Methodology and Campaign Design for the Evaluation of Semantic Search Tools

Stuart N. Wrigley[1], Dorothee Reinhard[2], Khadija Elbedweihy[1], Abraham Bernstein[2], Fabio Ciravegna[1]

[1]University of Sheffield, UK
[2]University of Zurich, Switzerland

SEVENTH FRAMEWORK PROGRAMME
e-infrastructure

Outline

- SEALS initiative
- Evaluation design
  - Criteria
  - Two phase approach
  - API
  - Workflow
- Data
- Results and Analyses
- Conclusions

SEALS INITIATIVE

SEALS goals

- Develop and diffuse best practices in evaluation of semantic technologies
- Create a lasting reference infrastructure for semantic technology evaluation
  - This infrastructure will be the SEALS Platform
- Organise two worldwide Evaluation Campaigns
  - One this summer
  - Next in late 2011 / early 2012
- Facilitate the continuous evaluation of semantic technologies
- Allow easy access to both:
  - evaluation results (for developers and researchers)
  - technology roadmaps (for non-technical adopters)
- Transfer all infrastructure to the community

Targeted technologies

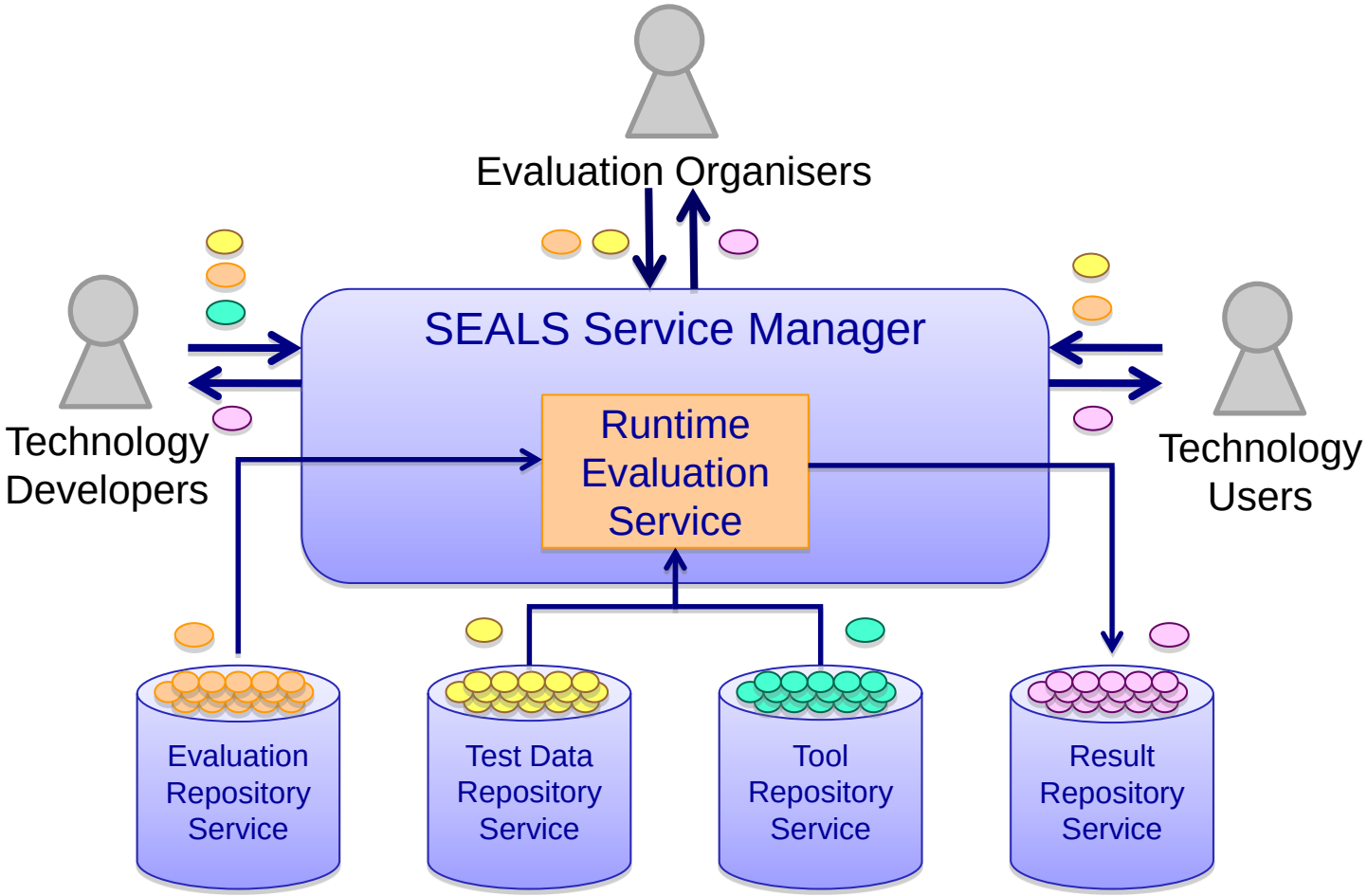
Five different types of semantic technologies:

- Ontology Engineering tools
- Ontology Storage and Reasoning Systems
- Ontology Matching tools
- Semantic Web Service tools
- **Semantic Search tools**

What's our general approach?

