 7.1.

The fragility functions can also be defined as continuous functions, through the use of cumulative lognormal distribution functions. In Figure 7.2, a continuous fragility model is presented.

An example fragility model comprising one discrete fragility function and one continuous fragility function is shown in Listing 15.

The initial portion of the schema contains general information that describes some general aspects of the fragility model. The information in this metadata section is common

```xml
1   <?xml version='1.0' encoding='UTF-8'?>
2   <nrml xmlns:gml="http://www.opengis.net/gml"
3         xmlns="http://openquake.org/xmlns/nrml/0.5">
4
5   <exposureModel id="exposure_example"
6                  category="buildings"
7                  taxonomySource="GEM_Building_Taxonomy_2.0">
8     <description>Exposure model example with occupants</description>
9     <conversions>
10      <costTypes>
11        <costType name="structural" type="aggregated" unit="USD" />
12        <costType name="nonstructural" type="per_area" unit="USD" />
13        <costType name="contents" type="per_asset" unit="USD" />
14        <costType name="business_interruption" type="per_asset" unit="USD/month" />
15      </costTypes>
16      <area type="per_asset" unit="SQM" />
17    </conversions>
18    <assets>
19      <asset id="a1" taxonomy="Adobe" number="5" area="200" >
20        <location lon="-122.000" lat="38.113" />
21        <costs>
22          <cost type="structural" value="20000" />
23          <cost type="nonstructural" value="15" />
24          <cost type="contents" value="2400" />
25          <cost type="business_interruption" value="1500" />
26        </costs>
27        <occupancies>
28          <occupancy occupants="6" period="day" />
29          <occupancy occupants="10" period="transit" />
30          <occupancy occupants="20" period="night" />
31        </occupancies>
32      </asset>
33    </assets>
34  </exposureModel>
35
36  </nrml>
```

Listing 14 – *Example exposure model specifying the aggregate number of occupants per asset* (*Download example*)

```
1   <?xml version='1.0' encoding='UTF-8'?>
2   <nrml xmlns="http://openquake.org/xmlns/nrml/0.5">
3
4   <fragilityModel id="fragility_example"
5                   assetCategory="buildings"
6                   lossCategory="structural">
7
8     <description>Fragility Model Example</description>
9     <limitStates>slight moderate extensive complete</limitStates>
10
11    <fragilityFunction id="Woodframe_TwoStorey" format="discrete">
12      <imls imt="PGA" noDamageLimit="0.05">0.005 0.2 0.4 0.6 0.8 1.0 1.2</imls>
13      <poes ls="slight">0.00 0.01 0.15 0.84 0.99 1.00 1.00</poes>
14      <poes ls="moderate">0.00 0.00 0.01 0.12 0.35 0.57 0.74</poes>
15      <poes ls="extensive">0.00 0.00 0.00 0.08 0.19 0.32 0.45</poes>
16      <poes ls="complete">0.00 0.00 0.00 0.06 0.17 0.26 0.35</poes>
17    </fragilityFunction>
18
19    <fragilityFunction id="RC_LowRise" format="continuous" shape="logncdf">
20      <imls imt="SA(0.3)" noDamageLimit="0.05" minIML="0.0" maxIML="5.0"/>
21      <params ls="slight" mean="0.50" stddev="0.10"/>
22      <params ls="moderate" mean="1.00" stddev="0.40"/>
23      <params ls="extensive" mean="1.50" stddev="0.90"/>
24      <params ls="complete" mean="2.00" stddev="1.60"/>
25    </fragilityFunction>
26
27  </fragilityModel>
28
29  </nrml>
```

Listing 15 – *Example fragility model comprising one discrete fragility function and one continuous fragility function (Download example)*
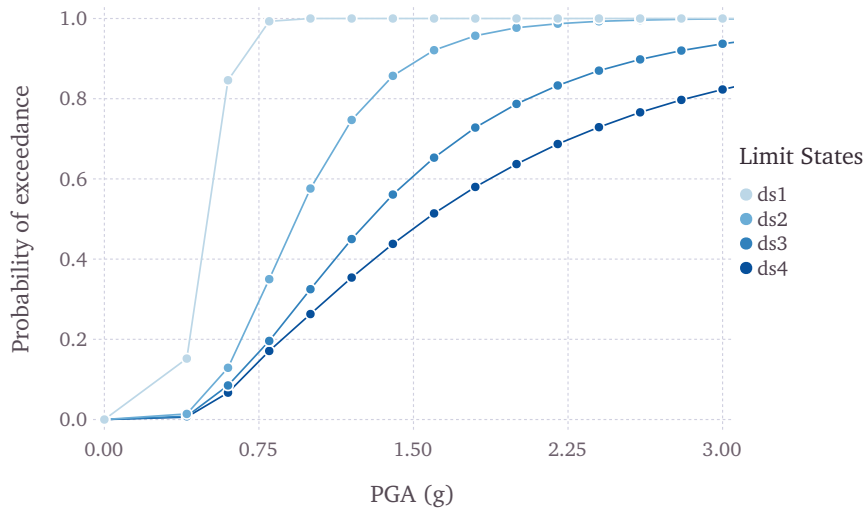
Figure 7.1 – *Graphical representation of a discrete fragility model*

to all of the functions in the fragility model and needs to be included at the beginning of every fragility model file. The parameters of the metadata section are shown in the snippet below and described after the snippet:

```
4   <fragilityModel id="fragility_example"
5                   assetCategory="buildings"
6                   lossCategory="structural">
7
8     <description>Fragility Model Example</description>
9     <limitStates>slight moderate extensive complete</limitStates>
```

- `id`: mandatory; a unique string used to identify the fragility model. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_), with a maximum of 100 characters.
- `assetCategory`: an optional string used to specify the type of assets for which fragility functions will be defined in this file (e.g: buildings, lifelines).
- `lossCategory`: mandatory; valid strings for this attribute are "structural", "nonstructural", "contents", and "business_interruption".
- `description`: mandatory; a brief string (ASCII) with further relevant information about the fragility model, for example, which building typologies are covered or the source of the functions in the fragility model.
- `limitStates`: mandatory; this field is used to define the number and nomenclature of each limit state. Four limit states are employed in the example above, but it is possible to use any number of discrete states, as long as a fragility curve is always defined for each limit state. The limit states must be provided as a set of strings separated by whitespaces between each limit state. Each limit state string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_). Please ensure that
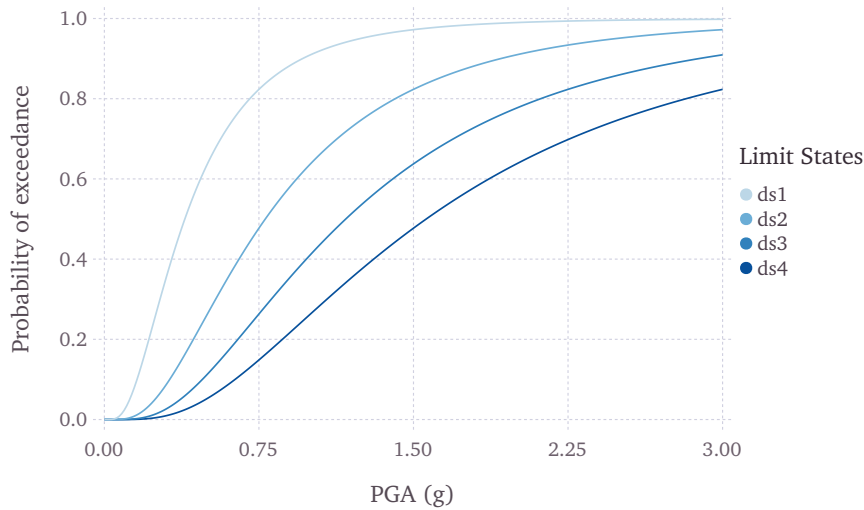
Figure 7.2 – *Graphical representation of a continuous fragility model*

there is no whitespace within the name of any individual limit state.

The following snippet from the above fragility model example file defines a discrete fragility function:

```
11    <fragilityFunction id="Woodframe_TwoStorey" format="discrete">
12      <imls imt="PGA" noDamageLimit="0.05">0.005 0.2 0.4 0.6 0.8 1.0 1.2</imls>
13      <poes ls="slight">0.00 0.01 0.15 0.84 0.99 1.00 1.00</poes>
14      <poes ls="moderate">0.00 0.00 0.01 0.12 0.35 0.57 0.74</poes>
15      <poes ls="extensive">0.00 0.00 0.00 0.08 0.19 0.32 0.45</poes>
16      <poes ls="complete">0.00 0.00 0.00 0.06 0.17 0.26 0.35</poes>
17    </fragilityFunction>
```

The following attributes are needed to define a discrete fragility function:

- `id`: mandatory; a unique string used to identify the taxonomy for which the function is being defined. This string is used to relate the fragility function with the relevant asset in the exposure model. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_), with a maximum of 100 characters.

- `format`: mandatory; for discrete fragility functions, this attribute should be set to "`discrete`".

- `imls`: mandatory; this attribute specifies the list of intensity levels for which the limit state probabilities of exceedance will be defined. In addition, it is also necessary to define the intensity measure type (`imt`). Optionally, a `noDamageLimit` can be specified, which defines the intensity level below which the probability of exceedance for all limit states is taken to be zero.

- `poes`: mandatory; this field is used to define the probabilities of exceedance (`poes`) for each limit state for each discrete fragility function. It is also necessary to specify which limit state the exceedance probabilities are being defined for using the attribute

ls. The probabilities of exceedance for each limit state must be provided on a separate line; and the number of exceedance probabilities for each limit state defined by the poes attribute must be equal to the number of intensity levels defined by the attribute imls. Finally, the number and names of the limit states in each fragility function must be equal to the number of limit states defined earlier in the metadata section of the fragility model using the attribute limitStates.

The following snippet from the above fragility model example file defines a continuous fragility function:

```
19    <fragilityFunction id="RC_LowRise" format="continuous" shape="logncdf">
20      <imls imt="SA(0.3)" noDamageLimit="0.05" minIML="0.0" maxIML="5.0"/>
21      <params ls="slight" mean="0.50" stddev="0.10"/>
22      <params ls="moderate" mean="1.00" stddev="0.40"/>
23      <params ls="extensive" mean="1.50" stddev="0.90"/>
24      <params ls="complete" mean="2.00" stddev="1.60"/>
25    </fragilityFunction>
```

The following attributes are needed to define a continuous fragility function:

- id: mandatory; a unique string used to identify the taxonomy for which the function is being defined. This string is used to relate the fragility function with the relevant asset in the exposure model. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_), with a maximum of 100 characters.

- format: mandatory; for continuous fragility functions, this attribute should be set to "continuous".

- shape: mandatory; for continuous fragility functions using the lognormal cumulative distrution, this attribute should be set to "logncdf". At present, only the lognormal cumulative distribution function can be used for representing continuous fragility functions.

- imls: mandatory; this element specifies aspects related to the intensity measure used by the the fragility function. The range of intensity levels for which the continuous fragility functions are valid is specified using the attributes minIML and maxIML. In addition, it is also necessary to define the intensity measure type imt. Optionally, a noDamageLimit can be specified, which defines the intensity level below which the probability of exceedance for all limit states is taken to be zero.

- params: mandatory; this field is used to define the parameters of the continuous curve for each limit state for this fragility function. For a lognormal cumulative distrbution function, the two parameters required to specify the function are the mean and standard deviation of the intensity level. These parameters are defined for each limit state using the attributes mean and stddev respectively. The attribute ls specifies the limit state for which the parameters are being defined. The parameters for each limit state must be provided on a separate line. The number and names of the limit states in each fragility function must be equal to the number of limit states defined earlier in the

metadata section of the fragility model using the attribute `limitStates`.

Note that the schema for representing fragility models has changed between NRML v0.4 (used prior to OpenQuake-engine v1.7 (oq-engine 1.7)) and NRML v0.5 (introduced in oq-engine 1.7).

A deprecation warning is printed every time you attempt to use a fragility model in the old NRML v0.4 format in an oq-engine 1.7 (or later) risk calculation. To get rid of the warning you must upgrade the old fragility models files to NRML v0.5. You can use the command `upgrade_nrml` with oq-lite to do this as follows:

```
user@ubuntu:~\$ oq-lite upgrade_nrml <directory-name>
```

The above command will upgrade all of your old fragility model files to NRML v0.5. The original files will be kept, but with a .bak extension appended. Notice that you will need to set the `lossCategory` attribute to its correct value manually. This is easy to do, since if you try to run a computation you will get a clear error message telling the expected value for the `lossCategory` for each file.

Several methodologies to derive fragility functions are currently being evaluated by GEM and have been included as part of the Risk Modeller's Toolkit, the code for which can be found on a public repository at GitHub at the following address: http://github.com/gemsciencetools/rmtk.

A web-based tool to build a fragility model in the NRML schema are also under development, and can be found at the OpenQuake platform at the following address: https://platform.openquake.org/ript

7.3   Consequence Models

Starting from OpenQuake-engine v1.7, the Scenario Damage calculator also accepts consequence models in addition to fragility models, in order to estimate consequences based on the calculated damage distribution. The user may provide one consequence model file corresponding to each loss type (amongst structural, nonstructural, contents, and business interruption) for which a fragility model file is provided. Whereas providing a fragility model file for at least one loss type is mandatory for running a Scenario Damage calculation, providing corresponding consequence model files is optional.

This section describes the schema currently used to store consequence models, which are optional inputs for the Scenario Damage Calculator. A consequence model defines a set of consequence functions, describing the distribution of the loss (or consequence) ratio conditional on a set of discrete limit (or damage) states. These consequence function can be currently defined in OpenQuake-engine by specifying the parameters of the continuous distribution of the loss ratio for each limit state specified in the fragility model for the corresponding loss type, for each taxonomy defined in the exposure model.

An example consequence model is shown in Listing 16.

The initial portion of the schema contains general information that describes some general

```xml
1  <?xml version='1.0' encoding='utf-8'?>
2  <nrml xmlns="http://openquake.org/xmlns/nrml/0.5">
3
4  <consequenceModel id="consequence_example"
5                    assetCategory="buildings"
6                    lossCategory="structural">
7
8    <description>Consequence Model Example</description>
9    <limitStates>slight moderate extensive complete</limitStates>
10
11   <consequenceFunction id="RC_LowRise" dist="LN">
12     <params ls="slight" mean="0.04" stddev="0.00"/>
13     <params ls="moderate" mean="0.16" stddev="0.00"/>
14     <params ls="extensive" mean="0.32" stddev="0.00"/>
15     <params ls="complete" mean="0.64" stddev="0.00"/>
16   </consequenceFunction>
17
18  </consequenceModel>
19
20  </nrml>
```

Listing 16 – *Example consequence model ([Download example](#))*

aspects of the consequence model. The information in this metadata section is common to all of the functions in the consequence model and needs to be included at the beginning of every consequence model file. The parameters are described below:

- id: a unique string used to identify the consequence model. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_), with a maximum of 100 characters.
- assetCategory: an optional string used to specify the type of assets for which consequence functions will be defined in this file (e.g: buildings, lifelines).
- lossCategory: mandatory; valid strings for this attribute are "structural", "nonstructural", "contents", and "business_interruption".
- description: mandatory; a brief string (ASCII) with further information about the consequence model, for example, which building typologies are covered or the source of the functions in the consequence model.
- limitStates: mandatory; this field is used to define the number and nomenclature of each limit state. Four limit states are employed in the example above, but it is possible to use any number of discrete states. The limit states must be provided as a set of strings separated by whitespaces between each limit state. Each limit state string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_). Please ensure that there is no whitespace within the name of any individual limit state. The number and nomenclature of the limit states used in the consequence model should

match those used in the corresponding fragility model.

```
4    <consequenceModel id="consequence_example"
5                      assetCategory="buildings"
6                      lossCategory="structural">
7
8      <description>Consequence Model Example</description>
9      <limitStates>slight moderate extensive complete</limitStates>
```

The following snippet from the above consequence model example file defines a consequence function using a lognormal distribution to model the uncertainty in the consequence ratio for each limit state:

```
11     <consequenceFunction id="RC_LowRise" dist="LN">
12       <params ls="slight" mean="0.04" stddev="0.00"/>
13       <params ls="moderate" mean="0.16" stddev="0.00"/>
14       <params ls="extensive" mean="0.32" stddev="0.00"/>
15       <params ls="complete" mean="0.64" stddev="0.00"/>
16     </consequenceFunction>
```

The following attributes are needed to define a consequence function:

- `id`: mandatory; a unique string used to identify the taxonomy for which the function is being defined. This string is used to relate the consequence function with the relevant asset in the exposure model. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_), with a maximum of 100 characters.

