Semantic Web Services

OWL-S Coalition

Semantic Web Services Initiative

David Martin, Drew McDermott,
Grit Denker, Katia Sycara, Rick Hull, Mark
Burstein

Outline

- OWL-S Status
  - Overview & Recent Milestones [David, 10]
  - Recent Evolution of Process Model [Drew, 10]
- Security Extensions [Grit, 15]
- Outreach, Tools, Standards [Katia, 15]
- SWSI Status
  - SWSL Overview [David, 8]
  - SWSL Ontology (FLOWS) [Rick,15]
  - SWSL Rules will be presented tomorrow
  - SWSA [Mark, 10]
- Open Issues and Roadmap [David, 10]

Outline

- OWL-S Status
  - **Overview & Recent Milestones [David, 10]** 🡐
  - Recent Evolution of Process Model [Drew, 10, 9:10]
- Security Extensions [Grit, 15]
- Outreach, Tools, Standards [Katia, 15]
- SWSI Status
  - SWSL Overview [David, 8]
  - SWSL Ontology (FLOWS) [Rick,12]
  - SWSA [Mark, 10]
- Open Issues and Roadmap [David, 10]

Participants

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

Yale: Drew McDermott*

A Few Highlights

- W3C member submission
- 1.1 release finalized
- Book published: “Developing Semantic Web Services“ by Peter H. Alesso & Craig F. Smith (A.K. Peters)
- Successful activities at ISWC
  - SWS workshop & tutorial had by far the best attendance
- Papers at ISWC main conference
  - Many employing / extending OWL-S
- SWS workshop planned for WWW 2005
- Significant "real-world" applications
  - Fujitsu Task Computing
  - FCS and other Army work at TARDEC
- Many more examples in Katia’s presentation

Recent Milestones

- W3C member submission (Nov. 2)
  - 9 sponsoring members
    - France Telecom, MIND Lab at the University of Maryland, National Institute of Standards and Technology (NIST), Network Inference, Nokia, SRI International, Stanford University, Toshiba Corporation, University of Southampton
  - Update planned (next 1 - 3 months)
    - Add surface syntax
    - Add 2 or 3 additional commercial sponsors (likely)
    - A few tweaks
  - Further updates possible with additional features
  - Planning workshop on Semantic Web Services

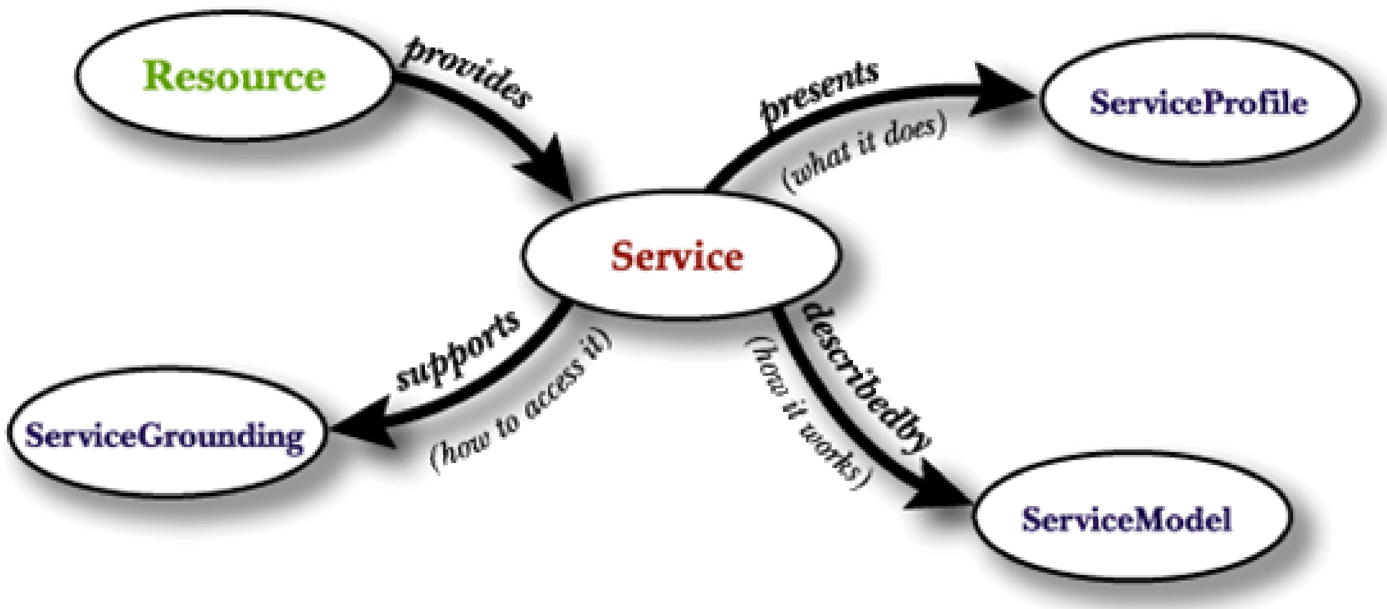
Recent Milestones (2)

