

Standardization of seismic tomographic models and earthquake focal mechanisms data sets based on web technologies, visualization with keyhole markup language[☆]

Luca Postpischl^{*}, Peter Danecsek, Andrea Morelli, Silvia Pondrelli

Istituto Nazionale di Geofisica e Vulcanologia, sezione di Bologna, Via Creti 12, 40128 Bologna, Italy.

ARTICLE INFO

Article history:

Received 8 June 2009

Received in revised form

29 April 2010

Accepted 7 May 2010

Keywords:

Seismology

Geophysics

JSON

Google Earth

KML

Semantic data

ABSTRACT

We present two projects in seismology that have been ported to web technologies, which provide results in Keyhole Markup Language (KML) visualization layers. These use the Google Earth geo-browser as the flexible platform that can substitute specialized graphical tools to perform qualitative visual data analyses and comparisons. The Network of Research Infrastructures for European Seismology (NERIES) Tomographic Earth Model Repository contains data sets from over 20 models from the literature. A hierarchical structure of folders that represent the sets of depths for each model is implemented in KML, and this immediately results into an intuitive interface for users to navigate freely and to compare tomographic plots. The KML layer for the European-Mediterranean Regional Centroid-Moment Tensor Catalog displays the focal mechanism solutions or moderate-magnitude Earthquakes from 1997 to the present. Our aim in both projects was to also propose standard representations of scientific data sets. Here, the general semantic approach of an XML framework has an important impact that must be further explored, although we find the KML syntax to more emphasis on aspects of detailed visualization. We have thus used, and propose the use of, Javascript Object Notation (JSON), another semantic notation that stems from the web-development community that provides a compact, general-purpose, data-exchange format.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction: towards a standard format for Earth model data

Earth models that result from seismic tomographic studies are published by means of their expansion in spherical harmonics functions.¹ Authors typically share the raw coefficients in ascii files, following personal conventions,² but having a common formalism would greatly enhance the data dissemination and their usage by other researchers. In practice, any Earth model can be represented by a set of values on a latitude, longitude and depth three-dimensional (3D) grid covering the interior of the Earth. Most models are based on p-wave and s-wave propagation velocities, but in general, many other physical measurables can also be exploited to gain insight into the Earth's interior.

A model typically consists of large quantities of data. Within the *Network of Research Infrastructures for European Seismology*

(NERIES) European Union Research Project 'JRA1' activity,³ which aims at defining a unified reference Earth model for the European region, twenty of the most popular existing global models are being reviewed. The need for an efficient representation of the data sets that can also serve as a common base to visualize and compare models without installing and running Fortran executables provided by original authors have become evident.

We have therefore searched for solutions that are capable of being *language independent*, *easy to parse*, and *semantic*, i.e. solutions that result in self-describing structures that integrate the data and metadata into a single resource. We believe that the *Javascript Object Notation (JSON)* formalism fits these general requirements well, and we have proposed its adoption for the standardization of Earth models (Postpischl et al., 2008).

JSON is a subset of the ECMA-262 specification (Crockford, 2006), and it is based on a very minimal and clean notation. This is currently supported by most of the major programming languages that should already be familiar to most programmers (ActionScript, C, C++, C#, Cold Fusion, D, Delphi, E, Erlang, Haskell, Java, Lisp, LotusScript, Lua, Perl, Objective-C, OCAML, PHP, Python,

[☆] Code available from server at <http://www.bo.ingv.it/NeriesDataFormats/viewToModels.kml> or at <http://www.iamg.org/CGEditor/index.htm>.

^{*} Corresponding author. Tel: +39 051 4151495; fax: +39 051 4151498.

E-mail address: postpischl@bo.ingv.it (L. Postpischl).

¹ <http://mathworld.wolfram.com/SphericalHarmonic.html>

² <http://geodynamics.usc.edu/~becker/tomography/node12.html>.

³ <http://www.neries-eu.org/>.

Rebol, Ruby, Scheme, and Squeak), given its derivation from the basic structures of C/C++.

Compared with an equivalent extensible markup language (XML) implementation, a JSON object is significantly more lightweight in file size (Lawrence, 2004). Moreover JSON-formatted structures are fully defined Javascript objects, so their elements are directly parsable by the browser javascript engine without the need for extra middle-ware layers, such as an SOAP, XPath, and SAX that are typically needed for XML processing. JSON is thus very efficient, and it is becoming the preferred data-exchange format for many representation state transfer (REST)ful web services (Richardson and Ruby, 2007). All of the open-source Ajax frameworks developed through the Web Standards Community offer advanced support for JSON, so adopting JSON also means bringing this huge arsenal of software tools into the hands of scientific researchers, potentially transforming a web browser into an advanced data visualization and analysis tool. Moreover the support of JSON by many programming languages guarantees that the conversion of scientific data towards higher level formats, such as the Network Common Data Form (NETCDF, 2007) and Hierarchical Data Format (HDF5, 2010), is easy to implement. This allows advanced plotting with specialized visualization tools, such as the GEON Integrated Data Viewer (IDV, 2007) and the Generic Mapping Tools (GMT, 1988).

KML-generating routines are also easily implemented from the JSON formalism with many programming languages, hence geo-browsers can be directly exploited for visualization of JSON data structures; in our opinion KML, being heavily targeted for presentational tasks, is not a suitable format for general spatial-data storage and transfer.

All revisions of the JSON standardization format proposal for tomographic Earth models are published and discussed openly on the '/wiki/' pages of the Istituto Nazionale di Geofisica e Vulcanologia (INGV) Bologna branch website (Eurorem, 2008). (Graphic 1).