- Low overhead to the participant
  - Automate as far as possible
  - We provide the compute
  - We initiate the actual evaluation run
  - We perform the analysis
- Provide infrastructure for more than simply running high profile evaluation campaigns
  - reuse existing evaluations for your personal testing
  - create new ones evaluations
  - store / publish / download test data sets
- Encourage participation in evaluation campaign definitions and design
- Open Source (Apache 2.0)

SEALS Platform



SEARCH EVALUATION DESIGN

What do we want to do?

- Evaluate / benchmark semantic search tools
  - with respect to their semantic peers.
- Allow as wide a range of interface styles as possible
- Assess tools on basis of a number of criteria including usability
- Automate (part) of it

Evaluation criteria

User-centred search methodologies will be evaluated according to the following criteria:

- Query expressiveness
- U
- S
- C
- P

- Is the style of interface suited to the type of query?
- How complex can the queries be?
- How easy is the tool to use?
- Ability to cope with a large ontolo
- Is it easy to understand?
- Is it well structured?

Resource consumption:
- execution time (speed)
- CPU load
- memory required

Two phase approach

- Semantic search tools evaluation demands a user-in-the-loop phase
  - usability criterion

- Two phases:
  - User-in-the-loop
  - Automated

Evaluation criteria

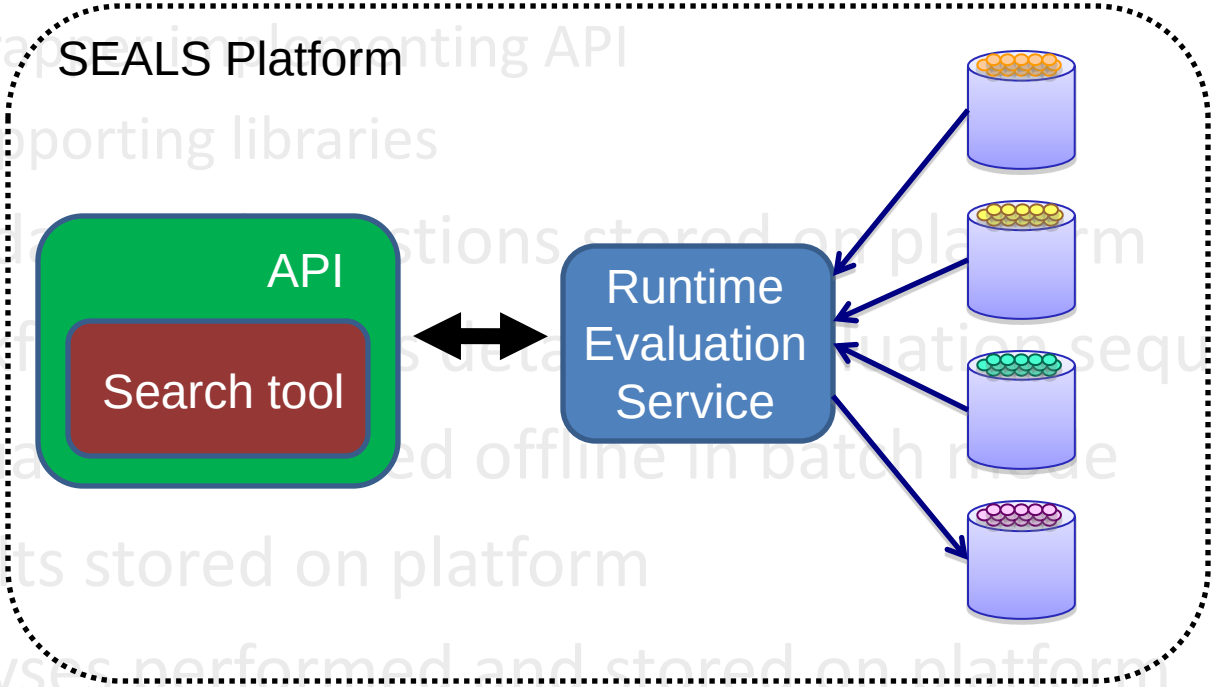
Each phase will address a different subset of criteria.