- 1.1 release finalized
- New features:
  - Development of presentation syntax (Drew)
  - More expressive examples
  - Profile: Added properties serviceClassification and serviceProduct.
  - Process: Added the "Produce" control construct.
  - Process: Made Repeat-Until and Repeat-While subclasses of Iterate.
  - Process: Added "valueType" as a property of Binding.
  - Process: Refined the definitions of InputBinding, OutputBinding.
  - Process: Eliminated "chosen" property (of Choose).
  - Process: Changed domain of "components" (from ControlConstruct to the union of selected control constructs).
  - Process: Renamed Unordered to Any-Order and clarified its definition.
- Note: other materials on OWL-S Web site

OWL-S (and SWSL) Objectives

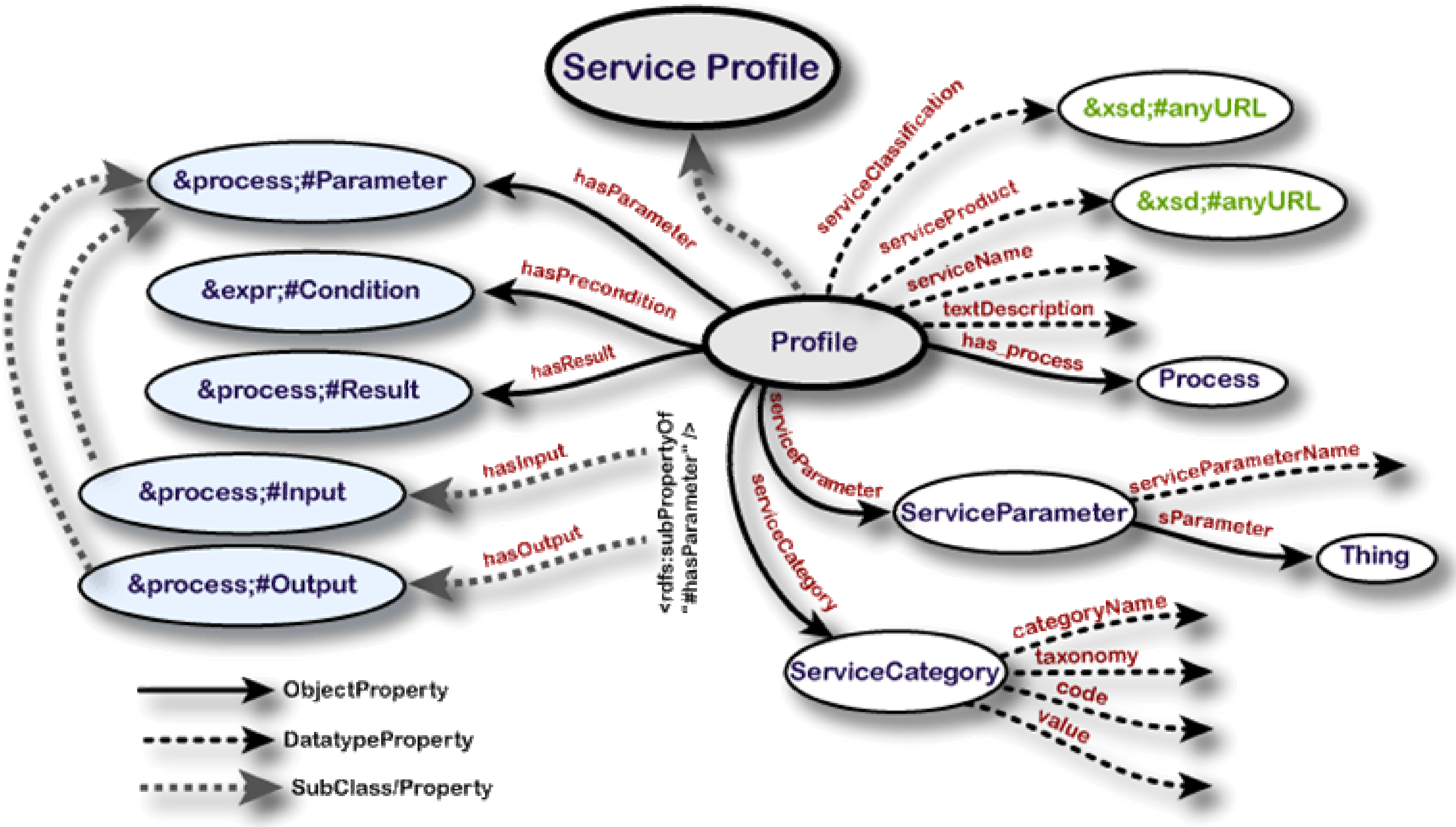
- Automation of service use by software agents
  - Ideal: full-fledged use of services never before encountered:
    - Discovery, selection, composition, invocation, monitoring, ..
- Useful in the “real world”
  - Compatible with industry standards
  - Incremental exploitation
- Enable reasoning/planning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

