OWL-S /SWSL Briefing

DAML Web Services Coalition

Presented by:
David Martin (SRI)

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

Top-level Outline

- **Language status (25 min.)**
  - **OWL-S Status & Evolution (David Martin)**
  - **New features of process model; surface language (Mark Burstein)**
- **Security extensions (Tim Finin) (20)**
- **Supporting products (Katia Sycara) (10)**
  - **Tools, demos, use cases**
- **Outreach & uptake (Katia) (10)**
  - **Standardization efforts & strategies**
  - **Users, workshops, books, papers**
- **Open issues & challenges (David) (10)**
- **Roadmap for language evolution (David) (15)**
  - **Transition to SWSL**

DAML Services Coalition

BBN: Mark Burstein*

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

De Montfort University: Monika Solanki

Maryland / College Park: Bijan Parsia, Evren Sirin

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

Top-level Outline

- **Language status**
  - **OWL-S 1.1**
    - **Profile, Process Model, Grounding: Overview, recent evolution, next steps for each**
    - **Release status**
- **Security extensions**
- **Supporting products**
- **Outreach & uptake**
- **Open issues & challenges**
- **Roadmap for language evolution**

OWL-S Objectives

Fuller automation of service use by agents

- Ideal: full-fledged use of services never before encountered:

    - Discovery, selection, composition, invocation, monitoring, ..

Useful in the "real world"

- Compatible with industry standards
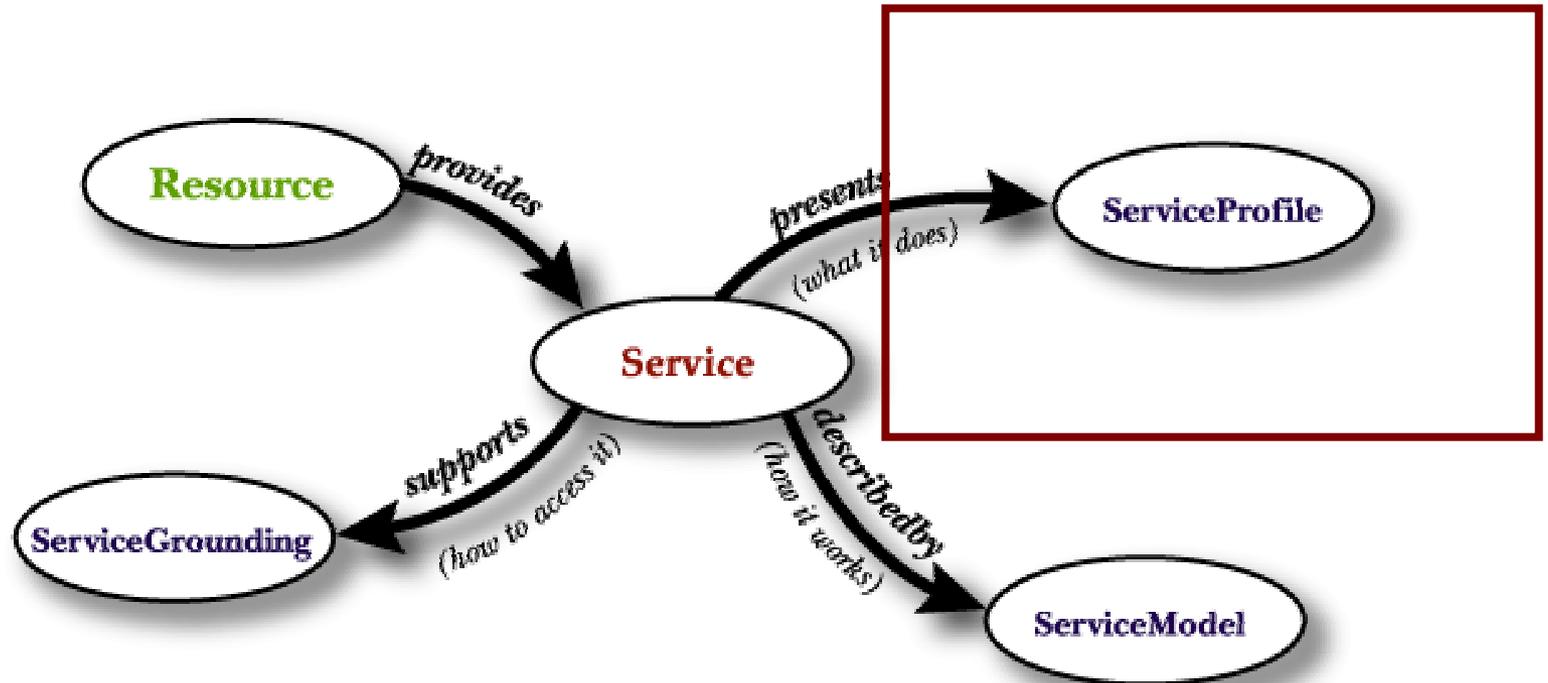- Incremental exploitation

