

The hazard component of OpenQuake: The calculation engine of the Global Earthquake Model

D. Monelli

ETH, Zurich

M. Pagani & G. Weatherill & V. Silva & H. Crowley

GEM Foundation, Pavia, Italy



SUMMARY

We present OpenQuake-Hazard: the seismic hazard calculator of the OpenQuake engine, the seismic risk platform promoted and developed by the Global Earthquake Model initiative. OpenQuake-Hazard is designed to be an open, transparent, and community driven tool for seismic hazard analysis. The source code is hosted on public repositories that allow for distributed development and it's implemented following modern software design principles like test driven development and continuous integration. It uses an XML-based data exchange format (called NRML, 'Natural hazard' Risk Markup Language) and its design reflects the analysis of seismic hazard models produced for different regions of the world (it's therefore not tight to the needs of a particular geographical/tectonic region).

Keywords: PSHA, GEM, open source

1. INTRODUCTION

Homogeneous global-scale probabilistic seismic hazard analysis (hereinafter PSHA) requires open, transparent, and well-documented definition of data and algorithms. The Global Earthquake Model (Pinho 2012) answers to this need, and one concrete step in this direction is the development of OpenQuake: a platform for seismic hazard and risk assessment (Silva *et al.* 2012). By following current standards in software developments, like test-driven development (Beck 2003) and continuous integration (Duvall *et al.* 2007), OpenQuake aims a becoming an open, and community-driven tool for seismic hazard and risk analysis. Open source: the source code is hosted in a public repository allowing for distributed development. Contributions to the source code follow a public review process. Key parameters in judging code quality are code documentation and code test coverage.

OpenQuake main features are:

- Based on XML (eXtensible Markup Language) data schema: XML is a widely used technology for data exchange format definition developed by the World Wide Web Consortium (W3C). OpenQuake uses an XML dialect called NRML ('Natural hazard' Risk Mark-up Language), which is able to describe a variety of data structures needed for seismic hazard and risk assessment.
- Designed by analysing seismic hazard models developed for different regions of the world (and therefore not tight to a particular geographical/tectonic area), and updated according to the needs of current regional seismic hazard/risk programs.

The present article aims at providing an overview of the OpenQuake-Hazard module, providing information on the software structure, modelling capabilities and available calculators.

2. OPENQUAKE-HAZARD

OpenQuake-Hazard is currently structured as a Python-based module (<https://github.com/gem/oq-engine>) which uses OpenSHA-lite [<https://github.com/gem/opensha>, a reduced version of OpenSHA, (Field *et al.* 2003)) for modelling earthquake ruptures and calculating hazard results such as hazard curves, stochastic event sets, ground motion fields.

2.1 NRML

OpenQuakes uses a XML-based format, called NRML, for describing input and output data. The NRML schema defines several elements. Among the most important, there is the Source Model element (Figure 1), which allows the definition of a seismic source model as a collection of seismic sources (of 4 possible typologies: area, point, simple fault and complex fault). Each source typology is defined by a unique ID, a human readable name, and a tectonic region type (to indicate to which tectonic region typology the source belongs to).

Area and point sources are additionally defined by a hypocentral depth, a top of rupture depth distribution (representing the top of rupture depth for different magnitude bins), and one or more frequency magnitude distribution (FMD)-focal mechanism pairs. This allows specifying multiple faulting styles in the same area (or point), each associated to a specific FMD. An area source is then defined by a polygon (representing the area boundary), and a point source by a single geographical location.

A simple fault is represented by a fault trace, upper and lower seismogenic depths, dip and rake angles, and a single FMD.

A complex fault is represented by fault top and bottom edges coordinates, rake angle and a single FMD.

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

Figure 1. NRML schema for source model definition.

NRML provides also an element for Logic Tree definition. In the NRML schema, a logic tree is defined in terms of three main objects:

- branching level

- branch set
- individual branch

A schematic representation of these three objects is provided in Figure 2. A branching level identifies the position in a tree where branching occurs while a branch set identifies a collection of branches (i.e. individual branches) whose weights sum to 1. In the NRML schema, a logic tree structure is defined through the logicTree element (Figure 3). A logicTree is defined as a sequence of logicTreeBranchingLevel elements. The position in the sequence specifies in which level of the tree the branching level 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. A logicTreeBranchingLevel is defined as a sequence of logicTreeBranchSet elements. Each logicTreeBranchSet defines a particular epistemic uncertainty inside a branching level. A branch set has two required attributes, branchSetID and uncertaintyType (defining the type of epistemic uncertainty the branch set is defining). A branchSet is defined as a sequence of logicTreeBranch elements, each defined 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). Logic trees are currently used in OpenQuake to model epistemic uncertainties in the source model as well as in the ground motion prediction equations.