- **Automated evaluation**: query expressiveness, scalability, performance, quality of documentation
- **User-in-the-loop**: usability, query expressiveness

RUNNING THE EVALUATION

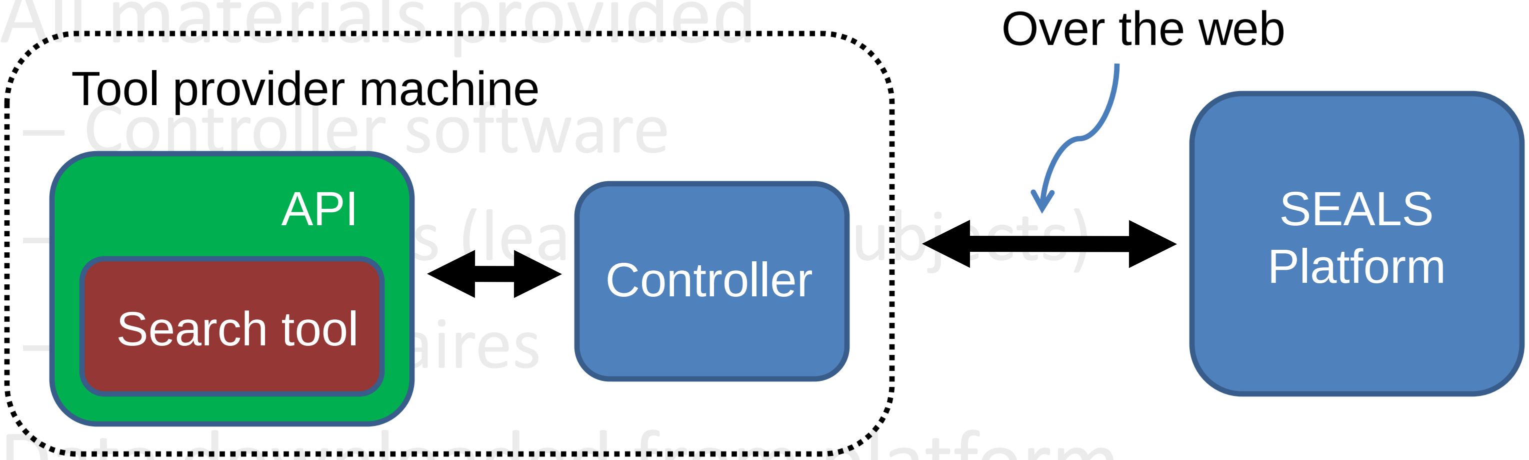
Automated evaluation

- Tools uploaded to platform. Includes:
  - wrapper implementing API
  - supporting libraries
- Test data and configurations stored on platform
- Workflow specifies data and evaluation sequence
- Evaluations executed offline in batch mode
- Results stored on platform
- Analyses performed and stored on platform



User in the loop evaluation

- Performed at tool provider site
- All materials provided
  – Controller software
  – …s (lea… …ubjects)
  – …aires



- Data downloaded from platform
- Results uploaded to platform

API

- A range of information needs to be acquired from the tool in both phases
- In automated phase, the tool has to be executed and interrogated with **no** human assistance.
- Interface between the SEALS platform and the tool must be formalised

API – common

- **`Load ontology`**
  - success / failure informs the interoperability
- **`Determine result type`**
  - ranked list or set?
- **`Results ready?`**
  - used to determine execution time
- **`Get results`**
  - list of URIs
  - number of results to be determined by developer

API – user in the loop

- **`User query input complete?`**
  - used to determine input time
- **`Get user query`**
  - String representation of user's query
  - if NL interface, same as text inputted
- **`Get internal query`**
  - String representation of the internal query
  - for use with…

API – automated

- **`Execute query`**
  - mustn't constrain tool type to particular format
  - tool provider given questions shortly before evaluation is executed
  - tool provider converts those questions into some form of 'internal representation' which can be serialised as a String
  - serialised internal representation passed to this method

DATA

Data set – user in the loop

- Mooney Natural Language Learning Data
  - used by previous semantic search evaluation
  - simple and well-known domain
  - using geography subset
    - 9 classes
    - 11 datatype properties
    - 17 object properties and
    - 697 instances
  - 877 questions already available

Data set – automated

- EvoOnt
  - set of object-oriented software source code ontologies
  - easy to create different ABox sizes given a TBox
  - 5 data set sizes: 1k, 10k, 100k, 1M, 10M triples
  - questions generated by software engineers