2. Visualization of tomographic maps with KML

The tomographic models available through the current NERIES JRA1 activity are all expressed in layers of data points that correspond to various depths from ground level down to 600 km below the surface of the Earth. For each depth level, a regular 2D grid with two degrees of resolution of latitude and longitude is defined for the European-Mediterranean region. The visualization of any aspect of the model tightly follows this general organizational scheme of the JSON data sets: in our main KML file for the tomographic models⁴ the depth levels are all listed in the 'places' sidebar of Google Earth, and by selecting the corresponding check-boxes, the tomographic map for that particular level is displayed, clamped to ground level. We have found this to be the most effective and user-friendly way to actually carry out comparisons across the depth levels, and it is clearly preferable to displaying all of the depths layered according to different altitudes (N.B. Google Earth does not support negative values).

As common in tomographic studies, the data values that are color coded into the KML polygons correspond to the percentage variations from the mean values computed for the depth levels of the single data points. This scalar quantity serves as a common parameter with which to compare different levels, and also different models, even if these are based on physical measurables that are not related.

The most interesting element of the implementation is the use of '< NetworkLink >' KML entities inside the main hierarchy of

< Folders > (Fig. 1). In this way, all of the KML code generating a particular tomographic map is retrieved asynchronously from the remote server only when the user explicitly requests it, which optimizes bandwidth use. As displayed in the code snippet of Graphic 2, each < NetworkLink > passes two parameters to a remote server-side script (Graphic 3) that parses the JSON data structure for that model and returns the set of KML < Polygon > s corresponding to the depth level to the Google Earth client application.

It would be equally easy to use this same technique based on < networkLink > s for importing high-resolution graphical files, batch-produced with specialized scientific routines, as < groundOverlay > s KML entities, an approach which transforms Google Earth into a high-quality, cross-platform software tool for quick, interactive, comparisons of any two models.

As a proof-of-concept of the capabilities of the proposed JSON format, we have here chosen a 'raw' implementation from the original data. This has also given us insights concerning the scaling of the performance of the Google Earth platform for the rendering of layers of several thousands of polygons in real time.

As we have not had performance issue limitations in the drawing of the horizontal layers, we believe that our approach can be further exploited for the creation of other kinds of maps that are commonly found in tomographic studies: vertical cross-sections. In this case, the import of batch-produced images would be limiting, because the user would want to create these maps interactively, and along many directions. The maps have to be created in real time once the coordinates of the path start and end points have been specified by the user. To perform such a selection, the standard Google Earth user interface based on the clickable hierarchies of folders in the 'places' sidebar would not be sufficient. Yamagishi et al. (2008) used a form on a web page to define the path and the output KML files for cross-sections,⁵ although these files have to be imported into Google Earth manually.

The Google Earth plug-in and the corresponding javascript application programming interface (Earth API, 2008) that brings the geo-browser KML support inside the web browser can be exploited to implement vertical cross-sections with a fully interactive user experience. Based on the plug-in, we intend to develop a tool that will let users select a tomographic model from a standard HTML form, and then allow them to define a path by dragging the mouse onto a 3D globe, whereby the mouse-up event of this line-drawing action will trigger the processing of the tomographic data for that particular cross-section.

3. The European-Mediterranean regional centroid-moment tensors

The European-Mediterranean regional centroid-moment tensor (RCMT) catalog collects seismic moment-tensor solutions that have been routinely computed for earthquakes with moderate magnitudes ($4.5 < M < 5.5$) in the European-Mediterranean regions (Pondrelli et al., 2007). We now have a catalog of centroid-moment tensors that includes more than 900 RCMTs that all together represent the time span from 1997 to the present. This database represents an extension for smaller magnitudes of the Global CMT catalog⁶ for the European-Mediterranean area (Ekström et al., 2005).

An RCMT computation is based on the analysis of seismograms that are recorded at regional distances and on the modeling of

⁵ <http://www.jamstec.go.jp/pacific21/TMGonGE/kmlgenerator/tomography.html>.

⁶ <http://www.globalcmt.org>.

⁴ <http://www.bo.ingv.it/NeriesDataFormats/viewToModels.kml>

```

1 {
2   "filetype": "Tomographic Earth Model",
3   "version": "0.5beta",
4   "model": "mk12wm13p",
5   "authors": [
6     {
7       "name": "",
8       "surname": "",
9       "institution": ""
10    },
11    <...>
12  ],
13  "doi": "",
14  "title": "Simultaneous inversion for 3-D variations in shear and
15  bulk velocity in the mantle",
16  "format": "pointdata",
17  "dataset": {
18    "mode": "compact",
19    "mapping": "latlon",
20    "reference": "WGS84/sphere",
21    "depth": {
22      "min": 50,
23      "max": 1518,
24      "unit": "km",
25      "values": [ 50, 250, 450, 650, 850, 1050, 1250, 1450, 1518 ]
26    },
27    "lat": {
28      "min": 10,
29      "max": 90,
30      "unit": "degree/radian",
31      "step": 2,
32      "values": "null"
33    },
34    "lon": {
35      "min": -40,
36      "max": 80,
37      "unit": "degree/radian",
38      "step": 2,
39      "values": "null"
40    },
41    "measurables": [
42      {
43        "name": "p-wave velocity",
44        "abbr": "Vp",
45        "units": "km/s",
46        "error": "null",
47        "values": [
48          [ 8.11107, 8.10290, 8.09699, 8.09306, 8.09081, 8.08999, 8.09036,
49            8.09178, 8.09415, 8.09734, 8.10122, 8.10557, 8.11008, 8.11437,
50            8.11800, 8.12055, 8.12167, 8.12116, 8.11901, 8.11543, 8.11088,
51            8.10594, 8.10135, 8.09780, 8.09590, 8.09605, 8.09839, 8.10275,
52            8.10866, 8.11545, 8.12232, 8.12846, 8.13318, 8.13604, 8.13692,
53            8.13601, 8.13387, 8.13123, 8.12900, 8.12804, 8.12904, 8.10734,

```

Graphic 1. Sample tomographic model data structure in JSON format.

intermediate period surface waves (Arvidsson and Ekström, 1998). Over the past few years, we also inverted simultaneously for body and surface waves, although only for those seismic events with a magnitude greater than 5.0—i.e. when the signal-to-noise ratio at 40–100 s of period is significant enough to contribute to the inversion.

