XBRL analytics with OIM

Mark Goodhand

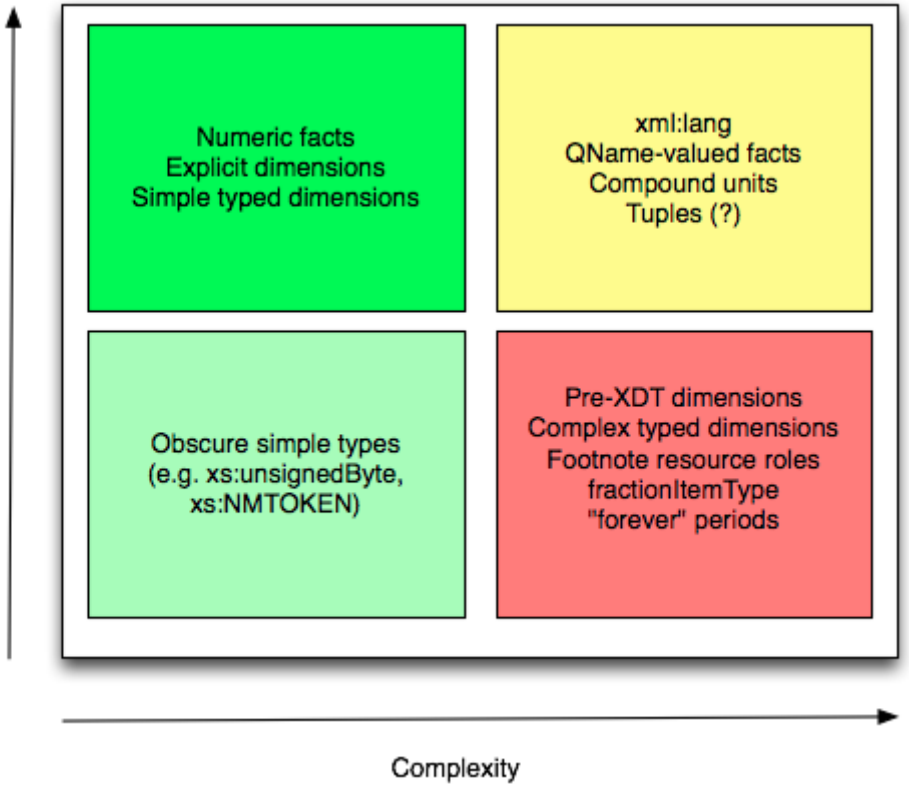
2017-06-08

Why OIM?

- Many details of the original XML representation are not semantically significant
  - Context ids
  - Segment vs Scenario
  - Ordering of facts and dimensions
- XML is not always the most convenient syntax
  - JSON better for web; CSV better for bulk

OIM simplifies

- XBRL is 14 years old
- Some features are more trouble than they're worth



Unsupported features

- 2.1.1 Non-dimensional segment/scenario content
- 2.1.2 Mixing segment and scenario elements
- 2.1.3 Complex-typed dimensions
- 2.1.4/2.15 Unsupported data types (fraction, float, double)
- 2.1.6 non-standard footnote resource roles
- 2.1.7 Use of zero-precision numeric facts

See *OIM CR-2017-05-02 §2.1: Constraints*

Unsupported features

- Custom attributes on facts (e.g. @find:filed)

```
<xs:element name="filingIndicator" id="find_filingIndicator"
            type="xbrli:stringItemType" substitutionGroup="xbrli:item"
            nillable="true" xbrli:periodType="instant"/>
<xs:attribute default="true" name="filed" type="xs:boolean"/>
```

  - Eurofiling-2016-05-11/www.eurofiling.info/eu/fr/xbrl/ext/filing-indicators.xsd
- Agreed yesterday to pursue new filing indicator representation, subject to EBA/EIOPA blessing

  See *OIM CR-2017-05-02 §2.2:* Other unsupported features

Supported features

- Tuples (still)
  - xbrl:tupleParent
  - xbrl:tupleOrder
- Forever periods (NEW)
  - Campbell Pryde: “Just when you thought that no one uses forever we have a taxonomy where we need it all over the place.”

