OWL-S Briefing

DAML Web Services Coalition

Presented by:

David Martin (SRI)

http://www.daml.org/services/

Objectives

- **Status & roadmap**
- **Mini-Tutorial**
- **Gather requirements**
- **Invite / stimulate new contributions to Semantic Web Services**
- **Create a larger discussion around the hard problems**

Top-level Outline

- **Language status (40 min.)**
  - **OWL-S 1.0 (David Martin)**
  - **Security extensions (Grit Denker)**
- **Supporting products (Massimo Paolucci) (30)**
  - **Tools, demos, use cases**
- **Outreach & uptake (20)**
  - **Standardization efforts & strategies (Katia Sycara)**
  - **Users, workshops, books, papers (Terry Payne)**

--- Break ---

- **Open issues & challenges (Mark Burstein) (40)**
- **Roadmap for language evolution (David) (20)**

DAML Services Coalition

BBN: Mark Burstein

CMU: Katia Sycara*, Massimo Paolucci*, Naveen Srinivasan

De Montfort University: Monika Solanki

ICSI: Srini Narayanan

Maryland / College Park: Bijan Parsia

Nokia: Ora Lassila

SRI: David Martin*

Stanford KSL: Deb McGuiness

Southampton: Terry Payne*

Univ. of Toronto: Sheila McIlraith*

USC-ISI: Jerry Hobbs

Vrije Universiteit Amsterdam: Marta Sabou

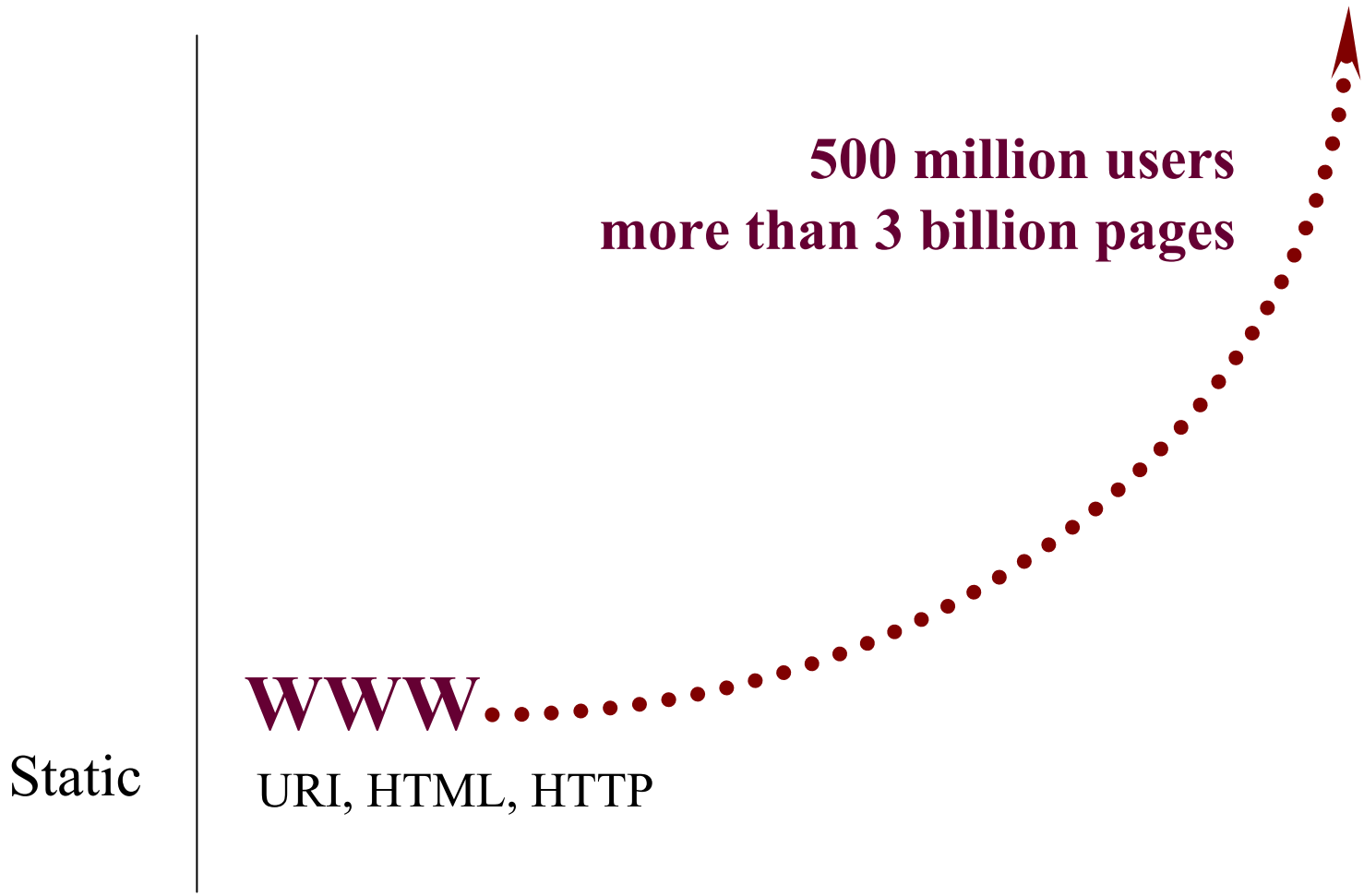
Yale: Drew McDermott

*Contributor to these slides

Semantic Web Services Initiative (SWSI)

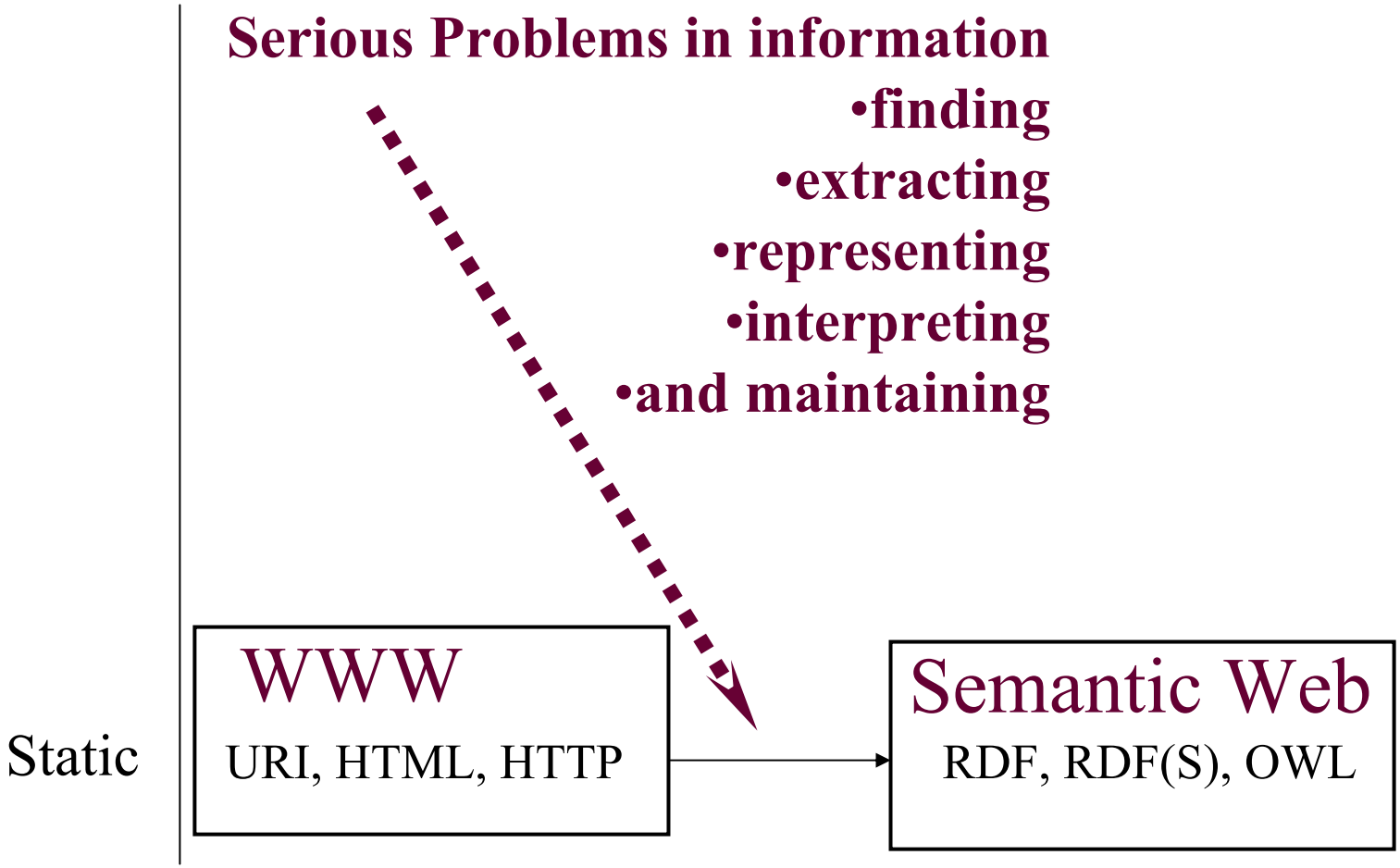
- Bring together US and European Semantic Web Services researchers
- Engage in collaborative standardization efforts
- Somewhat broader technology perspective
  - Semantic Web Services Language (SWSL)
    - OWL-S as a primary input
    - More attention to working with industry standards efforts
  - Semantic Web Services Architecture (SWSA)
    - WSMF as a primary input

Why Semantic Web Services?