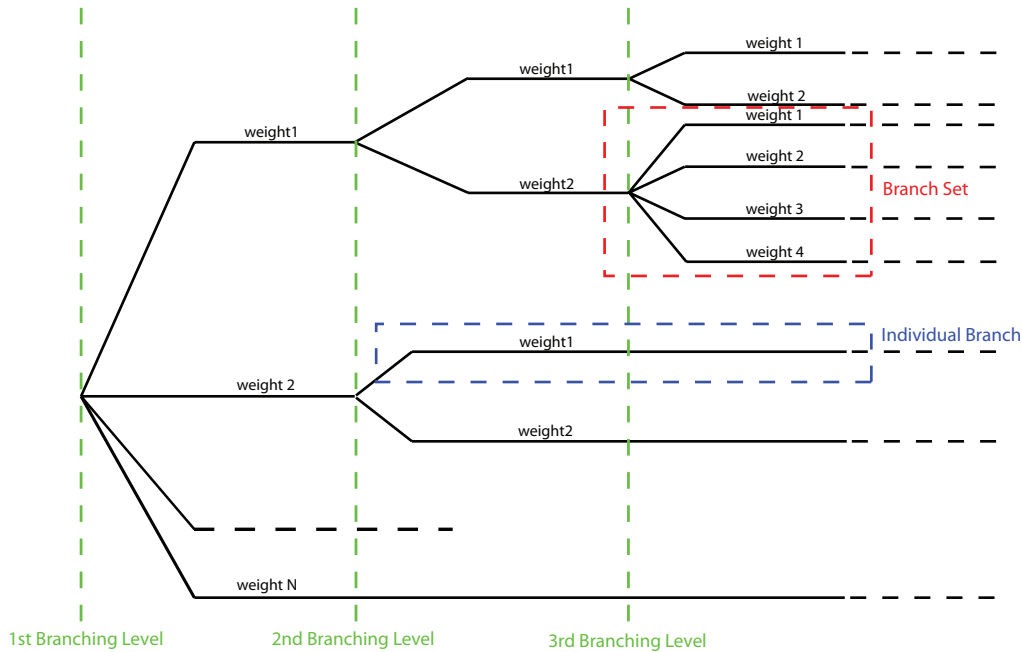


Figure 2. Logic Tree structure as defined in terms of branching levels, branch sets, and individual branches.

```

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

```

Figure 3. NRML schema for logic tree definition.

2.2 Seismic Source Modelling

An essential part in any seismic hazard analysis is the earthquake rupture modelling. It determines the site-to-rupture distance, and therefore the ground shaking distribution at the sites of interest.

For area sources, ruptures' hypocenters are placed inside the area boundary on a regular grid of nodes, whose spacing is determined by an area discretization value. Two main options are available for modeling earthquake ruptures: point or line. If the point rupture option is chosen, earthquake ruptures (independently of their magnitude values) are modeled as points (red dots) located at a depth equal to the hypocentral depth (Figure 4). If the line rupture option is chosen, ruptures' hypocenters are modeled as points (red dots) (again at a depth equal to the hypocentral depth). However, for ruptures with magnitudes in the range specified by the rupture depth distribution, the rupture's top edge is modeled as a line placed at a depth associated to the rupture's magnitude (green and yellow lines represent ruptures' top edges for two magnitude bins in the rupture depth distribution) (Figure 5). The finite rupture option allows modeling the rupture's hypocenter and top edge, and therefore to honor those GMPEs that make use of both these two pieces of information [for instance focal depth and distance to the rupture (closest distance or JB distance)]. The distance to the rupture is correctly computed for vertical ruptures (dip = 90°), however for dipping ruptures (dip < 90°) the distance calculation is based only on the rupture top edge and ignore the rupture extension along dip.

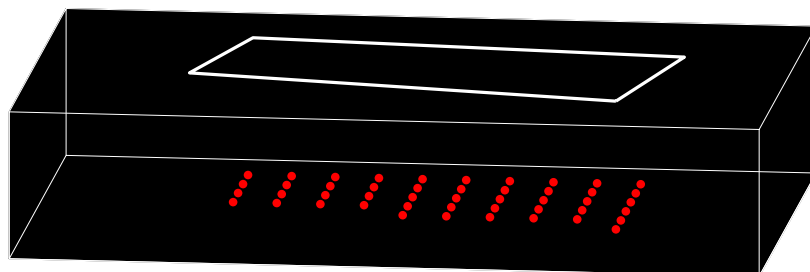


Figure 4. Point ruptures in area source.