OIM JSON

- Most effort in the working group has focused on the core model and the JSON representation
- Many syntax variations were considered

OIM JSON - 2016-09-29

- Object for period

```
"xbrl:period":  {
    "start": "2015-01-01T00:00:00",
    "end": "2016-01-01T00:00:00"
}
```

- Object for taxonomy-defined aspects

```
"tax:RegionDimension": {
        "value": "tax:Europe",
        "baseType": "QName"
}
```

OIM JSON – desirable properties

- Uniform representation for aspects (QName key, string value)
- Space for inline type annotations (“augmentations”)
- Statically determinable datatype at a given JSON path
- “Simple”

OIM JSON - CR-2017-05-02


```
{ "id": "f923",
   "value": "1234",
   "aspects": {
      "xbrl:concept": "tax:NumericConcept",
      "xbrl:entity": "cid:123456789",
      "xbrl:periodStart": "2015-01-01T00:00:00",
      "xbrl:periodEnd": "2016-01-01T00:00:00",
      "xbrl:unit": "iso4217:GBP",
      "tax:RegionDimension": "tax:Europe"
   }
}
```

JSON defaults

- Current
  - Absent periodStart & periodEnd means *forever*
  - Absent accuracy means *INF*
- Proposed
  - Absent units means *xbrli:pure* [bug 620]
  - Absent entity means “nobody” [bug 619]

OIM CSV

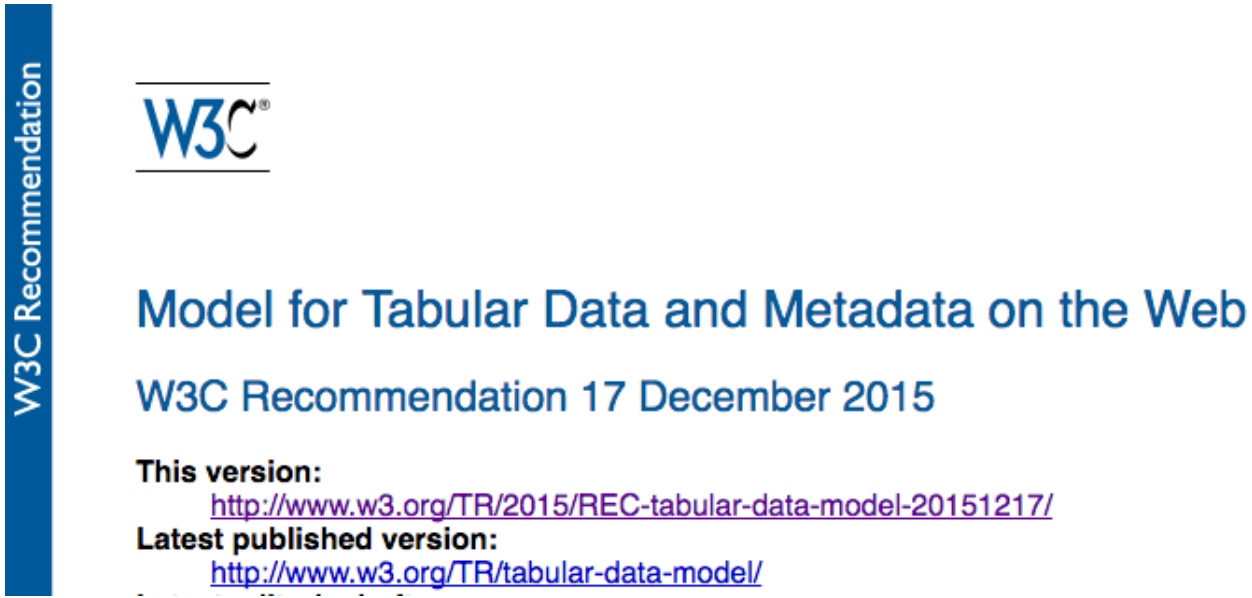
- First PWD published 2017-05-02
- Significant reductions in file size

					measured in kB					
Number of facts	explicit	1 typed dim	2 typed dim		XBRL	CSV	metadata	total		reduction
6884	3	2942	3939		2722	176	12	188		14,5
87651	5	27446	60200		41621	1918	19	1937		21,5

- Well suited to production from backend databases

OIM CSV

- Based on W3C's Model for Tabular Data and Metadata on the Web



OIM RDF?