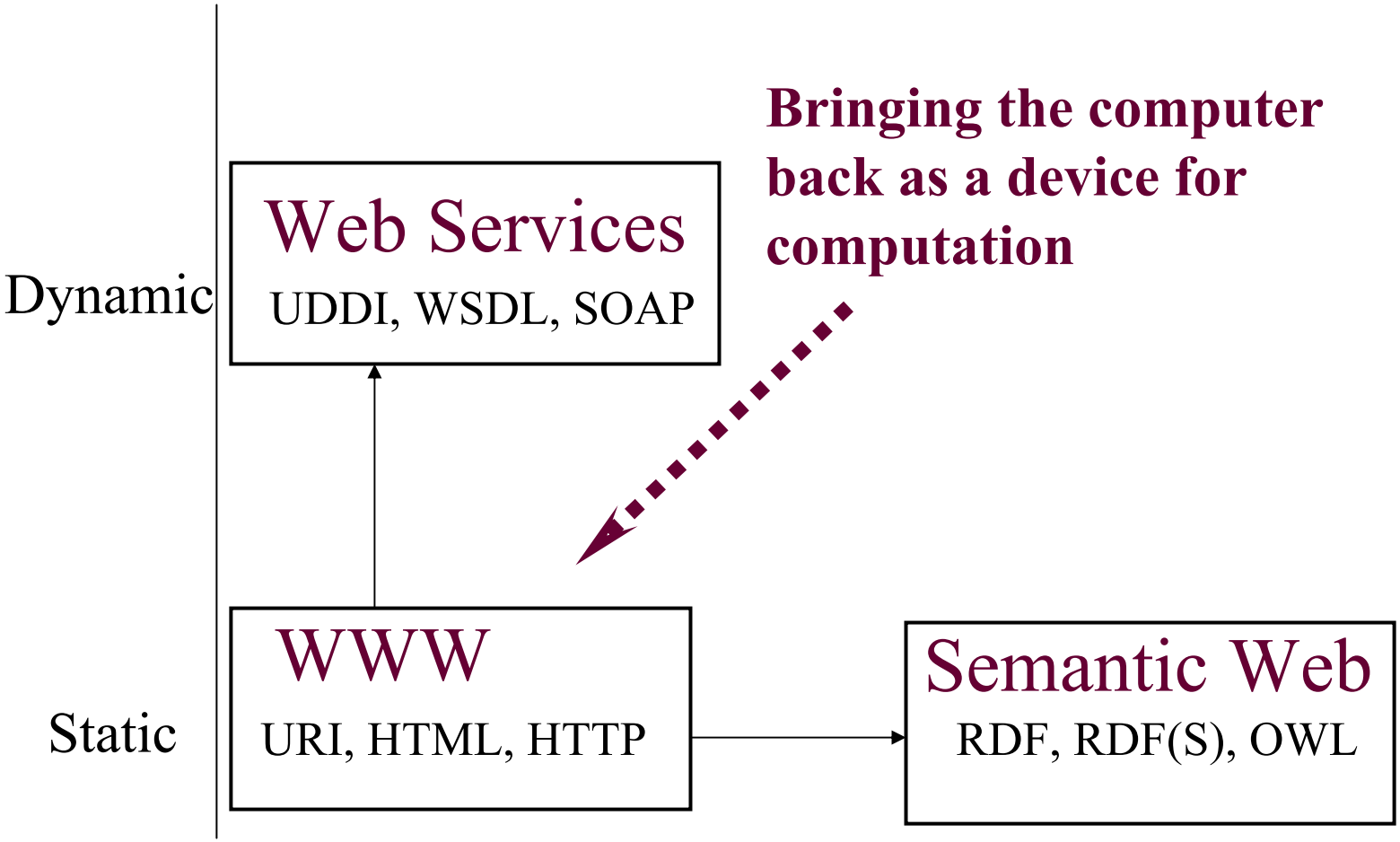
Thanks to Dieter Fensel (U. of Innsbruck)
for use of this material

Why Semantic Web Services?



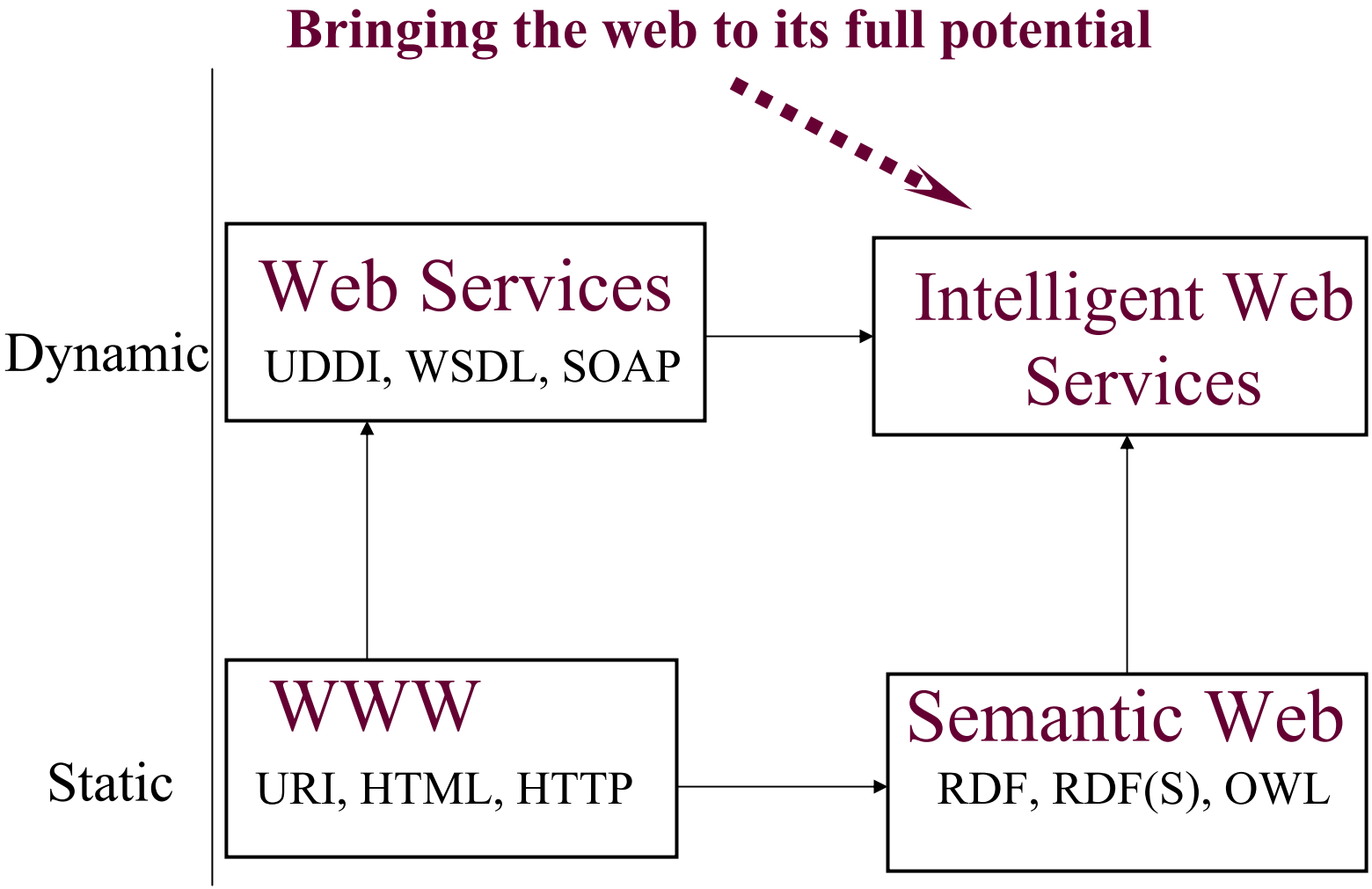
Thanks to Dieter Fensel (U. of Innsbruck) for use of this material

Why Semantic Web Services?



Thanks to Dieter Fensel (U. of Innsbruck) for use of this material

Why Semantic Web Services?



Thanks to Dieter Fensel (U. of Innsbruck) for use of this material

Top-level Outline

- **Language status**
  - **OWL-S 1.0 (David Martin)**
    - **Profile, Process Model, Grounding: Overview, recent evolution, next steps for each**
    - **Release status**
    - **For more detail: ISWC Tutorial (Katia & Terry)**
  - **Security extensions (Grit Denker)**
- **Supporting products**
- **Outreach & uptake**

--- Break ---

- **Open issues & challenges**
- **Roadmap for language evolution**

What is OWL-S?

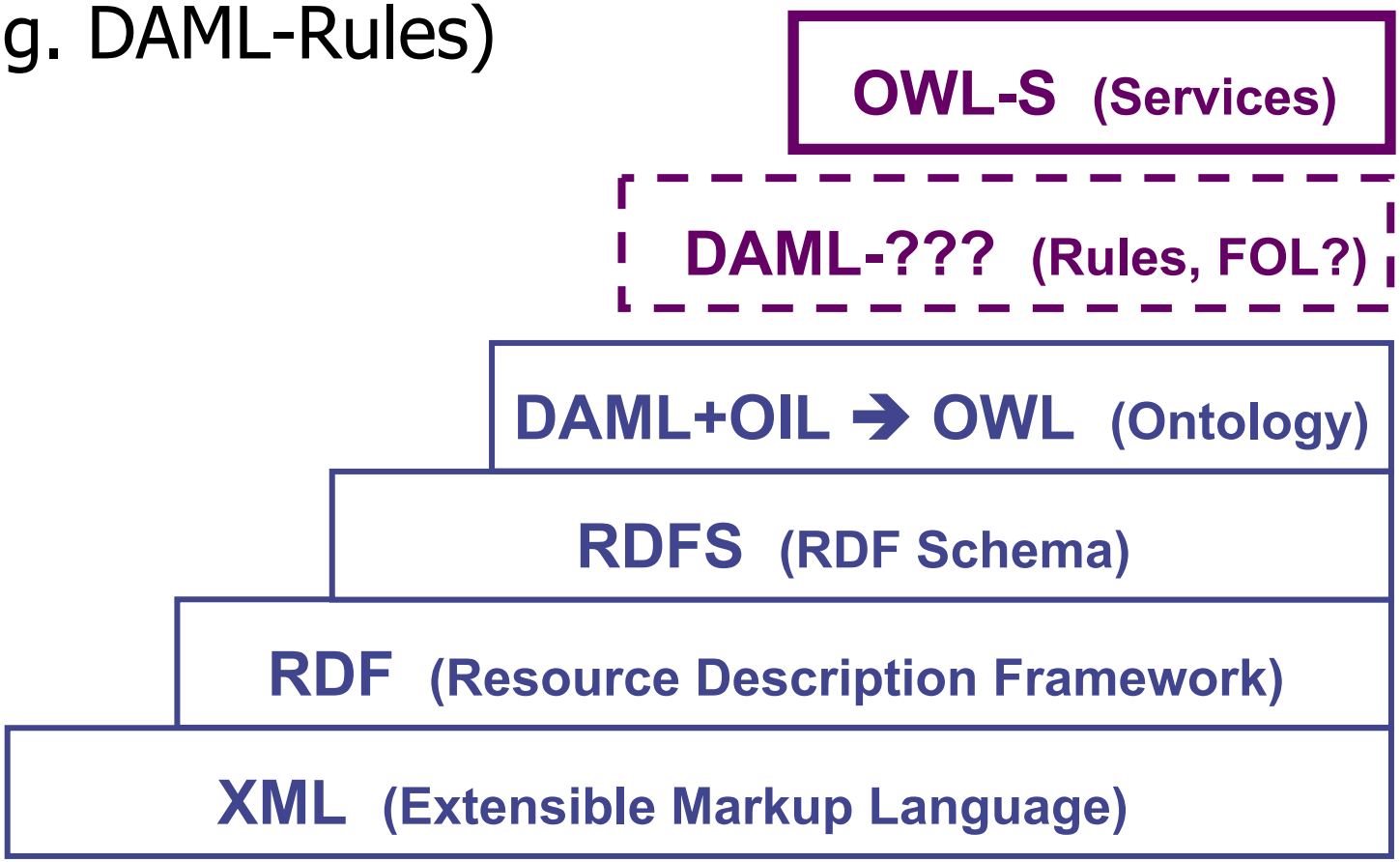
- Ontology Web Language for Services
- An OWL ontology/language for (formally) describing properties and capabilities of Web services
- An approach that draws on many sources
  - Description logic
  - AI planning
  - Workflow
  - Formal process modeling
  - Agents
  - Web services

http://www.daml.org/services/

Layered Approach to Language Development

OWL-S: a major application of OWL

Future versions will build upon emerging layers (e.g. DAML-Rules)