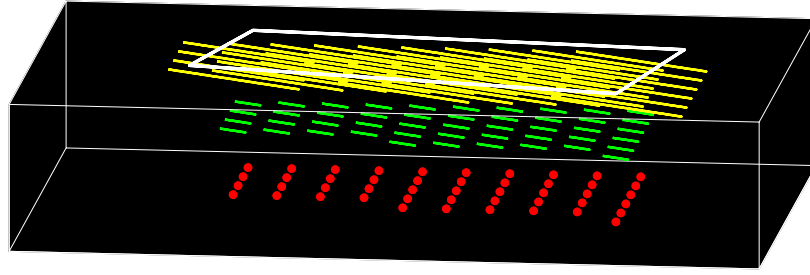


Figure 5. Finite ruptures in area source.

Rupture modeling options for a point source are the same of the ones for an area source, the only difference being that ruptures are modeled on a single geographical location rather than distributed over a regular grid of points inside a polygonal region.

A simple fault is modeled through a regular mesh, whose spacing is given by a fault discretization parameter [Figure 6 (a)]. Ruptures are modeled as portions of the fault surface mesh [Figure 6 (b) and (c)]. The rupture hypocenter is set in the centroid of the rupture surface. Rupture dimensions are derived by using a magnitude scaling relationship and an aspect ratio (rupture length/ rupture width) value. In case the rupture width implied for a given magnitude exceeds the down-dip width of the fault surface, then the rupture length is increased accordingly and the rupture width is set as the down dip width. Once rupture dimensions are determined, ruptures are placed over the entire fault surface [Figure 6 (d)].

A complex fault surface is modeled through a pseudo-regular mesh (Figure 7), that is mesh with a grid spacing that is not uniform but varies depending on the position on the surface. The actual surface discretization is 'approximately' equal to a user-specified surface discretization. In other words, the grid spacing in the actual mesh fluctuates around the specified surface discretization value. The intensity of these fluctuations depends on the degree of complexity of the surface geometry. Higher is the geometrical complexity, larger is the difference between the actual and specified grid spacing. As in a simple fault, ruptures are defined (by using scaling relationship and aspect ratio) as portions of the fault surface mesh.

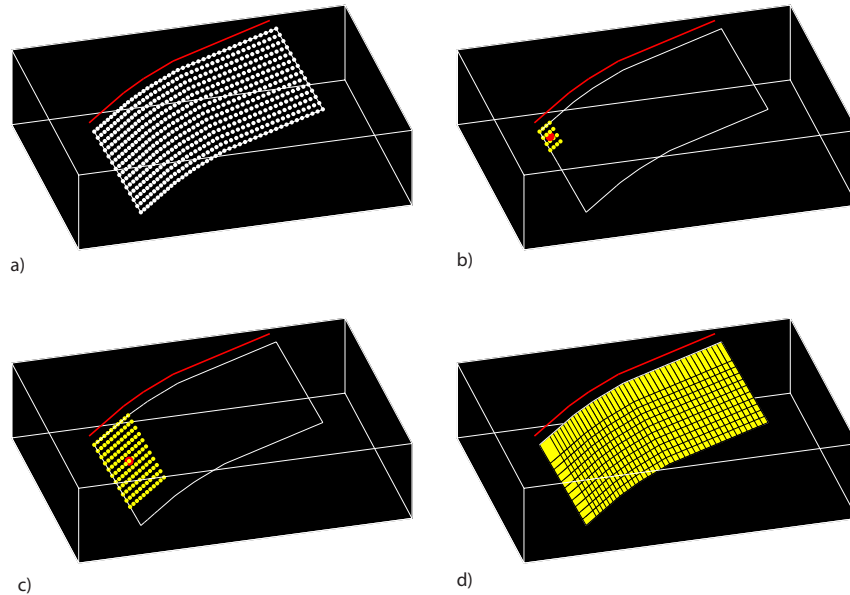


Figure 6. Simple fault mesh, as derived from fault trace (red line), dip angle, upper and lower seismogenic depths (a). Magnitude 5 and 6 ruptures (b and c) as derived by scaling relationship and aspect ratio (equal to 1). Rupture distribution along strike and dip (d).

