

# <QuakeML>

## QuakeML: an XML-based data exchange format for seismology

F. Euchner<sup>1</sup>, D. Schorlemmer<sup>2</sup>, J. Becker<sup>3</sup>, A. Heinloo<sup>3</sup>, P. Kästli<sup>1</sup>, J. Saul<sup>3</sup>, B. Weber<sup>3</sup>, and the QuakeML working group

<sup>1</sup> ETH Zürich, Switzerland. <sup>2</sup> University of Southern California, Los Angeles, USA. <sup>3</sup> GFZ Potsdam, Germany.

### Abstract

QuakeML is a new XML-based data exchange format for seismology. Adopting the XML standard family, it is designed for flexibility, extensibility and modularity.

The first release of QuakeML will cover a basic description of seismic events including picks, arrivals, amplitudes, magnitudes, origins, focal mechanisms, and moment tensors. Further extensions are in progress or planned, e.g., for macroseismic information, location probability density functions, slip distributions, and ground motion information.

As an application of QuakeML, ETH Zurich currently develops a Python-based seismicity analysis toolkit as a contribution to CSEP (Collaboratory for the Study of Earthquake Predictability).

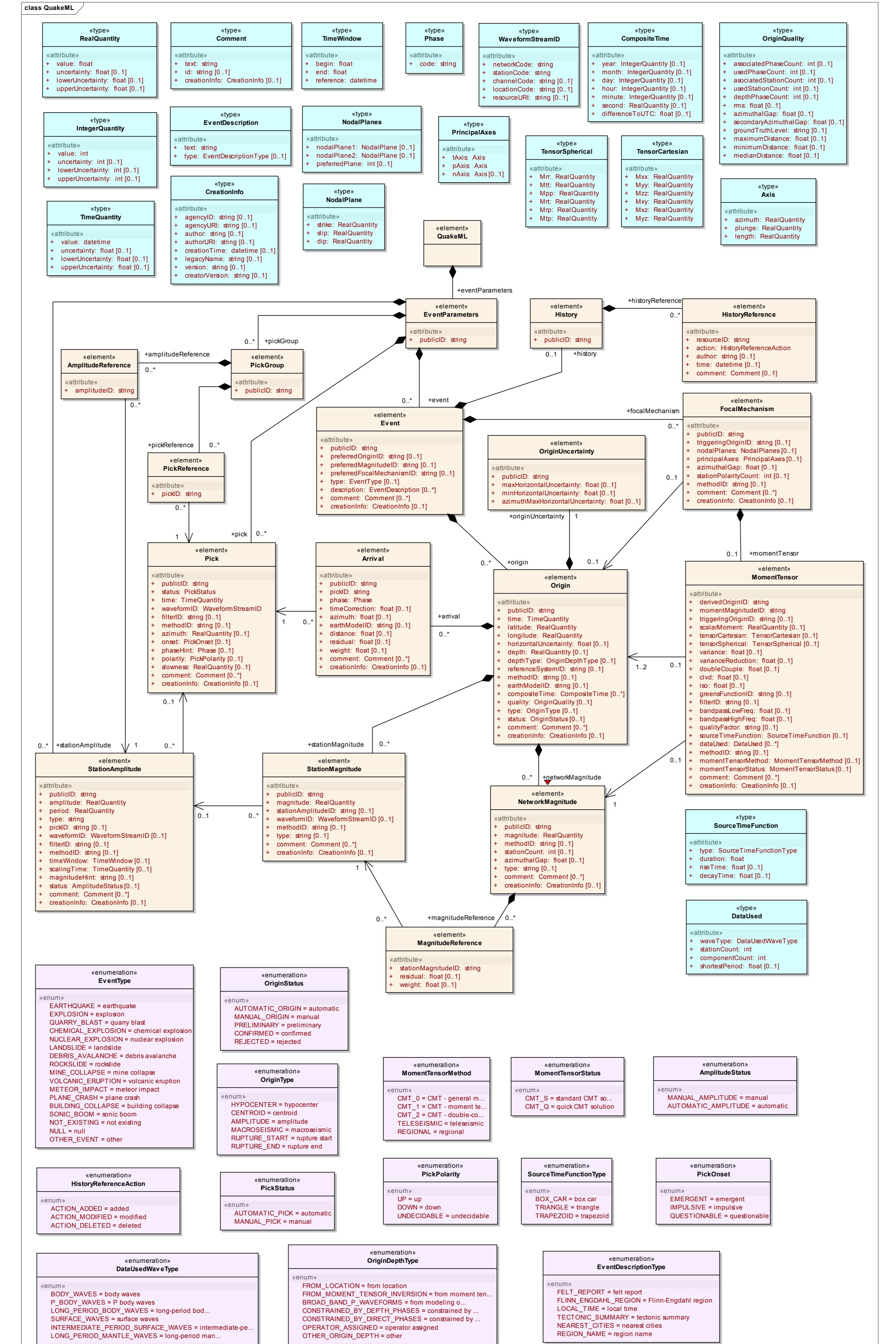
We follow a collaborative and transparent development approach along the lines of the procedures of the World Wide Web Consortium (W3C). QuakeML currently is in working draft status. The standard description will be subjected to a public Request for Comments (RFC) process and eventually reach the status of a recommendation. Current collaborators are from ETH, GFZ, USC, USGS, IRIS DMC, EMSC, ORFEUS, and ISTI.