- `dist`: mandatory; for vulnerability function which use a continuous distribution to model the uncertainty in the conditional loss ratios, this attribute should be set to either "LN" if using the lognormal distribution, or to "BT" if using the Beta distribution [2].

- `params`: mandatory; this field is used to define the parameters of the continuous distribution used for modelling the uncertainty in the loss ratios for each limit state for this consequence function. For a lognormal distrbution, the two parameters required to specify the function are the mean and standard deviation of the consequence ratio. These parameters are defined for each limit state using the attributes `mean` and `stddev` respectively. The attribute `ls` specifies the limit state for which the parameters are being defined. The parameters for each limit state must be provided on a separate line. The number and names of the limit states in each consequence function must be equal to the number of limit states defined in the corresponding fragility model using the attribute `limitStates`.

---

[2]Note that as of OpenQuake-engine v1.8, the uncertainty in the consequence ratios is ignored, and only the mean consequence ratios for the set of limit states is considered when computing the consequences from the damage distribution. Consideration of the uncertainty in the consequence ratios is planned for future releases of the OpenQuake-engine.

7.4 Vulnerability Models

In order to perform probabilistic or scenario risk calculations, it is necessary to define a vulnerability function for each building typology present in the exposure model. In this section, the schema for the vulnerability model is described in detail. A graphical representation of a vulnerability model (mean loss ratio for a set of intensity measure levels) is illustrated in Figure 7.3.
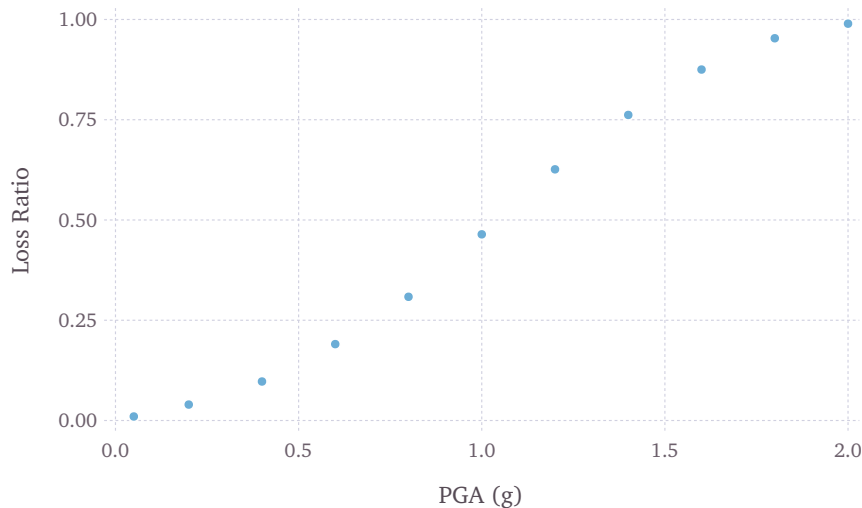
Figure 7.3 – *Graphical representation of a vulnerability model*

Note that although the uncertainty for each loss ratio is not represented in Figure 7.3, it can be considered in the input file, by means of a coefficient of variation per loss ratio and a probabilistic distribution, which can currently be set to lognormal (LN), Beta (BT); or by specifying a discrete probability mass (PM)[3] distribution of the loss ratio at a set of intensity levels. An example of a vulnerability function that models the uncertainty in the loss ratio at different intensity levels using a lognormal distribution is illustrated in Figure 7.4.

In general, defining vulnerability functions requires the user to specify the distribution of the loss ratio for a set of intensity levels. The loss ratio distributions can be defined using either a discrete or a continuous format, and the vulnerability model file can include a mix of both types of vulnerability functions. It is also possible to define a vulnerability function using a set of deterministic loss ratios corresponding to a set of intensity levels (i.e., ignoring the uncertainty in the conditional loss ratios).

An example vulnerability model comprising three vulnerability functions is shown in Listing 17. This vulnerability model contains one function that uses the lognormal distribution to represent the uncertainty in the loss ratio at different intensity levels, one function that

---

[3]As of OpenQuake-engine v1.8, the "PM" option for defining vulnerability functions is supported by the Scenario Risk and the Stochastic Event-Based Probabilistic Risk Calculators, but not by the Classical Probabilistic Risk Calculator.
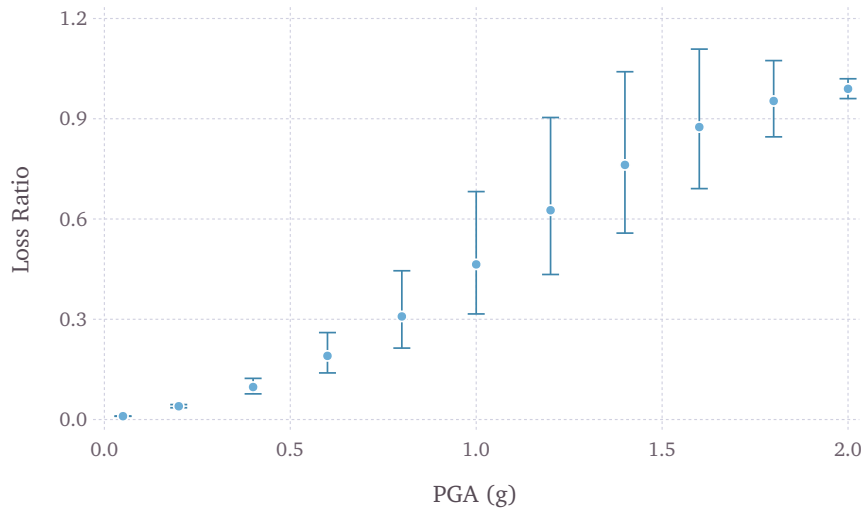
Figure 7.4 – *Graphical representation of a vulnerability function that models the uncertainty in the loss ratio using a lognormal distribution. The mean loss ratios and coefficients of variation are illustrated for a set of intensity levels.*

uses the Beta distribution, and one function that is defined using a discrete probability mass distribution.

The initial portion of the schema contains general information that describes some general aspects of the vulnerability model. The information in this metadata section is common to all of the functions in the vulnerability model and needs to be included at the beginning of every vulnerability model file. The parameters are illustrated in the snippet shown and described below:

```
4   <vulnerabilityModel id="vulnerability_example"
5                       assetCategory="buildings"
6                       lossCategory="structural">
7
8     <description>vulnerability model</description>
```

- id: a unique string (ASCII) used to identify the vulnerability model. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_), with a maximum of 100 characters.
- assetCategory: an optional string (ASCII) used to specify the type of assets for which vulnerability functions will be defined in this file (e.g: buildings, lifelines).
- lossCategory: mandatory; valid strings for this attribute are "structural", "nonstructural", "contents", "business_interruption", and "occupants".
- description: mandatory; a brief string with further information about the vulnerability model, for example, which building typologies are covered or the source of the functions in the vulnerability model.

The following snippet from the above vulnerability model example file defines a vulnera-

```xml
<?xml version='1.0' encoding='utf-8'?>
<nrml xmlns="http://openquake.org/xmlns/nrml/0.5">

<vulnerabilityModel id="vulnerability_example"
                    assetCategory="buildings"
                    lossCategory="structural">

  <description>vulnerability model</description>

  <vulnerabilityFunction id="W1_Res_LowCode" dist="LN">
   <imls imt="PGA">0.005 0.15 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0</imls>
   <meanLRs>0.01 0.04 0.10 0.20 0.33 0.50 0.67 0.80 0.90 0.96 0.99</meanLRs>
   <covLRs>0.03 0.12 0.24 0.32 0.38 0.40 0.38 0.32 0.24 0.12 0.03</covLRs>
  </vulnerabilityFunction>


  <vulnerabilityFunction id="S1_Res_HighCode" dist="BT">
   <imls imt="SA(0.3)">0.05 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0</imls>
   <meanLRs>0.01 0.03 0.07 0.15 0.24 0.37 0.50 0.60 0.67 0.72 0.75</meanLRs>
   <covLRs>0.03 0.12 0.24 0.32 0.38 0.40 0.38 0.32 0.24 0.12 0.03</covLRs>
  </vulnerabilityFunction>


  <vulnerabilityFunction id="ATC13_URM_Res" dist="PM">
   <imls imt="MMI">6 7 8 9 10 11 12</imls>
   <probabilities lr="0.000">0.95 0.49 0.30 0.14 0.03 0.01 0.00</probabilities>
   <probabilities lr="0.005">0.03 0.38 0.40 0.30 0.10 0.03 0.01</probabilities>
   <probabilities lr="0.050">0.02 0.08 0.16 0.24 0.30 0.10 0.01</probabilities>
   <probabilities lr="0.200">0.00 0.02 0.08 0.16 0.26 0.30 0.03</probabilities>
   <probabilities lr="0.450">0.00 0.02 0.03 0.10 0.18 0.30 0.18</probabilities>
   <probabilities lr="0.800">0.00 0.01 0.02 0.04 0.10 0.18 0.39</probabilities>
   <probabilities lr="1.000">0.00 0.01 0.01 0.02 0.03 0.08 0.38</probabilities>
  </vulnerabilityFunction>

</vulnerabilityModel>

</nrml>
```

Listing 17 – *Example vulnerability model (Download example)*

bility function modelling the uncertainty in the conditional loss ratios using a (continuous) lognormal distribution:

```
10    <vulnerabilityFunction id="W1_Res_LowCode" dist="LN">
11     <imls imt="PGA">0.005 0.15 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0</imls>
12     <meanLRs>0.01 0.04 0.10 0.20 0.33 0.50 0.67 0.80 0.90 0.96 0.99</meanLRs>
13     <covLRs>0.03 0.12 0.24 0.32 0.38 0.40 0.38 0.32 0.24 0.12 0.03</covLRs>
14    </vulnerabilityFunction>
```

The following attributes are needed to define a vulnerability function which uses a continuous distribution to model the uncertainty in the conditional loss ratios:

- `id`: a unique string (ASCII) used to identify the taxonomy for which the function is being defined. This string is used to relate the vulnerability function with the relevant asset in the exposure model. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_), with a maximum of 100 characters.
- `dist`: mandatory; for vulnerability functions which use a continuous distribution to model the uncertainty in the conditional loss ratios, this attribute should be set to either "LN" if using the lognormal distribution, or to "BT" if using the Beta distribution.
- `imls`: mandatory; this attribute specifies the list of intensity levels for which the parameters of the conditional loss ratio distributions will be defined. In addition, it is also necessary to define the intensity measure type (`imt`).
- `meanLRs`: mandatory; this field is used to define the mean loss ratios for this vulnerability function for each of the intensity levels defined by the attribute `imls`. The number of mean loss ratios defined by the `meanLRs` attribute must be equal to the number of intensity levels defined by the attribute `imls`.
- `covLRs`: mandatory; this field is used to define the coefficient of variation for the conditional distribution of the loss ratios for this vulnerability function for each of the intensity levels defined by the attribute `imls`. The number of coefficients of variation of loss ratios defined by the `covLRs` attribute must be equal to the number of intensity levels defined by the attribute `imls`. The uncertainty in the conditional loss ratios can be ignored by setting all of the `covLRs` for a given vulnerability function to zero.

The next snippet from the vulnerability model example file of Listing 17 defines a vulnerability function which models the uncertainty in the conditional loss ratios using a (discrete) probability mass distribution:

```
24      <vulnerabilityFunction id="ATC13_URM_Res" dist="PM">
25      <imls imt="MMI">6 7 8 9 10 11 12</imls>
26      <probabilities lr="0.000">0.95 0.49 0.30 0.14 0.03 0.01 0.00</probabilities>
27      <probabilities lr="0.005">0.03 0.38 0.40 0.30 0.10 0.03 0.01</probabilities>
28      <probabilities lr="0.050">0.02 0.08 0.16 0.24 0.30 0.10 0.01</probabilities>
29      <probabilities lr="0.200">0.00 0.02 0.08 0.16 0.26 0.30 0.03</probabilities>
30      <probabilities lr="0.450">0.00 0.02 0.03 0.10 0.18 0.30 0.18</probabilities>
31      <probabilities lr="0.800">0.00 0.01 0.02 0.04 0.10 0.18 0.39</probabilities>
32      <probabilities lr="1.000">0.00 0.01 0.01 0.02 0.03 0.08 0.38</probabilities>
33     </vulnerabilityFunction>
```

The following attributes are needed to define a vulnerability function which uses a discrete probability mass distribution to model the uncertainty in the conditional loss ratios:

- `id`: a unique string (ASCII) used to identify the taxonomy for which the function is being defined. This string is used to relate the vulnerability function with the relevant asset in the exposure model. This string can contain letters (a–z; A–Z), numbers (0–9), dashes (-), and underscores (_), with a maximum of 100 characters.

- `dist`: mandatory; for vulnerability functions which use a discrete probability mass distribution to model the uncertainty in the conditional loss ratios, this attribute should be set to "PM".

- `imls`: mandatory; this attribute specifies the list of intensity levels for which the parameters of the conditional loss ratio distributions will be defined. In addition, it is also necessary to define the intensity measure type (`imt`).

- `probabilities`: mandatory; this field is used to define the probability of observing a particular loss ratio (specified for each row of `probabilities` using the attribute `lr`), conditional on the set of intensity levels specified using the attribute `imls`. for this vulnerability function. Thus, the number of probabilities defined by each `probabilities` attribute must be equal to the number of intensity levels defined by the attribute `imls`. On the other hand, there is no limit to the number of loss ratios for which `probabilities` can be defined. In the example shown here, notice that the set of probabilities conditional on any particular intensity level, say, $MMI = 8$, sum up to one.

Note that the schema for representing vulnerability models has changed between NRML v0.4 (used prior to oq-engine 1.7) and NRML v0.5 (introduced in oq-engine 1.7).

A deprecation warning is printed every time you attempt to use a vulnerability model in the old NRML v0.4 format in an oq-engine 1.7 (or later) risk calculation. To get rid of the warning you must upgrade the old vulnerability models files to NRML v0.5. You can use the command `upgrade_nrml` with oq-lite to do this as follows:

```
user@ubuntu:~\$ oq-lite upgrade_nrml <directory-name>
```

The above command will upgrade all of your old vulnerability model files to NRML v0.5. The original files will be kept, but with a .bak extension appended. Notice that you will need

to set the `lossCategory` attribute to its correct value manually. This is easy to do, since if you try to run a computation you will get a clear error message telling the expected value for the `lossCategory` for each file.

Several methodologies to derive vulnerability functions are currently being evaluated by GEM and have been included as part of the Risk Modeller's Toolkit, the code for which can be found on a public repository at GitHub at: http://github.com/gemsciencetools/rmtk.

A web-based tool to build an vulnerability model in the NRML schema are also under development, and can be found at the OpenQuake platform at the following address: https://platform.openquake.org/ript/.