Enable reasoning/planning about services

- e.g., On-the-fly composition

Integrated use with information resources

Ease of use; powerful tools

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
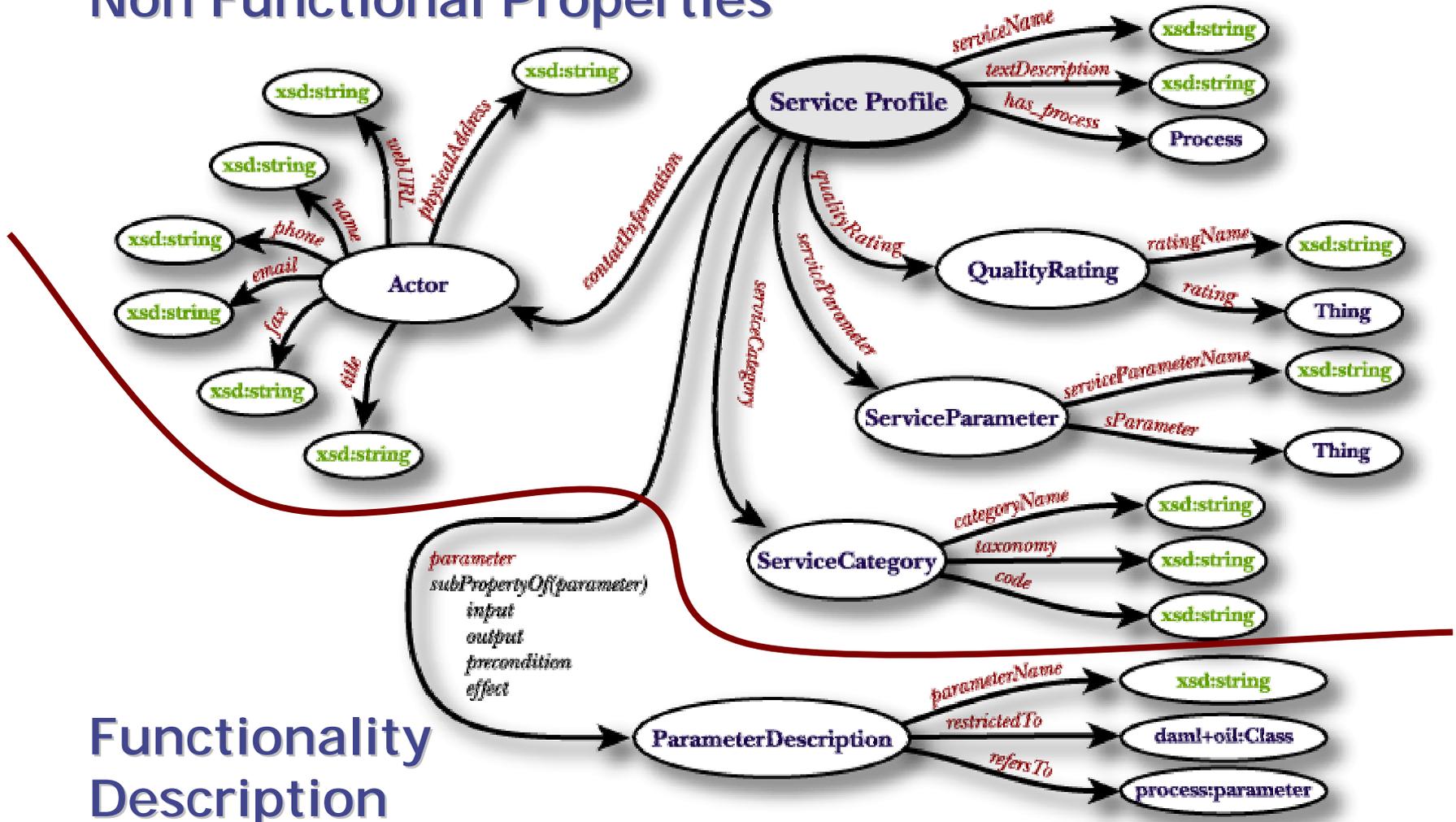
Used for

