THE UNIVERSITY of EDINBURGH
informatics

Semantic Web Systems

Web Services – Part 2

Jacques Fleuriot

School of Informatics

In the previous lecture

- Web Services (WS) can be thought of as Remote Procedure Calls.
- Messages from a client will specify the operation to be called, and will supply arguments for the operation.
- The service responds (typically) with the result of the operation on those arguments.
- The messages are standardly sent over HTTP as the body of a SOAP document; the SOAP header contains addressing information.
- Services are standardly described using WSDL. This specifies
  - types;
  - operations and their inputs and outputs;
  - a binding for each operation which specifies the allowed protocol and the service endpoints.

In this lecture

- Semantic Web Services
- OWL-S view of services:
  - Service profile
  - Service model
  - Service grounding

Motivation for Semantic Web Services

- Standard Web Service technology provides virtualisation for distributed computing:
  - Abstraction from specific platforms and programming languages.
  - Promotes interoperability of diverse service implementations.
- But foundation for **automating** Web Services still lacking.
- Semantic WS intended to supplement standard WS.
- By providing semantically explicit metadata for WS:
  - Software can interpret descriptions of unfamiliar WS.
  - Carry out discovery, composition, etc.
- OWL-S builds on OWL to provide OWL descriptions of Services.

OWL Digression

- RDFS allows us to build simple class hierarchies for describing ontological structure.
- OWL (Web Ontology Language) gives us a richer framework:
  - Syntactically layered on RDF.
  - Uses theoretical framework of Description Logic (decidable fragment of First Order Logic).
  - A language for describing 'concepts' (classes of instances).
  - Provides negation, and standard notion of logical consistency.
  - Provides operators for defining classes as well as introducing primitive classes.
  - Provides a **limited** form of quantification.

Syntax and Semantics of DL Concepts

Simple Concepts

Giraffe	$\{x \mid \text{Giraffe}(x)\}$

Composed Concepts

Brother ⊔ Sister	$\{x \mid \text{Brother}(x) \lor \text{Sister}(x)\}$
Adult ⊓ Male	$\{x \mid \text{Adult}(x) \land \text{Male}(x)\}$
¬ Married	$\{x \mid \neg\text{Married}(x)\}$

Subsumption

Giraffe ⊑ Mammal	$\forall x\, (\text{Giraffe}(x) \longrightarrow \text{Mammal}(x))$