8. Using the Risk Module

This Chapter summarises the structure of the information necessary to define the different input data to be used with the OpenQuake-engine risk calculators. Input data for scenario-based and probabilistic seismic damage and risk analysis using the OpenQuake-engine are organised into:

- An exposure model file in the NRML format, as described in Section 7.1.
- A file describing the vulnerability model (Section 7.4) for loss calculations, or a file describing the fragility model (Section 7.2) for damage calculations. Optionally, a file describing the consequence model (Section 7.3) can also be provided in order to calculate losses from the estimated damage distributions.
- A general calculation configuration file.
- Hazard inputs. These include hazard curves for the classical probabilistic damage and risk calculators, ground motion fields for the scenario damage and risk calculators, or stochastic event sets for the probabilistic event based calculators. As of OpenQuake-engine v1.7, in general, there are five different ways in which hazard calculation parameters or results can be provided to the OpenQuake-engine in order to run the subsequent risk calculations:
  - Use a single configuration file for running the hazard and risk calculations sequentially
  - Use separate configuration files for running the hazard and risk calculations sequentially
  - Use a configuration file for the risk calculation along with all hazard outputs from a previously completed, compatible OpenQuake-engine hazard calculation
  - Use a configuration file for the risk calculation along with a specific hazard output from a previously completed, compatible OpenQuake-engine hazard calculation
  - Use a configuration file for the risk calculation along with hazard input files in

the OpenQuake NRML format

The file formats for exposure models, fragility models, consequence models, and vulnerability models have been described earlier in Chapter 7. The configuration file is the primary file that provides the OpenQuake-engine information regarding both the definition of the input models (e.g. exposure, site parameters, fragility, consequence, or vulnerability models) as well as the parameters governing the risk calculation.

Information regarding the configuration file for running hazard calculations using the OpenQuake-engine can be found in Section 3.4. Some initial mandatory parameters of the configuration file common to all of the risk calculators are presented in Listing 18. The remaining parameters that are specific to each risk calculator are discussed in subsequent sections.

```
1   [general]
2   description = Example risk calculation
3   calculation_mode = scenario_risk
4
5   [exposure]
6   exposure_file = exposure_model.xml
7
8   [vulnerability]
9   structural_vulnerability_file = structural_vulnerability_model.xml
```

Listing 18 – *Example minimal risk calculation configuration file (Download example)*

- `description`: a parameter that can be used to include some information about the type of calculations that are going to be performed.
- `calculation_mode`: this parameter specifies the type of calculation to be run. Valid options for the `calculation_mode` for the risk calculators are: `scenario_damage`, `scenario_risk`, `classical_damage`, `classical_risk`, `event_based_risk`, and `classical_bcr`.
- `exposure_file`: this parameter is used to specify the path to the exposure model file.

Depending on the type of risk calculation, other parameters besides the aforementioned ones may need to be provided. We illustrate in the following sections different examples of the configuration file for the different risk calculators.

8.1 Scenario Damage Calculator

For this calculator, the parameter `calculation_mode` should be set to `scenario_damage`.

Example 1

This example illustrates a scenario damage calculation which uses a single configuration file to first compute the ground motion fields for the given rupture model and then calculate

damage distribution statistics based on the ground motion fields. A minimal job configuration file required for running a scenario damage calculation is shown in Listing 19.

```ini
                                    ── job.ini ──
1   [general]
2   description = Scenario damage using a single config file
3   calculation_mode = scenario_damage
4
5   [exposure]
6   exposure_file = exposure_model.xml
7
8   [rupture]
9   rupture_model_file = rupture_model.xml
10  rupture_mesh_spacing = 2.0
11
12  [site_params]
13  site_model_file = site_model.xml
14
15  [hazard_calculation]
16  random_seed = 42
17  truncation_level = 3.0
18  maximum_distance = 200.0
19  gsim = BooreAtkinson2008
20  number_of_ground_motion_fields = 1000
21
22  [fragility]
23  structural_fragility_file = structural_fragility_model.xml
```

Listing 19 – *Example combined configuration file for running a scenario damage calculation (Download example)*

The general parameters `description` and `calculation_mode`, and `exposure_file` have already been described earlier. The other parameters seen in the above example configuration file are described below:

- `rupture_model_file`: a parameter used to define the path to the earthquake rupture model file describing the scenario event.
- `rupture_mesh_spacing`: a parameter used to specify the mesh size (in km) used by the OpenQuake-engine to discretize the rupture. Note that the smaller the mesh spacing, the greater will be (1) the precision in the calculation and (2) the computational demand.
- `structural_fragility_file`: a parameter used to define the path to the structural fragility model file.

In this case, the ground motion fields will be computed at each of the locations of the assets in the exposure model. Ground motion fields will be generated for each of the intensity measure types found in the provided set of fragility models. The above calculation can be

run using the command line:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

After the calculation is completed, a message similar to the following will be displayed:

```
Calculation 2680 completed in 13 seconds. Results:
  id | output_type | name
5069 | datastore    | dmg_by_asset_and_collapse_map
5070 | datastore    | dmg_by_taxon
5071 | datastore    | dmg_total
```

Note that one or more of the following parameters can be used in the same job configuration file to provide the corresponding fragility model files:

- `structural_fragility_file`: a parameter used to define the path to a structural fragility model file
- `nonstructural_fragility_file`: a parameter used to define the path to a nonstructural fragility model file
- `contents_fragility_file`: a parameter used to define the path to a contents fragility model file
- `business_interruption_fragility_file`: a parameter used to define the path to a business interruption fragility model file

It is important that the `lossCategory` parameter in the metadata section for each provided fragility model file ("structural", "nonstructural", "contents", or "business_interruption") should match the loss type defined in the configuration file by the relevant keyword above.

Example 2

This example illustrates a scenario damage calculation which uses separate configuration files for the hazard and risk parts of a scenario damage assessment. The first configuration file shown in Listing 20 contains input models and parameters required for the computation of the ground motion fields due to a given rupture. The second configuration file shown in Listing 21 contains input models and parameters required for the calculation of the damage distribution for a portfolio of assets due to the ground motion fields.

In this example, the set of intensity measure types for which the ground motion fields should be generated is specified explicitly in the configuration file using the parameter `intensity_measure_types`. If the hazard calculation outputs are intended to be used as inputs for a subsequent scenario damage or risk calculation, the set of intensity measure types specified here must include all intensity measure types that are used in the fragility or vulnerability models for the subsequent damage or risk calculation.

In the hazard configuration file illustrated above (Listing 20), the list of sites at which the ground motion values will be computed is provided in a CSV file, specified using the `sites_csv` parameter. The sites used for the hazard calculation need not be the same as the locations of the assets in the exposure model used for the following risk calculation. In such cases, it is recommended to set a reasonable search radius (in km) using

```ini
                              ── job_hazard.ini ──
1   [general]
2   description = Scenario hazard example
3   calculation_mode = scenario
4
5   [rupture]
6   rupture_model_file = rupture_model.xml
7   rupture_mesh_spacing = 2.0
8
9   [sites]
10  sites_csv = sites.csv
11
12  [site_params]
13  site_model_file = site_model.xml
14
15  [hazard_calculation]
16  random_seed = 42
17  truncation_level = 3.0
18  maximum_distance = 200.0
19  gsim = BooreAtkinson2008
20  intensity_measure_types = PGA, SA(0.3)
21  number_of_ground_motion_fields = 1000
22  ground_motion_correlation_model = JB2009
23  ground_motion_correlation_params = {"vs30_clustering": True}
```

Listing 20 – *Example hazard configuration file for a scenario damage calculation (Download example)*

```ini
                              ── job_damage.ini ──
1   [general]
2   description = Scenario damage example
3   calculation_mode = scenario_damage
4
5   [exposure]
6   exposure_file = exposure_model.xml
7
8   [boundaries]
9   region_constraint = -123.0 38.3, -121.0 38.3, -121.0 36.5, -123.0 36.5
10
11  [hazard]
12  asset_hazard_distance = 20
13
14  [fragility]
15  structural_fragility_file = structural_fragility_model.xml
```

Listing 21 – *Example risk configuration file for a scenario damage calculation (Download example)*

the `asset_hazard_distance` parameter for the OpenQuake-engine to look for available hazard values, as shown in the job_damage.ini example file above.

The only new parameters introduced in risk configuration file for this example (Listing 21) are the `region_constraint` and `asset_hazard_distance` parameters, which are described below; all other parameters have already been described in earlier examples.

- `region_constraint`: this is an optional parameter, applicable only to risk calculations, which defines the polygon that will be used for filtering the assets from the exposure model. Assets outside of this region will not be considered in the risk calculations. This region is defined using pairs of coordinates that indicate the vertices of the polygon, which should be listed in the Well-known text (WKT) format:

  region_constraint = lon_1 lat_1, lon_2 lat_2, ..., lon_n lat_n

  For each point, the longitude is listed first, followed by the latitude, both in decimal degrees. The list of points defining the polygon can be provided either in a clockwise or counter-clockwise direction.

  If the `region_constraint` is not provided, all assets in the exposure model are considered for the risk calculation.

  This parameter is useful in cases where the exposure model covers a region larger than the one that is of interest in the current calculation.

- `asset_hazard_distance`: this parameter indicates the maximum allowable distance between an asset and the closest hazard input. Hazard inputs can include hazard curves or ground motion intensity values. If no hazard input site is found within the radius defined by the `asset_hazard_distance`, the asset is skipped and a message is provided mentioning the id of the asset that is affected by this issue.

  If multiple hazard input sites are found within the radius defined by the this parameter, the hazard input site with the shortest distance from the asset location is associated with the asset. It is possible that the associated hazard input site might be located outside the polygon defined by the `region_constraint`.

Now, the above calculations described by the two configuration files "job_hazard.ini" and "job_damage.ini" can be run sequentially using the command line as follows:

```
user@ubuntu:~\$ oq-engine --run job_hazard.ini,job_damage.ini
```

The hazard and risk calculations can also be run separately. In that case, the calculation id for the hazard calculation or the output id for the specific ground motion fields output generated by the hazard calculation should be provided to the OpenQuake-engine while running the risk calculation using the options `--hazard-calculation-id` (or `--hc`) and `--hazard-output-id` (or `--ho`) respectively. This is shown below:

```
user@ubuntu:~\$ oq-engine --run job_hazard.ini
```

After the hazard calculation is completed, a message similar to the one below will be displayed in the terminal:

```
Calculation 2681 completed in 4 seconds. Results:
  id | output_type | name
5072 | datastore   | gmfs
```

In the example above, the calculation id of the hazard calculation is 2681. There is only one output from this calculation, i.e., the GMFs. The output id for the GMFs generated by the above calculation is 5072.

The risk calculation for computing the damage distribution statistics for the portfolio of assets can now be run using either:

```
user@ubuntu:~\$ oq-engine --run job_damage.ini --hc 2681
```

or

```
user@ubuntu:~\$ oq-engine --run job_damage.ini --ho 5072
```

After the calculation is completed, a message similar to the one listed above in Example 1 will be displayed.

In order to retrieve the calculation id of a previously run hazard calculation, the option `--list-hazard-calculations` (or `--lhc`) can be used to display a list of all previously run hazard calculations:

```
job_id |      status |         last_update |          description
  2609 | successful | 2015-12-01 14:14:14 | Mid Nepal earthquake
  ...
  2681 | successful | 2015-12-12 10:00:00 | Scenario hazard example
```

The option `--list-outputs` (or `--lo`) can be used to display a list of all outputs generated during a particular calculation. For instance,

```
user@ubuntu:~\$ oq-engine --lo 2681
```

will produce the following display:

```
  id | output_type | name
5072 | datastore   | gmfs
```

Example 3

The example shown in Listing 22 illustrates a scenario damage calculation which uses a file listing a precomputed set of GMFs. These GMFs can be computed using the OpenQuake-engine or some other software. The GMFs must be provided in the NRML schema as presented in Section 4.3.3. The damage distribution is computed based on the provided GMFs.

- `gmfs_file`: a parameter used to define the path to the GMFs file in the NRML schema. This file must define GMFs for all of the intensity measure types used in the fragility model.

The above calculation can be run using the command line:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

```ini
┌─ job.ini ─────────────────────────────────────────────────────────────
1   [general]
2   description = Scenario damage example using user-defined ground motion fields
3   calculation_mode = scenario_damage
4
5   [hazard]
6   gmfs_file = gmfs.xml
7
8   [exposure]
9   exposure_file = exposure_model.xml
10
11  [fragility]
12  structural_fragility_file = structural_fragility_model.xml
└───────────────────────────────────────────────────────────────────────
```

Listing 22 – *Example configuration file for a scenario damage calculation using a precomputed set of ground motion fields (Download example)*

Example 4

This example illustrates a the hazard job configuration file for a scenario damage calculation which uses two GMPEs instead of only one. Currently, the set of GMPEs to be used for a scenario calculation can be specified using a logic tree file, as demonstrated in 3.3.1. As of OpenQuake-engine v1.8, the weights in the logic tree are ignored, and a set of GMFs will be generated for each GMPE in the logic tree file. Correspondingly, damage distribution statistics will be generated for each set of GMF.

The file shown in Listing 23 lists the two GMPEs to be used for the hazard calculation:

The only change that needs to be made in the hazard job configuration file is to replace the gsim parameter with `gsim_logic_tree_file`, as demonstrated in Listing 24.

Example 5

This example illustrates a scenario damage calculation which specifies fragility models for calculating damage to structural and nonstructural components of structures, and also specifies consequence model files for calculation of the corresponding losses.

A minimal job configuration file required for running a scenario damage calculation followed by a consequences analysis is shown in Listing 25.

Note that one or more of the following parameters can be used in the same job configuration file to provide the corresponding consequence model files:

- `structural_consequence_file`: a parameter used to define the path to a structural consequence model file
- `nonstructural_consequence_file`: a parameter used to define the path to a nonstructural consequence model file
- `contents_consequence_file`: a parameter used to define the path to a contents consequence model file
- `business_interruption_consequence_file`: a parameter used to define the

```
─────────────── gsim_logic_tree.xml ───────────────
1   <?xml version='1.0' encoding='utf-8'?>
2   <nrml xmlns:gml="http://www.opengis.net/gml"
3         xmlns="http://openquake.org/xmlns/nrml/0.5">
4
5   <logicTree logicTreeID="lt1">
6      <logicTreeBranchingLevel branchingLevelID="bl1">
7        <logicTreeBranchSet uncertaintyType="gmpeModel"
8                            branchSetID="bs1"
9                            applyToTectonicRegionType="Active Shallow Crust">
10
11         <logicTreeBranch branchID="b1">
12           <uncertaintyModel>BooreAtkinson2008</uncertaintyModel>
13           <uncertaintyWeight>0.75</uncertaintyWeight>
14         </logicTreeBranch>
15
16         <logicTreeBranch branchID="b2">
17           <uncertaintyModel>ChiouYoungs2008</uncertaintyModel>
18           <uncertaintyWeight>0.25</uncertaintyWeight>
19         </logicTreeBranch>
20
21       </logicTreeBranchSet>
22      </logicTreeBranchingLevel>
23   </logicTree>
24
25   </nrml>
```

Listing 23 – *Example ground motion logic tree for a scenario calculation (Download example)*

```
                                ── job_hazard.ini ──
 1   [general]
 2   description = Scenario hazard example using multiple GMPEs
 3   calculation_mode = scenario
 4
 5   [rupture]
 6   rupture_model_file = rupture_model.xml
 7   rupture_mesh_spacing = 2.0
 8
 9   [sites]
10   sites_csv = sites.csv
11
12   [site_params]
13   site_model_file = site_model.xml
14
15   [hazard_calculation]
16   random_seed = 42
17   truncation_level = 3.0
18   maximum_distance = 200.0
19   gsim_logic_tree_file = gsim_logic_tree.xml
20   intensity_measure_types = PGA, SA(0.3)
21   number_of_ground_motion_fields = 1000
22   ground_motion_correlation_model = JB2009
23   ground_motion_correlation_params = {"vs30_clustering": True}
```

Listing 24 – *Example configuration file for a scenario damage calculation using a logic-tree file*
(*Download example*)

```ini
                                    ─── job.ini ───
1    [general]
2    description = Scenario damage and consequences
3    calculation_mode = scenario_damage
4
5    [exposure]
6    exposure_file = exposure_model.xml
7
8    [rupture]
9    rupture_model_file = rupture_model.xml
10   rupture_mesh_spacing = 2.0
11
12   [site_params]
13   site_model_file = site_model.xml
14
15   [hazard_calculation]
16   random_seed = 42
17   truncation_level = 3.0
18   maximum_distance = 200.0
19   gsim = BooreAtkinson2008
20   number_of_ground_motion_fields = 1000
21   ground_motion_correlation_model = JB2009
22   ground_motion_correlation_params = {"vs30_clustering": True}
23
24   [fragility]
25   structural_fragility_file = structural_fragility_model.xml
26   nonstructural_fragility_file = nonstructural_fragility_model.xml
27
28   [consequence]
29   structural_consequence_file = structural_consequence_model.xml
30   nonstructural_consequence_file = nonstructural_consequence_model.xml
```

Listing 25 – *Example configuration file for a scenario damage calculation followed by a consequences analysis (Download example)*

path to a business interruption consequence model file

It is important that the `lossCategory` parameter in the metadata section for each provided consequence model file ("structural", "nonstructural", "contents", or "business_interruption") should match the loss type defined in the configuration file by the relevant keyword above.