- 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
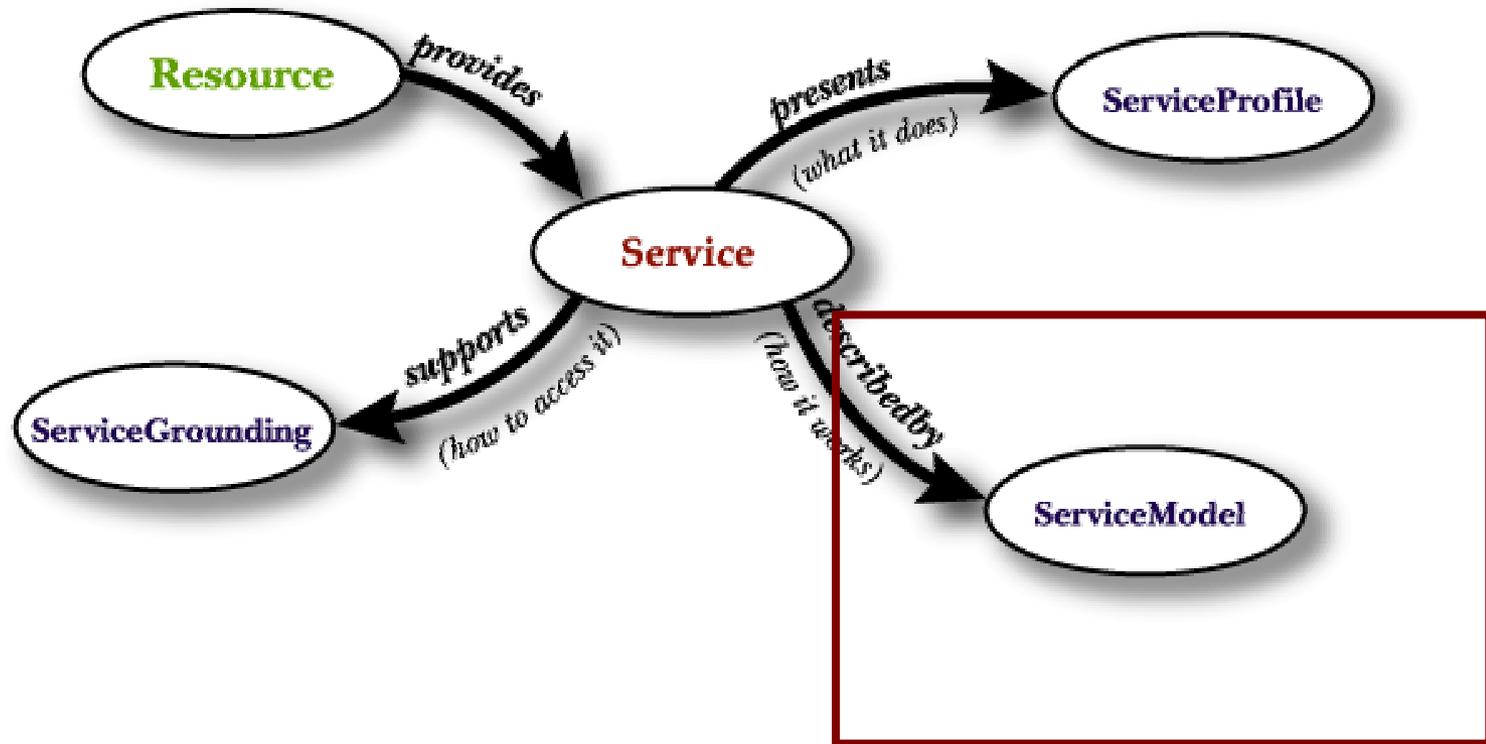
Functionality Description

