The Software Ontology (SWO):
A Case Study in Agile Ontology Engineering Method

James Malone, Maria Copeland, Andy Brown, Helen Parkinson, Robert Stevens
Motivation - The Software Ontology Project
http://theswo.sourceforge.net/

- 6-month Community driven ontology (with some extra recently)
- Captures formal descriptions of software used in:
  - The production of data
  - The analysis of data
  - For curation and preservation
- To promote:
  - Standardisation of knowledge
  - Reusability of knowledge
There is a LOT of stuff
Where do we start?

Amount of Stuff

Loss of sanity

Stuff
Bio-ontologies

- Many bio-ontologies similarly have a lot of potential ‘stuff’
- Many bio-ontologies are touted as ‘community driven’ or collaborative
- The process has challenges and is never really complete;

Requirements change (grants, trends, data shift)

Engaging with diverse set of current and future users

The underlying science change

Ontologies can be scary
Bio-ontology and software methodologies

- Up front requirements have known problems
- Users change their mind because their needs change
- Evaluation at end means little can be done to fix things

Sure, the Higgs boson is great but where is the cup holder?

Great, here’s $100 millions, I can’t wait to see the product
Our Agile Ontology Engineering method

- Modern methods use Agile development
- Iterations are short and often
- Test driven: each iteration is evaluated with user
- Requirements and priorities can change between iterations
- Teams are self-organising
- Talk to user in user language not ontology speak

Mr Ontologist, can you add word processing please?

Of course user, I’ll add a disjoint and a covering axiom about a necessary realizable entity which will allow inferred subsumption

Hmm maybe we should ask Siri?
Flow of Events

- Feature Request
- Competency Questions
- Requirements
- Priority Poker
- Implementation of top requirements
- Modular Development
- Assemble using Reasoning
- Testing with Defined Classes
- Evaluation
- Release

Compositional Approach
Requirements – General Competency questions
Requirements – General Competency questions

- What software works best with my dataset?
- Does it do what I want or need it do e.g. render a gif?
- Which software tool created this data?
- What software can perform task x?
- What are the primary inputs and outputs?
- Is this software available as a web service?
- What open source, maintained software can I use to process these in this format?
- Where can I get the software?
- Is there a mailing list?

- http://goo.gl/YLIjz
Priority poker

- Used to reach consensus on effort of adding feature
- Features derived from competency questions e.g.
  - Software, Data input and output, License, Architecture
- Each person estimates effort
- Discrepancies are discussed and a revote taken
- Users and developers participate
- Ontologies are not mentioned at all (and sadly no cigars)
Requirements priority “buy a feature”

- Poker gives us ‘cost’ for each feature
- Users given ‘money’ to spend as fraction of total cost
- Most features too expensive for one person to buy – requires multiple stakeholders jointly bidding
- Finish with list of top priority requirements
Knowledge Acquisition

- Multiple ways to suit different user
- Submit term requests via tracker
- Survey
- Preferred method was spreadsheet using Populous
- Populous structures knowledge so it can be readily added to ontology
- Columns auto-complete on existing parts of ontologies
- Looks like normal spreadsheet
### Eliciting Knowledge – Populous