4. The RCMT web-search interface

The RCMT catalog is updated every few months, and reports are published regularly. However, moment-tensor solutions are

also being computed on the basis of data that are available in quasi-real time. These preliminary solutions are available within a few hours after the occurrence of an earthquake, and they are published immediately as ‘Quick RCMTs’ in the online version.⁷

To provide full search capabilities over the moment-tensor solutions, the data set that was previously available as static ASCII files has been imported into a MySQL relational database. This thus provides a PHP web application (Fig. 2) with enhanced user interface controls that allow users to submit queries as a

⁷ <http://www.bo.ingv.it/RCMT/searchRCMT.html>.

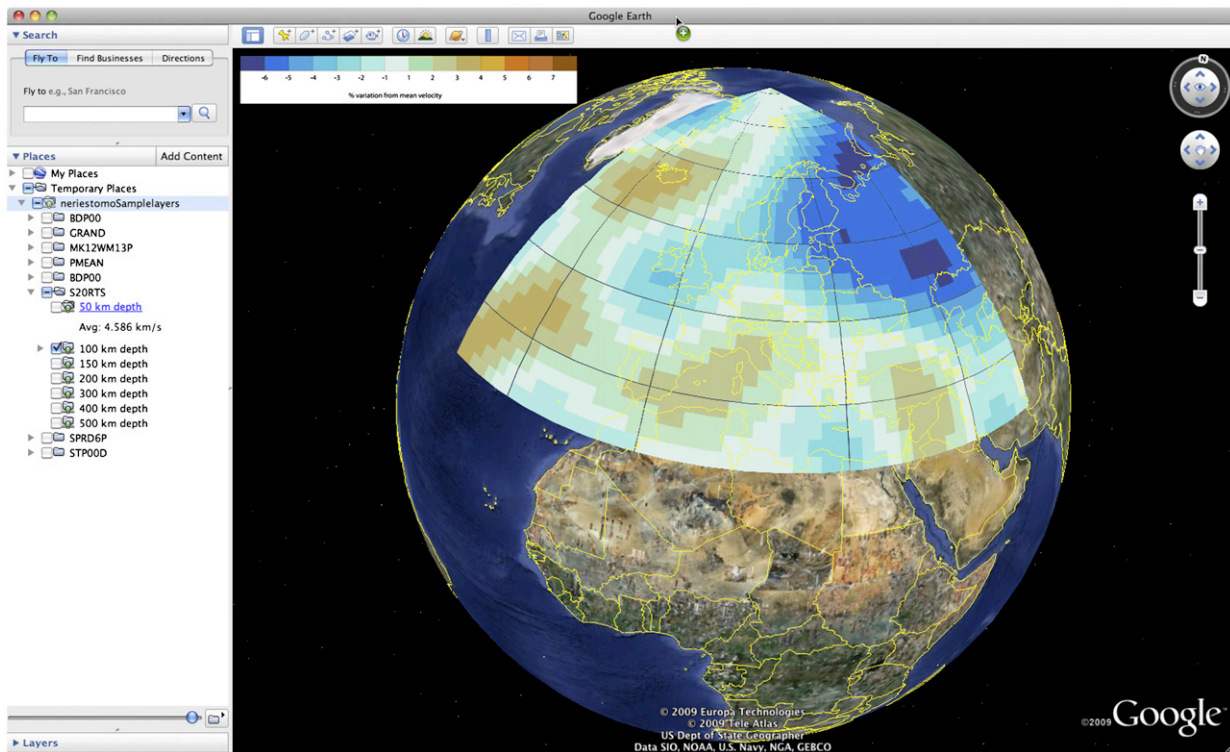


Fig. 1. Comparison of tomographic models. Tomographic models appear as a hierarchical tree in the Google Earth location sidebar, which provides a simple user interface for comparisons between different depth levels within one model or between corresponding depth levels of multiple models. Each depth level entry is actually a networkLink that triggers an execution of a remote PHP script only on user request, parsing JSON data into KML polygons.

combination of date, magnitude, depth and geographic coordinate ranges (Pondrelli et al., 2008). These events can be further filtered by two flags: one to distinguish between quick and definitive solutions, and one for the ‘quality’ categorization that we are using for the solutions.

While selecting the latitude and longitude ranges with the slide-bars of the form, the corresponding area of interest is drawn in real time on a zoomable map (Google Maps), which provides precise visual feedback to the user. To guarantee that this can be operated even with older browsers, there are standard input field elements which are synchronized with the slide-bar controls. Once the search parameters have been set and the query is submitted, the map is updated with beach-ball representations of the focal mechanisms,⁸ which are positioned at each earthquake epicenter location. A mouse-over event is defined on each beach ball, which generates a dynamic information box that contains the full solution for the event.

Immediately below the map, there is a visual characterization of the resulting data set, given as magnitude, depth and time frequency in-line distribution histograms⁹ as in Tufte, 2006. Finally, the full data set is provided and the user can convert it between various formats within the web browser, thus allowing it to be exported directly into other applications. The default output format for the data set through this web service is again JSON, as it intuitively and conveniently integrates the search parameter metadata with the actual records into a single machine and human-readable resource/file.