### QuakeML development website (trac)

This screenshot shows the QuakeML Trac interface. The main page displays a large diagram of the QuakeML class hierarchy. Below the diagram, there are sections for 'Wiki', 'Timeline', 'Roadmap', 'Browse Source', 'View Tickets', 'New Ticket', 'Settings', 'Help/Guide', and 'About Trac'. The 'Wiki' section contains links to 'Start Page', 'Index by Title', 'Index by Date', and 'Last Change'. The 'Roadmap' section has a link to 'Event Parameters'. The 'Browse Source' section shows the 'quakeML' module with various classes like 'EventParameters', 'Event', 'Arrival', 'Origin', etc. The 'View Tickets' section lists several open tickets. The 'New Ticket' section allows users to create new issues. The 'Settings' section includes links to 'E-Mail Addresses' and 'Request for Comments (RFC) Process'. The 'Help/Guide' section has links to 'Open Issues' and 'Missing Features'. The 'About Trac' section provides general information about the Trac system.

This screenshot shows the 'TasksTypes' page of the QuakeML Trac interface. It displays a table of task types with their descriptions and status. The tasks include: WaveformStreamID (locationCode and channelCode), RealQuantity, IntegerQuantity, TimeQuantity (add symmetric uncertainty), NodalPlane (preferred nodal plane), Axis, TensorCartesian, TensorSpherical (support both types?), Comment (deleting comments), CreationInfo (legacyName), CreationInfo (creatorVersion), CompositeTime and TimeQuantity, OriginalQuantity, and ProductID. Below the table, there are links to 'Edit this page' and 'Attach file'.

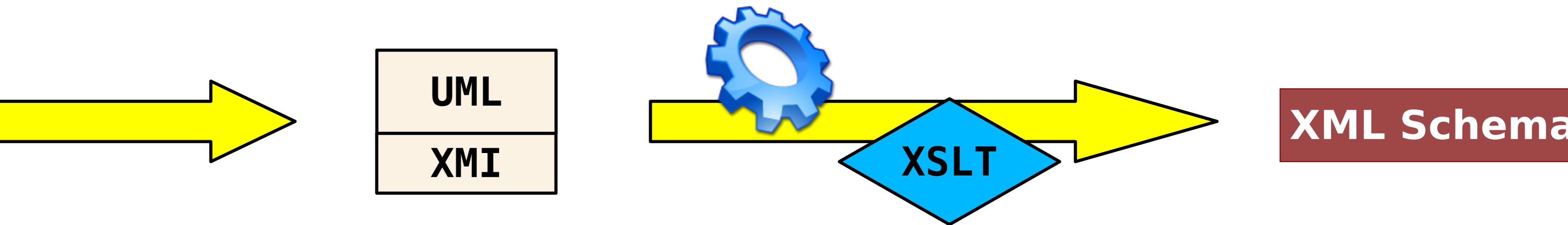
### QuakeML class diagram, version 0.7.3



### Acknowledgements

We thank Rémy Bossu, Ray Buland, John Clinton, Torild van Eck, Paul Friberg, Stéphanie Godey, Winfried Hanka, Linus Kamb, Silvio Maraini, Alessandro Spinuso, Stefan Wiemer, Jochen Wössner, and Adrian Wyss for their contributions to QuakeML.