Service Profile: 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

Profile: Recent Evolution

- DLization

  - Now editable using Protege

- (otherwise, stable)

Upper Ontology of Services

Resource —provides→ Service

Service —presents (what it does)→ ServiceProfile

Service —supports (how to access it)→ ServiceGrounding

Service —describedby (how it works)→ ServiceModel

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
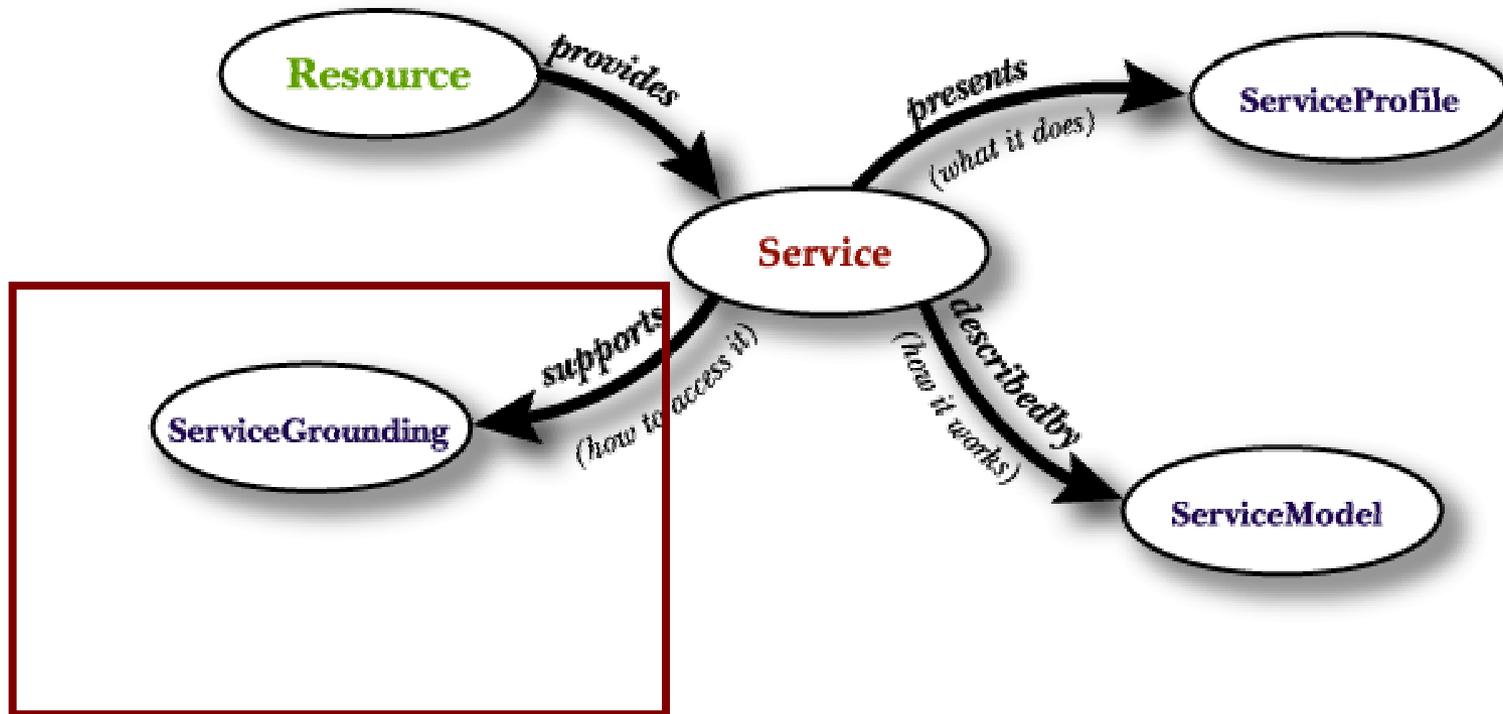
• Composite processes
- Control flow
- Data flow