The other output formats we provide are all generated dynamically on the client-side from the JSON data-object, without

further connections to the server. Currently, the form includes the Psmeca and Pselomeca GMT formats, basic comma-separated values (CSV) for easy integration in spreadsheet applications, and KML for Google Earth. We plan to also include QuakeML, an XML format specifically developed for seismic data exchanges (QuakeML, 2007), and GEON IDV ASCII Point-data format (GIAP, 2007), an ASCII format for which netCDF/HDF5 converters are available.

5. The RCMT KML output

The place-mark icons used in the KML output for the moment tensors are the same 2D bottom-half projections of the beach balls displayed on the in-page Google Map, and they retain the visualization of the further details concerning each earthquake in the dynamic information boxes that are shown on a user click. The beach balls are scaled in proportion to the magnitude of the corresponding earthquake events. (Graphic 4).

This conventional representation that corresponds to the classic one found in the literature poses new problems in the Google Earth 3D-rendering environment: when the user changes the point of view by rotating around the vertical axis or by tilting the view, the orientation of the place-marks remains fixed, and thus no longer correct.

KML would allow the importing of 3D-sphere Collada model files (.dae) for the beach balls, and to further orient them in space into the reference system used in Google Earth, with a simple transformation of the strike, dip and rake angles contained in the data records (i.e. the < heading >, < tilt >, < roll > KML entities) (De Paor, 2008; De Paor and Pinan-Llomas, 2006). (Graphic 5).

Within the 3D-viewing engine, this implementation would clearly display the fault-plane intersection with the ground, a

⁸ http://en.wikipedia.org/wiki/Focal_mechanism

⁹ <http://en.wikipedia.org/wiki/Spakline>


```

1 <?xml version='1.0' encoding='UTF-8'?>
2 <kml xmlns = 'http://earth.google.com/kml/2.1'>
3   <Folder>
4     <name>NERIES-JRA1 reviewed Earth Models</name>
5     <open>1</open>
6     <description>Sample tomographic layers created by processing the
7     json data files of the models.
8     Values of the p/s wave propagation velocity anomalies on a 2-
9     degrees Lat-Long regular grid.
10    </description>
11    <ScreenOverlay>
12      <visibility>1</visibility>
13      <name>Color table - Legend</name>
14      <Icon>
15        <href>http://www.bo.ingv.it/NeriesDataFormats/assets/
16        meanVelocityPercVariationLegend.jpg</href>
17      </Icon>
18      <overlayXY x="0" y="1" xunits="fraction" yunits="fraction"/>
19      <screenXY x="0" y="1" xunits="fraction" yunits="fraction"/>
20    </ScreenOverlay>
21    <Folder>
22      <name>PMEAN</name>
23      <NetworkLink>
24        <visibility>0</visibility>
25        <name>71 km depth</name>
26        <description></description>
27        <open>1</open>
28        <Link>
29          <href>http://www.bo.ingv.it/NeriesDataFormats/
30          NeriesJsonGE.php?depth=71&model=pmean</href>
31        </Link>
32      </NetworkLink>
33      <NetworkLink>
34        <name>215 km depth</name>
35        <visibility>0</visibility>
36        <Link>
37          <href>http://www.bo.ingv.it/NeriesDataFormats/
38          NeriesJsonGE.php?depth=215&model=pmean</href>
39        </Link>
40      </NetworkLink>
41      <NetworkLink>
42        <name>358 km depth</name>
43        <visibility>0</visibility>
44        <Link>
45          <href>http://www.bo.ingv.it/NeriesDataFormats/
46          NeriesJsonGE.php</href>
47          <httpQuery><![CDATA[depth=358&model=pmean]]></httpQuery>
48        </Link>
49      </NetworkLink>
50      <NetworkLink>
51        <name>502 km depth</name>
52        <visibility>0</visibility>
53        <Link>
54          <href>http://www.bo.ingv.it/NeriesDataFormats/

```

Graphic 2. KML snippet for networkLink entity passing variables to a remote PHP script.

feature that would be immensely useful in educational contexts. However, after some experimentation and discussion, we finally chose to only provide the classic 2D projection of the beach balls instead, as commonly found for printed maps in the literature (Fig. 3). This is to ease the analysis by seismologists, who will be accustomed to the very counter-intuitive convention for moment-tensor solutions, i.e. when viewed from above; beach balls are displayed as the horizontal projection of their bottom-half, whereas from the same point of view, the 3D spheres would instead be seen as their top half. Providing both representations as two distinct KML

folders might be the only solution to this dichotomy, whereby users would be able to choose their preferred representation from the locations sidebar. Also, to give users a better sense of the distinction between these two representations, the 3D spheres can be substituted by other custom, more intuitive, 3D models of the fault planes, as has been used by Labay and Haeussler (2007) and by De Paor and Williams (2006). This will be implemented into further revisions of the RCMT project.

To further simplify and enhance the readability of the map in Google Earth, as for the tomographic model implementation, we

again chose not to position the solutions vertically, and instead we kept them clamped to ground level. In this way, the beach balls are always clearly displayed at their precise epicenter localization points, and they are not affected by perspective parallax effects introduced by the 3D rendering engine on tilted semi-horizontal views. This also by-passes lack of support of Google Earth for negative altitude values, which would otherwise require a specific solution based on extra 3D Collada models, as for that developed by De Paor (2007). Such an implementation is not practicable in our case, since the data sets are formed dynamically at the request of the user, while the .dae Collada files would have to be created

explicitly within the Google Sketchup desktop application (Sketchup, 2007), and then referenced in the KML.