RESULTS AND ANALYSES

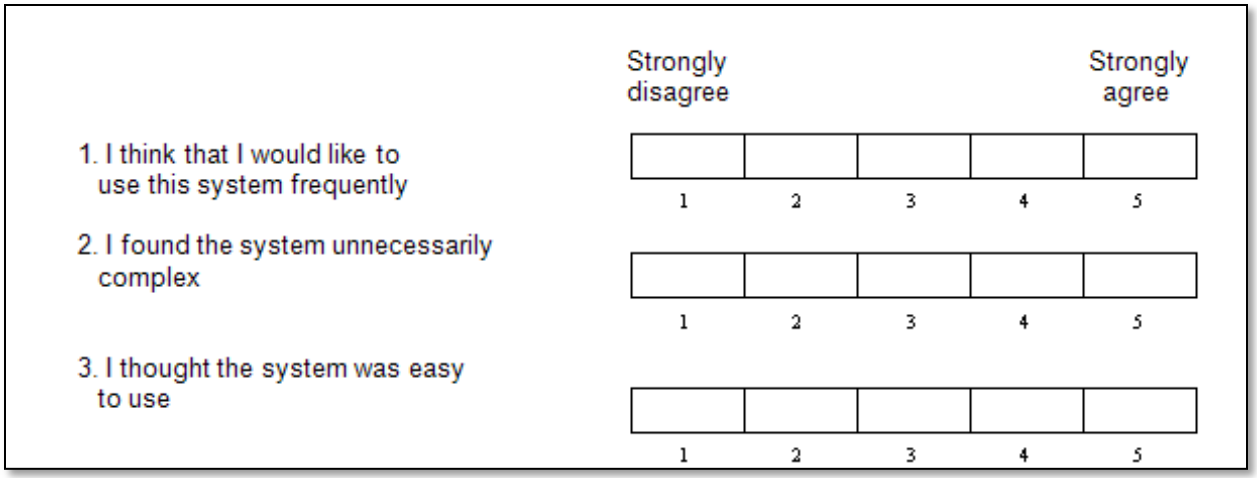
Questionnaires

3 questionnaires:

- SUS questionnaire
- Extended questionnaire
  - similar to SUS in terms of type of question but more detailed
- Demographics questionnaire

System Usability Scale (SUS) score

- SUS is a *Likert* scale
- 10-item questionnaire
- Each question has 5 levels (*strongly disagree* to *strongly agree*)
- SUS scores have a range of 0 to 100.
- A score of around **60** and above is generally considered as an indicator of good usability.



Demographics

- Age
- Gender
- Profession
- Number of years in education
- Highest qualification
- Number of years in employment
- knowledge of informatics
- knowledge of linguistics
- knowledge of formal query languages
- knowledge of English
- …

Automated

Results

- Execution success (OK / FAIL / PLATFORM ERROR)
- Triples returned
- Time to execute each query
- CPU load, memory usage

Analyses

- Ability to load ontology and query (interoperability)
- Precision and Recall (search accuracy and query expressiveness)
- Tool robustness: ratio of all benchmarks executed to number of failed executions

User in the loop

Results (other than core results similar to automated phase)

- Query captured by the tool
- Underlying query (e.g., SPARQL)
- Is answer in result set? (user may try a number of queries before being successful)
- time required to obtain answer
- number of queries required to answer question

Analyses

- Precision and Recall
- Correlations between results and SUS scores, demographics, etc

Dissemination

- Results browsable on the SEALS portal
- Split into three areas:
  - performance
  - usability
  - comparison between tools

CONCLUSIONS

Conclusions

- Methodology and design of a semantic search tool evaluation campaign
- Exists within the wider context of the SEALS initiative
- First version
  - feedback from participants and community will drive the design of the second campaign
- Emphasis on the user experience (for search)
  - Two phase approach

THANK YOU

Get involved!

- First Evaluation Campaign in all SEALS technology areas this Summer
- Get involved – your input and participation is crucial
- Workshop planned for ISWC 2010 after campaign

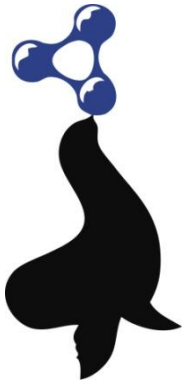
- Find out more (and take part!) at:

  http://www.seals-project.eu

  or talk to me, or email me (s.wrigley@dcs.shef.ac.uk)

Timeline

- May 2010: Registration opens
- May-June 2010: Evaluation materials and documentation are provided to participants
- July 2010: Participants upload their tools
- August 2010: Evaluation scenarios are executed
- September 2010: Evaluation results are analysed
- November 2010: Evaluation results are discussed at ISWC 2010 workshop (tbc)



Best paper award

SEALS is proud to be sponsoring the best paper award here at SemSearch2010

Congratulations to the winning authors!