Semantic Web

XLink

Why RDF?

- Syntax independent model
  - JSON LD
  - CSVW
- Highly expressive
- SPARQL
- OWL
- More widely supported than XLink
- Supports XML Schema data types

OIM RDF?

Reification

Ontologies

Blank nodes

N-Quads

N-Triples

TriG

RDFa

IRIs

Turtle

Why not RDF (yet)

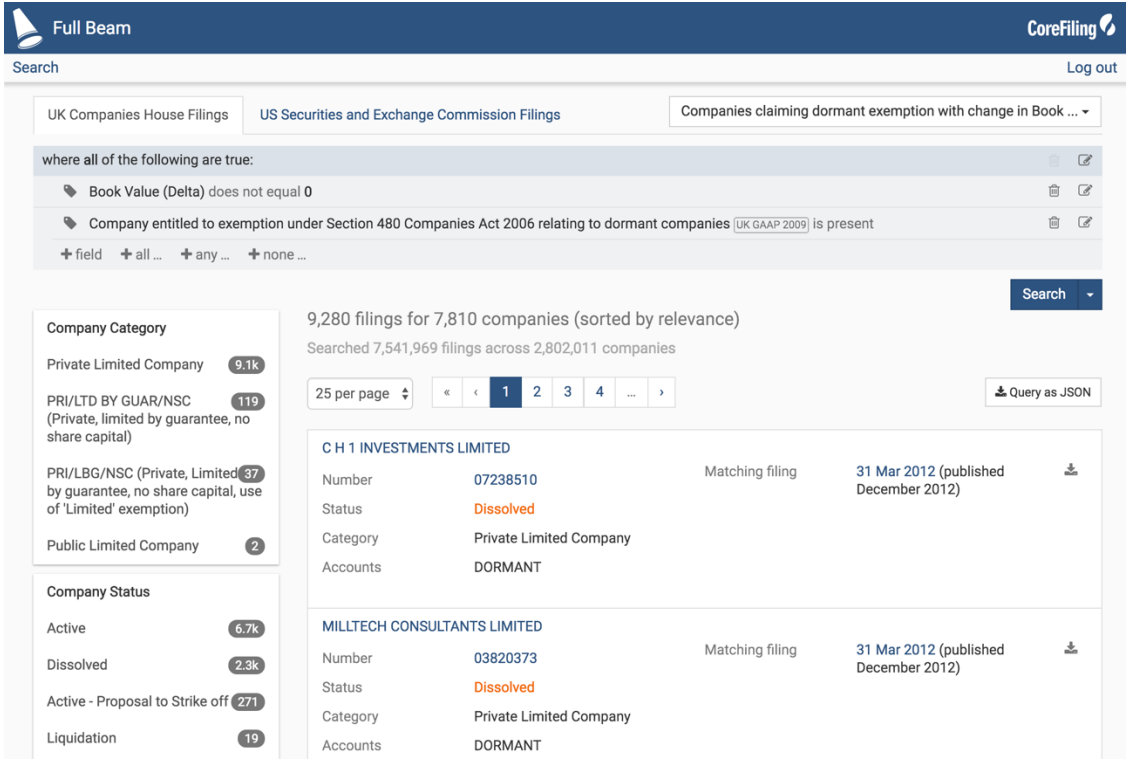
- No demand for RDF from regulators or filers
- Desire to publish something simple ASAP
- Lack of RDF expertise within working group

Reasons to revisit

- Taxonomy side of OIM not yet defined
- RDF can represent both instance and taxonomy
- JSON-LD helps to hide complexity
- Related technologies already mapped (SDMX)
- Interest in Semantic Web increasing?