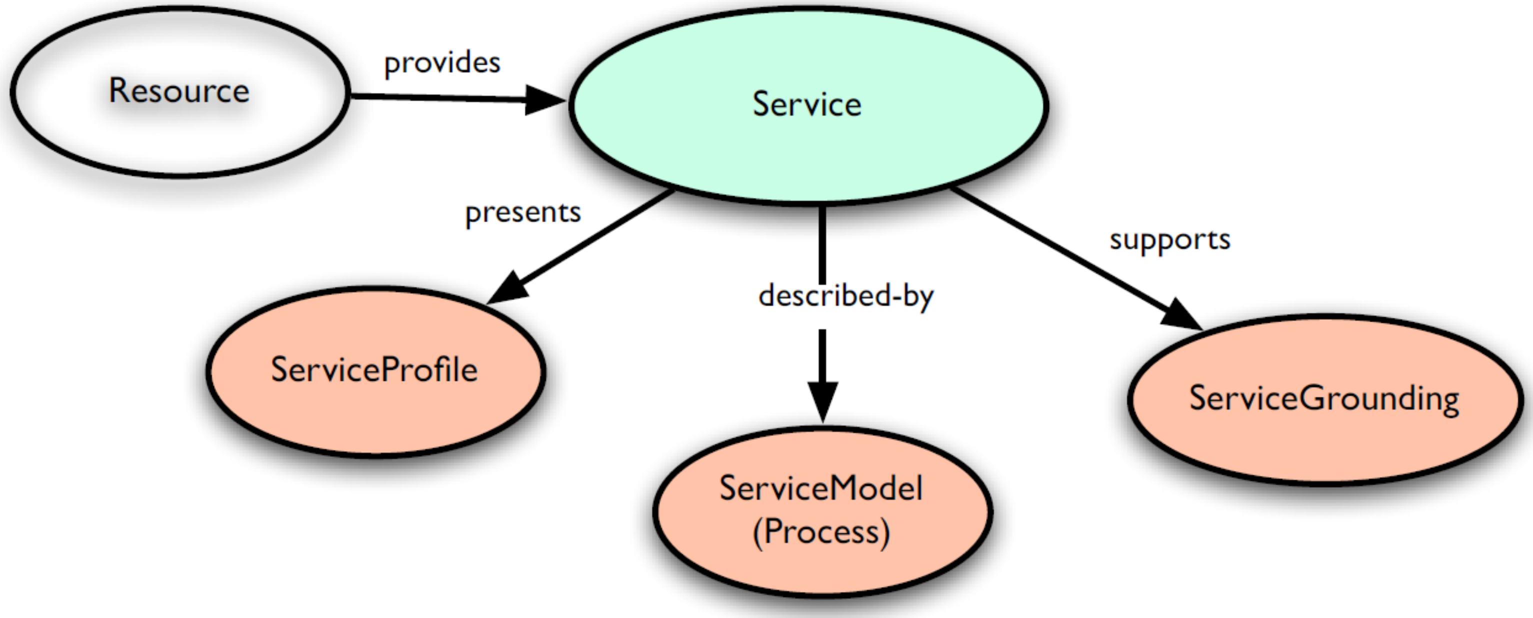
Definitional Equivalence

Sibling ≐ Brother ⊔ Sister	$\forall x\, (\text{Sibling}(x) \longleftrightarrow \text{Brother}(x) \lor \text{Sister}(x))$

OWL-S View of Services

- Based on DAML (Darpa Agent Markup Language) and DAML-S.
- Provides an ontology for web services that consists of three sub-ontologies:
  1. Service Profile: How the service presents itself to the external world.
  2. Service Model: What the service does, and how the client interacts with it.
  3. Service Grounding: How the service is **realised** – analogous to WSDL binding.