The above calculation can be run using the command line:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

After the calculation is completed, a message similar to the following will be displayed:

```
Calculation 1579 completed in 37 seconds. Results:
  id | output_type | name
8990 | datastore | csq_by_asset
8991 | datastore | csq_by_taxon
8992 | datastore | csq_total
8993 | datastore | dmg_by_asset_and_collapse_map
8994 | datastore | dmg_by_taxon
8995 | datastore | dmg_total
```

8.2 Scenario Risk Calculator

In order to run this calculator, the parameter `calculation_mode` needs to be set to `scenario_risk`.

Most of the job configuration parameters required for running a scenario risk calculation are the same as those described in the previous section for the scenario damage calculator. The remaining parameters specific to the scenario risk calculator are illustrated through the examples below.

Example 1

This example illustrates a scenario risk calculation which uses a single configuration file to first compute the ground motion fields for the given rupture model and then calculate loss statistics for structural losses, nonstructural losses, and insured structural losses, based on the ground motion fields. The job configuration file required for running this scenario risk calculation is shown in Listing 26.

Whereas a scenario damage calculation requires one or more fragility and/or consequence models, a scenario risk calculation requires the user to specify one or more vulnerability model files. Note that one or more of the following parameters can be used in the same job configuration file to provide the corresponding vulnerability model files:

- `structural_vulnerability_file`: this parameter is used to specify the path to the structural vulnerability model file
- `nonstructural_vulnerability_file`: this parameter is used to specify the path to the nonstructuralvulnerability model file
- `contents_vulnerability_file`: this parameter is used to specify the path to the contents vulnerability model file

```
┌─────────────────────────────── job.ini ───────────────────────────────┐
1    [general]
2    description = Scenario risk using a single config file
3    calculation_mode = scenario_risk
4
5    [exposure]
6    exposure_file = exposure_model.xml
7
8    [rupture]
9    rupture_model_file = rupture_model.xml
10   rupture_mesh_spacing = 2.0
11
12   [site_params]
13   site_model_file = site_model.xml
14
15   [hazard_calculation]
16   random_seed = 42
17   truncation_level = 3.0
18   maximum_distance = 200.0
19   gsim = BooreAtkinson2008
20   number_of_ground_motion_fields = 1000
21   ground_motion_correlation_model = JB2009
22   ground_motion_correlation_params = {"vs30_clustering": True}
23
24   [vulnerability]
25   structural_vulnerability_file = structural_vulnerability_model.xml
26   nonstructural_vulnerability_file = nonstructural_vulnerability_model.xml
27
28   [risk_calculation]
29   master_seed = 24
30   asset_correlation = 0.7
31
32   [risk_outputs]
33   insured_losses = true
```

Listing 26 – *Example combined configuration file for a scenario risk calculation (Download example)*

- `business_interruption_vulnerability_file`: this parameter is used to specify the path to the business interruption vulnerability model file
- `occupants_vulnerability_file`: this parameter is used to specify the path to the occupants vulnerability model file

It is important that the `lossCategory` parameter in the metadata section for each provided vulnerability model file ("structural", "nonstructural", "contents", "business_interruption", or "occupants") should match the loss type defined in the configuration file by the relevant keyword above.

The remaining new parameters introduced in this example are the following:

- `master_seed`: this parameter is used to control the random number generator in the loss ratio sampling process. If the same `master_seed` is defined at each calculation run, the same random loss ratios will be generated, thus allowing reproducibility of the results.
- `asset_correlation`: if the uncertainty in the loss ratios has been defined within the vulnerability model, users can specify a coefficient of correlation that will be used in the Monte Carlo sampling process of the loss ratios, between the assets that share the same taxonomy. If the `asset_correlation` is set to one, the loss ratio residuals will be perfectly correlated. On the other hand, if this parameter is set to zero, the loss ratios will be sampled independently. Any value between zero and one will lead to increasing levels of correlation. If this parameter is not defined, the OpenQuake-engine will assume zero correlation in the vulnerability. As of OpenQuake-engine v1.8, `asset_correlation` applies only to continuous vulnerability functions using the lognormal or Beta distribution; it does not apply to vulnerability functions defined using the PMF distribution.
- `insured_losses`: this parameter specifies whether insured losses should be calculated; the default value of this parameter is `false`. In order for the OpenQuake-engine to be able to compute insured losses, the insurance limits and deductibles must be listed for each asset in the exposure model, as described in Example 5 in Section 7.1.

In this case, the ground motion fields will be computed at each of the locations of the assets in the exposure model and for each of the intensity measure types found in the provided set of vulnerability models. The above calculation can be run using the command line:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

After the calculation is completed, a message similar to the following will be displayed:

```
Calculation 2735 completed in 10 seconds. Results:
  id | output_type | name
5328 | datastore | agglosses-rlzs
5329 | datastore | loss_map-rlzs
```

All of the different ways of running a scenario damage calculation as illustrated through the examples of the previous section are also applicable to the scenario risk calculator, though

the examples are not repeated here.

8.3 Classical Probabilistic Seismic Damage Calculator

In order to run this calculator, the parameter `calculation_mode` needs to be set to `classical_damage`.

Most of the job configuration parameters required for running a classical probabilistic damage calculation are the same as those described in the section for the scenario damage calculator. The remaining parameters specific to the classical probabilistic damage calculator are illustrated through the examples below.

Example 1

This example illustrates a classical probabilistic damage calculation which uses a single configuration file to first compute the hazard curves for the given source model and ground motion model and then calculate damage distribution statistics based on the hazard curves. A minimal job configuration file required for running a classical probabilistic damage calculation is shown in Listing 27.

The general parameters `description` and `calculation_mode`, and `exposure_file` have already been described earlier in Section 8.1. The parameters related to the hazard curves computation have been described earlier in Section 3.4.1.

In this case, the hazard curves will be computed at each of the locations of the assets in the exposure model, for each of the intensity measure types found in the provided set of fragility models. The above calculation can be run using the command line:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

After the calculation is completed, a message similar to the following will be displayed:

```
Calculation 2741 completed in 12 seconds. Results:
  id |     output_type | name
5359 | Damage Per Asset | Damage distribution for hazard=5357
```

Example 2

This example illustrates a classical probabilistic damage calculation which uses separate configuration files for the hazard and risk parts of a classical probabilistic damage assessment. The first configuration file shown in Listing 28 contains input models and parameters required for the computation of the hazard curves. The second configuration file shown in Listing 29 contains input models and parameters required for the calculation of the probabilistic damage distribution for a portfolio of assets based on the hazard curves and fragility models.

Now, the above calculations described by the two configuration files "job_hazard.ini" and "job_damage.ini" can be run sequentially or separately, as illustrated in Example 2 in Section 8.1. The new parameters introduced in the above example configuration file are described below:

```
                                 ─── job.ini ───
1    [general]
2    description = Classical probabilistic damage using a single config file
3    calculation_mode = classical_damage
4
5    [exposure]
6    exposure_file = exposure_model.xml
7
8    [erf]
9    width_of_mfd_bin = 0.1
10   rupture_mesh_spacing = 2
11   area_source_discretization = 20
12
13   [site_params]
14   site_model_file = site_model.xml
15
16   [logic_trees]
17   source_model_logic_tree_file = source_model_logic_tree.xml
18   gsim_logic_tree_file = gsim_logic_tree.xml
19   number_of_logic_tree_samples = 0
20
21   [hazard_calculation]
22   random_seed = 42
23   investigation_time = 1
24   truncation_level = 3.0
25   maximum_distance = 200.0
26
27   [fragility]
28   structural_fragility_file = structural_fragility_model.xml
```

Listing 27 – *Example combined configuration file for a classical probabilistic damage calculation*
(Download example)

```ini
                          ─── job_hazard.ini ───
1   [general]
2   description = Classical probabilistic hazard
3   calculation_mode = classical
4
5   [sites]
6   region = -123.0 38.3, -121.0 38.3, -121.0 36.5, -123.0 36.5
7   region_grid_spacing = 0.5
8
9   [erf]
10  width_of_mfd_bin = 0.1
11  rupture_mesh_spacing = 2
12  area_source_discretization = 20
13
14  [site_params]
15  site_model_file = site_model.xml
16
17  [logic_trees]
18  source_model_logic_tree_file = source_model_logic_tree.xml
19  gsim_logic_tree_file = gsim_logic_tree.xml
20  number_of_logic_tree_samples = 0
21
22  [hazard_calculation]
23  random_seed = 42
24  investigation_time = 1
25  truncation_level = 3.0
26  maximum_distance = 200.0
```

Listing 28 – *Example hazard configuration file for a classical probabilistic damage calculation*
(Download example)

```
job_damage.ini
1   [general]
2   description = Classical probabilistic damage example
3   calculation_mode = classical_damage
4
5   [exposure]
6   exposure_file = exposure_model.xml
7
8   [hazard]
9   asset_hazard_distance = 20
10
11  [fragility]
12  structural_fragility_file = structural_fragility_model.xml
13
14  [risk_calculation]
15  risk_investigation_time = 50
16  steps_per_interval = 4
```

Listing 29 – *Example risk configuration file for a classical probabilistic damage calculation (Download example)*

- `risk_investigation_time`: an optional parameter that can be used in probabilistic damage or risk calculations where the period of interest for the risk calculation is different from the period of interest for the hazard calculation. If this parameter is not explicitly set, the OpenQuake-engine will assume that the risk calculation is over the same time period as the preceding hazard calculation.
- `steps_per_interval`: an optional parameter that can be used to specify whether discrete fragility functions in the fragility models should be discretized further, and if so, how many intermediate steps to use for the discretization. Setting

  steps_per_interval = n

  will result in the OpenQuake-engine discretizing the discrete fragility models using (n - 1) linear interpolation steps between each pair of intensity level, poe points.

  The default value of this parameter is one, implying no interpolation.

8.4   Classical Probabilistic Seismic Risk Calculator

In order to run this calculator, the parameter `calculation_mode` needs to be set to `classical_risk`.

Most of the job configuration parameters required for running a classical probabilistic risk calculation are the same as those described in the previous section for the classical probabilistic damage calculator. The remaining parameters specific to the classical probabilistic risk calculator are illustrated through the examples below.

Example 1

This example illustrates a classical probabilistic risk calculation which uses a single configuration file to first compute the hazard curves for the given source model and ground motion model and then calculate loss exceedance curves based on the hazard curves. An example job configuration file for running a classical probabilistic risk calculation is shown in Listing 30.

```
─────── job.ini ───────
1  [general]
2  description = Classical probabilistic risk using a single config file
3  calculation_mode = classical_risk
4
5  [exposure]
6  exposure_file = exposure_model.xml
7
8  [erf]
9  width_of_mfd_bin = 0.1
10 rupture_mesh_spacing = 2
11 area_source_discretization = 20
12
13 [site_params]
14 site_model_file = site_model.xml
15
16 [logic_trees]
17 source_model_logic_tree_file = source_model_logic_tree.xml
18 gsim_logic_tree_file = gsim_logic_tree.xml
19 number_of_logic_tree_samples = 0
20
21 [hazard_calculation]
22 random_seed = 42
23 investigation_time = 1
24 truncation_level = 3.0
25 maximum_distance = 200.0
26
27 [vulnerability]
28 structural_vulnerability_file = structural_vulnerability_model.xml
29 nonstructural_vulnerability_file = nonstructural_vulnerability_model.xml
30
31 [risk_outputs]
32 insured_losses = true
```

Listing 30 – *Example combined configuration file for a classical probabilistic risk calculation (Download example)*

Apart from the calculation mode, the only difference with the example job configuration file shown in Example 1 of Section 8.3 is the use of a vulnerability model instead of a fragility model.

As with the Scenario Risk calculator, it is possible to specify one or more vulnerability

model files in the same job configuration file, using the parameters:

- `structural_vulnerability_file`,
- `nonstructural_vulnerability_file`,
- `contents_vulnerability_file`,
- `business_interruption_vulnerability_file`, and/or
- `occupants_vulnerability_file`

It is important that the `lossCategory` parameter in the metadata section for each provided vulnerability model file ("structural", "nonstructural", "contents", "business_interruption", or "occupants") should match the loss type defined in the configuration file by the relevant keyword above.

In this case, the hazard curves will be computed at each of the locations of the assets in the exposure model, for each of the intensity measure types found in the provided set of vulnerability models. The above calculation can be run using the command line:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

After the calculation is completed, a message similar to the following will be displayed:

```
Calculation 2749 completed in 24 seconds. Results:
  id | output_type | name
5373 | Loss Curve  | loss curves. type=structural, hazard=5371
5374 | Loss Curve  | loss curves. type=nonstructural, hazard=5371
5375 | Loss Curve  | insured loss curves. type=structural hazard=5371
```

Example 2

This example illustrates a classical probabilistic risk calculation which uses separate configuration files for the hazard and risk parts of a classical probabilistic risk assessment. The first configuration file shown in Listing 31 contains input models and parameters required for the computation of the hazard curves. The second configuration file shown in Listing 32 contains input models and parameters required for the calculation of the loss exceedance curves and probabilistic loss maps for a portfolio of assets based on the hazard curves and vulnerability models.