OIM in production

- XBRL and iXBRL transformed into OIM JSON
- Augmented
- Indexed in Elasticsearch
- Initially built using 2016-01-03 PWD
- 7,541,969 UK filings
- 180,728 US filings



OIM potential: lightweight, open BI

- Taxonomies are standard, open meta-models
  - Reuse data cubes rather than redefining
- OIM datasets are semantic and easily query-able
  - NoSQL databases cope with size and dimensionality
  - REST APIs support querying by aspects
- Web components for visualisation

Horizontal scaling and wide columns

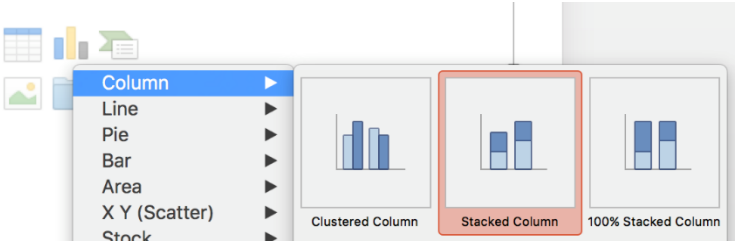
Google BigTable	Apache Cassandra	Apache HBase
“sparsely populated NoSQL database which can scale to billions of rows, thousands of columns, and petabytes of data”	“Some of the largest production deployments include Apple's, with over 75,000 nodes storing over 10 PB of data, Netflix (2,500 nodes, 420 TB, over 1 trillion requests per day”	“This project's goal is the hosting of very large tables -- billions of rows X millions of columns -- atop clusters of commodity hardware”
+ BigQuery	+ Spark SQL	+ Phoenix / Hive / Impala

OIM queries

- OIM-based queries insulate users from database representation

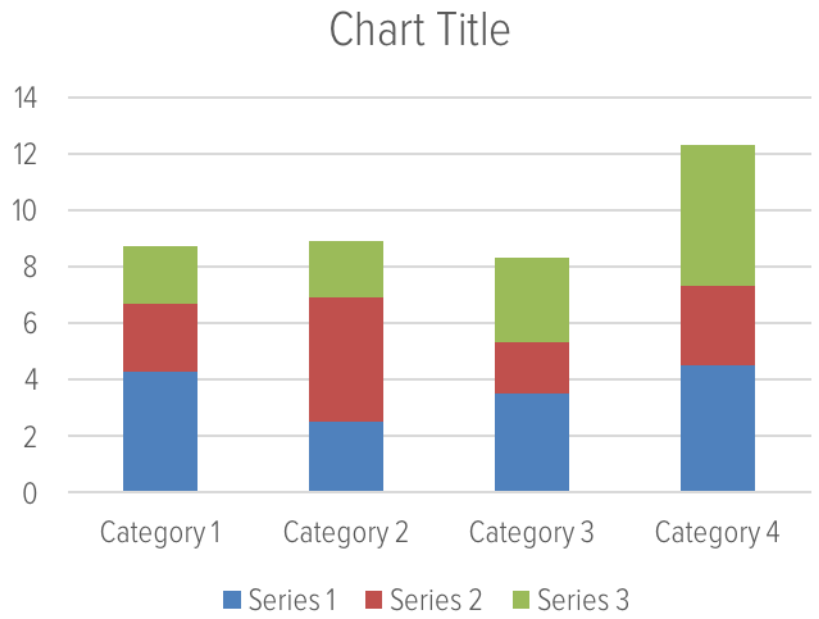
```
{
    "prefixes": {
        "my": "http://taxonomies.corefiling.com/mrg/samples/mgmt-reports"
    },
    "aspects": {
        "xbrl:concept": "my:Revenue",
        "xbrl:periodStart": "2015-01-01T00:00:00",
        "xbrl:periodEnd": "2015-12-31T00:00:00"
    }
}
```

Charts the old fashioned way



A6

	A	B	C	D
1		Series 1	Series 2	Series 3
2	Category 1	4.3	2.4	2
3	Category 2	2.5	4.4	2
4	Category 3	3.5	1.8	3
5	Category 4	4.5	2.8	5
6				
7				