Service Model / Process Model
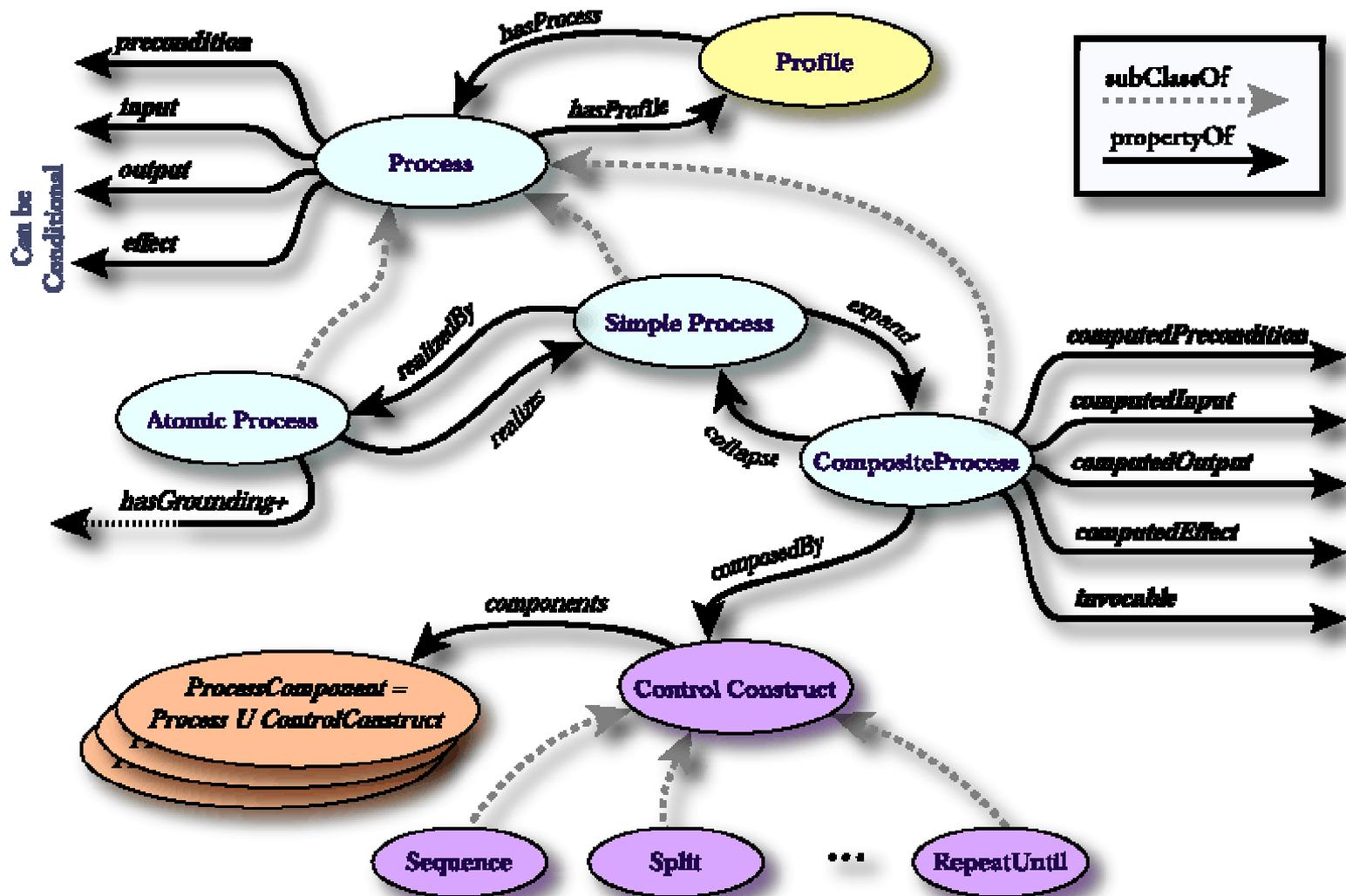
Process Model: Recent Evolution