The last important feature implemented in these KML code snippets is the inclusion of the '<TimeStamp><when> ...', as used by De Paor and Pinan-Llomas (2006). This very simple KML entity auto-triggers the appearance of the time slide-bar in Google Earth, allowing the user to interactively animate and analyze the data set by setting a reference time window and dragging it back and forth within the global 1997-to-present time-frame. This feature is especially important in the full export version of the data set that we

```

1  <?php
2  //get the parameters from the URL (as found in each <NetworkLink>
3  of the kml file)
4  $depth = $_GET["depth"];
5  $model = $_GET["model"];
6  //set the hex values for the color-index scale
7  $color = array
8  ("565D79", "2C5F9D", "3F81AD", "68B5C8", "96CEC5", "CCEEC0", "EEF5DF", "E9
9  F1AD", "E4E37A", "E6CC4C", "E5AB34", "E6760D", "CD7334");
10 //retrieve the json file for the requested model
11 //URL sanitation checks omitted here for brevity...
12 $site = "http://www.bo.ingv.it";
13 $path = "/sample/path/eurorem/models/json/";
14 $json_file = $path.$model.".json.txt";
15 $string= file_get_contents($json_file);
16 //parsing of the JSON string into a PHP object
17 $obj = json_decode($string);
18 //select the index inside the array of depths corresponding to the
19 requested depth value
20 $depths= $obj->{'dataset'}->{'depth'}->{'values'};
21 $count = 0;
22 $depthIndex = null;
23
24 function average($a){
25     return array_sum($a)/count($a);
26 }
27
28 foreach ($depths as $value) {
29     if ($depths[$count] == $depth) {
30         $depthIndex = $count;
31     }
32     $count++;
33 }
34
35 $avgSpeed = average($obj->{'dataset'}->{'measurables'}[0]->
36 {'values'}[$depthIndex]);
37
38 //start building the KML output
39 $kml = "<?xml version='1.0' encoding='UTF-8'>
40 <kml xmlns='http://www.opengis.net/kml/2.2'><Document><Folder id=
41 \"Main Folder\"><name>Avg. speed $avgSpeed</name>";
42
43 $color = array
44 ("565D79", "2C5F9D", "3F81AD", "68B5C8", "96CEC5", "CCEEC0", "EEF5DF", "E9
45 F1AD", "E4E37A", "E6CC4C", "E5AB34", "E6760D", "CD7334");
46
47 $latMin = $obj->{'dataset'}->{'lat'}->{'min'};
48 $latMax = $obj->{'dataset'}->{'lat'}->{'max'};
49 $lonMin = $obj->{'dataset'}->{'lon'}->{'min'};
50 $lonMax = $obj->{'dataset'}->{'lon'}->{'max'};
51 $latStep = $obj->{'dataset'}->{'lat'}->{'step'};
52 $lonStep = $obj->{'dataset'}->{'lon'}->{'step'};
53 $iVal=0;
54

```

Graphic 3. PHP snippet receiving parameters from networkLink and processing JSON data into KML color coded polygons.

```

47 for ($i= $lonMin; $i <= $lonMax; $i += $lonStep) {
48     for ($j= $latMin; $j <= $latMax; $j += $latStep) {
49
50         $vs = $obj->{'dataset'}->{'measurables'}[0]->{'values'}
51         [$depthIndex][$iVal];
52         //echo "$iVal ($j,$i) $vs<br>";
53
54         $percVar = ($vs - $avgSpeed)/$avgSpeed;
55
56         if ($percVar <= -.07) {$colorIndex = $color[0];}
57         elseif (($percVar > -.07) && ($percVar <= -.055))
58         {$colorIndex = $color[1];}
59         elseif (($percVar > -.055) && ($percVar <= -.045))
60         {$colorIndex = $color[2];}
61         elseif (($percVar > -.045) && ($percVar <= -.035))
62         {$colorIndex = $color[3];}
63         elseif (($percVar > -.035) && ($percVar <= -.025))
64         {$colorIndex = $color[4];}
65         elseif (($percVar > -.025) && ($percVar <= -.015))
66         {$colorIndex = $color[5];}
67         elseif (($percVar > -.015) && ($percVar <= -.005))
68         {$colorIndex = $color[6];}
69         elseif (($percVar > -.005) && ($percVar <= .005))
70         {$colorIndex = $color[7];}
71         elseif (($percVar > .005) && ($percVar <= .015))
72         {$colorIndex = $color[8];}
73         elseif (($percVar > .015) && ($percVar <= .025))
74         {$colorIndex = $color[9];}
75         elseif (($percVar > .025) && ($percVar <= .035))
76         {$colorIndex = $color[10];}
77         elseif (($percVar > .035) && ($percVar <= .045))
78         {$colorIndex = $color[11];}
79         elseif (($percVar > .045) && ($percVar <= .055))
80         {$colorIndex = $color[12];}
81         $megaVs = $vs*1000;
82         $iVal++;
83         $boxLonA = $i;
84         $boxLatA = $j;
85         $boxLonB = $i + 1.99;
86         $boxLatB = $j;
87         $boxLonC = $i + 1.99;
88         $boxLatC = $j + 1.99;
89         $boxLonD = $i;
90         $boxLatD = $j + 1.99;
91         $kml .= "<Placemark><name>Lat: $j Lon: $i</
92         name><visibility>1</visibility><Style><PolyStyle><color>ff
93         $colorIndex</color><outline>0</outline><fill>1</fill></PolyStyle></
94         Style><Polygon><extrude>0</extrude><altitudeMode>clampedMode</
95         altitudeMode><outerBoundaryIs><LinearRing>
96         <coordinates>
97         $boxLonA,$boxLatA,$megaVs
98         $boxLonB,$boxLatB,$megaVs
99         $boxLonC,$boxLatC,$megaVs
100        $boxLonD,$boxLatD,$megaVs
101        $boxLonA,$boxLatA,$megaVs
102        </coordinates>
103        </LinearRing></outerBoundaryIs></Polygon>
104        </Placemark>";
105    }
106 }
107
108 $kml .="</Folder></Document></kml>";
109 print $kml;
110 ?>