OWL-S Objectives

- Automation of service use by software agents
  - Ideal: full-fledged use of services never before encountered:
    - Discovery, selection, composition, invocation, monitoring, ..
- Useful in the “real world”
  - Compatible with industry standards
  - Incremental exploitation
- Enable reasoning/planning about services
  - e.g., On-the-fly composition
- Integrated use with information resources
- Ease of use; powerful tools

Automation Enabled by OWL-S

- Web service discovery

  Find me a shipping service that transports goods to Dubai.

- Web service invocation

  Buy me 500 lbs. powdered milk from www.acmemoo.com

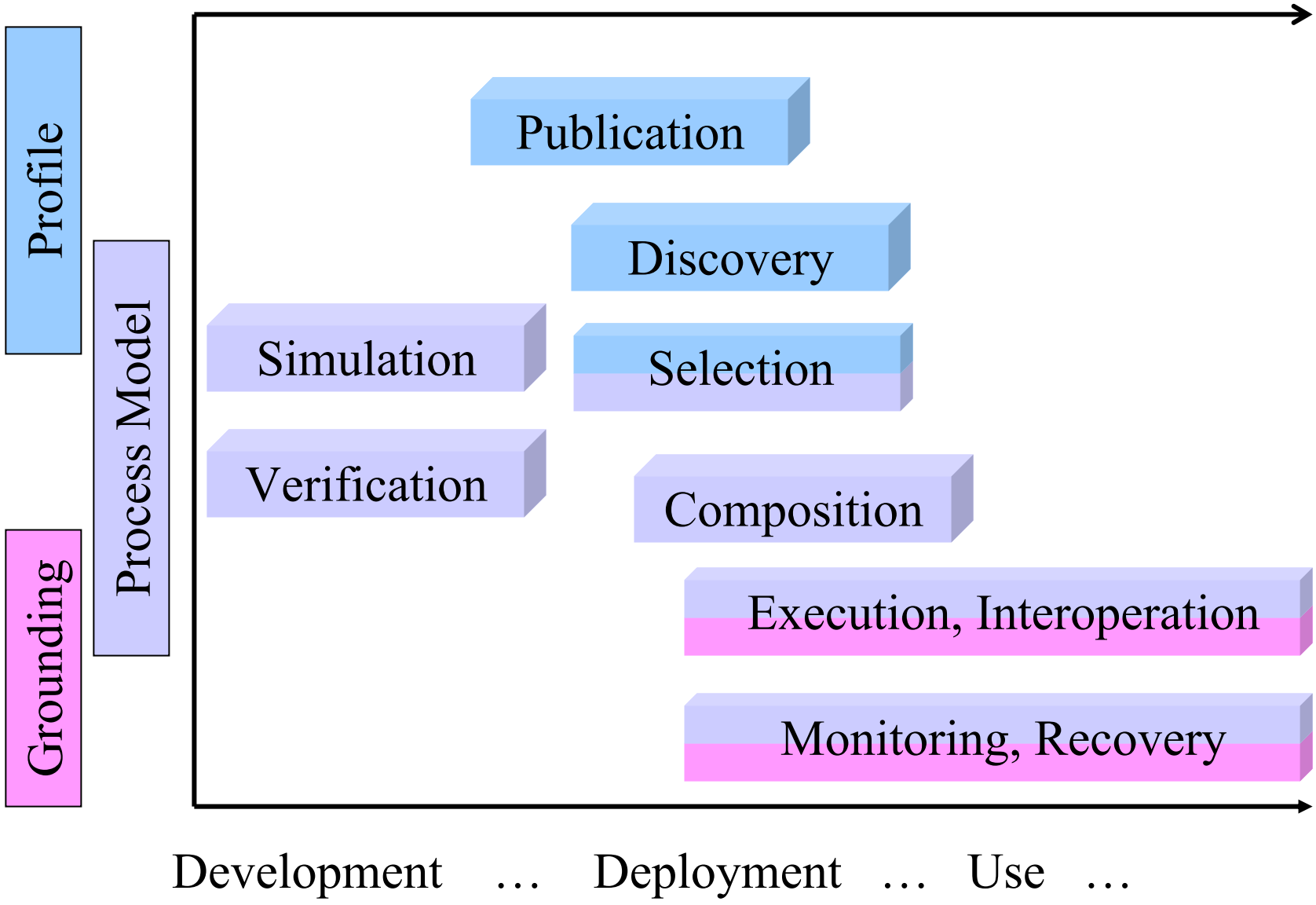
- Web service selection & composition

  Arrange food for 500 people for 2 weeks in Dubai.

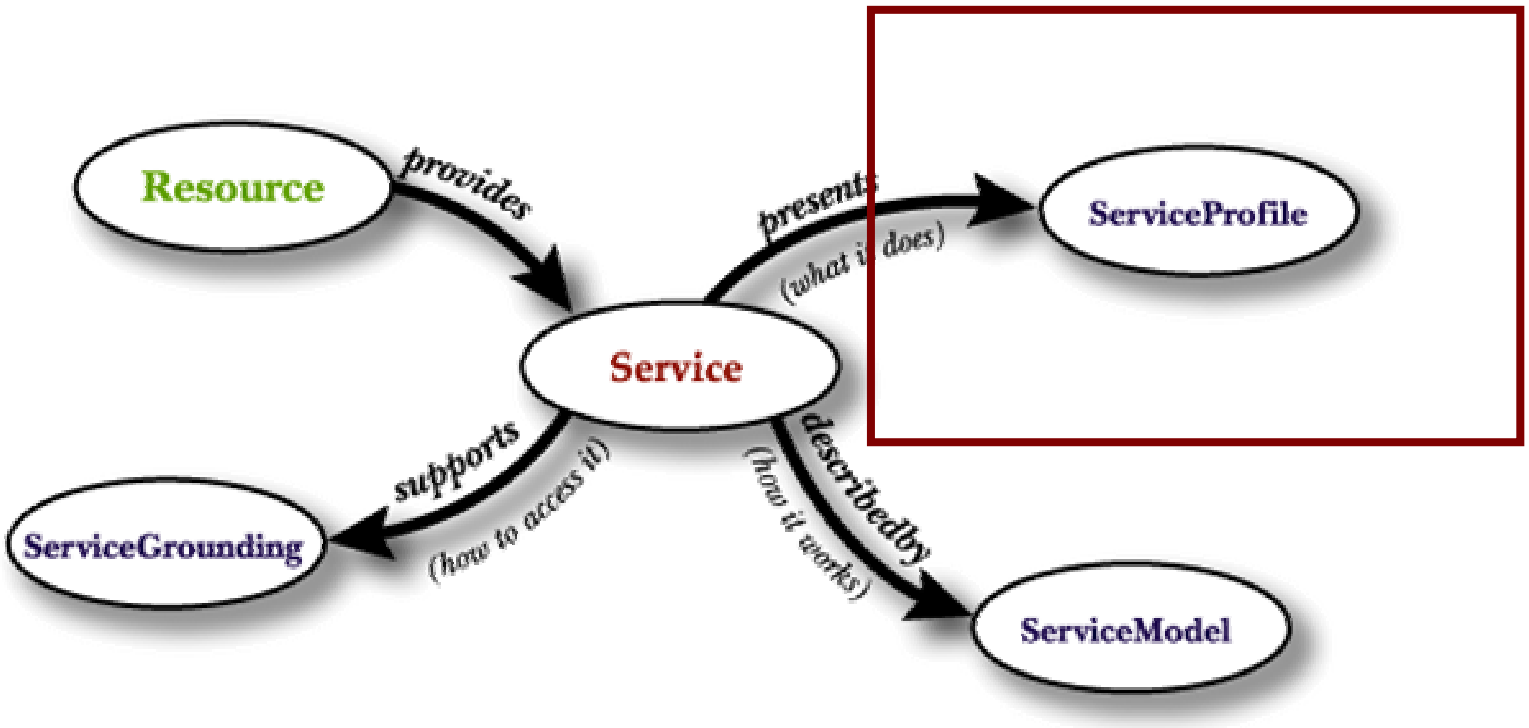
- Web service execution monitoring

  Has the powdered milk been ordered and paid for yet?

Key:


Profile
Process Model
Grounding
Publication
Discovery
Simulation
Selection
Verification
Composition
Execution, Interoperation
Monitoring, Recovery
Development … Deployment … Use …

Upper Ontology of Services



Ontology images compliments of Terry Payne, University of Southampton

Service Profile: "What does it do?"