- DLization
- Expression language
- Conditionalizing outputs and effects
- *Perform* construct
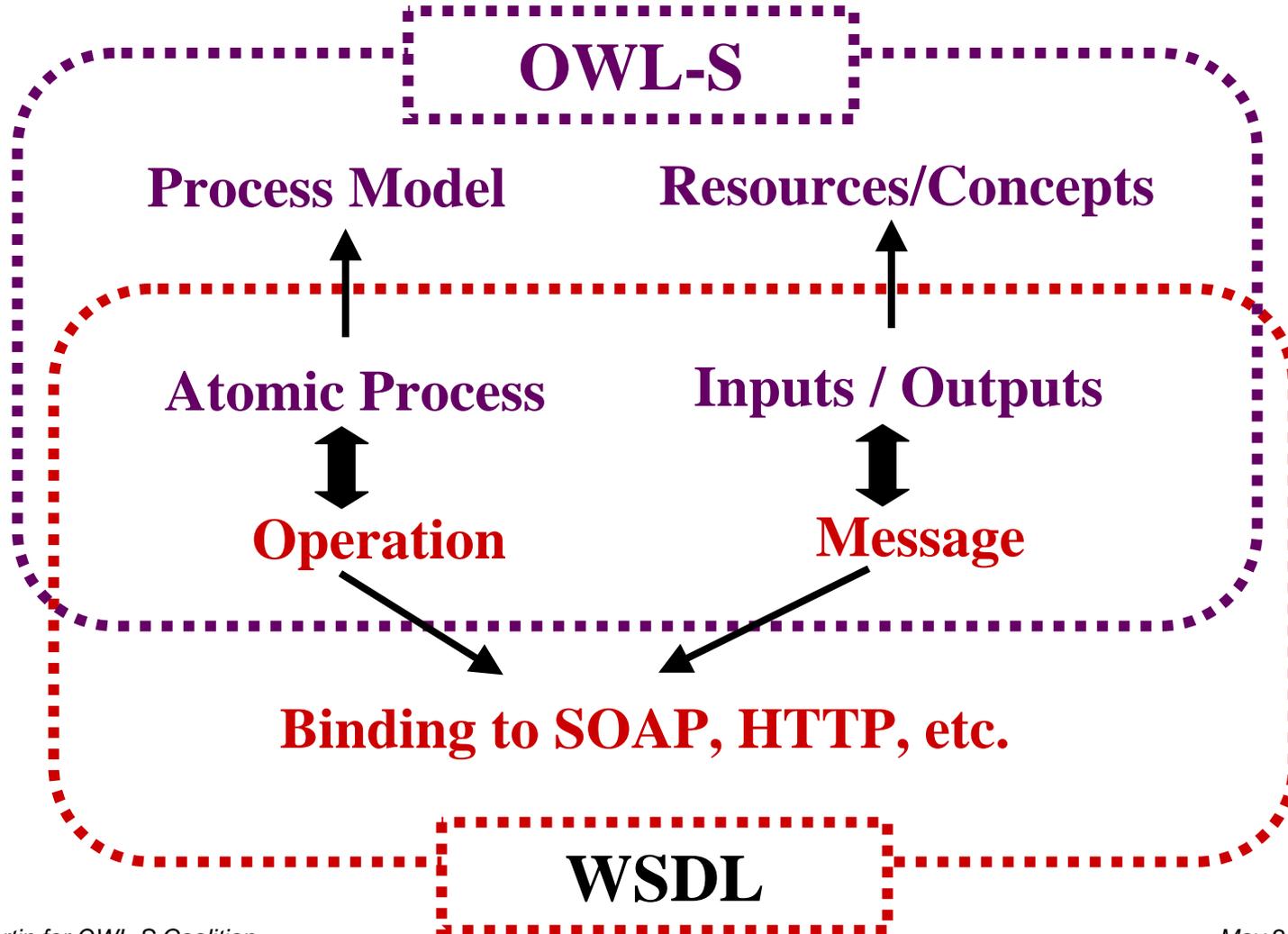