Now, the above calculations described by the two configuration files "job_hazard.ini" and "job_risk.ini" can be run sequentially or separately, as illustrated in Example 2 in Section 8.1. The new parameters introduced in the above risk configuration file example (Listing 32) are described below:

- `lrem_steps_per_interval`: this parameter controls the number of intermediate values between consecutive loss ratios (as defined in the vulnerability model) that are considered in the risk calculations. A larger number of loss ratios than those defined in each vulnerability function should be considered, in order to better account for the uncertainty in the loss ratio distribution. If this parameter is not defined in the configuration file, the OpenQuake-engine assumes the `lrem_steps_per_interval`

```
──────────── job_hazard.ini ────────────
1    [general]
2    description = Classical probabilistic hazard
3    calculation_mode = classical
4
5    [sites]
6    region = -123.0 38.3, -121.0 38.3, -121.0 36.5, -123.0 36.5
7    region_grid_spacing = 0.5
8
9    [erf]
10   width_of_mfd_bin = 0.1
11   rupture_mesh_spacing = 2
12   area_source_discretization = 20
13
14   [site_params]
15   site_model_file = site_model.xml
16
17   [logic_trees]
18   source_model_logic_tree_file = source_model_logic_tree.xml
19   gsim_logic_tree_file = gsim_logic_tree.xml
20   number_of_logic_tree_samples = 0
21
22   [hazard_calculation]
23   random_seed = 42
24   investigation_time = 1
25   truncation_level = 3.0
26   maximum_distance = 200.0
```

Listing 31 – *Example hazard configuration file for a classical probabilistic risk calculation (Download example)*

```ini
                                    ──── job_risk.ini ────
1   [general]
2   description = Classical probabilistic risk
3   calculation_mode = classical_risk
4
5   [exposure]
6   exposure_file = exposure_model.xml
7
8   [hazard]
9   asset_hazard_distance = 20
10
11  [vulnerability]
12  structural_vulnerability_file = structural_vulnerability_model.xml
13  nonstructural_vulnerability_file = nonstructural_vulnerability_model.xml
14
15  [risk_calculation]
16  risk_investigation_time = 50
17  lrem_steps_per_interval = 2
18
19  [risk_outputs]
20  quantile_loss_curves = 0.15, 0.50, 0.85
21  conditional_loss_poes = 0.02, 0.10
```

Listing 32 – *Example risk configuration file for a classical probabilistic risk calculation (Download example)*

to be equal to 5. More details are provided in the OpenQuake Book (Risk).

- `quantile_loss_curves`: this parameter can be used to request the computation of quantile loss curves for computations involving non-trivial logic trees. The quantiles for which the loss curves should be computed must be provided as a comma separated list. If this parameter is not included in the configuration file, quantile loss curves will not be computed.

- `conditional_loss_poes`: this parameter can be used to request the computation of probabilistic loss maps, which give the loss levels exceeded at the specified probabilities of exceedance over the time period specified by `risk_investigation_time`. The probabilities of exceedance for which the loss maps should be computed must be provided as a comma separated list. If this parameter is not included in the configuration file, probabilistic loss maps will not be computed.

8.5 Stochastic Event Based Probabilistic Seismic Risk Calculator

The parameter `calculation_mode` needs to be set to `event_based_risk` in order to use this calculator.

Most of the job configuration parameters required for running a stochastic event based risk calculation are the same as those described in the previous sections for the scenario risk calculator and the classical probabilistic risk calculator. The remaining parameters specific to the stochastic event based risk calculator are illustrated through the example below.

Example 1

This example illustrates a stochastic event based risk calculation which uses a single configuration file to first compute the SESs and GMFs for the given source model and ground motion model, and then calculate event loss tables, loss exceedance curves and probabilistic loss maps for structural losses, nonstructural losses, insured structural losses, and occupants, based on the GMFs. The job configuration file required for running this stochastic event based risk calculation is shown in Listing 33.

Similar to that the procedure described for the Scenario Risk calculator, a Monte Carlo sampling process is also employed in this calculator to take into account the uncertainty in the conditional loss ratio at a particular intensity level. Hence, the parameters `asset_correlation` and `master_seed` may be defined as previously described for the Scenario Risk calculator in Section 8.2. This calculator is also capable of estimating insured losses and therefore, setting the `insured_losses` attribute to `true` will generate all results (loss tables, loss curves, loss maps) for insured losses as well. The parameter "risk_investigation_time" specifies the time period for which the event loss tables and loss exceedance curves will be calculated, similar to the Classical Probabilistic Risk calculator. If this parameter is not provided in the risk job configuration file, the time period used is the same as that specifed in the hazard calculation using the parameter "investigation_time".

The new parameters introduced in this example are described below:

```
┌─────────────────────────────── job.ini ───────────────────────────────┐
 1  [general]
 2  description = Stochastic event based risk using a single job file
 3  calculation_mode = event_based_risk
 4
 5  [exposure]
 6  exposure_file = exposure_model.xml
 7
 8  [site_params]
 9  site_model_file = site_model.xml
10
11  [erf]
12  width_of_mfd_bin = 0.1
13  rupture_mesh_spacing = 2.0
14  area_source_discretization = 10
15
16  [logic_trees]
17  source_model_logic_tree_file = source_model_logic_tree.xml
18  gsim_logic_tree_file = gsim_logic_tree.xml
19
20  [correlation]
21  ground_motion_correlation_model = JB2009
22  ground_motion_correlation_params = {"vs30_clustering": True}
23
24  [hazard_calculation]
25  random_seed = 24
26  truncation_level = 3
27  maximum_distance = 200.0
28  investigation_time = 1
29  number_of_logic_tree_samples = 0
30  ses_per_logic_tree_path = 100000
31
32  [vulnerability]
33  structural_vulnerability_file = structural_vulnerability_model.xml
34  nonstructural_vulnerability_file = nonstructural_vulnerability_model.xml
35
36  [risk_calculation]
37  master_seed = 42
38  risk_investigation_time = 50
39  asset_correlation = 0.0
40  loss_curve_resolution = 10
41  loss_ratios = {"structural": [0.00010, 0.00016, 0.00025, 0.00040, 0.00065,
42    0.0010, 0.0016, 0.0025, 0.0040, 0.0065, 0.010, 0.016, 0.025, 0.040, 0.065,
43    0.10, 0.16, 0.25, 0.40, 0.65, 1.0], "nonstructural": [0.00010, 0.00016,
44    0.00025, 0.00040, 0.00065, 0.0010, 0.0016, 0.0025, 0.0040, 0.0065, 0.010,
45    0.016, 0.025, 0.040, 0.065, 0.10, 0.16, 0.25, 0.40, 0.65, 1.0]}
46
47  [risk_outputs]
48  avg_losses = true
49  insured_losses = true
50  asset_loss_table = true
51  quantile_loss_curves = 0.15, 0.50, 0.85
52  conditional_loss_poes = 0.02, 0.10
└────────────────────────────────────────────────────────────────────────┘
```

Listing 33 – *Example combined configuration file for running a stochastic event based risk calculation*

- `loss_curve_resolution`: this parameter specifies the number of points on the aggregate loss curve. The loss levels on the aggregate loss curve are obtained by dividing the interval between the minimum and maximum portfolio losses in the portfolio loss table into 'n' equispaced intervals, where 'n' is the value specified for the `loss_curve_resolution`. If this parameter is not set, the OpenQuake-engine uses a default value of 20 for the `loss_curve_resolution`.

- `loss_ratios`: this parameter specifies the set of loss ratios at which the individual asset loss curves will be computed. If `loss_ratios` is not set in the configuration file, the individual asset loss curves will not be computed; and only the aggregate loss curve for the portfolio of assets will be computed. Furthermore, if `loss_ratios` is not set in the configuration file, loss maps will also **not** be computed.

- `avg_losses`: this boolean parameter specifies whether the average asset losses over the time period "risk_investigation_time" should be computed. The default value of this parameter is `false`.

$$
\begin{aligned}
average\_loss = {} & sum(event\_losses) \\
& \div (hazard\_investigation\_time \times ses\_per\_logic\_tree\_path) \\
& \times risk\_investigation\_time
\end{aligned}
$$

- `asset_loss_table`: this boolean parameter specifies whether the individual asset event loss tables should be saved to the datastore. The default value of this parameter is `false`.

The above calculation can be run using the command line:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

Computation of the loss tables, loss curves, and average losses for each individual asset in the exposure model can be resource intensive, and thus these outputs are not generated by default, unless instructed to by using the parameters described above.

8.6 Retrofit Benefit-Cost Ratio Calculator

As previously explained, this calculator uses loss exceedance curves which are calculated using the Classical Probabilistic risk calculator. In order to run this calculator, the parameter `calculation_mode` needs to be set to `classical_bcr`.

Most of the job configuration parameters required for running a classical retrofit benefit-cost ratio calculation are the same as those described in the previous section for the classical probabilistic risk calculator. The remaining parameters specific to the classical retrofit benefit-cost ratio calculator are illustrated through the examples below.

Example 1

This example illustrates a classical probabilistic retrofit benefit-cost ratio calculation which uses a single configuration file to first compute the hazard curves for the given source model and ground motion model, then calculate loss exceedance curves based on the hazard curves using both the original vulnerability model and the vulnerability model for the retrofitted structures, then calculate the reduction in average annual losses due to the retrofits, and finally calculate the benefit-cost ratio for each asset. A minimal job configuration file required for running a classical probabilistic retrofit benefit-cost ratio calculation is shown in Listing 34.

```ini
                                  ── job.ini ──
1   [general]
2   description = Classical cost-benefit analysis using a single config file
3   calculation_mode = classical_bcr
4
5   [exposure]
6   exposure_file = exposure_model.xml
7
8   [erf]
9   width_of_mfd_bin = 0.1
10  rupture_mesh_spacing = 2
11  area_source_discretization = 20
12
13  [site_params]
14  site_model_file = site_model.xml
15
16  [logic_trees]
17  source_model_logic_tree_file = source_model_logic_tree.xml
18  gsim_logic_tree_file = gsim_logic_tree.xml
19  number_of_logic_tree_samples = 0
20
21  [hazard_calculation]
22  random_seed = 42
23  investigation_time = 1
24  truncation_level = 3.0
25  maximum_distance = 200.0
26
27  [vulnerability]
28  structural_vulnerability_file = structural_vulnerability_model.xml
29  structural_vulnerability_retrofitted_file = retrofit_vulnerability_model.xml
30
31  [risk_calculation]
32  interest_rate = 0.05
33  asset_life_expectancy = 50
34  lrem_steps_per_interval = 1
```

Listing 34 – *Example configuration file for a classical probabilistic retrofit benefit-cost ratio calculation (Download example)*

The new parameters introduced in the above example configuration file are described below:

- `vulnerability_retrofitted_file`: this parameter is used to specify the path to the vulnerability model file containing the vulnerability functions for the retrofitted asset
- `interest_rate`: this parameter is used in the calculation of the present value of potential future benefits by discounting future cash flows
- `asset_life_expectancy`: this variable defines the life expectancy or design life of the assets, and is used as the time-frame in which the costs and benefits of the retrofit will be compared

The above calculation can be run using the command line:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

After the calculation is completed, a message similar to the following will be displayed:

```
Calculation 2776 completed in 25 seconds. Results:
  id | output_type | name
5422 | Benefit-cost ratio distribution | BCR Map. type=structural, hazard=5420
```

8.7 Exporting Risk Results

To obtain a list of all risk calculations that have been previously run (successfully or unsuccessfully), or are currently running, the following command can be employed:

```
user@ubuntu:~\$ oq-engine --list-risk-calculations
```

or simply:

```
user@ubuntu:~\$ oq-engine --lrc
```

Which will display a list of risk calculations as presented below.

```
job_id |      status |          last_update |     description
     1 | successful | 2015-12-02 08:50:30  | Scenario damage example
     2 |     failed | 2015-12-03 09:56:17  | Scenario risk example
     3 | successful | 2015-12-04 10:45:32  | Scenario risk example
     4 | successful | 2015-12-04 10:48:33  | Classical risk example
```

Then, in order to display a list of the risk outputs from a given job, the following command can be used:

```
user@ubuntu:~\$ oq-engine --list-outputs <risk_calculation_id>
```

or simply:

```
user@ubuntu:~\$ oq-engine --lo <risk_calculation_id>
```

which will display a list of outputs for the calculation requested, as presented below:

```
Calculation 4 results:
id | output_type | name
29 | loss_curve  | loss curves. type=structural, hazard=32
30 | loss_map    | loss maps. type=structural poe=0.1, hazard=32
```

Then, in order to export all of the risk calculation outputs in the appropriate xml format, the following command can be used:

```
user@ubuntu:~\$ oq-engine --export-outputs <risk_calculation_id> <output_directory>
```

or simply:

```
user@ubuntu:~\$ oq-engine --eos <risk_calculation_id> <output_directory>
```

If, instead of exporting all of the outputs from a particular calculation, only particular output files need to be exported, this can be achieved by using the `--export-output` option and providing the id of the required output:

```
user@ubuntu:~\$ oq-engine --export-output <risk_output_id> <output_directory>
```

or simply:

```
user@ubuntu:~\$ oq-engine --eo <risk_output_id> <output_directory>
```

9. Risk Results

This following sections describe the different output files produced by the risk calculators.

9.1 Scenario Damage Outputs

The Scenario Damage Calculator produces the following set of output files for all loss types (amongst "structural", "nonstructural", "contents", or "business_interruption") for which a fragility model file was provided in the configuration file:

1. `dmg_dist_per_asset`: this file contains the damage distribution statistics for each of the individual assets defined in the exposure model that fall within the `region_constraint` and have a computed GMF value available within the defined `asset_hazard_distance`. For each asset, the mean number of buildings (`mean`) and associated standard deviation (`stddev`) of the number of buildings in each damage state are listed in this file.

2. `dmg_dist_per_taxonomy`: this file contains the aggregated damage distribution statistics for each of the taxonomies defined in the exposure model. For each taxonomy, the mean number of buildings (`mean`) and associated standard deviation (`stddev`) of the number of buildings in each damage state are listed in this file.

3. `dmg_dist_total`: this file contains the aggregated damage distribution statistics for the entire portfolio of assets defined in the exposure model. The mean (`mean`) and associated standard deviation (`stddev`) of the total number of buildings in each damage state are listed in this file.

4. `collapse_map`: this file contains mean (`mean`) and associated standard deviation (`stddev`) of the number of buildings in the ultimate limit state for all assets at each of the unique locations in the exposure model.

In addition to the above output files which are produced for all Scenario Damage calculations, the following set of output files for all loss types (amongst "structural", "nonstructural", "contents", or "business_interruption") for which a consequence model file was also provided

in the configuration file:

5. `csq_by_asset`: this file contains the scenario consequence statistics for each of the individual assets defined in the exposure model that fall within the `region_constraint` and have a computed GMF value available within the defined `asset_hazard_distance`. For each asset, the mean consequences (`mean`) and associated standard deviation (`stddev`) are listed in this file.

6. `csq_by_taxon`: this file contains the aggregated scenario consequence statistics for each of the taxonomies defined in the exposure model. For each taxonomy, the mean consequences (`mean`) and associated standard deviation (`stddev`) are listed in this file.

7. `csq_total`: this file contains the aggregated scenario consequence statistics for the entire portfolio of assets defined in the exposure model. The mean consequences (`mean`) and associated standard deviation (`stddev`) are listed in this file.

If the calculation involves multiple GMPEs as described in Example 4 in Section 8.1, separate output files are generated for each of the above outputs, for each of the different GMPEs used in the calculation.

These different output files for Scenario Damage calculations are described in more detail in the following subsections.

9.1.1 Scenario damage statistics

9.1.1.1 Asset damage statistics

This output contains the damage distribution statistics for each of the individual assets defined in the exposure model that fall within the `region_constraint` and have a computed GMF value available within the defined `asset_hazard_distance`. An example output file for structural damage is shown in the file snippet in Listing 35.

The key fields in the above output file are the following:

- `damageStates`: this field serves the purposes of storing the set of damage states, as defined in the fragility model employed in the calculations
- `DDNode`: this attribute is used to store the damage distribution of a number of assets, at a given location (defined within the attribute `gml:Point`). For each asset, the mean number of buildings (`mean`) and associated standard deviation (`stddev`) of the number of buildings in each damage state are listed.

9.1.1.2 Taxonomy damage statistics

The Scenario Damage calculator also estimates the expected total number of buildings of a certain taxonomy in each damage state. This distribution of damage per building taxonomy is depicted in the example output file snippet in Listing 36.