High-level characterization/summary of a service

Used for

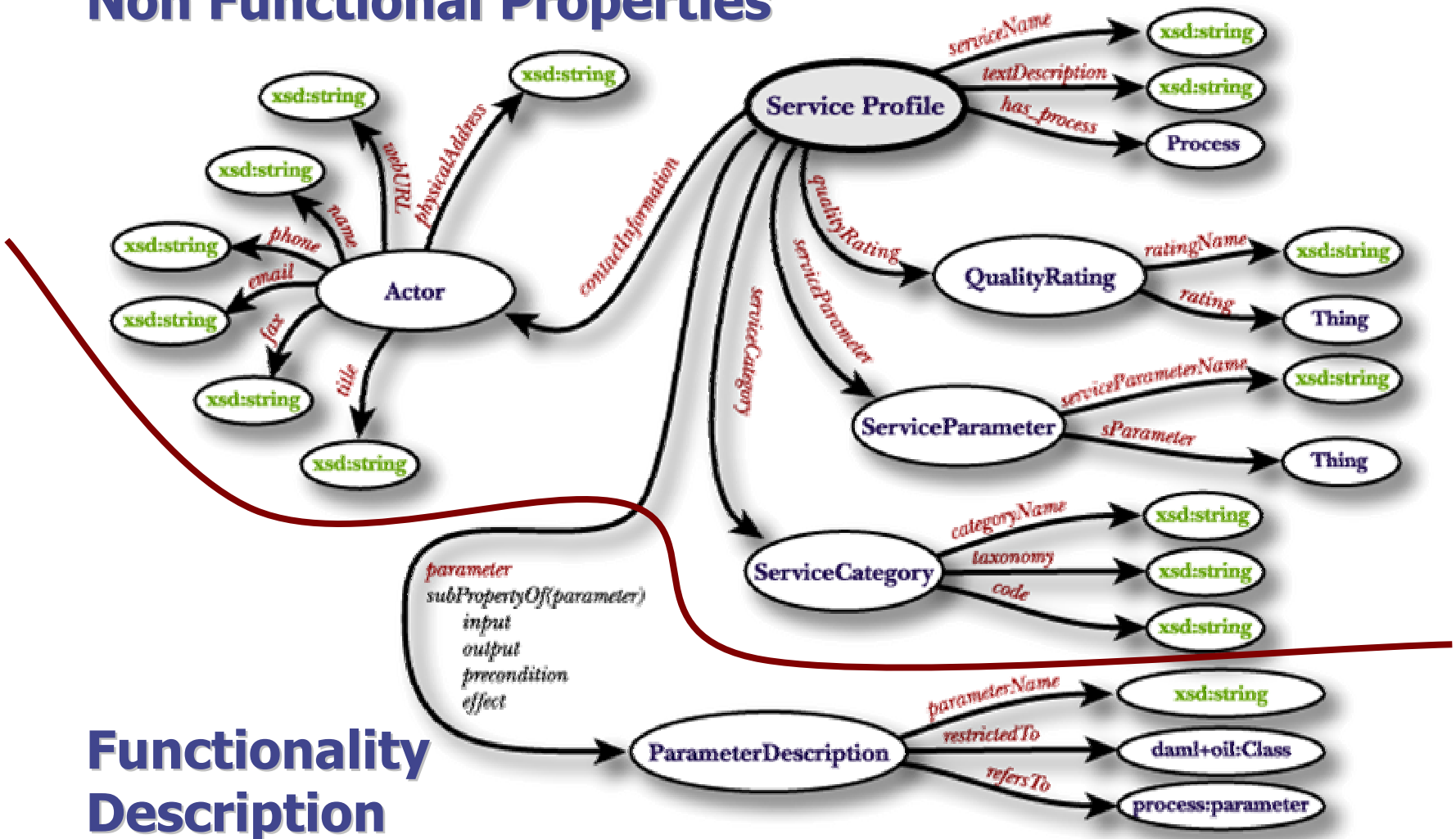
- Populating service registries
  - A service can have many profiles
- Automated service discovery
- Service selection (matchmaking)

One can derive:

- Service advertisements
- Service requests

Service Profile

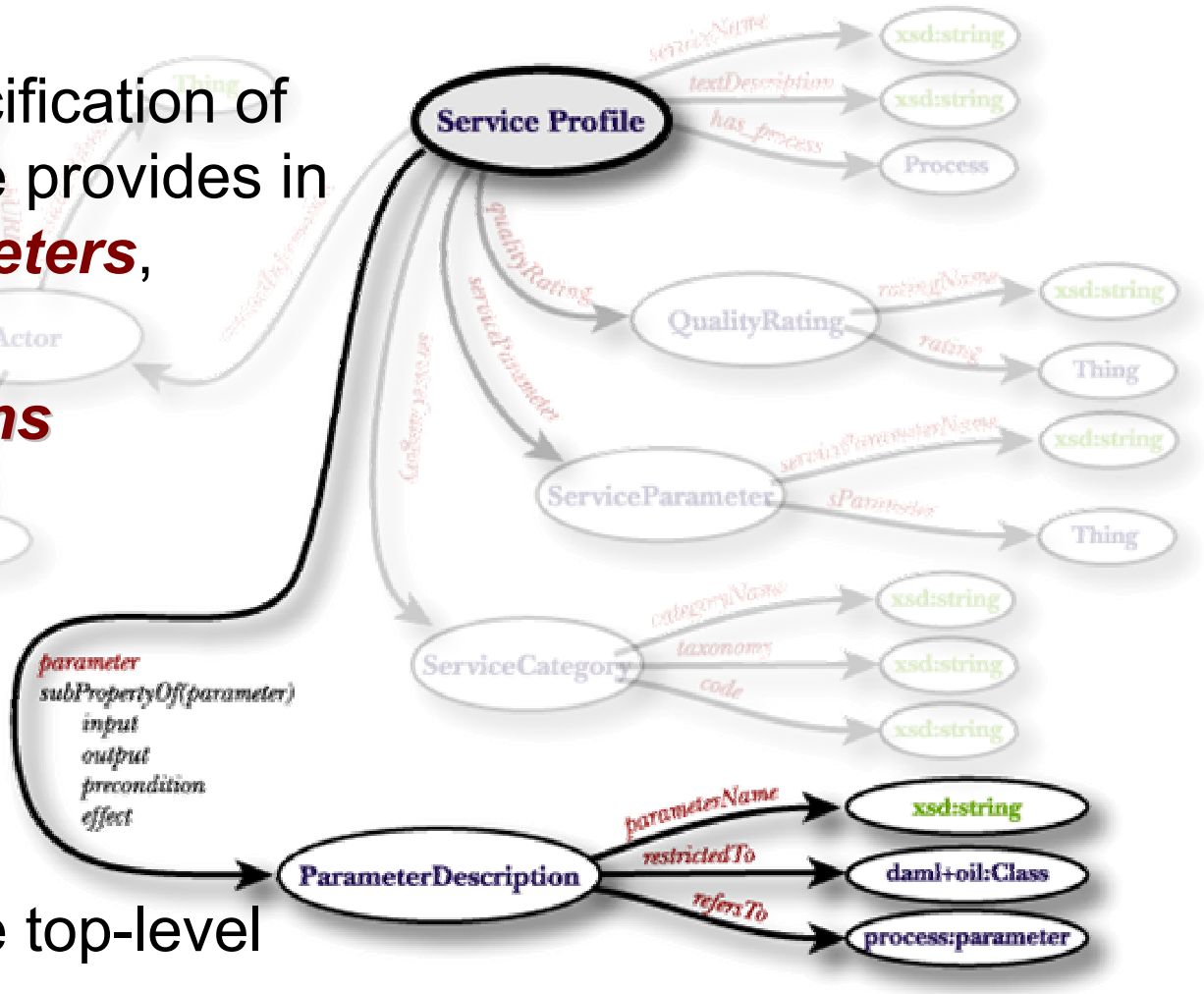
Non Functional Properties



Functionality Description

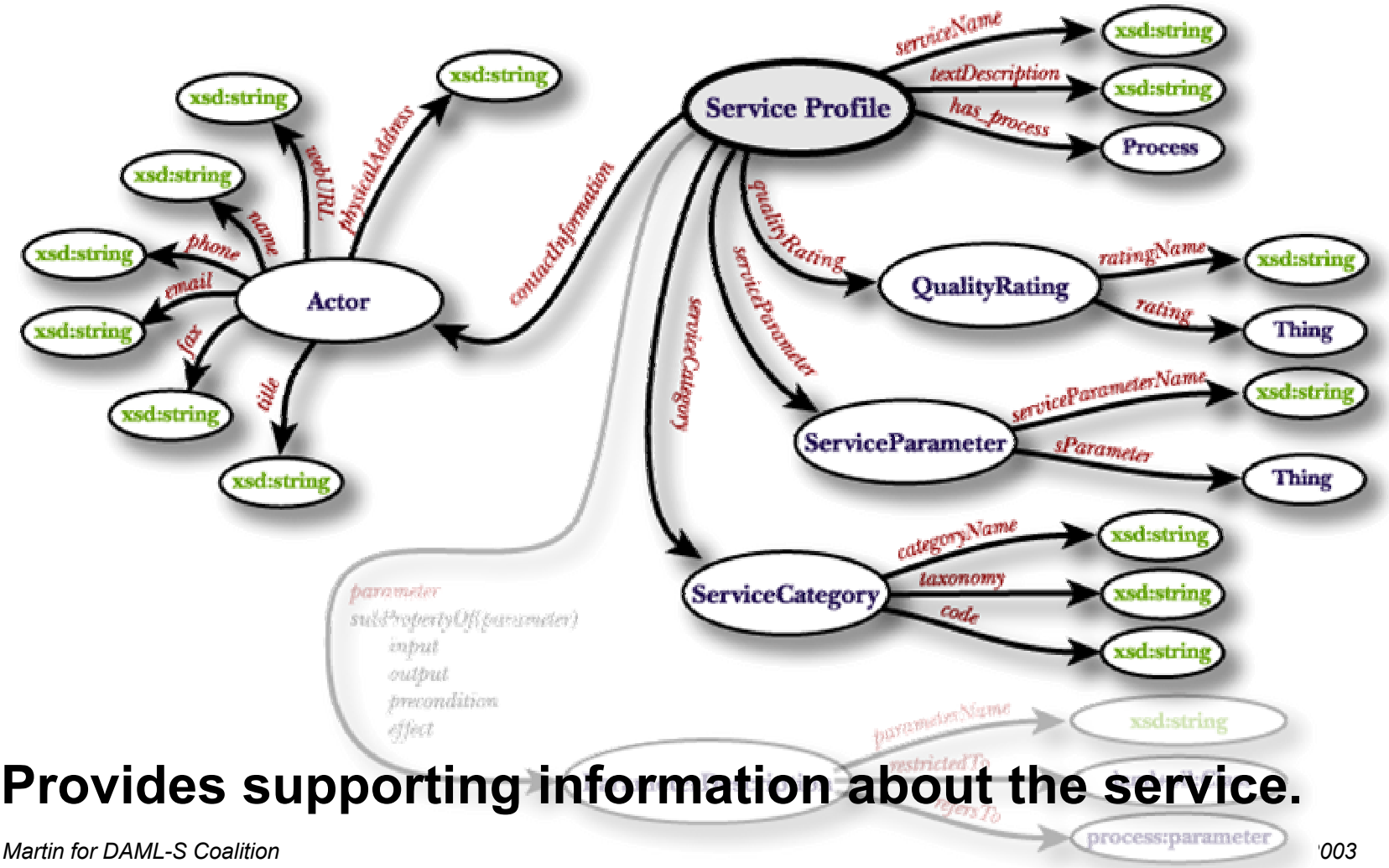
Service Profile: Functionality Description

