

Eurofiling XBRL 2018 Week

10th anniversary of
XBRL EUROPE



LIGHTWEIGHT BI WITH OIM DASHBOARDS

Mark Goodhand
CoreFiling

Warsaw,



CORPORATE DASHBOARDS

- View metrics from several data sources at a glance
- Charts beats grids of numbers
 - Trends and outliers immediately apparent
- Screens are cheap
 - Show data in every corner of the office
 - [Big Visible Charts](#)
- Don't require data warehouses



PERFORMANCE AND RESULTS

- KPI = Key Performance Indicator
 - Not financial in nature
 - Often leading or “forward-looking”
 - Understood by all members of the staff
 - Actionable
 - Measured frequently

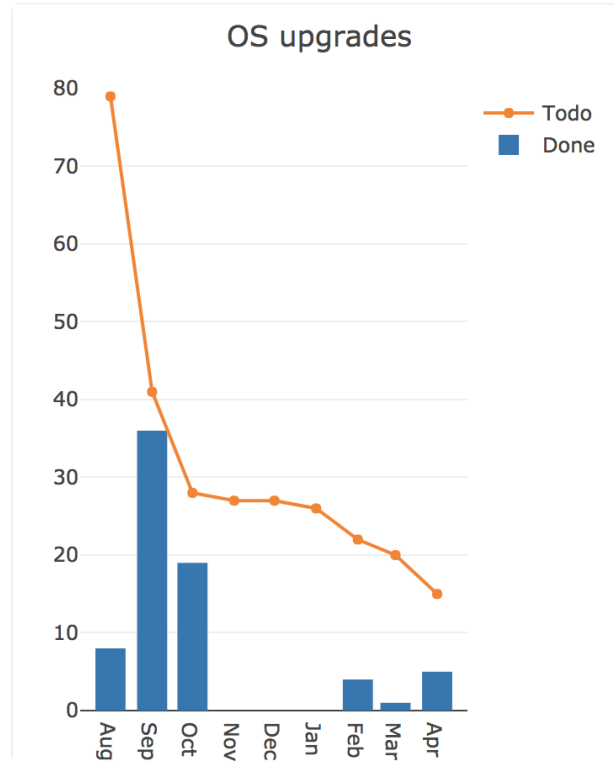
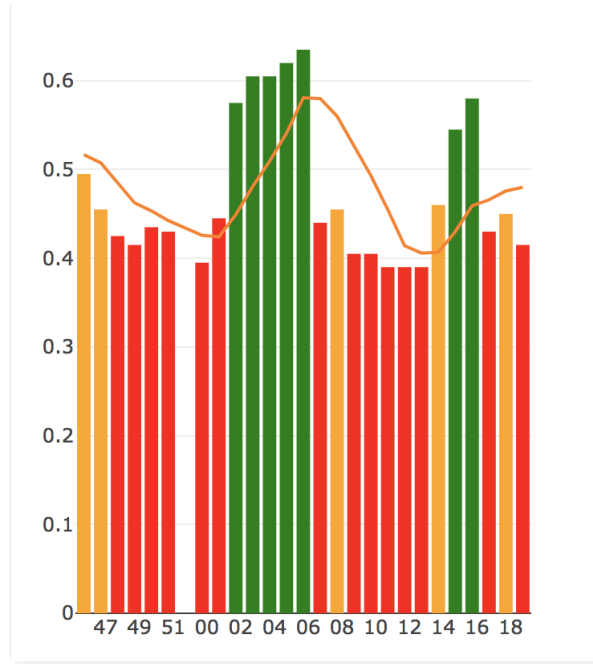
- KRI = Key Result Indicator
 - Frequently financial in nature
 - Often lagging or “backwards-looking”
 - Of great interest to shareholders
 - Not easily actionable

Source: [KPI vs. KRI - The Difference and the Importance](#)