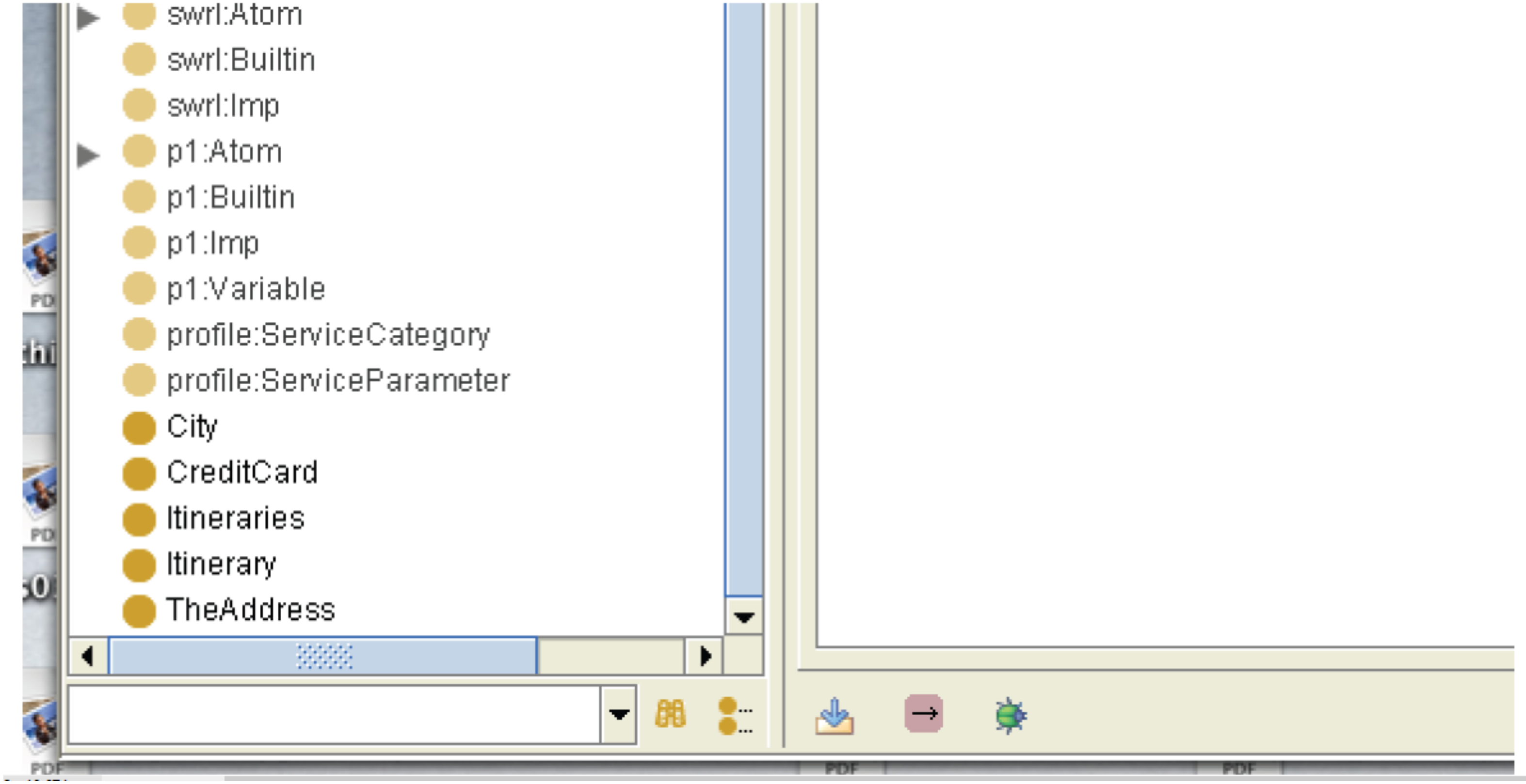
OWL-S Service Ontology



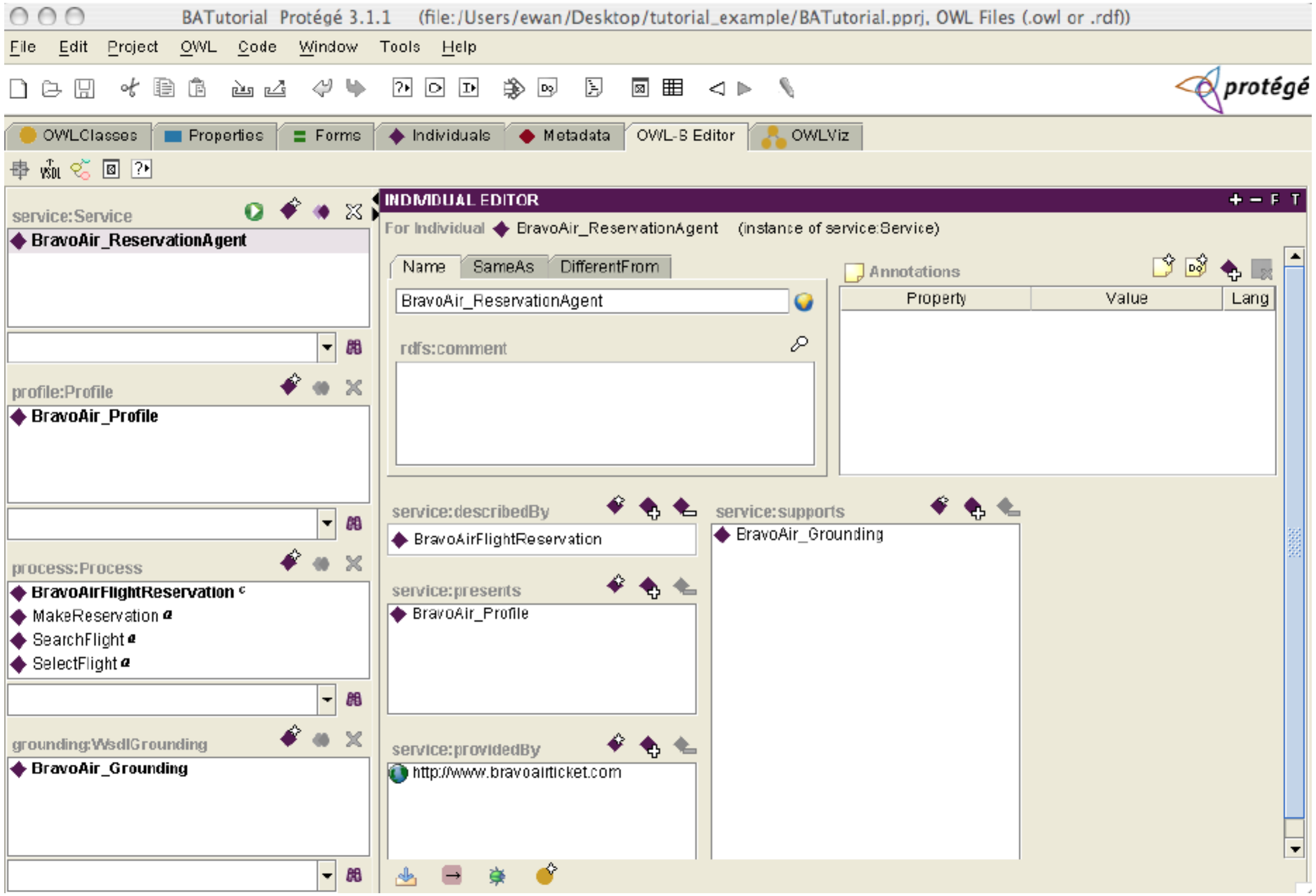
Service Model: Inputs and Outputs

- OWL-S functional description of services very similar to WSDL.
- Inputs and outputs specify the data transformation produced by the process.
- General notion of Parameter;
- The type of (values of) the Parameter is specified with a URI.
- Typically, this will be a pointer to an OWL class in a domain ontology.
  - Input,Output $\sqsubseteq$ Parameter
- Parameters are associated with services via property hasParameter:
  - hasInput, hasOutput sub-properties of hasParameter

OWL-S Plugin for Protégé: Domain Ontology



OWL-S Plugin for Protégé: OWL-S Service



Service Model: Participants

- A process involves two or more agents.
- Required agents:
  - TheClient – the service is described from the point of view of the client.
  - TheServer – principal element of the service that the client deals with.

State Transformations

Question: Can Web Services change the world?

Changing the world with WS

Before invoking Amazon: your net assets are £999.00.

After invoking Amazon: your net assets are £000.00 but you are now the proud owner of a Widescreen 4K LED TV.

Preconditions and Effects

OWL-S distinguishes two aspects of WS:

1. Transforming information – inputs and outputs
2. Transforming the world – preconditions and effects

Example Preconditions

valid(creditcard, t0) ∧ limit(creditcard) ≥ £999.00

Example Effect

(debt(creditcard, t1) = debt(creditcard, t0) - £999.00) ∧
own(i, TV, t1)

IOPEs

IOPE = Input, Output, Precondition, Effect

Expressing Preconditions and Effects

Expressing Truths about the World

Preconditions and effects need to be stated in terms of a reasonably expressive logical language. By themselves, RDF and OWL do not provide a good basis for such a language.