- Dataflow and bindings
- Surface syntax
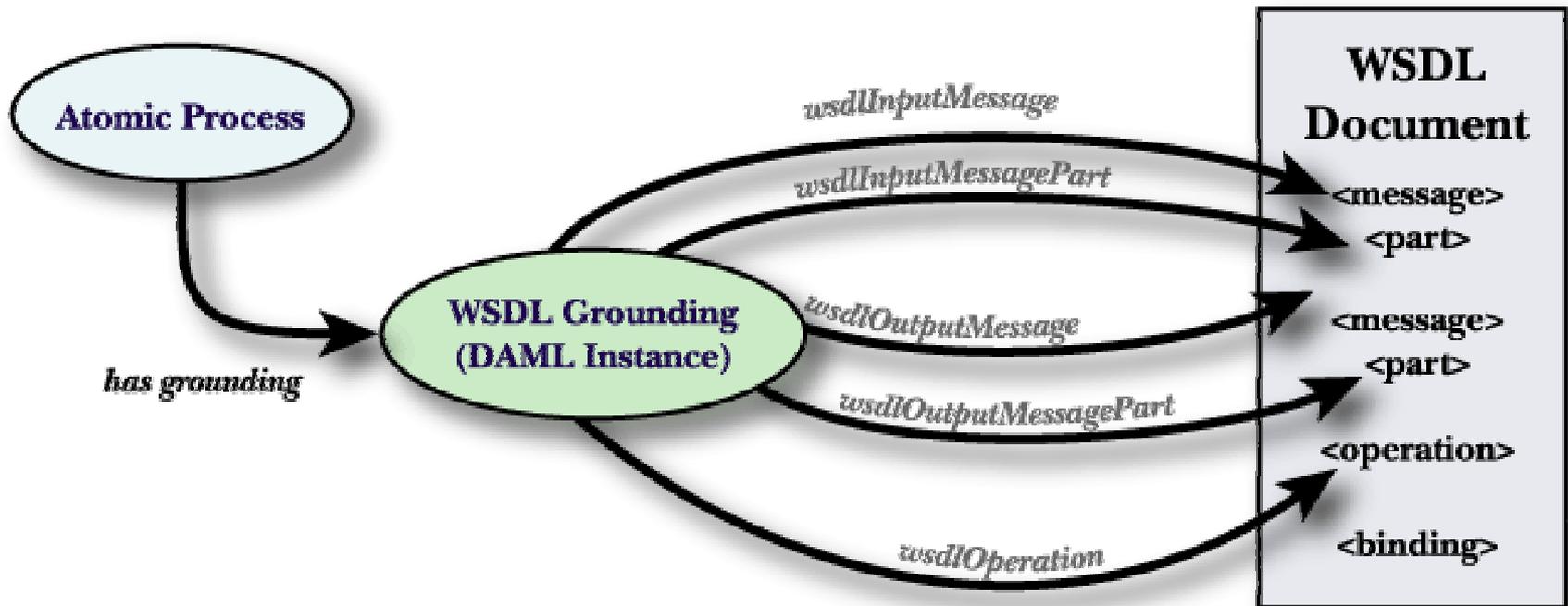
Upper Ontology of Services