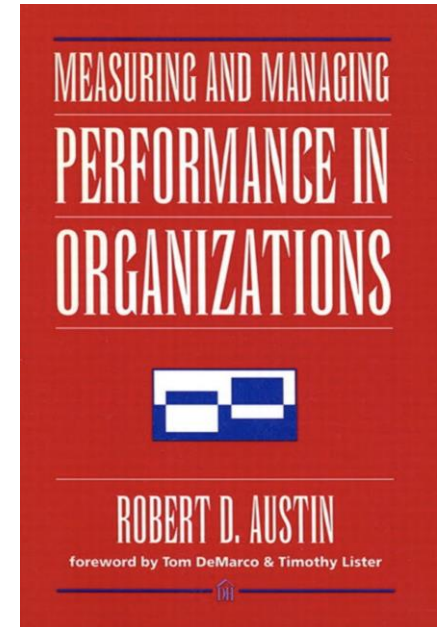


INFORMATIONAL & MOTIVATIONAL

- Charts can inform and motivate



- Though beware of dysfunction ...
 - Metrics intended purely to inform can motivate in unexpected ways
 - Humans respond to incentives





DIMENSIONAL DATA

- Dashboard data is dimensional
 - Leads per salesperson
 - Revenue by region
 - Bugs by product
 - Headcount over time



THE RISE OF THE API



GitLab



stripe



DERIVED DATA

- SUM, AVG, COUNT by period
- Moving averages
- Ratios and percentages
- Deltas



FREE, OPEN SOURCE FRAMEWORKS

Dashing 

freeboard

 Dash
by plotly



SOME CONCERNS

- Collect and transform for one-off display?
- Calculation logic in the dashboard?
- What if a source system is down?
- Is the data valid?
- Is the data sane?
- When will errors become apparent?
- Can I see the raw numbers?
- Can I comment on the data?
- Can I use the data in other contexts?
- Are known-good snapshots of the data available?



XBRL AND AUTOMATION

- Decouple data retrieval and derivation from display
- Catch bad data before it reaches the dashboard
- Save good data in a versioned data store
- Write business rules to flag potential concerns
- Derive data using formulas
- Use a standard, common format (not source-specific JSON/CSV)
- Take advantage of features in COTS software:
 - Table views
 - Search
 - Commenting
 - Comparison
 - Export to other formats



Dashboard data

✓ OK

Validation profile: Web access
Created: 24 May 2018 10:10
Uploaded by: Automated upload

New revision | Export as | Edit | Delete

Contents 5 | Validation 0 | History 91

- Invoiced revenue
- New business
- Renewals
- Sales leads
- Sales pipeline

Other accumulated results [member]

€148,063,000.00

148063000.0	
Concept	Profit (loss)
Dimensions	(1 non-defaulted)
Entity	529900SCZKCX0WMOCC24
Period	2016-01-01 to 2016-12-31
Value	148063000.0
Decimals	0
Unit	EUR
iXBRL	Scale: ×1,000

Validation | Discussions

Is this number right?

21/4000

Cancel Discuss

Contents 5 | Validation 0 | History 91

- ✓ Specification Conformance
 - ✓ XML and XML Schema
 - ✓ XBRL Specification
- ▶ XBRL Consistency
 - ✓ XBRL Dimensions
 - ✓ XBRL Generic Links, Labels and References
 - ✓ XBRL Table Linkbase
 - ✓ XBRL Formula

All issues

✓ - ⚠ - ✖ - 🚫 -

No validation messages to display



WHY NOT XBRL?

- Taxonomies are a pain to produce
- Instances are a pain to consume



TAXONOMY GENERATION

- Create taxonomy definition from the data
- Refine the inferred data types
- Enrich with labels, formulas and rules
- Invoke API to generate, validate and publish taxonomy



XML INSTANCE DATA

```
<dei:EntityCommonStockSharesOutstanding  
id="Fact-B90BB051582C5EE9E2AD8C6C79A5CE80"  
contextRef="context_2"  
decimals="INF"  
unitRef="unit">348952225  
</dei:EntityCommonStockSharesOutstanding>
```

```
<xbrli:unit id="unit">  
<xbrli:measure>xbrli:shares  
</xbrli:measure>  
</xbrli:unit>
```

```
<xbrli:context id="context_2">  
  <xbrli:entity>  
    <xbrli:identifier scheme="http://www.sec.gov/CIK">0001652044</xbrli:identifier>  
    <xbrli:segment>  
      <xbrldi:explicitMember dimension="us-  
gaap:StatementClassOfStockAxis">goog:CapitalClassCMember</xbrldi:explicitMember>  
    </xbrli:segment>  
  </xbrli:entity>  
<xbrli:period>  
  <xbrli:instant>2018-04-18</xbrli:instant>  
</xbrli:period>  
</xbrli:context>
```