- Functional Specification of what the service provides in terms of ***parameters***, subclassed as:
  - ***preconditions***
  - ***inputs***
  - ***outputs***
  - ***effects***



- Summarizes the top-level Process

Service Profile: NonFunctional Properties



- **Provides supporting information about the service.**

Service Profile: Styles of use

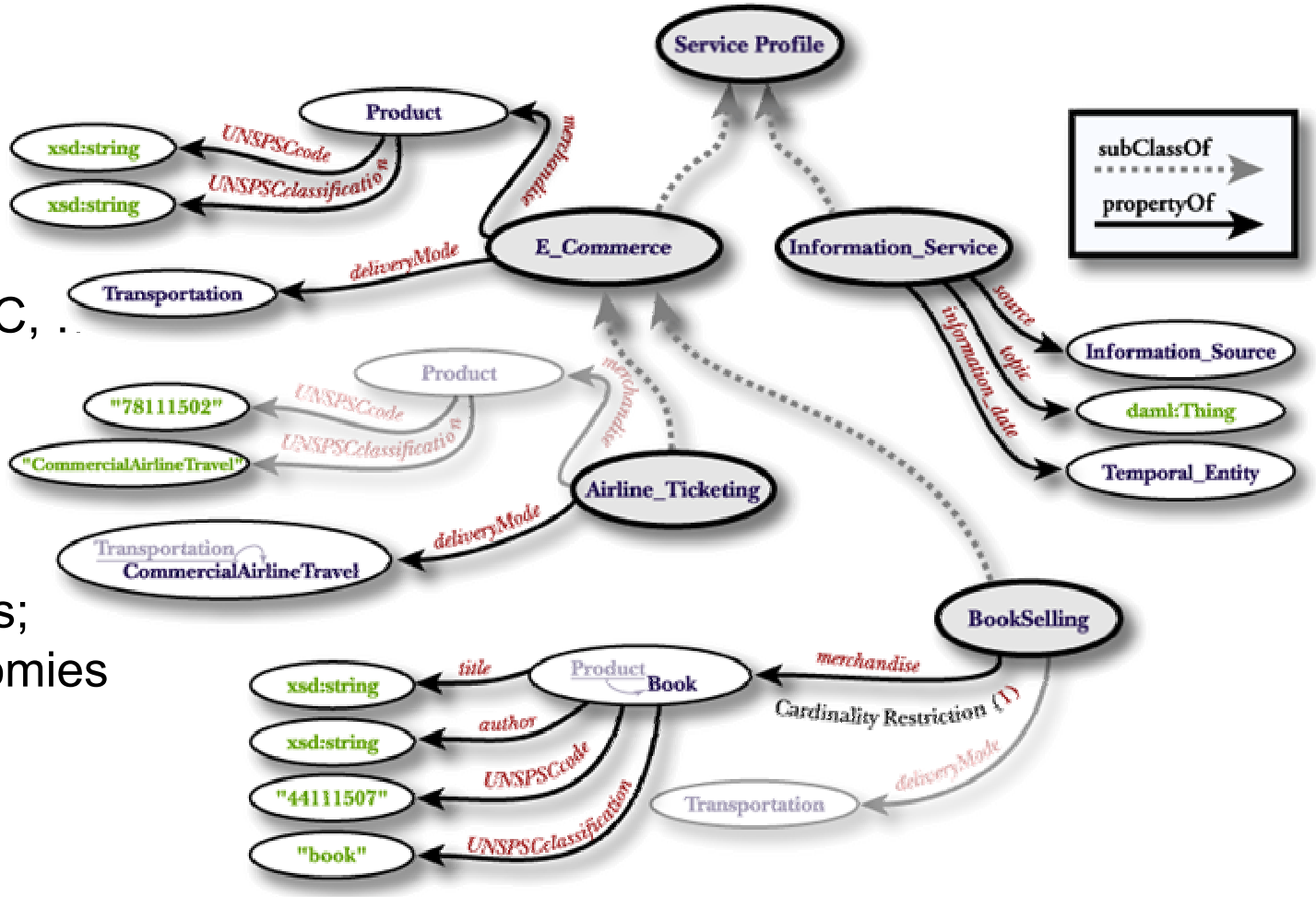
- Class hierarchical yellow pages
  - Implicit capability characterization
  - Arrangement of attributes on class hierarchy
  - Can use multiple inheritance
  - Relies primarily on “non-functional” properties
- Process summaries for planning purposes
  - More explicit
  - Inputs, outputs, preconditions, effects
  - Less reliance on formal hierarchical organization
  - Summarizes process model specs
  - Relies primarily on functional description

Exploiting Profile Hierarchies

Tie in with UDDI, UNSPSC, ..

DL Basis for matchmaking

Multiple profiles; multiple taxonomies



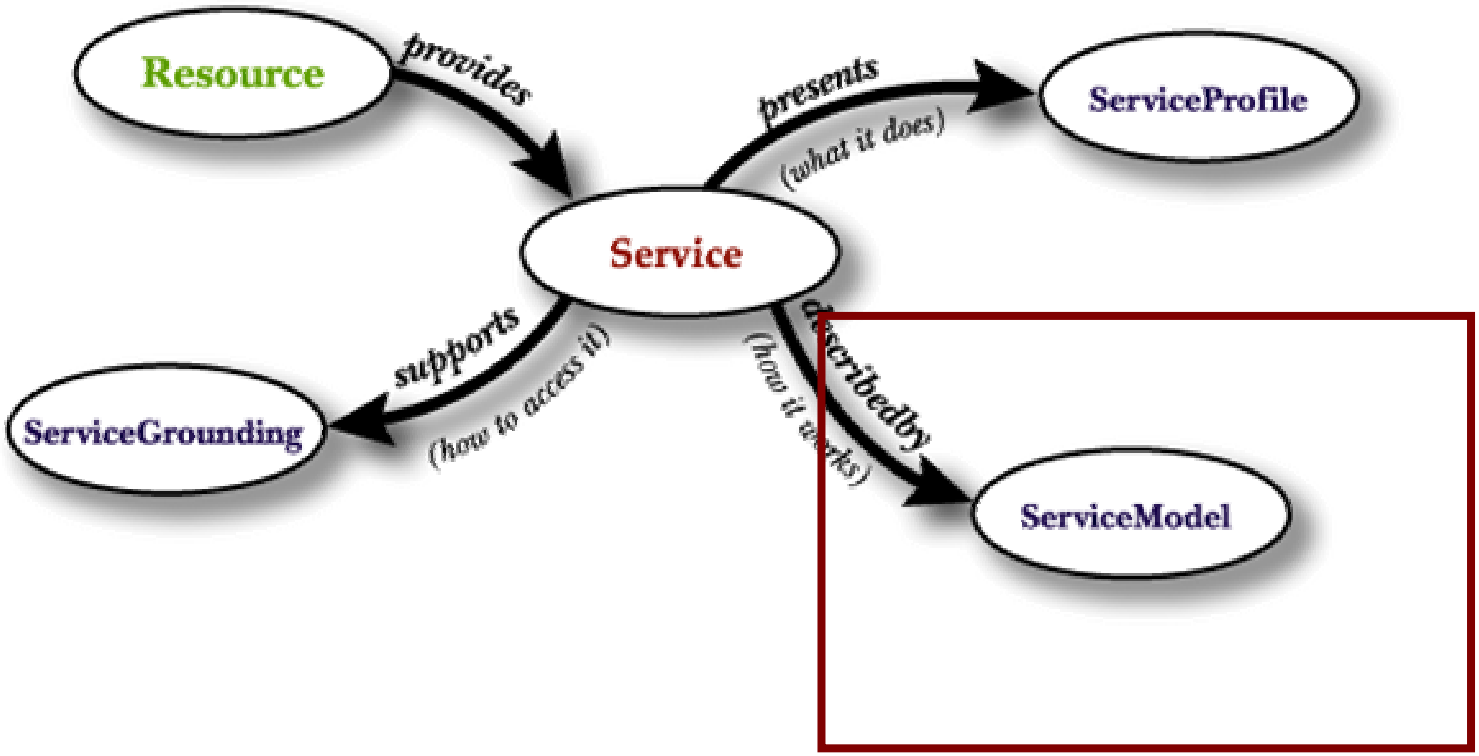
Profile: Recent Evolution

- IOPE Changes
  - Inputs, Outputs, Preconditions, Effects
  - Better integrated with Process Model
  - Relationship to Process Model clearer
- Better modularization

Profile: Issues

- Relationship to Process Model may need further clarification
- OWL is well-suited to characterizing & classifying services
- But greater expressiveness needed for many things (contracting & negotiation)