http://www.populous.org.uk

<table>
<thead>
<tr>
<th>Software Name</th>
<th>Licence e.g. CC-BY</th>
<th>SW Version e.g. 1.0</th>
<th>Organization e.g. Supplier</th>
<th>Algorithm E.g. RMA</th>
<th>Data Format e.g. MAGE-ML</th>
<th>Maturity -</th>
<th>Software Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe Acrobat Reader</td>
<td>proprietary</td>
<td>10.1 for mac</td>
<td>Adobe</td>
<td>does not matter</td>
<td>pdf</td>
<td>stable, maintained</td>
<td>GUI</td>
</tr>
<tr>
<td>Annotare</td>
<td>MIT</td>
<td>1</td>
<td>Stanford University</td>
<td>n/a</td>
<td>MAGE-TAB</td>
<td>mature, latest release</td>
<td>GUI</td>
</tr>
<tr>
<td>ArrayExpress Restful API</td>
<td></td>
<td></td>
<td>EBI</td>
<td>n/a</td>
<td>XML</td>
<td>mature, latest release</td>
<td>Restful service</td>
</tr>
<tr>
<td>BioJava</td>
<td>LGPL v2.1</td>
<td>3.0.1</td>
<td>BioJava</td>
<td></td>
<td></td>
<td>stable, maintained</td>
<td>API</td>
</tr>
<tr>
<td>BioPerl</td>
<td>Perl Artistic Licence</td>
<td>1.6.9</td>
<td>BioPerl</td>
<td></td>
<td></td>
<td>stable, maintained</td>
<td>API</td>
</tr>
<tr>
<td>DNDC</td>
<td>unknown</td>
<td>not recorded</td>
<td>Changshend Li, University Of New Hampshire</td>
<td>unknown</td>
<td>tab delimited flat file</td>
<td>Stable, maintained</td>
<td>GUI, command line</td>
</tr>
<tr>
<td>DROID (Digital Record Object Identification)</td>
<td>The BSD 2-Clause License</td>
<td>6.02</td>
<td>The National Archives</td>
<td>DROID byte pattern matching algorithm</td>
<td>Jar files,</td>
<td>Stable, not maintained, support provided where possible (politically complex!)</td>
<td>CLI, GUI</td>
</tr>
<tr>
<td>Dropbox</td>
<td>specific EULA</td>
<td>1.1.3.5 for windows</td>
<td>dropbox</td>
<td>n/a</td>
<td>files, API</td>
<td>stable, maintained</td>
<td>GUI, API</td>
</tr>
</tbody>
</table>
Requirements priorities changed

• In workshop one “hardware platform” was not prioritised
• In workshop two it was
• Some features always bought, e.g. data format
• Working through examples helped
• Add some “doing” to the “thinking”…. 

Forget everything I said last month, I want the newest new new new one. What do you mean they all look the same?
Results

• Six month project (plus some recent time), 1,000 classes
• Lots of axiomatisation as per prioritised requirements
• Tested using competency questions
• Three workshops with stakeholders to extract competency questions, requirements and priorities
• 80% of stakeholders had little or no ontology experience
Testing Results

- Test driven using defined classes

<table>
<thead>
<tr>
<th>Feature</th>
<th>Competency Question</th>
<th>Manchester OWL Test Question</th>
<th>Example Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Which software has MAGE-tab input?</td>
<td>has specified input some (data and has format specification some ‘MAGE tab format’)</td>
<td>ArrayExpress, Bioconductor</td>
</tr>
<tr>
<td>Function</td>
<td>What software performs sequencing analysis?</td>
<td>achieves objective some ‘molecular sequencing analysis’</td>
<td>EMBOSS</td>
</tr>
<tr>
<td>Version</td>
<td>Which version of Microsoft Excel came after 2007?</td>
<td>‘Microsoft Excel’ and (has version some (‘version name or number’ and (preceeded by value ‘Microsoft 2007 version’)))</td>
<td>Microsoft Excel 2010</td>
</tr>
</tbody>
</table>
Ontology Modules and EDAM

- Algorithms
- Data
- Objectives
- Interfaces
- Maturity
- Versions
- Licenses
- Organizations
Conclusions

• Attendees *enjoyed* experience and reusing method
• Democratic; everyone equal say in requirements
• Process is open and transparent
• Clear process for conflict resolution
• Competency questions gave us test cases
• Life sciences vast scope – prioritising provides cost benefit
• *Community driven ontologies* need users more than they need our ontologies; listen to them
Conclusions

- One user said: “Has all the merits and flaws of a democracy – e.g. some times hard decisions are needed that group aren’t willing to take”
- Priorities could change vastly at each iteration…
- …but at each iteration something is delivered according to requirements.
- Given rapid changes in life sciences, bio-ontology needs to be responsive
- And given ontology development money is limited, prioritising reqs is critical
Acknowledgements

All who participated in the SWO workshops

Allyson Lister, Duncan Hull and Jon Ison (recent work on SWO and EDAM integration)

Funding from:
  JISC
  EMBL
  University of Manchester