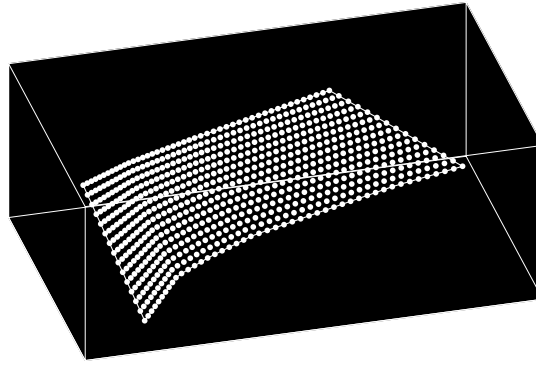


Figure 7. Complex Fault Mesh. From the top and bottom edges coordinates a pseudo-regular grid of points is generated, representing the fault surface.

2.3 Calculators

OpenQuake offers multiple calculators for different types of seismic hazard analysis:

- Classical Probabilistic Seismic Hazard Analysis, allowing calculation of hazard curves and hazard maps following the classical integration procedure [Cornell (1968), McGuire (1976)], as formulated as Field *et al.* (2003).
- Event Based Probabilistic Seismic Hazard Analysis, allowing calculation of ground motion fields from stochastic event sets.
- Scenario Seismic Hazard Analysis, allowing calculation of ground motion fields from a single earthquake scenario, taking into account ground motion aleatory variability.

2.3.1 The Classical Probabilistic Seismic Hazard Calculator

The calculation workflow for the classical calculator is depicted in Figure 8. The input data consists of a PSHA input model which defines a ‘Seismic Source System’, that is the collection of one or more seismic source models and their associated epistemic uncertainties, and a ‘GMPEs System’, representing the set of ground motion prediction equations (per tectonic region type) required for ground shaking modeling. The Logic Tree Processor is responsible for ‘harvesting’ the logic tree, and therefore extracting a source model and a set of GMPEs (one per tectonic region type defined in the source model) to be used for calculation. The selected source model is then used as input for the Earthquake Rupture Forecast calculator, which creates the collection of ruptures (with their associated probability of occurrence in a given time span) as derived by the source model. The rupture collection and the GMPEs serve then as input to the classical hazard curve calculator.

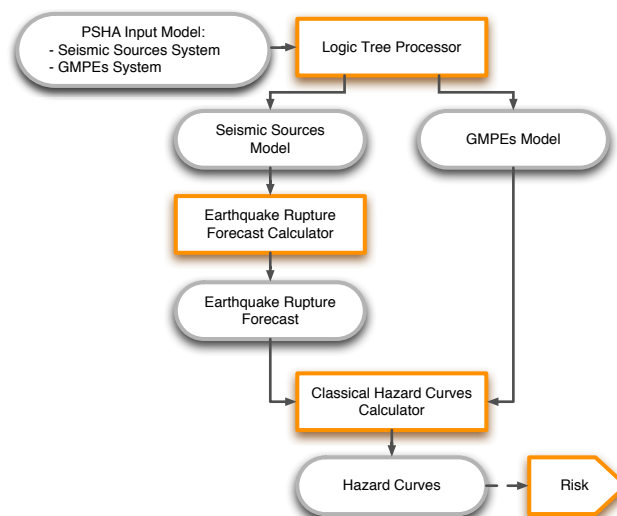


Figure 8. Classical PSHA workflow.

2.3.2 The Event Based Probabilistic Seismic Hazard Calculator

The calculation workflow for the Event Based calculator is shown in Figure 9. The starting input data and the first steps in the workflow are the same as for the Classical calculator. After the earthquake rupture forecast is computed, this is passed as input to the stochastic event set generator, which samples each rupture according to its probability of occurrence, therefore generating a stochastic event set representing a realization of seismicity for a given time span. Each event is then used to compute a ground motion field over the sites of interest, each ground motion field being a realization of the ground shaking distribution as predicted by the GMPEs.

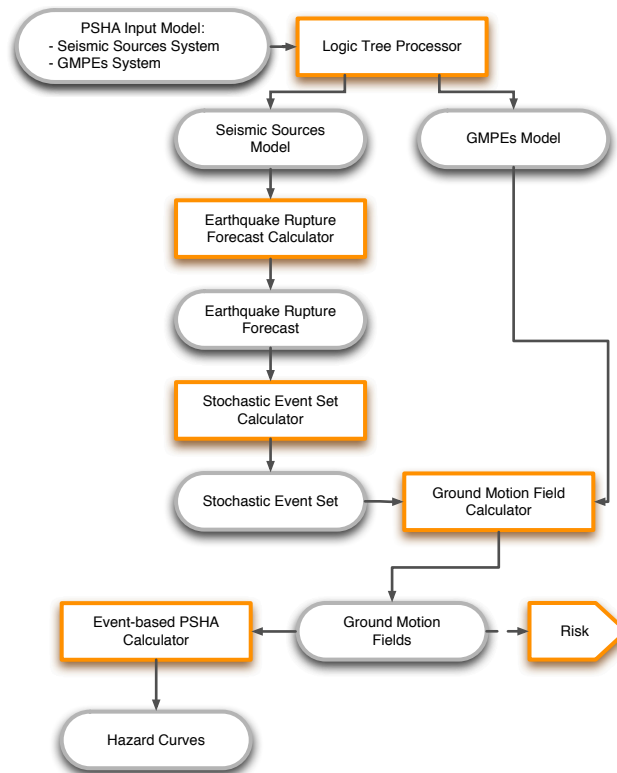


Figure 9. Calculation workflow for Event Based Calculator.