Charts on the web

Google Charts

```
google.charts.load('current', {packages: ['corechart', 'bar']});
google.charts.setOnLoadCallback(drawMultSeries);

function drawMultSeries() {
      var data = google.visualization.arrayToDataTable([
        ['City', '2010 Population', '2000 Population'],
        ['New York City, NY', 8175000, 8008000],
        ['Los Angeles, CA', 3792000, 3694000],
        ['Chicago, IL', 2695000, 2896000],
        ['Houston, TX', 2099000, 1953000],
        ['Philadelphia, PA', 1526000, 1517000]
      ]);

      var options = {
        title: 'Population of Largest U.S. Cities',
        chartArea: {width: '50%'},
        hAxis: {
          title: 'Total Population',
          minValue: 0
        },
        vAxis: {
          title: 'City'
        }
      };

      var chart = new
google.visualization.BarChart(document.getElementById('chart_div'));
      chart.draw(data, options);
    }
```

C3

```
chart = c3.generate({
bindto: '#chart',
data: {
  columns: [
    ['data1', 30, 200, 100, 400, 150, 250],
    ['data2', 50, 20, 10, 40, 15, 25]
  ],
  axes: {
    data2: 'y2'
  }
},
axis: {
  y: {
    label: { // ADD
      text: 'Y Label',
      position: 'outer-middle'
    }
  },
  y2: {
    show: true,
    label: { // ADD
      text: 'Y2 Label',
      position: 'outer-middle'
    }
  }
}
```

Chart.js

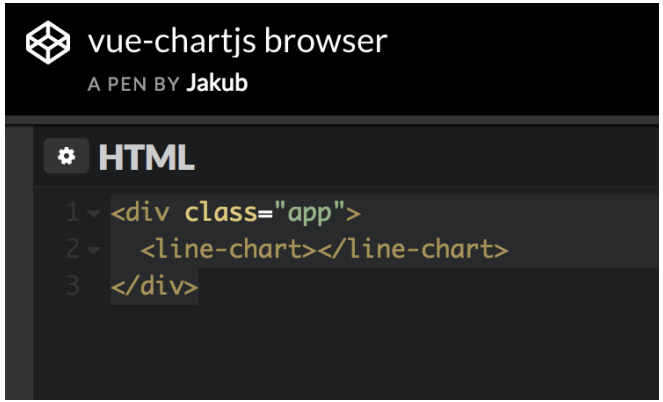
```
var ctx = document.getElementById('myChart').getContext('2d');
var chart = new Chart(ctx, {
    // The type of chart we want to create
    type: 'line',

    // The data for our dataset
    data: {
        labels: ["January", "February", "March", "April", "May"
        datasets: [{
            label: "My First dataset",
            backgroundColor: 'rgb(255, 99, 132)',
            borderColor: 'rgb(255, 99, 132)',
            data: [0, 10, 5, 2, 20, 30, 45],
        }]
    },

    // Configuration options go here
    options: {}
});
```

Charts as componen

[vue-chartjs](#)



[google-chart](#)

```
<google-chart
  type='pie'
  options='{"title": "Distribution of days in 2001Q1"}'
  cols='[{"label":"Month", "type":"string"}, {"label":"Days", "type":"number"}]'
  rows='[["Jan", 31],["Feb", 28],["Mar", 31]]'>
</google-chart>
```

```
ReactDOM.render(
  <BarChart
    title= {title}
    data= {generalChartData}
    width= {width}
    height= {height}
    chartSeries = {chartSeries}
    x= {x}
    xLabel= {xLabel}
    xScale= {xScale}
    yTicks= {yTicks}
    yLabel = {yLabel}
  />
, document.getElementById('data_bar')
)
```

[react-d3](#)

OIM charts

- Labels drawn from taxonomy
- Values and dimensions from OIM JSON
- No need to bend data into shape
- Can be used with individual documents or datasets
- Embeddable in Markdown and HTML

OIM charts

```
<oim-chart
  type='pie'
  title='Revenue by country, 2015'
  dataset='https://oim.corefiling.com/mrg/samples/mgmt-reports'
  filter='{ "aspects": { "xbrl:concept": "my:Revenue",
                         "xbrl:periodStart": "2015-01-01T00:00:00",
                         "xbrl:periodEnd": "2015-12-01T00:00:00"} }'
  group='my:Region'>
</oim-chart>
```

The future is soon

- We'll be opening up OIM JSON APIs in the next few weeks
- Expect a growing set of free tools for exploring the power of OIM
- Get in touch:
  - mrg@corefiling.com
  - @MarkGoodhand