The Semantic Web & Ontologies for ACCESS

Richard Newman

CS127 r.newman@reading.ac.uk

Outline

- What is the Semantic Web?
  - What are ontologies?
- Autonomic computing, markets, and description
- Ontologies for ACCESS

What is the Semantic Web?

- There are plenty of definitions!
- Here is mine:

  Web-centric knowledge representation.
- The Semantic Web uses URIs as globally-shared identifiers, HTTP for resource retrieval, and XML/Unicode (among others) for interchange.

What are ontologies?

- “Specification of a conceptualisation” (Gruber). A formal definition of the classes and relations in the domain.
- A theory of the world, its contents, and the relations between them.
- Allows for interoperability: two systems *subscribing* to the same ontology can communicate *meaning*.

Autonomic computing and markets

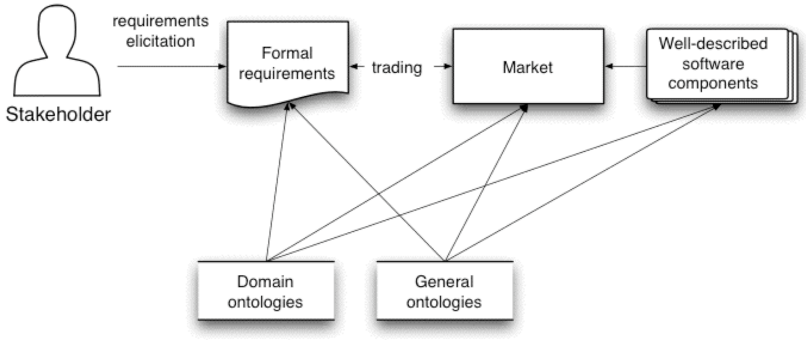
- ACCESS is taking a market-based approach to dynamically achieving requirements.
- Market mechanisms are used to match the requirements of a system to components in a software market.
- Of course, this needs to be done *automatically* — indeed, *autonomically*.

Ontologies in a market

- Markets already have languages and implicit ontologies
  - Impenetrable cattle market speech
  - 22SWF GSOH NS
  - 17in alloy, A/C, sat-nav, 140bhp, 28mpg
- Selling on the market is easier if you use the right “lingo”.
- Buying on the market often requires it in order to interact.

Ontologies for machines

- Humans are good at context and language.
- An agent market *requires* precise, shared terminology and language to describe the items being traded.
  - And the market itself?
- Unambiguous language.



Describing things

- The requirements of a system, its constraints, the components available, and the markets themselves need to be described in a machine-understandable way.
- Further, the structure and operation of components need to be understood in order to compose and evaluate them.
- Ontologies of domains and services are needed. These ontologies must be available to every agent that uses a market.

The Semantic Web & ACCESS

- Any versatile autonomic system needs an extensible, formal way of describing requirements and services (the things which fulfill the requirements).
- It also needs a language which can be reasoned about, in order to compose and select services or components.
- The Semantic Web provides languages for doing these, describing domains (in OWL) and processes (OWL-S).

IBM’s work

- “The Role of Ontologies in Autonomic Computing Systems” (Stojanovic *et al*, 2004)
  - Correlation engines: analysis of incoming data — rules engine.
  - Proposes use of ontologies at the system level.
  - Vital for autonomic computing — availability.
  - Can ontologies also achieve a more abstract kind of autonomic computing?

Tiny example

- Calculating MPG from litres and miles.
- Requirement: a process which takes L and M, producing MPG, for less than £0.10 per use.

```
:mpgService   a :RequiredService ;
      :in    [ a xsd:double ;
               :represents :LitreQuantity ] ;
      :in    [ a xsd:double ;
               :represents :MileQuantity ] ;
      :out   [ a xsd:double ;
               :represents :MilesPerGallon ] ;
      :maxCost [ rdf:value “0.1”; :unit :UKP ].
```

Components

```
:mgToMPG     a :ConversionService ;
:in    [ a xsd:double ; :represents :MileQuantity ] ;
:in    [ a xsd:double ; :represents :GallonQuantity ] ;
:out   [ a xsd:double ; :represents :MilesPerGallon ] ;
:cost  [ rdf:value “0.05”; :unit :UKP ].
:litToUKGal  a :ConversionService ;
:in    [ a xsd:double ; :represents :LitreQuantity ] ;
:out   [ a xsd:double ; :represents :GallonQuantity ] ;
:cost [ rdf:value “0.1”; :unit :USD ].
```

Conclusions

- Ontologies and the Semantic Web — machine-understandable description of “things”.
- ACCESS attempts to continually match requirements to available software components.
- The Semantic Web can probably be used to describe these software components.

Open question

- Can we also declaratively describe the requirements of the system stakeholders, allowing for ACCESS to satisfy their needs?
- I.e. can human requirements for software be described formally?

A more useful question

- Do the human requirements that we *can* describe suffice to allow an autonomic system such as ACCESS to achieve its goals?