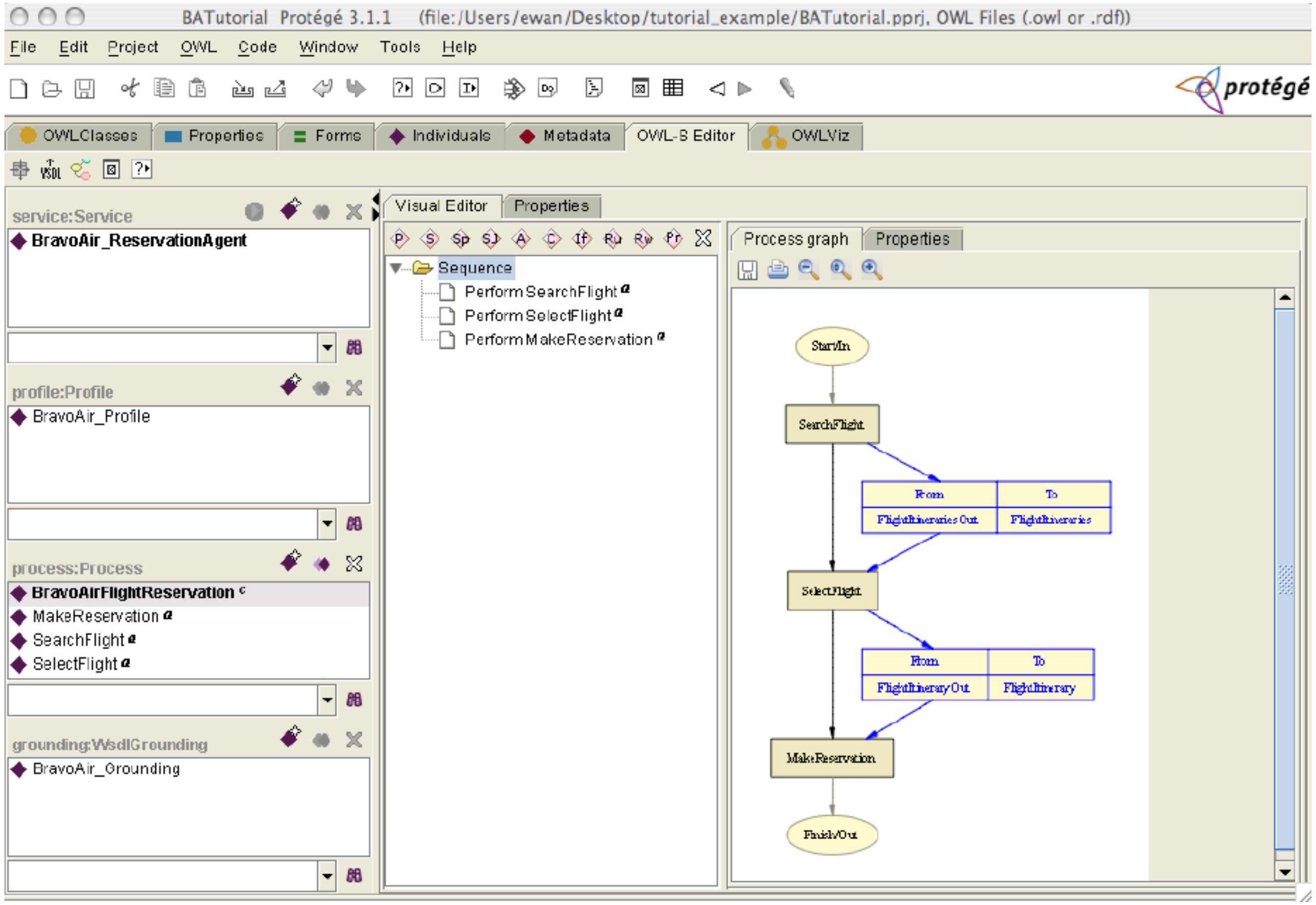
Embedding Logic in OWL-S

- Logic and the Semantic Web – rather messy!
  - http://www.w3.org/DesignIssues/Logic
  - Fensel & van Harmelen (2007)
- OWL-S tries to be non-committal about choice of logical language, makes a number of suggestions:
  - N3 Extensions beyond RDF for expressing logical rules.
  - RuleML http://www.ruleml.org – and broader than deductive logic; XML-based; somewhat orthogonal to other efforts.
  - SWRL (Semantic Web Rule Language) http://www.w3.org/Submission/SWRL/ – embeds OWL assertions in Horn-clause rules.
  - SWRL-FOL http://www.daml.org/2004/11/fol/proposal – extension of SWRL to arbitrary FOL formulas.
  - SPARQL: Partial specification of entailment over RDF(S) graphs.
- In OWL-S, expressions from these languages can be embedded as RDF literals.