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

OWL-S

Process Model   **Resources/Concepts**

Atomic Process   **Inputs / Outputs**

Operation   **Message**

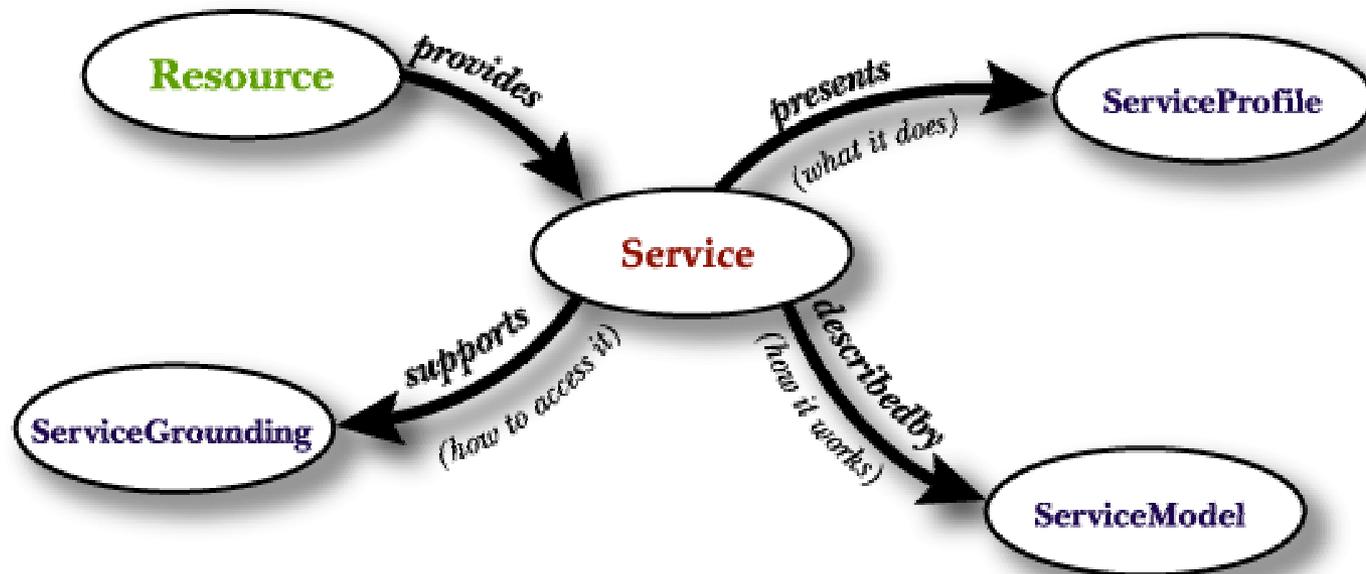**Binding to SOAP, HTTP, etc.**

WSDL

OWL-S / WSDL Grounding (cont'd)

Grounding: Recent Evolution

- DLization

Review: Upper Ontology of Services

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

Release 1.1 (June 2004)

Completion of features required for coherence of process model

Transition to SWSL

1.1 Release

- **Under construction**

  - Core ontology files mostly done

  - Examples under revision

  - Not all documentation is complete

- **Not yet announced**

  - Release scheduled for June 4

  www.daml.org/services/owl-s/1.1

Top-level Outline

- **Language status**
  - **OWL-S 1.1**
    - **Profile**, **Process Model**, **Grounding:**
      Overview, recent evolution, next steps for each
    - **Release status**
- **Security extensions**
- **Supporting products**
- **Outreach & uptake**
- **Open issues & challenges**
- **Roadmap for language evolution**

Top-level Outline

- **Language status (25 min.)**
  - **OWL-S Status & Evolution (David Martin)**
  - **New features of process model; surface language (Mark Burstein)**
- **Security extensions (Tim Finin) (20)**
- **Supporting products (Katia Sycara) (10)**
  - **Tools, demos, use cases**
- **Outreach & uptake (Katia) (10)**
  - **Standardization efforts & strategies**
  - **Users, workshops, books, papers**
- **Open issues & challenges (David) (10)**
- **Roadmap for language evolution (David) (15)**
  - **Transition to SWSL**