Upper Ontology of Services



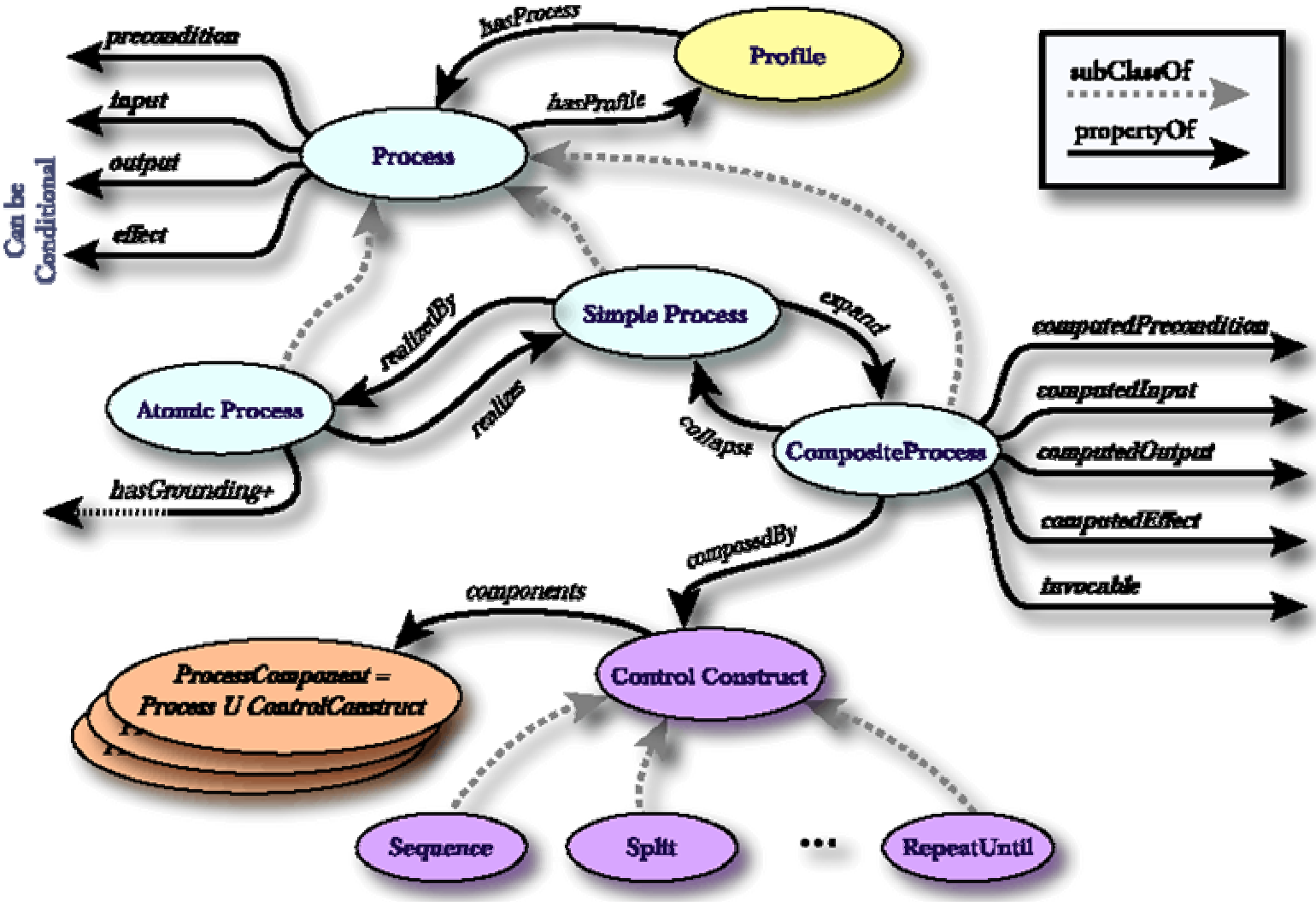
Process Model: "How does it work?"

Process
- Interpretable description of service provider's behavior
- Tells service user how and when to interact (read/write messages)

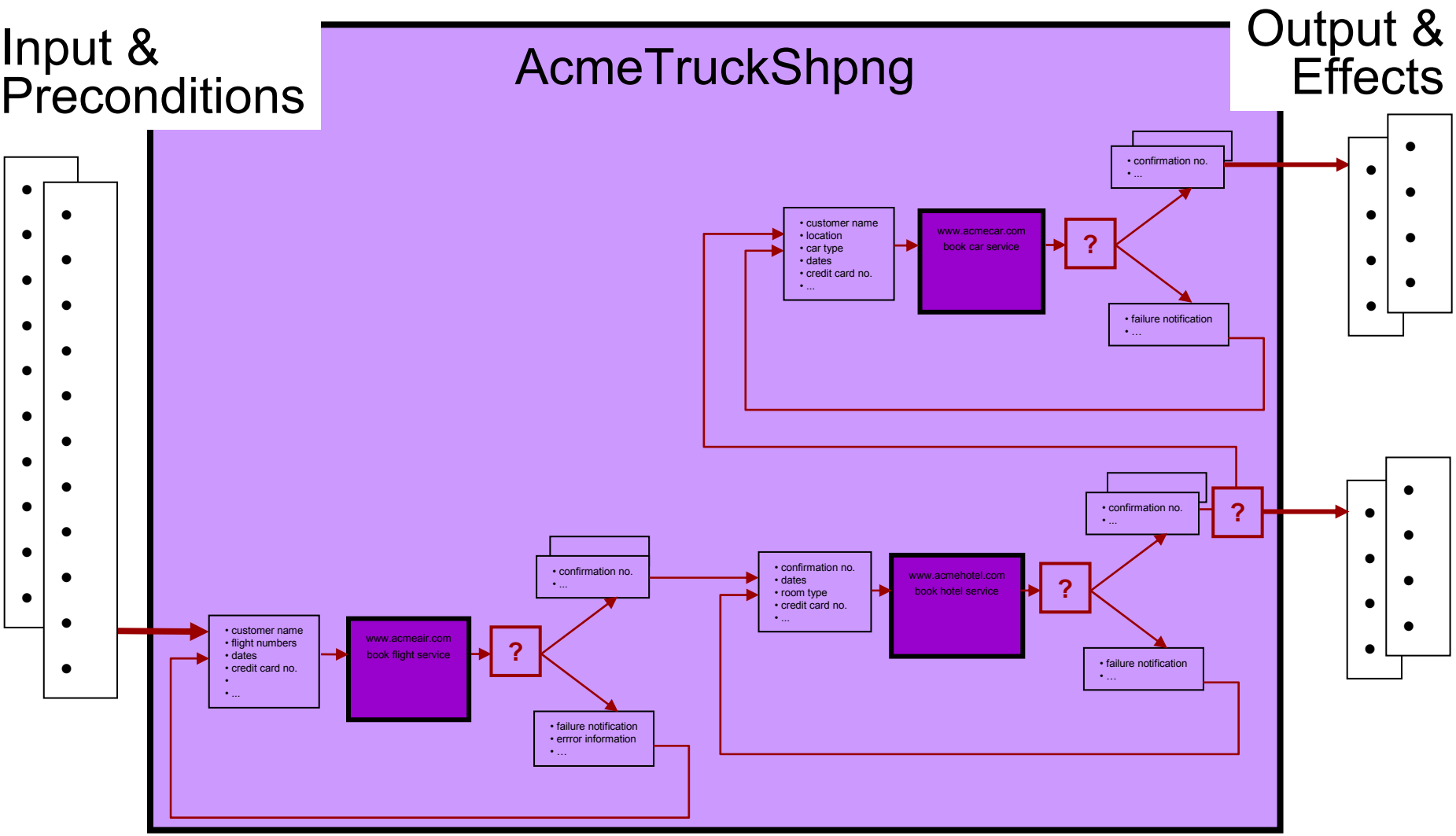
& Process control
- Ontology of process state; supports status queries
- (stubbed out at present)

- Used for:
  - Service invocation, planning/composition, interoperation, monitoring
- All processes have
  - Inputs, outputs, preconditions and effects
  - Function/dataflow metaphor; action/process metaphor
- Composite processes
  - Control flow
  - Data flow

Service Model / Process Model



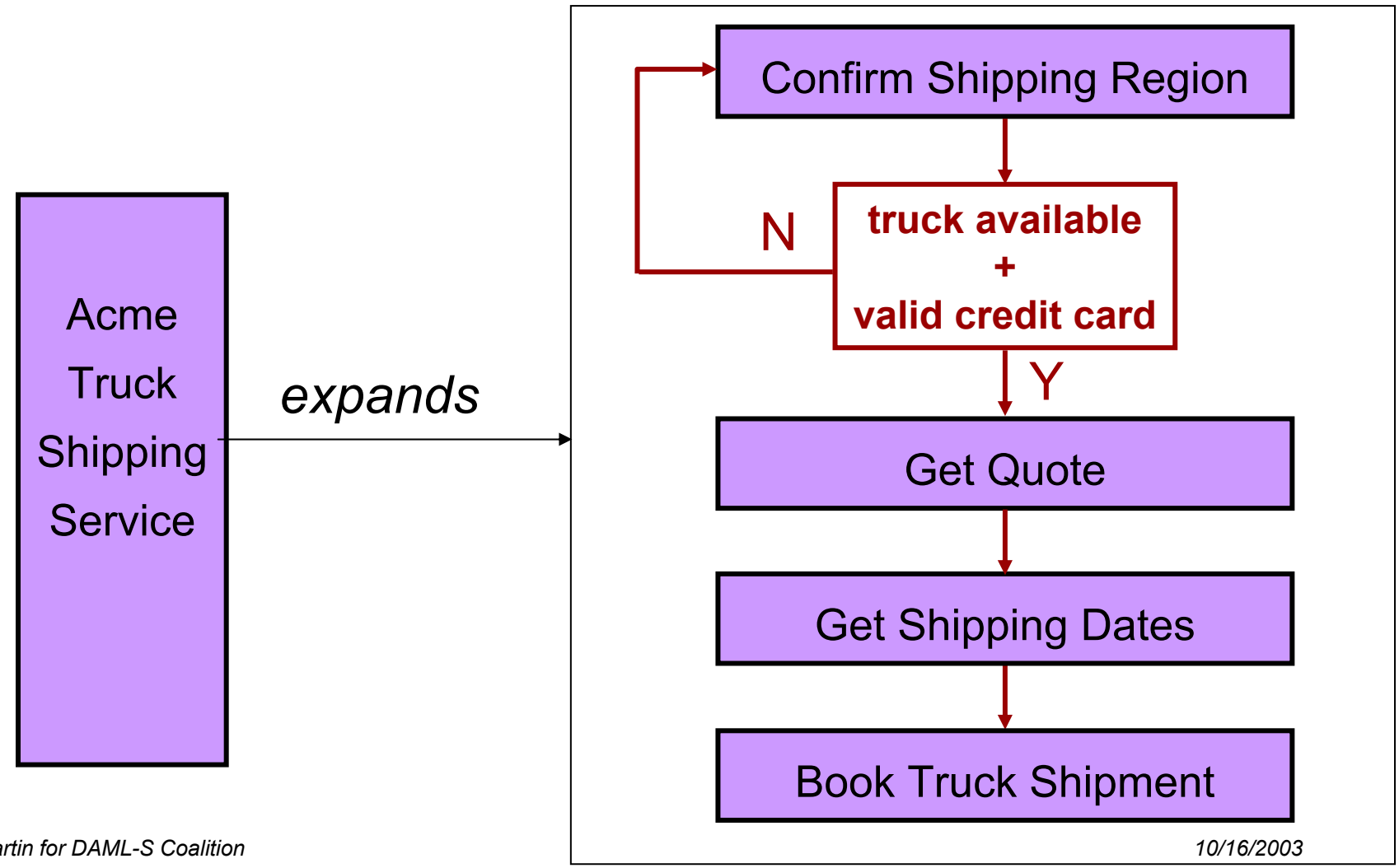
Composite Process



Simple and Composite Processes

AcmeTruckShpng

ExpandedAcmeTruckShpng



Process Model: Recent Evolution

- **Change to Processes-as-Instances**
  - **Pros**
    - **Simplified the means of expressing many things**
    - **Far more readable**
    - **More intuitive (for some of us)**
    - **PAC kept forcing us into OWL Full (or worse?)**
  - **Cons**
    - **Representation & reasoning of "execution traces" requires new work**
- **New IOPE constructs**
- **Simplification (IOPEs, deletion of synonyms)**