In the damage distribution per taxonomy, each `DDNode` contains the statistics of the number of buildings in each damage state, belonging to a given building class as specified by the `taxonomy` attribute.

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <dmgDistPerAsset>
        <damageStates>
            no_damage slight moderate extensive complete
        </damageStates>
        <DDNode>
            <gml:Point>
                <gml:pos>
                    -122.57 38.113
                </gml:pos>
            </gml:Point>
            <asset assetRef="a3">
                <damage ds="no_damage" mean="7.1795E-01" stddev="1.2221E-01"/>
                <damage ds="slight" mean="2.5959E-01" stddev="9.5708E-02"/>
                <damage ds="moderate" mean="2.2447E-02" stddev="3.1015E-02"/>
                <damage ds="extensive" mean="3.9972E-06" stddev="1.8029E-05"/>
                <damage ds="complete" mean="1.7999E-26" stddev="1.1334E-25"/>
            </asset>
        </DDNode>


        ...


        <DDNode>
            <gml:Point>
                <gml:pos>
                    -121.886 38.113
                </gml:pos>
            </gml:Point>
            <asset assetRef="a7">
                <damage ds="no_damage" mean="2.9778E-01" stddev="1.3390E-01"/>
                <damage ds="slight" mean="3.6392E-01" stddev="7.9219E-02"/>
                <damage ds="moderate" mean="3.0590E-01" stddev="1.5269E-01"/>
                <damage ds="extensive" mean="3.2343E-02" stddev="7.8265E-02"/>
                <damage ds="complete" mean="3.4134E-05" stddev="3.2795E-04"/>
            </asset>
        </DDNode>
    </dmgDistPerAsset>
</nrml>
```

Listing 35 – *Example scenario damage statistics per asset*

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <dmgDistPerTaxonomy>
        <damageStates>
            no_damage slight moderate extensive complete
        </damageStates>
        <DDNode>
            <taxonomy>
                wood
            </taxonomy>
            <damage ds="no_damage" mean="1.4548E+00" stddev="3.4801E-01"/>
            <damage ds="slight" mean="1.2070E+00" stddev="2.0861E-01"/>
            <damage ds="moderate" mean="1.0288E+00" stddev="2.5993E-01"/>
            <damage ds="extensive" mean="2.9099E-01" stddev="3.0502E-01"/>
            <damage ds="complete" mean="1.8247E-02" stddev="7.8220E-02"/>
        </DDNode>
        <DDNode>
            <taxonomy>
                concrete
            </taxonomy>
            <damage ds="no_damage" mean="1.0078E+00" stddev="3.5099E-01"/>
            <damage ds="slight" mean="2.9329E-01" stddev="5.4491E-02"/>
            <damage ds="moderate" mean="2.5675E-01" stddev="7.9949E-02"/>
            <damage ds="extensive" mean="1.8173E-01" stddev="7.7708E-02"/>
            <damage ds="complete" mean="2.6039E-01" stddev="1.9921E-01"/>
        </DDNode>
        <DDNode>
            <taxonomy>
                steel
            </taxonomy>
            <damage ds="no_damage" mean="4.2150E-01" stddev="1.9680E-01"/>
            <damage ds="slight" mean="7.6996E-02" stddev="2.0042E-02"/>
            <damage ds="moderate" mean="2.2407E-01" stddev="3.7714E-02"/>
            <damage ds="extensive" mean="1.2628E-01" stddev="5.5348E-02"/>
            <damage ds="complete" mean="1.5114E-01" stddev="1.2089E-01"/>
        </DDNode>
    </dmgDistPerTaxonomy>
</nrml>
```

Listing 36 – *Example scenario damage statistics per taxonomy*

9.1.1.3 Total damage statistics

Finally, a total damage distribution output file is also generated, which contains the mean and standard deviation of the total number of buildings in each damage state, as illustrated in the example file in Listing 37.

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <totalDmgDist>
        <damageStates>
            no_damage slight moderate extensive complete
        </damageStates>
        <damage ds="no_damage" mean="5.6572E+03" stddev="1.0439E+03"/>
        <damage ds="slight" mean="7.4325E+02" stddev="5.2130E+02"/>
        <damage ds="moderate" mean="3.3043E+02" stddev="2.8045E+02"/>
        <damage ds="extensive" mean="8.6711E+01" stddev="1.2493E+02"/>
        <damage ds="complete" mean="1.8231E+02" stddev="2.3540E+02"/>
    </totalDmgDist>
</nrml>
```

Listing 37 – *Example total damage statistics*

9.1.2 Scenario collapse maps

Collapse maps are part of the Scenario Damage calculator outputs. These results provide the spatial distribution of the number of the buildings in the ultimate limit state throughout the area of interest. An example snippet from an output file depicting a collapse map for structural damage is presented below.

The results for a number of assets at a given location are stored within the field `CMNode`. This field is associated with a location (defined within the `gml:Point` attribute) and it contains the mean number of buildings in the ultimate limit state (`mean`) and the corresponding standard deviation (`stdDev`) for each asset (identified by the parameter `assetRef`).

9.1.3 Scenario consequence statistics

9.1.3.1 Asset consequence statistics

This output contains the consequences statistics for each of the individual assets defined in the exposure model that fall within the `region_constraint` and have a computed GMF value available within the defined `asset_hazard_distance`. An example output file for structural damage consequences is shown in Table 9.1.

The output file lists consequence statistics for all loss types (amongst "structural", "non-structural", "contents", or "business_interruption") for which a consequence model file was

```
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <collapseMap>
        <CMNode>
            <gml:Point>
                <gml:pos>
                    -122.57 38.113
                </gml:pos>
            </gml:Point>
            <cf assetRef="a3" mean="4.0578E-20" stdDev="2.7091E-19"/>
            <cf assetRef="a2" mean="4.2933E-02" stdDev="5.1818E-02"/>
        </CMNode>
        <CMNode>
            <gml:Point>
                <gml:pos>
                    -122.0 38.225
                </gml:pos>
            </gml:Point>
            <cf assetRef="a6" mean="1.6218E-01" stdDev="1.4466E-01"/>
        </CMNode>
        ...
        <CMNode>
            <gml:Point>
                <gml:pos>
                    -121.886 38.113
                </gml:pos>
            </gml:Point>
            <cf assetRef="a7" mean="9.0050E-06" stdDev="8.8179E-05"/>
        </CMNode>
    </collapseMap>
</nrml>
```

Listing 38 – *Example collapse map*

asset_ref	lon	lat	nonstructural-mean	nonstructural-stddev
a3	-122.57000	38.11300	428.29	281.49
a2	-122.11400	38.11300	1220.84	1111.4
a5	-122.00000	37.91000	1390.59	859.10
a4	-122.00000	38.00000	2889.04	1663.33
a1	-122.00000	38.11300	3191.30	1707.41
a6	-122.00000	38.22500	3310.62	2069.87
a7	-121.88600	38.11300	1415.19	845.83

Table 9.1 – *Example of a scenario asset consequences output file*

also provided in the configuration file in addition to the corresponding fragility model file.

9.1.3.2 Taxonomy consequence statistics

The Scenario Damage calculator also estimates the expected total consequences for buildings of a certain taxonomy, as depicted in the sample output shown in Table 9.2.

taxonomy	contents-mean	contents-stddev	structural-mean	structural-stddev
concrete	6425.37	2965.17	1338.30	1661.46
steel	4531.46	2821.97	597.90	743.71
wood	2889.04	1663.33	334.09	386.80

Table 9.2 – *Example of a scenario taxonomy consequences output file*

9.1.3.3 Total consequence statistics

Finally, a total consequences output file is also generated, which contains the mean and standard deviation of the total consequences for the selected scenario, as illustrated in the example shown in Table 9.3.

contents-mean	contents-stddev	structural-mean	structural-stddev
13845.87	6517.61	2270.29	2440.90

Table 9.3 – *Example of a scenario total consequences output file*

9.2 Scenario Risk Outputs

The Scenario Risk Calculator produces the following set of output files for all loss types (amongst "structural", "nonstructural", "contents", "occupants", or "business_interruption") for which a vulnerability model file was provided in the configuration file:

1. `agg_loss`: this file contains the aggregated scenario loss statistics for the entire portfolio of assets defined in the exposure model. The mean (`mean`) and standard deviation (`stddev`) of the total loss for the portfolio of assets are listed in this file.

2. `loss_map`: this file contains mean (`mean`) and associated standard deviation (`stddev`) of the scenario loss for all assets at each of the unique locations in the exposure model.

If the calculation involves multiple GMPEs, separate output files are generated for each of the above outputs, for each of the different GMPEs used in the calculation.

These different output files for Scenario Risk calculations are described in more detail in the following subsections.

9.2.1 Scenario loss statistics

9.2.1.1 Total loss statistics

This output is always produced for a Scenario Risk calculation and comprises a mean total loss and associated standard deviation for the selected earthquake rupture. These results are stored in a comma separate value (.csv) file as illustrated in the example shown in Table 9.4.

LossType	Unit	Mean	Standard Deviation
structural	USD	8717775315.66	2047771108.36

Table 9.4 – *Example of a scenario total loss output file*

The important attributes in a scenario total loss statistics output file are described below:

- `LossType`: the type of losses that are being stored. This parameter is taken from the vulnerability model that was used in the loss calculations (e.g. fatalities, economic loss).
- `Unit`: this attribute defines the units in which the losses are being measured (e.g. USD or EUR). These units are the same as those defined in the exposure model used for the calculation.
- `Mean`: the mean total loss across the portfolio of assets for the selected earthquake rupture.
- `Standard Deviation`: the standard deviation of the total loss across the portfolio of assets for the selected earthquake rupture.

9.2.2 Scenario loss maps

A scenario loss map contains the spatial distribution of the losses throughout the region of interest. The scenario loss map comprises a mean loss and respective standard deviation for each asset for the selected earthquake rupture, as shown in the example file in Listing 39.

The important attributes in a scenario loss map are described below:

- `LossType`: the type of losses that are being stored. This parameter is taken from the vulnerability model that was used in the loss calculations (e.g. fatalities, economic loss).
- `unit`: this attribute defines the units in which the losses are being measured (e.g. EUR).

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <lossMap
    gsimTreePath="b1"
    investigationTime="None"
    lossType="nonstructural"
    poE="None"
    unit="USD"
    >
        <node>
            <gml:Point>
                <gml:pos>
                    -122.57 38.113
                </gml:pos>
            </gml:Point>
            <loss assetRef="a3" mean="455.43" stdDev="184.14"/>
            <loss assetRef="a2" mean="2769.42" stdDev="2801.90"/>
        </node>
        ...
        <node>
            <gml:Point>
                <gml:pos>
                    -122.0 38.225
                </gml:pos>
            </gml:Point>
            <loss assetRef="a6" mean="8286.74" stdDev="5219.20"/>
        </node>
        <node>
            <gml:Point>
                <gml:pos>
                    -121.886 38.113
                </gml:pos>
            </gml:Point>
            <loss assetRef="a7" mean="1301.43" stdDev="1171.17"/>
        </node>
    </lossMap>
</nrml>
```

Listing 39 – *Example scenario loss map*

- `node`: each loss map comprises various nodes, each node possibly containing a number of assets. The location of the node is defined by a latitude and longitude in decimal degrees within the field `gml:Point`. The mean loss (`mean`) and associated standard deviation (`stdDev`) for each asset (identified by the parameter `assetRef`) is stored in the `loss` field.

9.3 Probabilistic Damage Outputs

9.3.1 Probabilistic damage distribution

The Classical Probabilistic Damage calculator provides the expected damage distribution per asset as a csv file, an example of which is shown in Table 9.5.

asset_ref	no_damage	slight	moderate	extreme	complete
a1	4.4360E-06	6.3482E-03	3.4851E-01	4.7628E-01	1.6884E-01
a2	1.0391E-05	9.1856E-03	3.7883E-01	4.6140E-01	1.5056E-01
…	…	…	…	…	…
a998	6.9569E-02	6.4106E+00	7.4108E+01	5.7563E+01	1.7848E+01
a999	1.2657E-01	8.1294E+00	7.6249E+01	5.4701E+01	1.6792E+01

Table 9.5 – *Example of a probabilistic damage distribution output file*

This file lists the expected number of structural units in each damage state for each asset, for the time period specified by the parameter `risk_investigation_time`.

9.4 Probabilistic Risk Outputs

Probabilistic risk calculations can be run using either the Classical Probabilistic Risk Calculator or the Stochastic Event-Based Probabilistic Risk Calculator. The following set of outputs is generated by both calculators:

1. `loss_curves`: loss exceedance curves describe the probabilities of exceeding a set of loss ratios or loss values, within a given time span (or investigation interval).
2. `loss_map`: loss maps describe the loss (`value`) that is exceeded at the selected probability of exceedance (`poE`) within the specified time period for all assets at each of the unique locations in the exposure model.

In addition, with the Stochastic Event-Based Probabilistic Risk Calculator, it is also possible to calculate the following types of outputs:

1. `agg_curve`: aggregate loss curves describe the exceedance probabilities for a set of loss values for the entire portfolio of assets defined in the exposure model.
2. `event_loss_table`: an event loss table contains the aggregate loss across all assets in the exposure model for each of the simulated earthquake ruptures in the Stochastic Event Set.

9.4.1 Loss exceedance curves

Loss exceedance curves describe the probabilities of exceeding a set of loss ratios or loss values, within a given time span (or investigation interval). Depending upon the type of calculator used and the options defined before running a probabilistic risk calculation, one or more of the sets of loss exceedance curves described in the following subsections will be generated for all loss types (amongst "structural", "nonstructural", "contents", "occupants", or "business_interruption") for which a vulnerability model file was provided in the configuration file.

9.4.1.1 Asset loss exceedance curves

Individual asset loss exceedance curves for ground-up losses are always generated for the Classical Probabilistic Risk Calculator. On the other hand, individual asset loss exceedance curves are not generated for the Stochastic Event-Based Probabilistic Risk Calculator unless the parameter "loss_ratios" is defined in the configuration file. The file snippet included in Listing 40 shows an example asset loss exceedance curve output file.

Each `lossCurve` is associated with a location (defined within the `gml:Point` attribute) and a reference to the asset (`assetRef`) whose loss exceedance probabilities are being represented. Then, three lists of values are presented: the probabilities of exceedance (`poE`), levels of absolute loss (`losses`) and percentages of loss (`lossRatios`).

9.4.1.2 Insured loss exceedance curves

If the exposure model used in the calculation included insurance limits and deductibles for the assets and if the parameter `insured_losses` was set to `true` in the configuration file, individual asset insured loss exceedance curves will also be generated by the Classical Probabilistic Risk Calculator. On the other hand, individual asset insured loss exceedance curves will not be generated for the Stochastic Event-Based Probabilistic Risk Calculator unless the parameter "loss_ratios" is also defined in the configuration file. The file format for an insured loss exceedance curve file is identical to the individual asset loss exceedance curve output file described in the subsection above and is not repeated here.

9.4.1.3 Mean loss exceedance curves

For calculations involving multiple hazard branches, mean asset loss exceedance curves are also generated for both the Classical Probabilistic Risk Calculator and the Stochastic Event-Based Probabilistic Risk Calculator (if the parameter "loss_ratios" is defined in the configuration file). An example output file containing mean asset loss exceedance curves is shown in Listing 41. Apart from the metadata section, the structure of the file is identical to that of the individual asset loss exceedance curve output file.

