The OWL-S Experience
(are you experienced?)

Bijan Parsia, Evren Sirin,
James Hendler
W3C Workshop on SWSF
June 2006
A brief (conceptual) overview

• OWL-S is a collection of ontologies
  – Encoded in OWL (currently OWL-DL)
  – A loose upper ontology/framework for web services
    • Description oriented
  – A focus on capabilities
    • “Functional”: Inputs-Outputs-Preconditions-Effects (IOPEs)
    • “Other”: QoS, location, owner, price, yellow pages category
  – And behavior
    • Modeled with a process/action paradigm
    • Roots in the situation calculus
    • Primary composition model == planning
      – Ok, my primary composition model
OWL-S Concepts

• Main top level concepts:
  – Service
    • Connects the others
  – Profile
    • Discovery, matchmaking, selection
  – Process Model
    • Composition, execution, monitoring
  – Grounding
    • Execution
    • Where OWL-S descriptions touch the WS Stack
    • WSDL grounding built in, others (e.g., UPnP) developed
A world above

Profile

IOPE

Extras

Process Model

AP

AP

Grounding

Op  WSDL  Op  UPnP  Etc.

Execution (SOAP, HTTP, etc.)
Ascension

• The (justified) fear
  – It’s hard to add all this detail
  – Need loads of background knowledge
    • Both in you
      – (must be good at OWL-S, OWL, FOL, other scary things)
    • And in the computer
      – (need lots of background theories)

• The experience
  – OWL-S has a lot of Pointless Cruft
    • Partially from RDF/XML; partly from a (mis)use of OWL
  – Writing by hand isn’t that bad
  – Writing tools to ease the pain is quite straightforward
  – The descriptions don’t have to be that rich
    • To be moderately interesting
Description Enablers

• My favorite
  – A grad student, let’s call him, “Evren”
    • Cheap, effective

• Wizard like WSDL2OWL-S tools
  – There are a couple of these
    • Ours, CMU’s (which is now in the Protégé plugin)
    • Derive structure from WSDL
    • Let author add a-li’l’-semantics

• Machine learning approaches
  – E.g., ASSAM
  – Induce a-li’l’-semantics from a set of WSDL
(One kind of) Composition

If I...

A series of actions, aka service calls...

Composition/planning can focus on IO chaining, PE evaluation toward a goal, template instantiation, and consider PEs or other capabilities along the way...that do the job (in this case with these parameters)
Exercising the framework

• Discovery and selection
  – You had to *find* the services…
  – …with the right functional properties…
  – …and capabilities

• Composition
  – You built up a specific sequence
    • Perhaps manually; perhaps automatically

• Execution
  – The resultant composition can be invoked
    • And, if your descriptions are good and the circumstances right….everything Will Just Work
(Semi)Automating Composition

• People driven
  – Interactive workflow editors
    • “The Composer”
    • Task Computing
    • Several Grid projects
  – Tend to focus on matchmaking/selection
  – Some mixing of initiative

• Planner driven
  – IO chainers
  – JSHOP-DL
    • Treat a set of composite processes “with holes” as templates
    • Template holes are (recursively) filled in
    • Covers much of OWL-S
Some Lessons

• Using OWL-S
  – It can be used
    • Software infrastructure is maturing
    • OWL-S API key enabler
  – You don’t have to use a lot
    • To be interesting
  – E.g., Task computing
    • Fujistu Labs -- http://taskcomputing.org/
    • 10s of services, mostly from pervasive enviro
    • “End user” interaction
    • Wrote 90% of the manipulated services, all the descriptions

• Conceptual model fairly robust
More lessons

• “Expressive type systems”
  – A good idea…maybe
    • Not *any* ol’ expressivity

• Representing state change
  – Preconditions and effects
  – Aka Queries and Updates

• Process modeling
  – Hard to compete with programming
The usual places

- http://www.daml.org/services/owl-s/
- http://www.mindswap.org/2004/owl-s/
- http://www.mindswap.org/~evren/composer/