<http://www.quakeml.org>

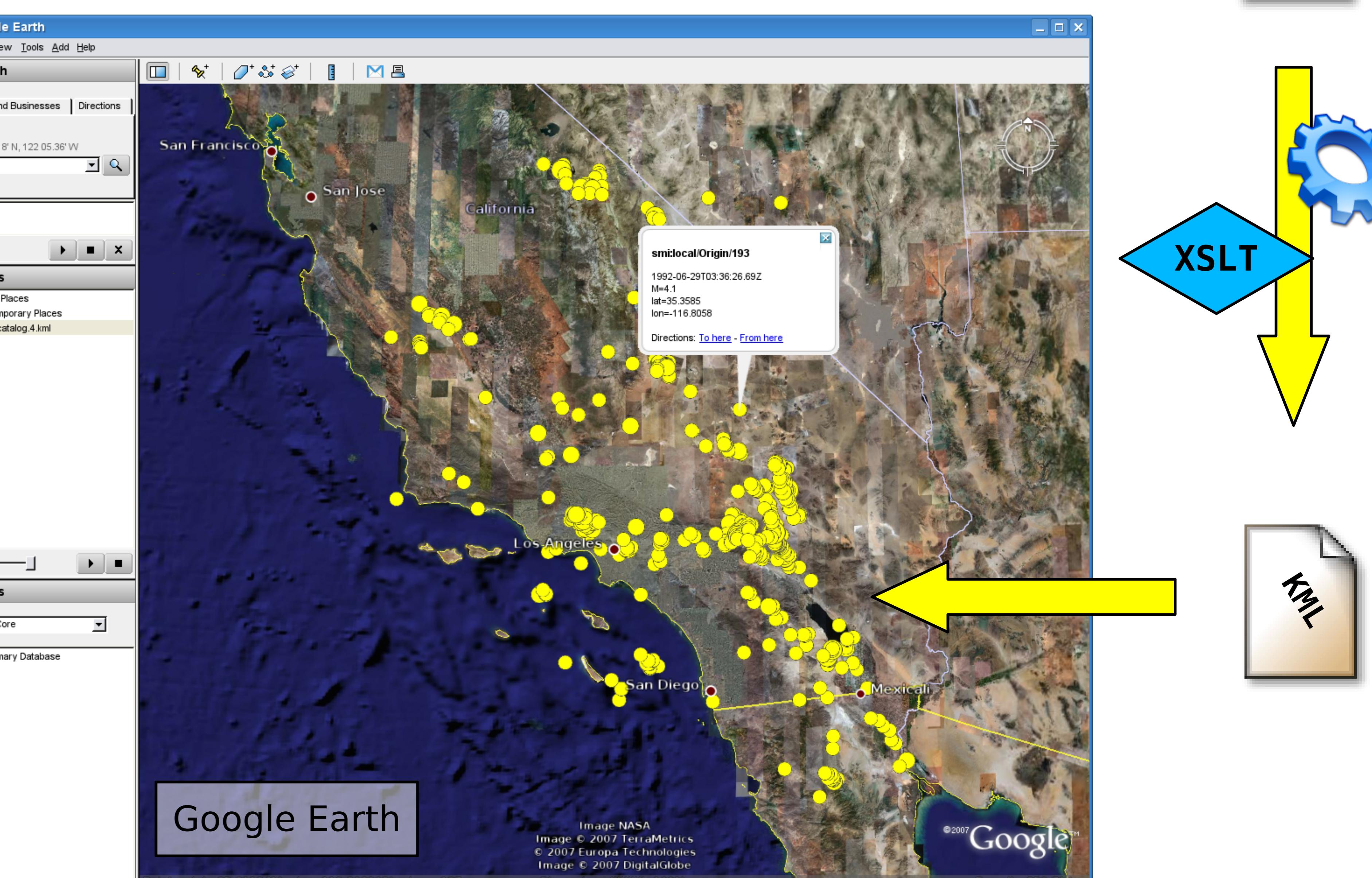


### QuakeML instance

```
<?xml version="1.0"?>
<quakeML>
  <eventParameters publicID="smi:local/Catalogue/001">
    <event publicID="smi:local/Event/1">
      <origin publicID="smi:local/Origin/1">
        <time>1981-04-19T09:02:10.33Z</value>
        <longitude>-117.7743</value>
        <latitude>35.833</value>
        <magnitude>4.8</magnitude>
      </origin>
    </event>
  </eventParameters>
</quakeML>
```

### QuakePy Python package

```
fab@desdemona:~/prog/pyprog/quakeml> python
>>> import quakeml as qp
>>> cat = qp.QPCatalog( 'socal.qml' )
>>> print len( cat.eventParameters.events )
14003
>>> cat.cut( minmag='4.0' )
>>> print len( cat.eventParameters.events )
470
>>> cat.writexml( 'socal.4.qml' )
```



Google Earth

KML