9.4.1.4 Quantile loss exceedance curves

For calculations involving multiple hazard branches, quantile asset loss exceedance curves can also be generated for both the Classical Probabilistic Risk Calculator and the Stochastic Event-

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <lossCurves
    gsimTreePath="b12_b22"
    investigationTime="50.0"
    lossType="structural"
    sourceModelTreePath="b2"
    unit="USD"
    >
        <lossCurve
        assetRef="a3"
        >
            <gml:Point>
                <gml:pos>
                    -122.57 38.113
                </gml:pos>
            </gml:Point>
            <poEs>
                0.305803 0.305803 0.305803 0.305803 0.305803 0.305803
                0.305803 0.305803 0.305803 0.305803 0.305803 0.201484
                0.143585 0.0582355 0.0246901 0.00995016 0.0 0.0 0.0 0.0 0.0
            </poEs>
            <losses>
                1.0 1.6 2.5 4.0 6.5 10.0 16.0 25.0 40.0 65.0 100.0 160.0
                250.0 400.0 650.0 1000.0 1600.0 2500.0 4000.0 6500.0 10000.0
            </losses>
            <lossRatios>
                0.0001 0.00016 0.00025 0.0004 0.00065 0.001
                0.0016 0.0025 0.004 0.0065 0.01 0.016 0.025
                0.04 0.065 0.1 0.16 0.25 0.4 0.65 1.0
            </lossRatios>
            <averageLoss>
                9.5571e+01
            </averageLoss>
        </lossCurve>

        <lossCurve
        assetRef="a2"
        >

        ...

    </lossCurves>
</nrml>
```

Listing 40 – *Example asset loss exceedance curves*

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <lossCurves
    investigationTime="50.0"
    lossType="structural"
    statistics="mean"
    unit="USD"
    >
        <lossCurve
        assetRef="a3"
        >
            <gml:Point>
                <gml:pos>
                    -122.57 38.113
                </gml:pos>
            </gml:Point>
            <poEs>
                0.275685 0.275685 0.275685 0.275685 0.275685 0.275685
                0.275685 0.275685 0.275685 0.275685 0.270581 0.174495
                0.102799 0.041552 0.018518 0.007463 0.004478 0.0 0.0 0.0 0.0 0.0
            </poEs>
            <losses>
                1.0 1.6 2.5 4.0 6.5 10.0 16.0 25.0 40.0 65.0 100.0 160.0
                250.0 400.0 650.0 1000.0 1600.0 2500.0 4000.0 6500.0 10000.0
            </losses>
            <lossRatios>
                0.0001 0.00016 0.00025 0.0004 0.00065 0.001
                0.0016 0.0025 0.004 0.0065 0.01 0.016 0.025
                0.04 0.065 0.1 0.16 0.25 0.4 0.65 1.0
            </lossRatios>
            <averageLoss>
                9.5571e+01
            </averageLoss>
        </lossCurve>

        <lossCurve
        assetRef="a2"
        >

        ...

    </lossCurves>
</nrml>
```

Listing 41 – *Example mean asset loss exceedance curves*

Based Probabilistic Risk Calculator (if the parameter "loss_ratios" is defined in the configuration file). The quantiles for which loss curves will be calculated should have been defined in the job configuration file for the calculation using the parameter `quantile_loss_curves`. An example output file containing mean asset loss exceedance curves is shown in Listing 42. Apart from the metadata section, the structure of the file is identical to that of the individual asset loss exceedance curve output file.

9.4.1.5 Aggregate loss exceedance curves

Aggregate loss exceedance curves are generated only by the Stochastic Event- Based Probabilistic Risk Calculator and describe the probabilities of exceedance of the total loss across the entire portfolio for a set of loss values within a given time span (or investigation interval). An example aggregate loss exceedance curve file is shown in Listing 43.

Same as described previously for individual assets, aggregate insured loss exceedance curves, mean aggregate loss exceedance curves, and quantile aggregate loss exceedance curves will also be generated when relevant.

9.4.2 Probabilistic loss maps

A probabilistic loss map contains the losses that have a specified probability of exceedance within a given time span (or investigation interval) throughout the region of interest. This result can be generated using either the Stochastic Event-Based Probabilistic Risk Calculator or the Classical Probabilistic Risk Calculator.

The file snippet included in Listing 44 shows an example probabilistic loss map output file.

The important attributes in a probabilistic loss map are described below:

- `sourceModelTreePath`: this parameter specifies the seismic source model used in this particular branch of the logic tree
- `gsimTreePath`: this parameter specifies the combination of ground motion models used in this particular branch of the logic tree
- `lossType`: the type of losses that are being stored. This parameter is taken from the vulnerability model that was used in the loss calculations (e.g. fatalities, economic loss).
- `investigationTime`: time span used to compute the probability of exceedance
- `poE`: parameter specifying the probability of exceedance (e.g. 0.1)
- `unit`: this attribute defines the units in which the losses are being measured (e.g. EUR).
- `node`: each loss map comprises various nodes, each node possibly containing a number of assets. The location of the node is defined by a latitude and longitude in decimal degrees within the field `gml:Point`. The mean loss (`mean`) and associated standard deviation (`stdDev`) for each asset (identified by the parameter `assetRef`) is stored in the `loss` field.

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <lossCurves
    investigationTime="50.0"
    lossType="structural"
    quantileValue="0.5"
    statistics="quantile"
    unit="USD"
    >
        <lossCurve
        assetRef="a3"
        >
            <gml:Point>
                <gml:pos>
                    -122.57 38.113
                </gml:pos>
            </gml:Point>
            <poEs>
                0.308977 0.308977 0.308977 0.308977 0.308977 0.308977
                0.308977 0.308977 0.308977 0.308977 0.307717 0.206570
                0.125464 0.039641 0.018289 0.007370 0.0 0.0 0.0 0.0 0.0
            </poEs>
            <losses>
                1.0 1.6 2.5 4.0 6.5 10.0 16.0 25.0 40.0 65.0 100.0 160.0
                250.0 400.0 650.0 1000.0 1600.0 2500.0 4000.0 6500.0 10000.0
            </losses>
            <lossRatios>
                0.0001 0.00016 0.00025 0.0004 0.00065 0.001
                0.0016 0.0025 0.004 0.0065 0.01 0.016 0.025
                0.04 0.065 0.1 0.16 0.25 0.4 0.65 1.0
            </lossRatios>
            <averageLoss>
                8.5800e+01
            </averageLoss>
        </lossCurve>

        <lossCurve
        assetRef="a2"
        >

        ...

    </lossCurves>
</nrml>
```

Listing 42 – *Example quantile asset loss exceedance curves*

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <aggregateLossCurve
    gsimTreePath="b12_b21"
    investigationTime="50.0"
    lossType="structural"
    sourceModelTreePath="b2"
    unit="USD"
    >
        <poEs>
            0.893542 0.464739 0.189416 0.104166 0.0582355
            0.0392106 0.0246901 0.014888 0.014888 0.0
        </poEs>
        <losses>
            0.0000 3352.3259 6704.6519 10056.9775 13409.3037
            16761.6289 20113.9551 23466.2812 26818.6074 30170.9336
        </losses>
        <averageLoss>
            4.5491e+03
        </averageLoss>
    </aggregateLossCurve>
</nrml>
```

Listing 43 – *Example aggregate loss exceedance curve*

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
    <lossMap
    gsimTreePath="b11_b21"
    investigationTime="50.0"
    lossType="structural"
    poE="0.10"
    unit="USD"
    >
        <node>
            <gml:Point>
                <gml:pos>
                    -122.57 38.113
                </gml:pos>
            </gml:Point>
            <loss assetRef="a3" value="256.854"/>
            <loss assetRef="a2" value="646.431"/>
        </node>
        ...
        <node>
            <gml:Point>
                <gml:pos>
                    -122.0 38.225
                </gml:pos>
            </gml:Point>
            <loss assetRef="a6" value="790.869"/>
        </node>
        <node>
            <gml:Point>
                <gml:pos>
                    -121.886 38.113
                </gml:pos>
            </gml:Point>
            <loss assetRef="a7" value="1269.7"/>
        </node>
    </lossMap>
</nrml>
```

Listing 44 – *Example probabilistic loss map*

9.4.3   Stochastic event loss tables

The Stochastic Event-Based Probabilistic Risk Calculator will also produce an aggregate event loss table. Each row of this table contains the rupture id, and aggregated loss (sum of the losses from the collection of assets within the region of interest), for each event in the stochastic event sets. The rupture id listed in this table is linked with the rupture ids listed in the stochastic event sets files.

The file snippet included in Table 9.6 shows an example stochastic event loss table output file.

rupture_tag	structural	structural_ins	nonstructural
col=00 ses=0001 src=1-1 rup=006-01	1.17E+03	3.50E+00	3.82E+03
col=00 ses=0002 src=1-0 rup=034-01	1.78E+03	3.50E+00	3.86E+03
col=00 ses=0002 src=1-1 rup=027-01	9.84E+02	3.50E+00	3.69E+03
col=00 ses=0002 src=1-3 rup=042-01	7.33E+02	2.80E+00	1.36E+04
col=00 ses=0002 src=1-5 rup=038-01	1.21E+03	3.50E+00	5.46E+03
col=00 ses=0005 src=1-2 rup=047-01	2.23E+03	3.50E+00	5.58E+03
. . .		. . .	. . .	. . .

Table 9.6 – *Example aggregate event loss table*

Asset event loss tables provide calculated losses for each of the assets in the exposure model, for each event within the stochastic event sets. In these tables, each row contains the rupture id, the asset id, magnitude and asset loss. Note that only assets that sustain non-zero losses in a rupture are listed in the asset event loss table. Considering that the amount of data usually contained in an asset event loss table is substantial, this table is not generated by default, unless the parameter `asset_loss_table` is set to `true` in the configuration file.

9.5   Benefit-Cost Ratio Outputs

9.5.1   Retrofitting benefit/cost ratio maps

Ratio maps from the Retrofitting Benefit/Cost Ratio calculator require loss exceedance curves, which can be calculated using the Classical Probabilistic Risk calculator. For this reason, the parameters `sourceModelTreePath` and `gsimTreePath` are also included in this NRML schema, so the whole calculation process can be traced back. The results for each asset are stored as depicted in Listing 45.

- `interestRate`: this parameter represents the interest rate used in the time-value of money calculations
- `assetLifeExpectancy`: this parameter specifies the life expectancy (or design life) of the assets considered for the calculations

```xml
<?xml version="1.0" encoding="utf-8"?>
<nrml
xmlns="http://openquake.org/xmlns/nrml/0.5"
xmlns:gml="http://www.opengis.net/gml"
>
  <bcrMap
  assetLifeExpectancy="50.0"
  gsimTreePath="b1"
  interestRate="0.05"
  lossType="structural"
  sourceModelTreePath="b1"
  >
    <node>
      <gml:Point><gml:pos>80.0888 28.8612</gml:pos></gml:Point>
      <bcr aalOrig="966606" aalRetr="53036" assetRef="a1846" ratio="1.7157"/>
      <bcr aalOrig="225788" aalRetr="26638" assetRef="a4119" ratio="1.4621"/>
      <bcr aalOrig="444595" aalRetr="16952" assetRef="a6444" ratio="1.3275"/>
      <bcr aalOrig="106907" aalRetr="10085" assetRef="a8717" ratio="0.3908"/>
    </node>
    <node>
      <gml:Point><gml:pos>80.0888 28.9362</gml:pos></gml:Point>
      <bcr aalOrig="964381" aalRetr="53007" assetRef="a1784" ratio="1.9248"/>
      <bcr aalOrig="106673" aalRetr="10080" assetRef="a8655" ratio="0.4386"/>
    </node>
    ...
    <node>
      <gml:Point><gml:pos>88.1138 26.9862</gml:pos></gml:Point>
      <bcr aalOrig="5273" aalRetr="612.9" assetRef="a2734" ratio="1.1607"/>
      <bcr aalOrig="23550" aalRetr="1259.3" assetRef="a461" ratio="1.3617"/>
      <bcr aalOrig="10799" aalRetr="393.6" assetRef="a5059" ratio="1.0529"/>
    </node>
  </bcrMap>
</nrml>
```

Listing 45 – *Example retrofitting benefit-cost ratio map*

- `node`: this schema follows the same `node` structure already presented for the loss maps, however, instead of losses for each asset, the benefit/cost ratio (`ratio`), the average annual loss considering the original vulnerability (`aalOrig`) and the average annual loss for the retrofitted (`aalRetr`) configuration of the assets are provided.

10. Demonstrative Examples

The following sections describe the set of demos that have been compiled to demonstrate some of the features and usage of the risk calculators of the OpenQuake-engine. These demos can be found in a public repository on GitHub at the following link: https://github.com/gem/oq-risklib/tree/master/demos/risk. Furthermore, a folder containing all of these demonstrative examples is provided when an OATS (OpenQuake Alpha Testing Service) account is requested, and it is also part of the oq-engine virtual image package.

These examples are purely demonstrative and are not intended to represent accurately the seismicity, vulnerability or exposure characteristics of the region of interest, but simply to provide example input files that can be used as a starting point for users planning to employ the OpenQuake-engine in seismic risk and loss estimation studies.

It is also noted that in the demonstrative examples presented in this section, illustrations about the various messages from the engine displayed in the command line interface are presented. These messages often contain information about the calculation id and output id, which will certainly be different for each user.

Following is the list of demos which illustrate how to use the oq-engine for various scenario-based and probabilistic seismic risk analyses:

- ClassicalBCR
- ClassicalDamage
- ClassicalRisk
- EventBasedRisk
- ScenarioDamage
- ScenarioRisk

These six demos use Nepal as the region of interest. An example exposure model has been developed for this region, comprising 9,144 assets distributed amongst 2,221 locations (due to the existence of more than one asset at the same location). A map with the distribution of

the number of buildings throughout Nepal is presented in Figure 10.1.
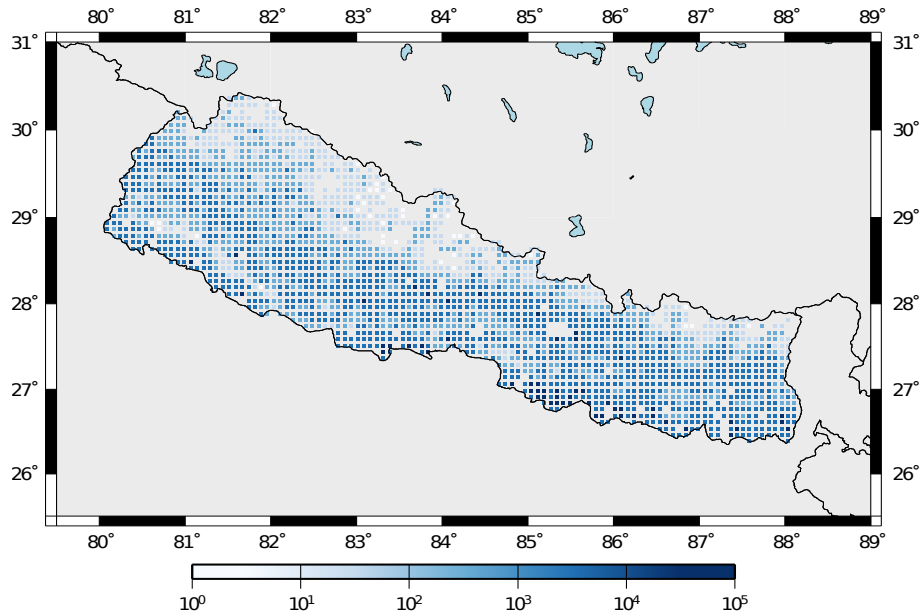
Figure 10.1 – *Distribution of number of buildings in Nepal*

The building portfolio was organised into four classes for the rural areas (adobe, dressed stone, unreinforced fired brick, wooden frames), and five classes for the urban areas (the aforementioned typologies, in addition to reinforced concrete buildings). For each one of these building typologies, vulnerability functions and fragility functions were collected from the published literature available for the region. These input models are only for demonstrative purposes and for further information about the building characteristics of Nepal, users are advised to contact the National Society for Earthquake Technology of Nepal (NSET - http:www.nset.org.np/).

The following sections include instructions not only on how to run the risk calculations, but also on how to produce the necessary hazard inputs. Thus, each demo comprises the configuration file, exposure model and fragility or vulnerability models fundamental for the risk calculations. Each demo folder also a configuration file and the input models to produce the relevant hazard inputs.

10.1   Scenario Damage Demos

A rupture of magnitude Mw 7 in the central part of Nepal is considered in this demo. The characteristics of this rupture (geometry, dip, rake, hypocentre, upper and lower seismogenic depth) are defined in the `fault_rupture.xml` file, and the hazard and risk calculation settings are specified in the `job.ini` file.

To run the Scenario Damage demo, users should navigate to the folder where the required files have been placed and employ following command:

```
user@ubuntu:~\$ oq-engine --run job_hazard.ini,job_risk.ini
```

The hazard calculation should produce the following outputs:

```
Calculation 8967 completed in 4 seconds. Results:
  id | output_type | name
9060 | datastore | gmfs
9061 | datastore | realizations
```

and the following outputs should be produced by the risk calculation:

```
Calculation 8968 completed in 16 seconds. Results:
  id | output_type | name
9062 | datastore | csq_by_asset
9063 | datastore | csq_by_taxon
9064 | datastore | csq_total
9065 | datastore | dmg_by_asset_and_collapse_map
9066 | datastore | dmg_by_taxon
9067 | datastore | dmg_total
```

10.2 Scenario Risk Demos

The same rupture described in the Scenario Damage demo is also used for this demo. In this case, a combined job file, job.ini, is used to specify the configuration parameters for the hazard and risk calculations.

To run the Scenario Risk demo, users should navigate to the folder where the required files have been placed and employ following command:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

and the following outputs should be produced:

```
Calculation 8970 completed in 16 seconds. Results:
  id | output_type | name
9070 | datastore | gmfs
9071 | datastore | realizations
9072 | datastore | loss_map-rlzs
9073 | datastore | agglosses-rlzs
```

10.3 Classical Probabilistic Seismic Damage Demos

The seismic source model developed within the Global Seismic Hazard Assessment Program (GSHAP) is used with the Chiou and Youngs, 2008 ground motion prediction equation to produce the hazard input for this demo. No uncertainties are considered in the seismic source model and since only one GMPE is being considered, there will be only one possible path in the logic tree. Therefore, only one set of seismic hazard curves will be produced. To run the hazard calculation, the following command needs to be employed:

```
user@ubuntu:~\$ oq-engine --run job_hazard.ini
```

which will produce the following sample hazard output:

```
Calculation 8971 completed in 34 seconds. Results:
  id | output_type | name
9074 | datastore | hcurves
9075 | datastore | realizations
```

The risk job calculates the probabilistic damage distribution for each asset in the exposure model starting from the above generated hazard curves. The following command launches the risk calculations:

```
user@ubuntu:~\$ oq-engine --run job_risk.ini --hc 8971
```

and the following sample outputs are obtained:

```
Calculation 8972 completed in 16 seconds. Results:
  id | output_type | name
9076 | datastore | damages-rlzs
```

10.4   Classical Probabilistic Seismic Risk Demos

The same hazard input as described in the Classical Probabilistic Damage demo is used for this demo. Thus, the workflow to produce the set of hazard curves described in Section 10.3 is also valid herein. Then, to run the Classical Probabilistic Risk demo, users should navigate to the folder containing the demo input models and configuration files and employ the following command:

```
user@ubuntu:~\$ oq-engine --run job_hazard.ini
```

which will produce the following hazard output:

```
Calculation 8971 completed in 34 seconds. Results:
  id | output_type | name
9074 | datastore | hcurves
9075 | datastore | realizations
```

In this demo, loss exceedance curves for each asset and two probabilistic loss maps (for probabilities of exceedance of 1% and 10%) are produced. The following command launches these risk calculations:

```
user@ubuntu:~\$ oq-engine --run job_risk.ini --hc 8971
```

and the following outputs are expected:

```
Calculation 8973 completed in 16 seconds. Results:
  id | output_type | name
9077 | datastore | loss_curves-rlzs
9078 | datastore | loss_maps-rlzs
```

10.5 Event Based Probabilistic Seismic Risk Demos

This demo uses the same probabilistic seismic hazard assessment (PSHA) model described in the previous examples in Section 10.3 and Section 10.4. However, instead of hazard curves, sets of ground motion fields will be generated by the hazard calculation of this demo. Again, since there is only one branch in the logic tree, only one set of ground motion fields will be used in the risk calculations. The hazard and risk jobs are defined in a single configuration file for this demo. To trigger the hazard and risk calculations the following command needs to be used:

```
user@ubuntu:~\$ oq-engine --run job.ini
```

and the following results are expected:

```
Calculation 8974 completed in 229 seconds. Results:
  id | output_type | name
9079 | datastore | agg_curve-rlzs
9080 | datastore | agg_loss_table
9081 | datastore | loss_maps-rlzs
9082 | datastore | rcurves-rlzs
9083 | datastore | realizations
9084 | datastore | sescollection
```

10.6 Retrofit Benefit-Cost Ratio Demos

The loss exceedance curves used within this demo are produced using the Classical Probabilistic Risk calculator. Thus, the process to produce the seismic hazard curves described in Section 10.4 can be employed here. Then, the risk calculations can be initiated using the following command:

```
user@ubuntu:~\$ oq-engine --run job_risk.ini --hc 8971
```

Alternatively, the hazard and risk jobs can be run sequentially using:

```
user@ubuntu:~\$ oq-engine --run job_hazard.ini,job_risk.ini
```

which should produce the following output:

```
Calculation 8976 completed in 14 seconds. Results:
  id | output_type | name
9087 | datastore | bcr-rlzs
```

Bibliography

Books

Aki, K. and P. G. Richards (2002). *Quantitative Seismology*. Sausalito, California: University Science Books (cited on pages 27, 32).

Articles

Chiou, B. S.-J. and R. R. Youngs (2008). "An NGA Model for the Average Horizontal Component of Peak Ground Motion and Response Spectra". In: *Earthquake Spectra* 24, pages 173–215 (cited on pages 53, 177).

Chiou, B. S.-J. and R. R. Youngs (2014). "Update of the Chiou and Youngs NGA model for the average horizontal component of peak ground motion and response spectra". In: *Earthquake Spectra* 30.3, pages 1117–1153 (cited on page 28).

Cornell, C. A. (1968). "Engineering Seismic Risk Analysis". In: *Bulletin of the Seismological Society of America* 58, pages 1583–1606 (cited on pages 40, 91, 94, 95).

Field, E. H., T. H. Jordan, and C. A. Cornell (2003). "OpenSHA - A developing Community-Modeling Environment for Seismic Hazard Analysis". In: *Seismological Research Letters* 74, pages 406–419 (cited on pages 40, 91).

Frankel, A. (1995). "Mapping Seismic Hazard in the Central and Eastern United States". In: *Seismological Research Letters* 66.4, pages 8–21 (cited on page 25).

Schwartz, D. P. and K. J. Coppersmith (1984). "Fault Behaviour and Characteristic Earthquakes: Examples from the Wasatch and San Andreas fault zones". In: *Journal of Geophysical Research* 89.B7, pages 5681–5698 (cited on page 27).

Strasser, F. O., M. C. Arango, and J. J. Bommer (2010). "Scaling of the Source Dimensions of Interface and Intraslab Subduction-zone Earthquakes with Moment Magnitude". In: *Seismological Research Letters* 81 (cited on page 39).

Wells, D. L. and K. J. Coppersmith (1994). "New Empirical Relationships among Magnitude, Rupture Length, Rupture Width, Rupture Area, and Surface Displacement". In: *Bulletin of the Seismological Society of America* 84.4, pages 974–1002 (cited on page 39).

Woo, G. (1996). "Kernel Estimation Methods for Seismic Hazard Area Source Modeling". In: *Bulletin of the Seismological Society of America* 86.2, pages 353–362 (cited on page 25).

Youngs, R. R. and K. J. Coppersmith (1985). "Implications of fault slip rates and earthquake recurrence models to probabilistic seismic hazard estimates". In: *Bulletin of the Seismological Society of America* 75, pages 939–964 (cited on pages 37, 38).

Other Sources

EPRI, E. P. R. I. (2011). *Technical Report: Central and Eastern United States Seismic Source Characterisation for Nuclear Facilities*. Report. EPRI, Palo Alto, CA. U. S. DoE, and U. S. NERC (cited on page 39).

McGuire, K. K. (1976). *FORTRAN computer program for seismic risk analysis*. Open-File report 76-67. 102 pages. United States Department of the Interior, Geological Survey (cited on pages 40, 91, 94, 95).

Petersen, M. D., A. D. Frankel, S. C. Harmsen, C. S. Mueller, K. M. Haller, R. L. Wheeler, R. L. Wesson, Y. Yzeng, O. S. Boys, D. M. Perkins, N. Luco, E. H. Field, C. J. Wills, and K. S. Rukstales (2008). *Documentation for the 2008 Update of the United States National Seismic Hazard Maps*. Open File Report 2008-1128. U.S. Department of the Interior, U.S. Geological Survey (cited on page 25).

Index

Glossary

List of Acronyms

GEM

      Global Earthquake Model.

GMF

      Ground Motion Field.

GMPE

      Ground Motion Prediction Equation.

MFD

      Magnitude-Frequency Distribution.

NRML

      Natural hazards' Risk Markup Language.

oq-engine

      OpenQuake-engine.

oq-engine 1.7

      OpenQuake-engine v1.7.

oq-engine 1.8

      OpenQuake-engine v1.8.

oq-hazardlib

      OpenQuake hazard library.

oq-risklib

      OpenQuake risk library.

PSHA

      Probabilistic Seismic Hazard Analysis.

SES

      Stochastic Event Set.

SSHA

      Scenario Based Seismic Hazard Analysis.

SSM

      Seismic Source Model.

List of Terms

area source

>   A source type usually adopted to model distributed seismicity. In an area source the seismicity occurrence rate is assumed uniform over the source area; this produces an hazard pattern with a plateau of constant hazard inside the polygon delimiting the area source and values of hazard that tend to decrease as we move away from the border of the source.

asset

>   An asset is an element with a certain value, which can include buildings or population. For example, an asset can include an individual building at a given location, or a number of buildings that are grouped, co- located at a single location and classified with the same taxonomy.

branch set

>   The structure describing the epistemic uncertainty on a specific parameter or model included in a logic tree structure. It ensembles a number of branches, each one representing a discrete alternative.

branching level

>   It indicates the position where a branch set or a branch is located in a logic tree structure. For example, the first branching level of the seismic source logic tree always contains one or several initial seismic source input models.

characteristic fault source

>   A fault source typology where ruptures always cover the entire fault surface.

complex fault source

>   A source typology usually adopted to model subduction interface faults.

consequence function

>   the distribution of the consequence (or loss) ratio conditional on a set of discrete limit states, defined for a particular taxonomy.

consequence model

>   A set of consequence functions used to model the consequence ratios of all the taxonomies in the exposure model.

deductible

>   A parameter used in the calculation of insured losses that establishes the economic value that needs to be deducted from the ground-up losses.

dip

>   The dip is the steepest angle of descent of the fault plane relative to a horizontal plane; it is measured in degrees [0,90].

earthquake rupture

A 3D surface - representing a portion or the entire fault surface - over which a slip event (i.e. an earthquake) occurs.

earthquake rupture forecast

A list of all possible ruptures generated by all the sources included in a seismic source model. Each element in the list contains: the rupture geometry and the rupture probability of occurrence in a given time span. See also the definition available on the OpenSHA website.

exposure model

A set of assets grouped according to their geographical location, taxonomy and value.

fault trace

A curve representing the intersection between the surface containing the fault surface (or its prolongation) and the topographic surface .
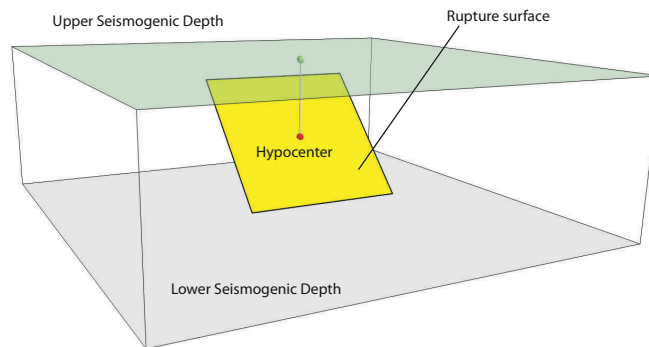
fragility function

the probability of exceeding a set of limit states, given an intensity measure level. These functions can be discrete or continuous.

fragility model

A set of vulnerability functions used to model the fragility of all the assets in the exposure model.

frequency-magnitude distribution

See magnitude-frequency distribution.

grid source

A source typology usually adopted to model distributed seismicity. It is routinely produced by a seismicity smoothing algorithm (one of the most famous algorithm is the one proposed by Frankel (1995)).

ground-motion logic tree

A method used to systematically describe the epistemic uncertainties related to the ground motion models used in the computation of hazard using a specific PSHA input

model.

ground-motion model

An object that given a rupture with specific properties computes the expected ground motion at the given site. In simplest case a ground motion model corresponds to a ground-motion prediction equation. In case of complex PSHA input models, the produced ground motion models contains a set of GMPEs, one for each tectonic region considered.

initial seismic source input model

It is the ensable of information needed to fully describe the seismic sources composing a seismic source input model. The initial seismic source input model is included in the first branching level of a seismic source logic tree.

limit

A parameter used in the calculation of insured losses that establishes the maximum economic amount that can be covered by the insurance industry, according to a certain insurance policy.

magnitude-frequency distribution

A distribution describing the frequency of earthquakes with a specific magnitude. It can be continuous or discrete. One frequency- magnitude distribution frequently adopted in PSHA is the double truncated Gutenberg-Richter distribution.

magnitude-scaling relationship

An empirical relationship linking the magnitude with a parameter describing the size of the corresponding rupture (e.g. the area of the rupture or the rupture length).

non-parametric source

A source typology in which the earthquake rupture forecast is described explicitly by a set of ruptures and the corresponding probabilities of occurrence.

point source

The elemental source typology used in the OpenQuake-engine to model distributed seismicity.

probabilistic seismic hazard analysis

A methodology to compute seismic hazard by taking into account the potential contributions coming from all the sources of engineering importance for a specified site.

PSHA input model

An object containing the information necessary to describe the seismic source and the ground motion models - plus the related epistemic uncertainties.

rake

The rake is the direction in which a hanging wall block moves during a rupture, measured relative to fault strike on the plane of the fault.

rupture model

An object containing the information necessary to describe a earthquake rupture, such as magnitude, hypocenter location, strike, dip, rake, and seismogenic depths.

seismic source

An object that can generate.

seismic source input model

An object containing a list of seismic source data. In the OpenQuake-engine a seismic source model doesn't contain epistemic uncertainty.

simple fault source

A source typology usually adopted to model shallow structures with an uncomplicated geometry.

strike

The strike direction correspond to the angle between the north and the direction you take so that when you walk along the fault trace the fault dips on your right.

taxonomy

Scheme used to classify the assets. For buildings, a classification scheme has been proposed by GEM which considers a number of attributes including lateral load resisting system and its material, height, year of construction. The taxonomy is currently used to link the assets in the exposure model to the relevant vulnerability function or fragility function.

vulnerability function

A function that describes the probability distribution of loss ratio, conditioned on an intensity measure level. Currently only discrete vulnerability functions are supported.

vulnerability model

A set of vulnerability functions used to model the physical vulnerability of all the assets in the exposure model.