```

Graphic 3. (Continued)

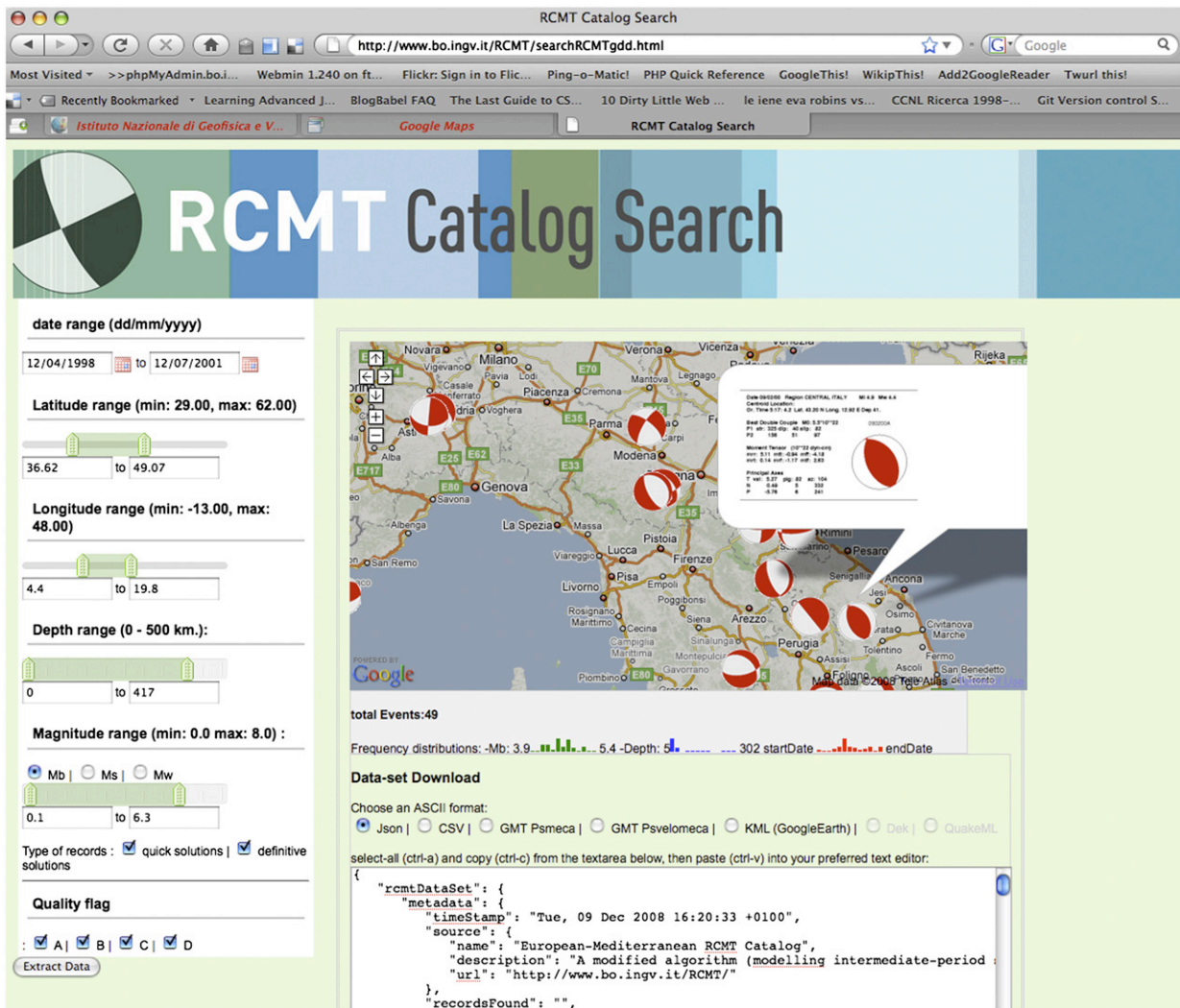


Fig. 2. Moment-tensor solution database search. Web page interface consists of a sidebar with an enhanced slide-bar form controls for setting search parameters, visual queues about resulting data set (dynamic map and in-line frequency distributions), and actual data set records, which can be freely and dynamically converted between several formats.

```

243 <Placemark>
244 <TimeStamp><when>2009-04-15T12:44:50Z</when></TimeStamp>
245 <name>M 2.5 - 2009-04-15</name>
246 <styleUrl>#orange</styleUrl>
247 <description><![CDATA[
248 <ul><li>Origin time: 2009-04-15 12:44:50.000</li>
249 <li>Depth: 9.6</li>
250 <li>Magnitude: 2.5</li>
251 <li>Source: SISBAS</li>
252 </ul>
253 ]]>
254 </description>
255 <Style><iconStyle><scale>0.3</scale></iconStyle></Style>
256 <Point><altitudeMode>absolute</altitudeMode>
257 <coordinates>13.501, 42.272</coordinates>
258 </Point>
259 </Placemark>
260

```

Graphic 4. KML snippet for a basic place-mark for RCMT KML layer.

provide as a standalone downloadable KMZ file.¹⁰ Indeed, this provides a less cluttered view of the sub-regional clusters of earthquakes.