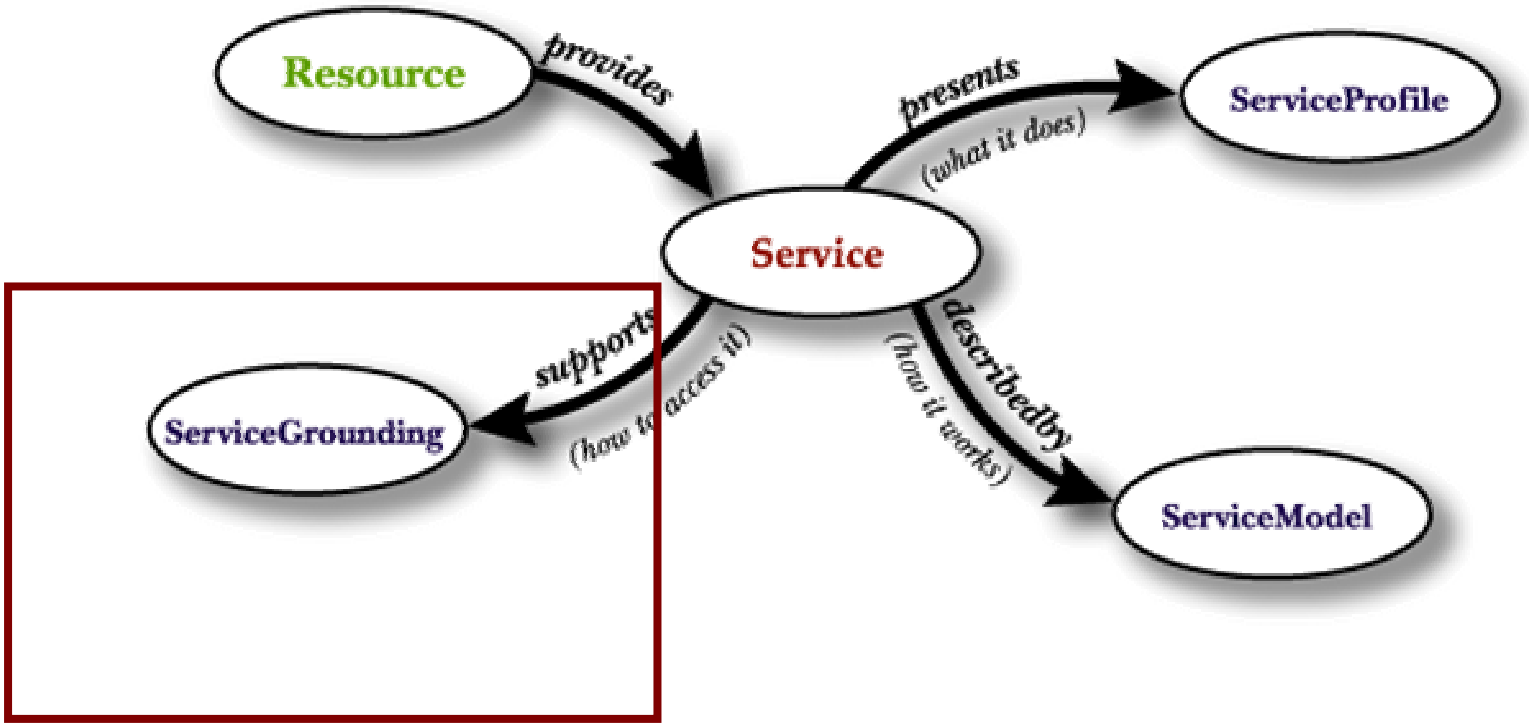
Process Model: Next Steps

- **New proposals for**
  - **Conditional (bundled) O/E**
    - **Faults**
  - **Nicer conditional statements**
  - **Synchronization constructs**
  - **More explicit messaging**
  - **Dataflow**
  - **Surface syntax**

Process Model: Other issues

- Standardization efforts
- Grid / OGSA tie-in
- More clarity needed on options for expressing conditions/ effects
- Execution traces
- Process control (lifecycle) / monitoring

Upper Ontology of Services



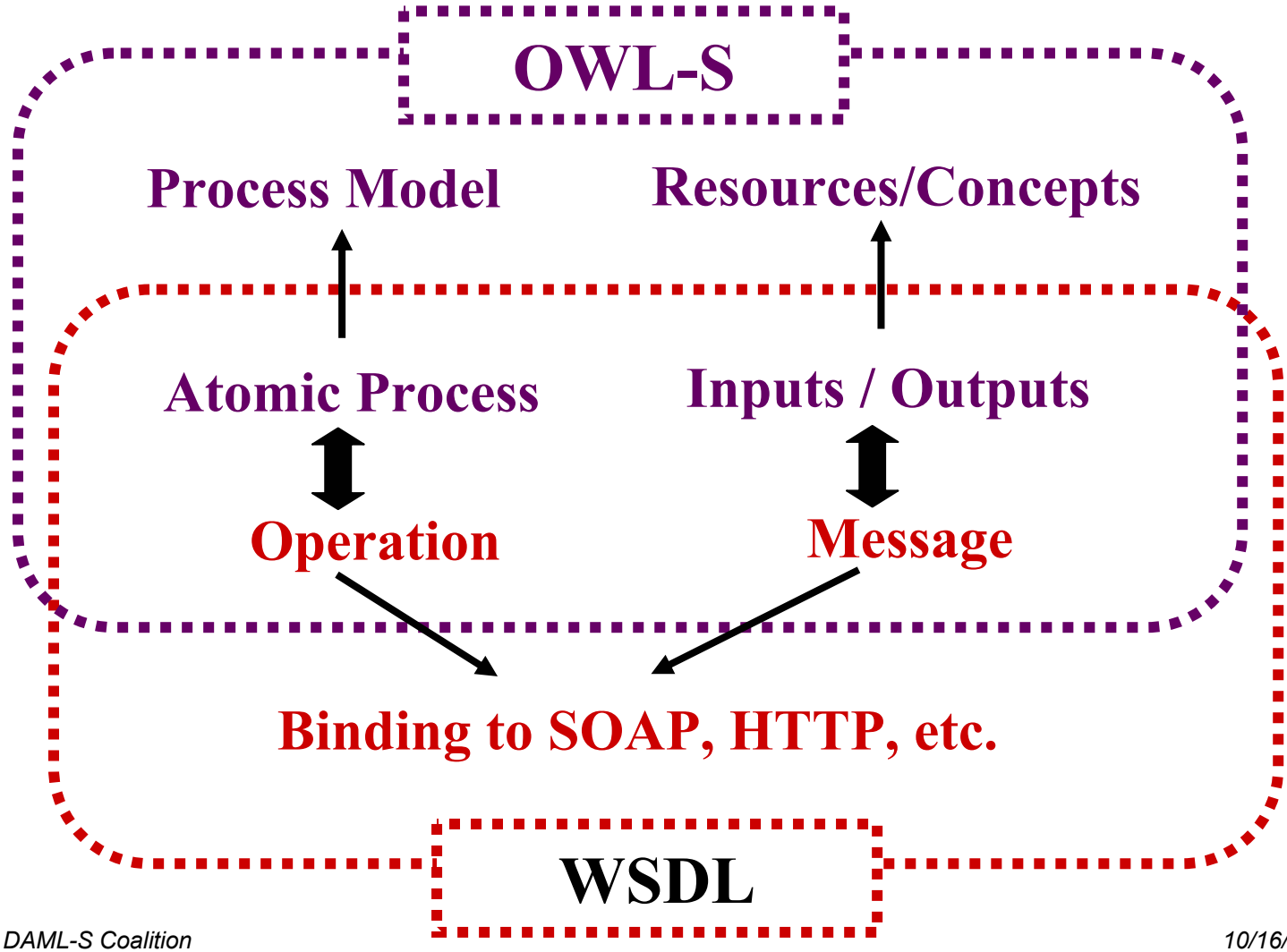
Service Grounding: "How to access it"

- Implementation-specific
- Message formatting, transport mechanisms, protocols, serializations of types
- Service Model + Grounding give everything needed for using the service
- Builds upon WSDL

