SERVICES AND THE WEB OF DATA

An Engineering Perspective

Carlos Pedrinaci, Maria Maleshkova
(The Open University)
Acknowledgements

- Dong Liu (OU)
- Ning Li (OU)
- Jacek Kopecky (OU)
- John Domingue (OU)
- SOA4All project
From Linked Data to Smart Applications

- Linked Data Principles
  - Simple, clear, well-established
- Linked Data Applications
  - Based on these simple principles and technologies
  - ... and a bunch of hacks on top
We need reusable Components

The level of complexity and refinement of Linked Data Apps will be proportional to our ability to reuse pre-existing solutions (i.e., components/functionality)
How much functionality are we sharing? And reusing?
...THIS WILL BE OFTEN OFFERED AS A SERVICE

For reasons of *scale* (or control), in the Semantic Web many of these components will *have to be* offered as *services* that gather and analyse large quantities of data to provide *advanced functionality* (or their results) online.
Developing and sharing software
Models for capturing services
Registry for sharing and finding these
Engine for invoking services and providing a linked data interface over these...
SOA4RE

Mashes Linked Data and Web APIs data

Finds and Invokes Linked Services on the Fly

Modularity and Extensibility as a core built-in feature
Where are all the Problem Solvers Gone?

- Applications require both static and dynamic knowledge
- “To build systems that solve real-world tasks, however, we must not only specify our conceptualizations, but also clarify how problem solving ideally will occur.”

M. A. Musen
Abstracting Problem-solving

- A pillar of semantic technologies is genericity. Let’s exploit it!
- This requires decoupling problem-solving knowledge from the domain
- Problem-Solving Methods research focused precisely on this
Applying PSMs over domain specific data

Engines based on problem-specific vocabularies
Used by feeding them with domain-specific data at invocation time
Suggestion

• We need to support the systematic and effective reuse of functionality
• We have devised some initial infrastructure to support this process
• Decoupling problem-solving knowledge from the domain is essential for further reuse
• PSMs research showed us how to do it
Challenges

- Keep the overhead low
- Exploit the cloud and other people’s work/services
- Performance vs genericity
- Full complexity of PSMs needs to be avoided but is also not necessary
- Surrounding solid tooling is necessary
... AND YOUR COMPONENTS!
“... THIS CAN’T WORK ...”
Bringing the Big Think to the Small Screen

Web Services and APIs in the Cloud

Domain & Task Models

Intelligence at the Interface

Slide by Tom Gruber, Siri