6. Conclusions

The general approach outlined above for the use of JSON as a semantic data-exchange format for scientific data brings the 'web-as-a-platform' paradigm into the hands of researchers. As well as fitting in with the latest theoretical trends in computer

¹⁰ <http://www.bo.ingv.it/RCMT/fullDataSet.kmz>


```

8806 <Placemark>
8807 <TimeStamp><when>2009-03-30T13:38:38Z</when></TimeStamp>
8808 <name>M 4.0 - 2009-03-30</name>
8809 <description><![CDATA[
8810 <ul><li>Origin time: 2009-03-30 13:38:38.000</li>
8811 <li>Depth: 10.6</li>
8812 <li>Magnitude: 4.0</li>
8813 <li>Source: SISBAS</li>
8814 </ul>
8815 ]]>
8816 </description>
8817 <Model>
8818 <altitudeMode>absolute</altitudeMode>
8819 <Location>
8820 <longitude>13.362</longitude>
8821 <latitude>42.326</latitude>
8822 </Location>
8823 <Orientation>
8824 <heading>88</heading>
8825 <tilt>-40</tilt>
8826 <roll>56</roll>
8827 </Orientation>
8828 <Scale>
8829 <x>3.4</x>
8830 <y>3.4</y>
8831 <z>3.4</z>
8832 </Scale>
8833 <Link>
8834 <href>http://www.bo.ingv.it/assets/files/GE/redBeachBall.dae</href>
8835 </Link>
8836 </Model>
8837 </Placemark>

```

Graphic 5. KML snippet for an enhanced place-mark with importation of 3D-sphere models.

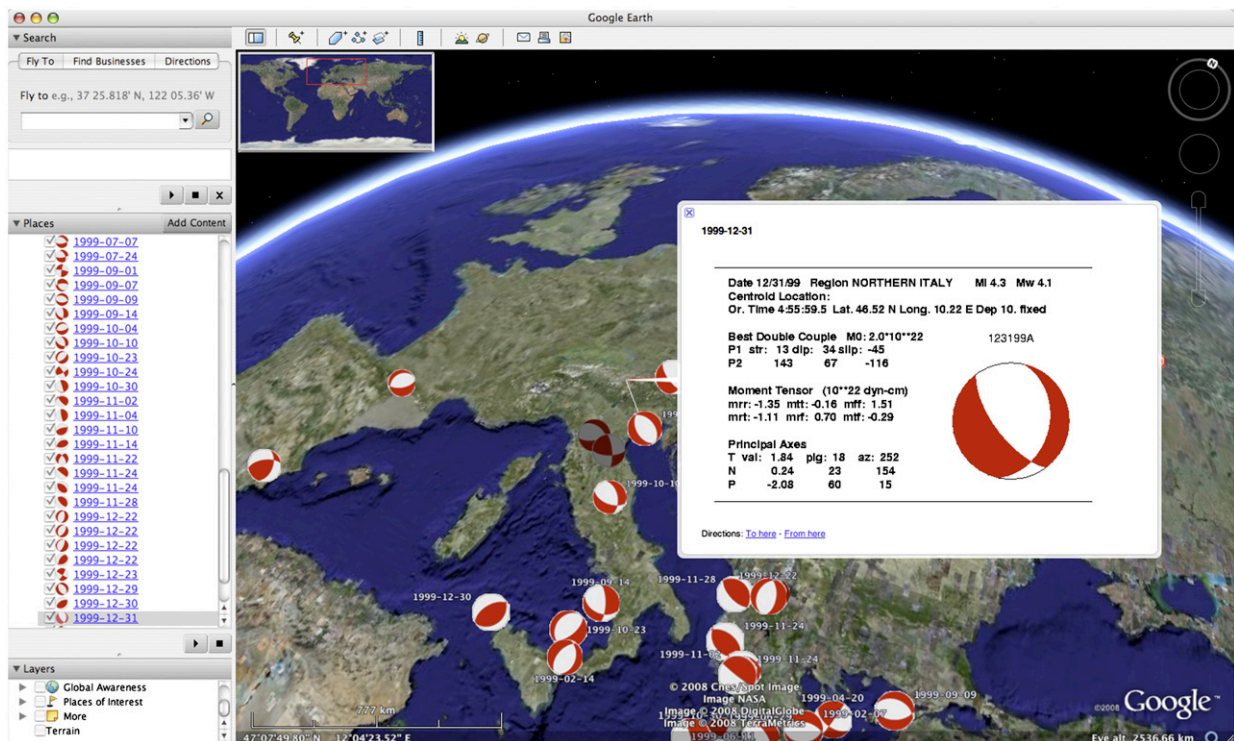


Fig. 3. Focal-mechanism visualization. D beach-ball representation of focal mechanisms is georeferenced in Google Earth in clampedToground mode. Tilted, semi-horizontal 3D views, such as that shown, would result in a poorly readable visualization of hypocenters if focal mechanisms were plotted at heights proportional to the original inverted depth of the event.

science, this has enormous practical advantages. Many online tools and advanced javascript frameworks are readily available to build interactive, cross-platform visualizations of data, with no need for users to install or configure anything. Google Earth is just

one of these options, and despite its current lack of support for negative values of altitudes, Google Earth still represents a very flexible 3D-rendering environment for the geosciences, as it provides good performance scaling when dealing with large data

sets. Also of note is the ease with which the geobrowser platform architecture overlays data sets from different sources, allowing users to carry out advanced integrations and comparisons.

Appendix A. Supporting material

Supplementary data associated with this article can be found in the online version at [doi:10.1016/j.ejcb.2009.08.001](https://doi.org/10.1016/j.ejcb.2009.08.001).