OIM INSTANCE DATA

```
{  
  "id": "Fact-B90BB051582C5EE9E2AD8C6C79A5CE80",  
  "value": "348952225",  
  "aspects": {  
    "xbrl:concept": "dei:EntityCommonStockSharesOutstanding",  
    "xbrl:entity":  
"cik:0001652044",  
    "xbrl:periodStart": "2018-04-19T00:00:00",  
    "xbrl:periodEnd": "2018-04-19T00:00:00",  
    "xbrl:unit": "xbrli:shares",  
    "us-gaap:StatementClassOfStockAxis": "goog:CapitalClassCMember"  
  }  
}
```

<https://github.com/mgoodhand/oim-samples>



OIM JSON BENEFITS

- Context and unit information is immediately available
- All aspects represented consistently
- 1:1 namespace-prefix bindings, at the top of the document
- Reserved prefix for built-in aspects
- Trivial deserialisation
- Simple, flat structure



WORKING WITH OIM JSON

```
static taxonomyDefinedAspects(facts) {  
  return this.aspects(facts).filter(a => !a.startsWith("xbrl:"))  
}  
  
static coreAspects(facts) {  
  return this.aspects(facts).filter(a => a.startsWith("xbrl:"))  
}
```



IMMUTABLE

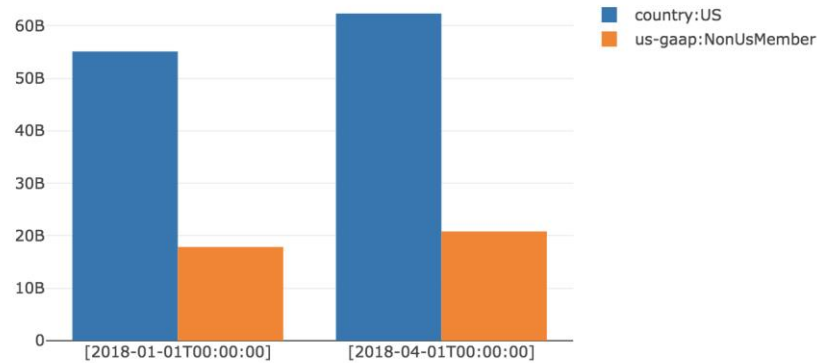
```
static aspectMatch(fact, filter) {  
  return Map(filter).every((aspectValue, aspect) => this.factHasMatchingAspect(fact, aspect, aspectValue))  
}
```

```
series(facts, label) {  
  const sortedFacts = List(facts).sortBy(f => f.aspects["xbrl:periodEnd"]).toArray()  
  return {  
    x : sortedFacts.map(f => this.formatPeriod(f.aspects["xbrl:periodStart"], f.aspects["xbrl:periodEnd"])),  
    y : sortedFacts.map(f => f.value),  
    type: 'bar',  
    name: label  
  }  
}
```



SAMPLE APP

xbrl:concept = us-gaap:AssetsNoncurrent
xbrl:entity = ns0:0001652044
xbrl:unit = iso4217:USD



us-gaap:StatementGeographicalAxis	xbrl:periodEnd	xbrl:periodStart	Value
country:US	2018-01-01T00:00:00	2018-01-01T00:00:00	55113000000
us-gaap:NonUsMember	2018-01-01T00:00:00	2018-01-01T00:00:00	17874000000
country:US	2018-04-01T00:00:00	2018-04-01T00:00:00	62347000000
us-gaap:NonUsMember	2018-04-01T00:00:00	2018-04-01T00:00:00	20827000000

mgoodhand / oim-chart-demo

Code

Issues 0

Pull requests 0

```
class App extends Component {
  render() {
    return (
      <div className="App">
        <header className="App-header">
          <img src={logo} className="App-logo" alt="logo" />
          <h1 className="App-title">OIM Charts</h1>
        </header>
        <ReportLoader src="https://raw.githubusercontent.com/mgoodhand/oim-samples/main"
          filters={{}} />
      </div>
    );
  }
}
```

<https://github.com/mgoodhand/oim-chart-demo>



UPCOMING CHANGES

- Tuples forbidden
- Ids required (generated if necessary)
- Facts keyed by id (unordered but easy to access)
- roleRef and arcroleRef handled automatically
- linkbaseRef forbidden
- Footnotes represented as facts
 - consistent treatment of fact-footnote and fact-explanatoryFact
- Entity is optional
- Unit is optional



QUESTIONS?

Mark Goodhand

Head of Research, CoreFiling

mrg@corefiling.com