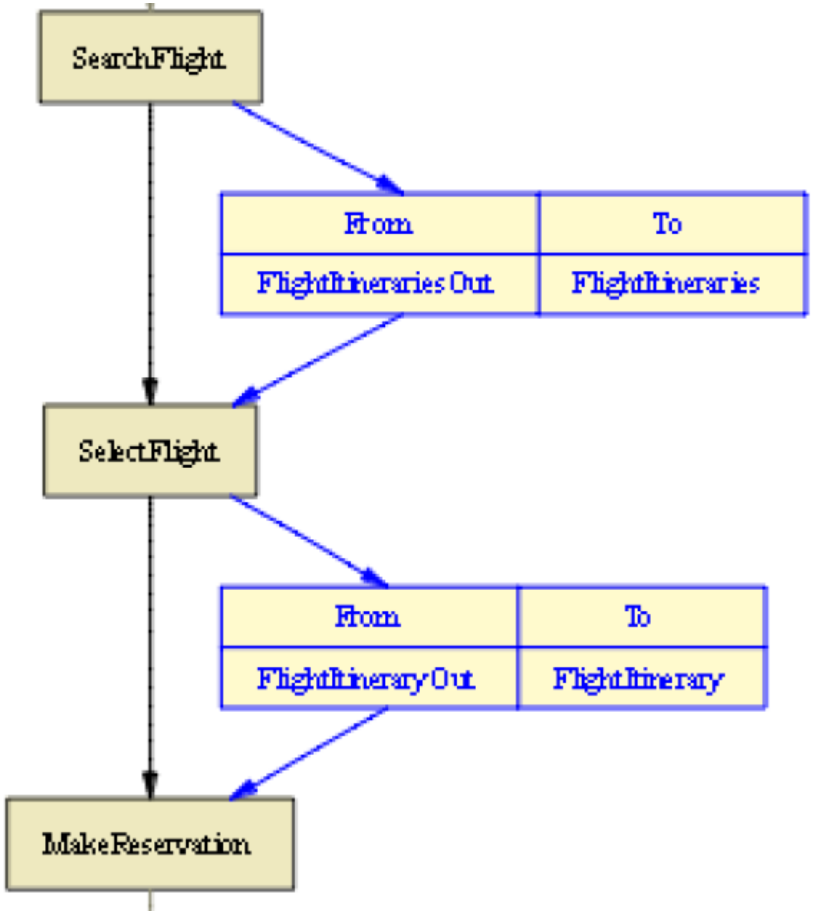
The Process Ontology

- OWL-S divides processes into
  - Atomic, and
  - Composite.
- Various constructors are provided for assembling composite processes out of component ones, e.g.,
  - Sequence,
  - Choice,
  - Iterate, etc.
- A composite process represents behaviour a client can perform by sending and receiving messages.
- Inputs of an standalone atomic process must come directly from client;
- Inputs of components of a composite process may come from preceding steps.

OWL-S Plugin for Protégé: Process 1



OWL-S Plugin for Protégé: Process 2



Abstracting over Composite Processes

- Composite processes can be viewed at a higher level of abstraction, as simple processes.
- Allows layering, i.e. composite processes can be incorporated as simple processes into further composites.

Service Profile

- Description of the service that can be used by registry or broker.
- Once a client has chosen to engage with a service, uses the Service Model, not the Profile.
- By default, Profile uses same IOPEs as the Model, but this is not mandatory.
- Can also include information such as Service Category and Quality of Service (QoS).

Grounding

- Mapping from abstract specification to a concrete specification of service;
- particularly, those service elements required for interaction.
- For OWL-S, main issue is relating inputs and outputs of atomic process to the input and outputs of a WSDL operation.
- WSDL by default specifies types using XML Schema,
- But OWL classes could be defined (using OWL namespace) in types section, or
- Referenced from within a WSDL operation definition using an owl-s-parameter attribute.

Summary

- OWL-S provides an upper ontology for web services:
  - Profile,
  - Process, and
  - Grounding.
- OWL-S allows service inputs and outputs to be typed in terms of OWL classes.
- Latter are typically drawn from a domain ontology.
- OWL-S supplements functional descriptions with preconditions and effects.
- The logic for these is embedded as RDF literals.
- Service Grounding is realised in terms of a mapping to WSDL.

Reading

- http://www.w3.org/Submission/OWL-S
- http://www.daml.org/services/owl-s/1.0/
- *Bringing Semantics to Web Services with OWL-S,* David Martin et al. (2007) World Wide Web Journal, Volume 10, Number 3, pp. 243-277.
- *Unifying Reasoning and Search to Web Scale*, Dieter Fensel and Frank van Harmelen (2007) Internet Computing, IEEE Volume 11, Issue 2, March-April, pp. 95–96.