References

- Arvidsson, R., Ekström, G., 1998. Global CMT analysis of moderate earthquakes $M_w \geq 4.5$ using intermediate period surface waves. *Bulletin of the Seismological Society of America* 88, 1003–1013.
- Crockford, D., 2006. JSON: the fat-free alternative to XML. In: *Proceedings of the XML*, Boston, USA, 54–58, <http://www.json.org/fatfree.htm> (accessed 16.06.2010.).
- De Paor, D.G., 2007. Embedding collada models in geo-browser visualizations: a powerful tool for geological research and teaching. In: *Proceedings of the EOS Transactions of the American Geophysical Union, Fall Meeting*, San Francisco, CA, USA. IN32A-08, <http://adsabs.harvard.edu/abs/2007AGUFMIN32A.08D> (accessed 16.06.2010.).
- De Paor, D.G., 2008. Enhanced visualization of seismic focal mechanisms and centroid-moment tensors using solid models, surface bump-outs, and Google Earth. In: Declan De Paor (Ed.), *Google Earth Science, Journal of the Virtual Explorer* 29 (Electronic Edition), paper 2, doi:10.3809/jvirtex.2008.00195, <http://virtualexplorer.com.au/article/2008/195/seismic-model-visualization> (accessed 16.06.2010.).
- De Paor, D.G., Pinan-Llomas, A., 2006. Application of novel presentation techniques to a structural and metamorphic map of the pampean orogenic belt, NW Argentina. In: *Proceedings of the Geological Society of America Annual Meeting*, Philadelphia, PA, USA, 38(7), 131–12, http://gsa.confex.com/gsa/2006AM/finalprogram/abstract_112392.htm (accessed 16.06.2010.).
- De Paor, D.G., Williams, N.R., 2006. Solid Modeling of Moment-tensor Solutions and Temporal Aftershock Sequences for the Kiholo Bay Earthquake using Google Earth with a Surface Bump-out. *EOS Transactions of the American Geophysical Union, Fall Meeting*, San Francisco, CA, USA S53E-05.
- Earth API, 2008. Google Earth API, Google Incorporated, Mountain View, CA, USA. <http://code.google.com/apis/earth/> (accessed 16.06.2010.).
- Ekström, G., Dziewonski, A.M., Maternovskaya, N.N., Nettles, M., 2005. Global seismicity of 2003: centroid-moment tensor solutions for 1087 earthquakes. *Physics of the Earth and Planetary Interiors* 148, 327–351.
- EUROREM, 2008. European Seismological Reference Model, NERIES—Network of Research Infrastructures for European Seismology. Istituto Nazionale di Geofisica e Vulcanologia, Bologna, Italy <<http://www.bo.ingv.it/eurorem/>> (accessed 16.06.2010.).
- GIAP, 2007. GEON IDV, UCAR, Boulder, CO, USA, <http://geon.unavco.org/unavco/IDV_for_GEON_GIAP_format.html> (accessed 16.06.2010.).
- GMT, 1988. The Generic Mapping Tools, SOEST. University of Hawaii, Honolulu, HI, USA (accessed 16.06.2010.).
- HDF5, 2010. HDF Group, University of Illinois Research Park, Champaign, IL, USA, <http://www.hdfgroup.org/HDF5/>, (accessed 16.06.2010.).
- IDV, 2007. The GEON Integrated Data Viewer, UCAR, Boulder, CO, USA, <<http://geon.unavco.org/>> (accessed 16.06.2010.).
- Labay, K.A., Haeussler, P. J., 2007. 3D visualization of earthquake focal mechanisms using arcscene. *United States Geological Survey Data Series* 241 (1.1), USGS, Reston, VI, 17pp, <<http://pubs.usgs.gov/ds/2007/241/ds-241.pdf>> (accessed 16.06.2010.).
- Lawrence, R., 2004. The space efficiency of XML. *Information and Software Technology* 46, 753–759.
- NETCDF, 2007. Network Common Data Form, UCAR, Boulder, CO, USA, <<http://www.unidata.ucar.edu/software/netcdf/>> (accessed 16.06.2010.).
- Pondrelli, S., Salimbeni, S., Morelli, A., Ekstrom, G., Boschi, E., 2007. European-Mediterranean regional centroid-moment tensor catalog: solutions for years 2003 and 2004. *Physics of the Earth and Planetary Interiors* 164, 90–112.
- Pondrelli, S., Salimbeni, S., Morelli, A., Ekstrom, G., Postpischl, L., 2008. 1997–2008: 11 years of European-Mediterranean regional centroid-moment tensors and their dissemination. In: *Proceedings of the Eos Transactions of the American Geophysical Union, Fall Meeting Supplement*, San Francisco, CA, USA, S43D-1903, <<http://www.bo.ingv.it/RCMT/Posters/RCMT-json-AGU08.pdf>> (accessed 16.06.2010.).
- Postpischl, L., Danecek, P., Morelli, A., 2008. A standard format to describe Earth models and improve their dissemination. In: *Proceedings of the Eos Transactions of the American Geophysical Union, Fall Meeting Supplement*, San Francisco, CA, USA. IN33A-1159, <<http://www.bo.ingv.it/RCMT/Posters/NERIES-json-AGU08.pdf>> (accessed 16.06.2010.).
- QuakeML, 2007. Swiss Seismological Service, ETH, Zurich, Switzerland, <<https://quake.ethz.ch/quakeml/>> (accessed 16.06.2010.).
- Richardson, L., Ruby, S., 2007. RESTful Web Services. O'Reilly Media, Sebastopol, CA 446pp.
- Sketchup, 2007. Google SketchUp, Google Incorporated, Mountain View, CA, USA, <<http://sketchup.google.com>> (accessed 16.06.2010.).
- Tufte, E., 2006. *Beautiful Evidence*, Graphics Press, Cheshire, CT 213pp.
- Yamagishi, Y., Yanaka, H., Suzuki, K., Tamura, H., Nagao, H., Tsuboi, S., 2008. Visualization of geoscience data on Google Earth: development of data converter system for seismic tomography models, geochemical data of rocks, and geomagnetic field models. In: *Proceedings of the Eos Transactions of the American Geophysical Union, Fall Meeting Supplement*, San Francisco, CA, USA. IN41B-1151.