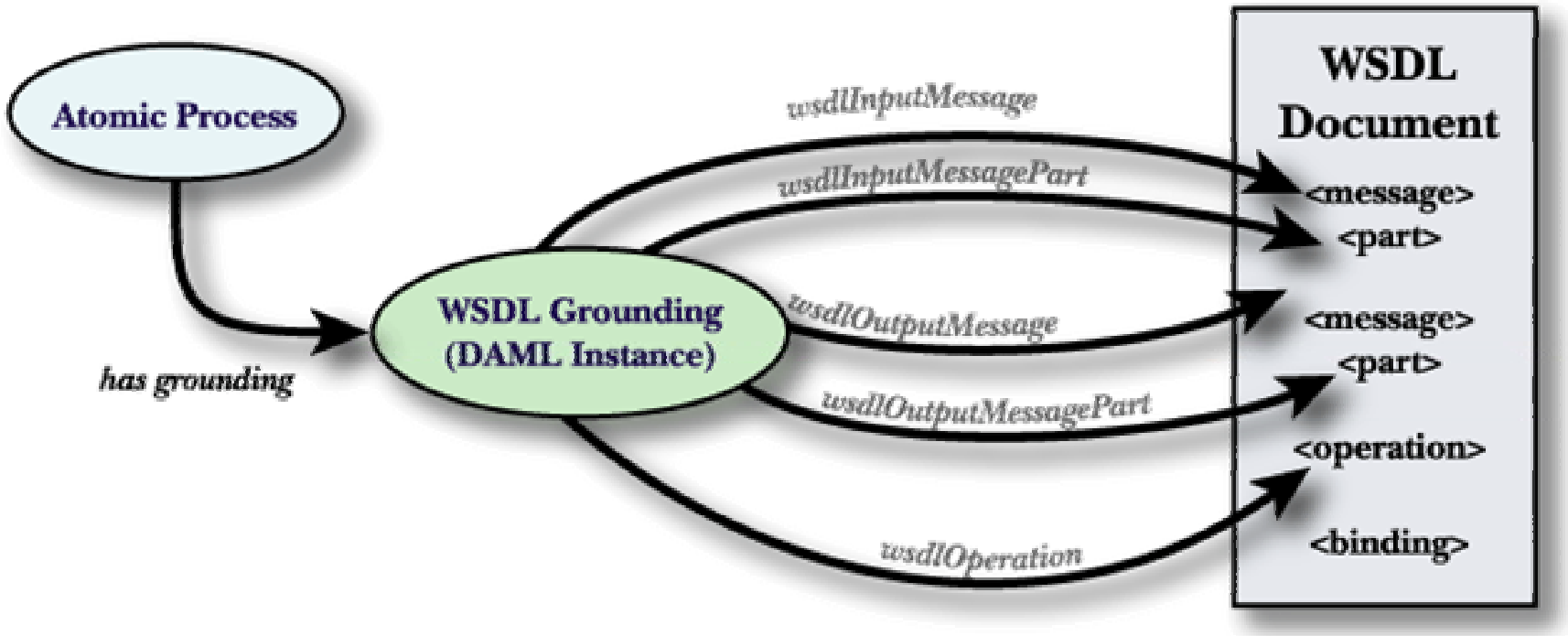
OWL-S / WSDL Grounding

- Web Services Description Language
  - Authored by IBM, Ariba, Microsoft
  - Focus of W3C Web Services Description WG
  - Commercial momentum
  - Specifies message syntax accepted/generated by communication ports
  - *Bindings* to popular message/transport standards (SOAP, HTTP, MIME)
  - Abstract “types”; extensibility elements
- *Complementary* with OWL-S

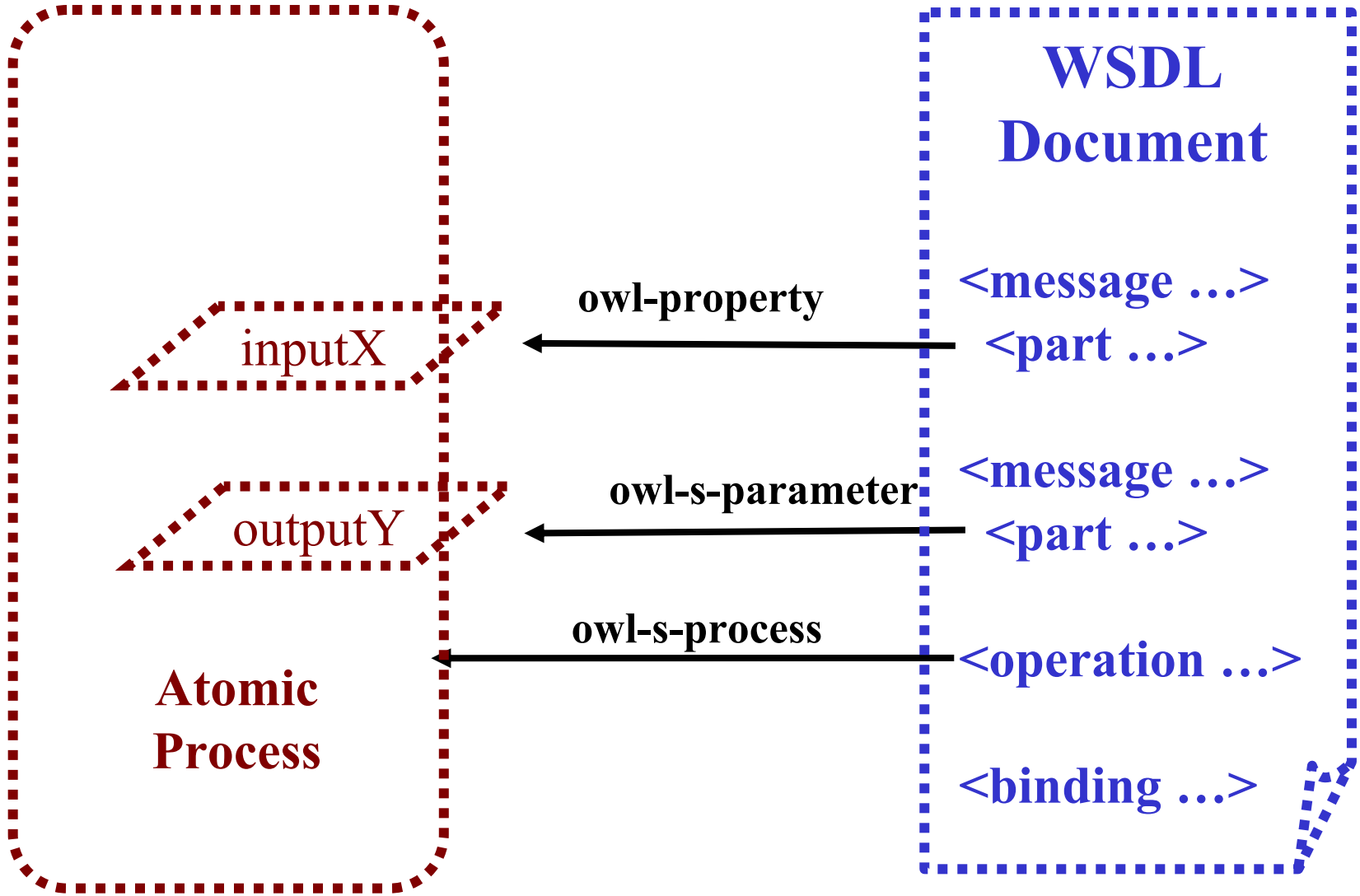
OWL-S / WSDL Grounding



OWL-S / WSDL Grounding (cont'd)



OWL-S / WSDL Grounding (cont'd)



Grounding: Recent Evolution

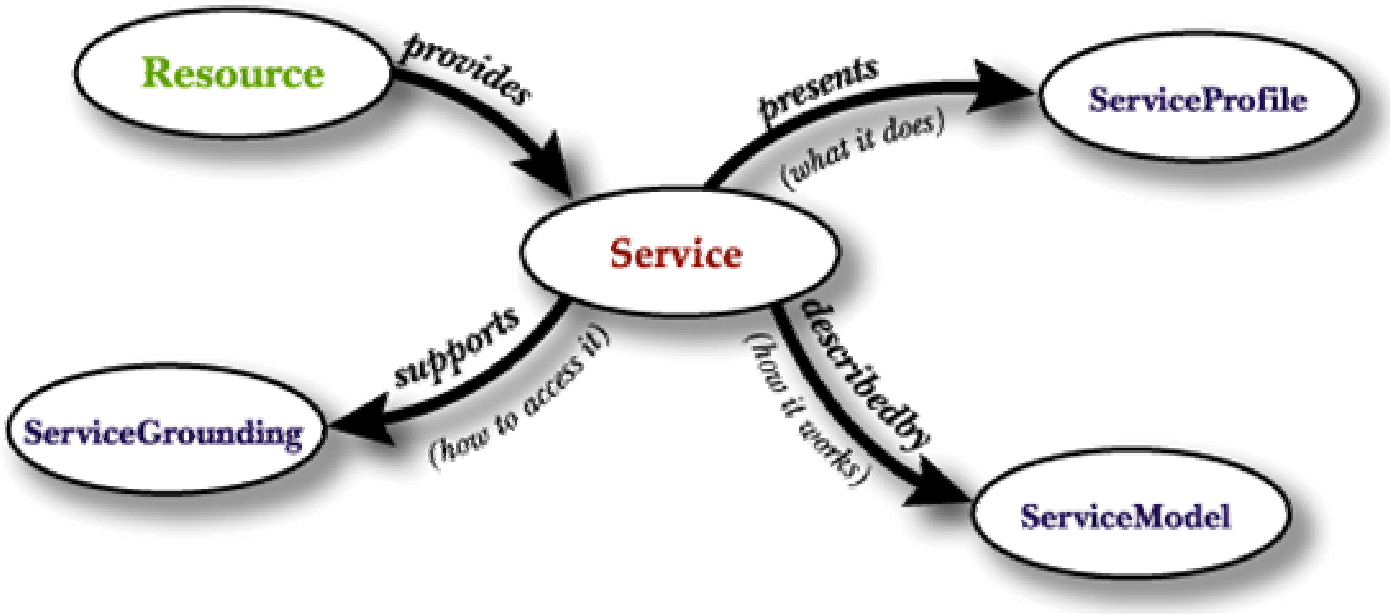
- Minor adjustments for Processes-as-Instances
- Proposal to correlate conditional outputs with WSDL fault messages

Grounding: Issues

Issue: waiting for WSDL

- May generate new WSDL requirements (e.g. for conditional inputs)

Review: Upper Ontology of Services



Global Issues

- How to use the language we have
  - Sometimes convoluted representations
  - How much DL-based reasoning do we need?
- How to go beyond the language we have
  - Modular approach
  - Does OWL Full use preclude extensions, and how significant is this?
- Compatibility with (commercial) Web Services
- Making the case for SWS

DAML-S/OWL-S Path of Evolution

Release 0.5 (May 2001)
- Initial Profile & Process ontologies

Release 0.6 (December 2001)
- Refinements to Profile & Process; Resources ontology

Release 0.7 (October 2002)
- Initial DAML-S/WSDL Grounding;
- Profile, Process Model refinements; more complete examples

Release 0.9 (May 2003)
- Grounding: greater generality, flexibility
- Initial work on expressing conditions, security
- More community support (contributions pages)

Release 1.0 (October 2003)
- DAML-S ➔ OWL-S completed
- Processes-as-instances
- New IOPE classes
- Initial version of surface syntax
- Profile reorganization

1.0 Beta Release

- Under construction
  - Core ontology files
  - Not all documentation is complete
- OWL-final
- Profile
- Process Model
- Grounding

www.daml.org/services/owl-s/1.0