2.3.3 The scenario based calculator

The Scenario Based calculator accepts as input a rupture scenario model consisting of a rupture model (rupture geometry and magnitude), and a GMPE. The ground motion field calculators allows then the computation of multiple ground motion field realizations, each sampling the ground motion variability as predicted by the GMPE.

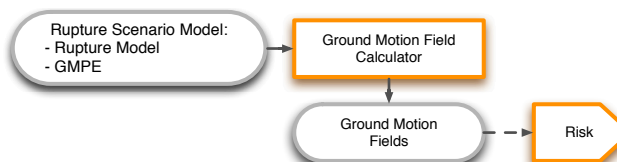


Figure 10. Calculation workflow for Scenario Calculator.

3. CONCLUSIONS

OpenQuake-Hazard is a fast evolving software, which attempts to be a transparent and open tool for seismic hazard assessment at a global scale. To achieve this goal, OpenQuake uses a standardized data model, expressed in a XML format, called NRML. Being specifically designed for data exchange, and for being easily extensible, this data format is particularly suitable for representing seismic hazard models, which often come in different file formats, and involve different parameters.

OpenQuake-Hazard offers multiple options for modeling seismicity: distributed seismicity, through area and point sources, and fault based seismicity, by means of simple and complex fault sources. This means that OpenQuake-Hazard can cope with different strategies for modeling seismic hazard, often linked to the tectonic regime(s) the hazard model is supposed to describe. Area and point sources are often useful for modeling seismicity where fault data are poor (as frequently happens in stable continental regions), while simple fault sources are definitely useful for modeling shallow crustal faults. The flexibility provided by the complex fault source typology can be instead appreciated when modeling complicated subduction environments.

OpenQuake-Hazard provides several calculators, which again reflect different approaches for hazard/risk assessment. The classical calculator represents the most efficient calculator for regional scale seismic hazard assessment. The event based calculator is particularly suitable for seismic risk assessment requiring ground motion spatial correlation to be taken into account. The scenario based calculator allows for scenario risk analysis, potentially taking into account ground motion spatial correlation.

At the present stage (April 2012), Openquake-Hazard is a blend of Java (OpenSHA) and Python code developed following the most common requirements of Open Source software development, such as a public repository, IRC channel and open mailing lists. The source code, released under an open source software license, is freely and openly accessible on a web based repository while the development process is managed so that the community can participate to the day by day development as well as in the mid- and long-term design process.

In December 2011 the GEM Governing Board decided a further improvement of the engine aimed at a complete harmonization of the hazard and risk components. Following this decision, in January 2012 the development of a new – python based – hazard component started, with the goal of being fully harmonized with the physical risk and IT modules. By the end of 2012, GEM will probably release a first version of the new OpenQuake engine.

REFERENCES

- Beck, K. (2003). Test-driven development by example, Addison Wesley.
- Cornell, C. A. (1968). Engineering seismic risk analysis. Bulletin of the Seismological Society of America, 58:1583–1606.
- Duvall, P.M., Matyas, S. and Glover, D. (2007). Continuous Integration – Improving Software Quality and Reducing Risk, Addison Wesley.
- Field, E. H., Jordan, T. H. and Cornell, C. A. (2003). OpenSHA: a developing community-modelling environment for seismic hazard analysis, *Seismological Research Letters*, **74**:4,406-419
- McGuire, K. K. (1976). Fortran computer program for seismic risk analysis. Open-File report 76-67, United States Department of the Interior, Geological Survey.
- Pinho, R. (2012). GEM: a participatory framework for open, state-of-the-art models and tools for earthquake risk assessment worldwide, *Proceedings of the 15th World Conference on Earthquake Engineering*, Lisbon, Portugal.
- Silva, V., Crowley, H., Pagani, M., Monelli, D., Pinho, R. (2012). Development and application of OpenQuake, an open source software for seismic risk assessment, *Proceedings of the 15th World Conference on Earthquake Engineering*, Lisbon, Portugal, paper n. 4917.