Upper Ontology of Services

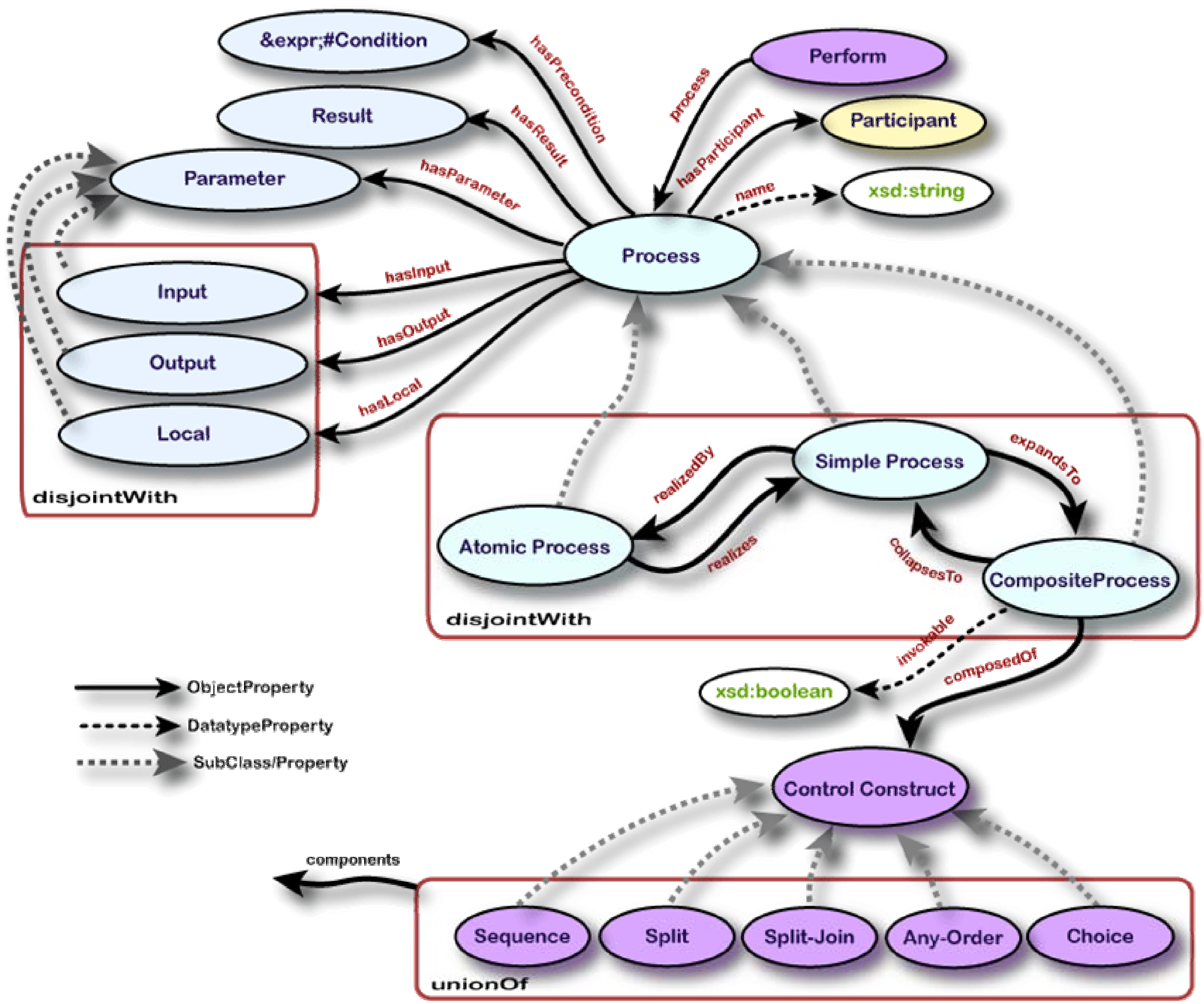


Ontology images compliments of Terry Payne, University of Southampton

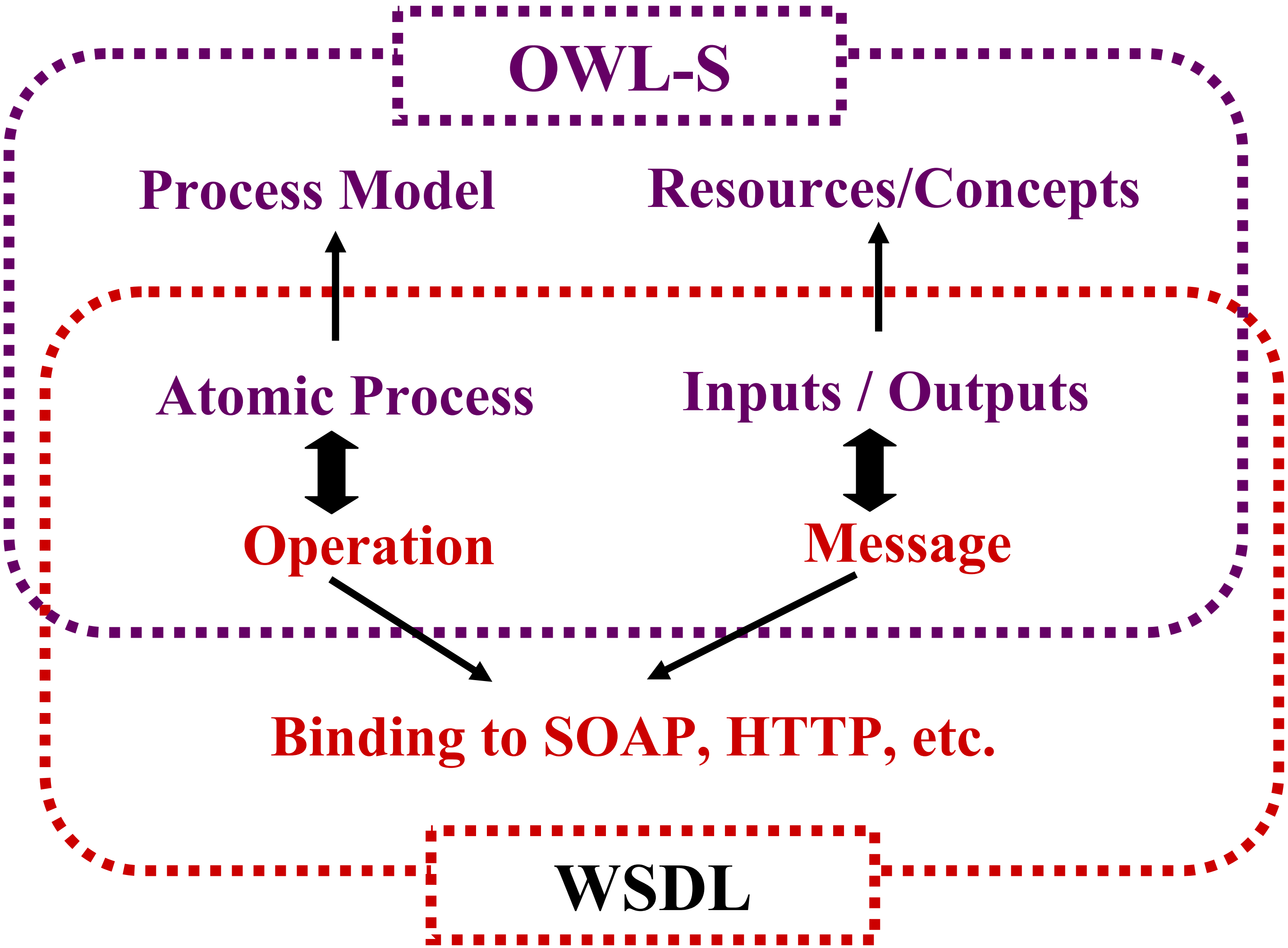
Service Profile



Process Model



OWL-S / WSDL Grounding



Outline

- OWL-S Status
  - Overview & Recent Milestones [David, 9:00]
  - **Recent Evolution of Process Model [Drew, 9:10]** 🡐
- Security Extensions [Grit, 9:20]
- Outreach, Tools, Standards [Katia, 15]
- SWSI Status
  - SWSL Overview [David, 8]
  - SWSL Ontology (FLOWS) [Rick,12]
  - SWSA [Mark, 10]
- Open Issues and Roadmap [Mark, 10]

Owl-S
Recent Changes to Process Model

What is the Process Model?

Sets out the inputs to and outputs from a web service, plus the events that will occur as a consequence of invoking it. (Key difference between Owl-S and more traditional systems.)

Terms in PM help link together terms from Profile, Grounding, and other components. The PM supports detailed reasoning about how to accomplish a goal by dealing with a web service.

Owl-S Process Model

Adoption of real condition notation ---

This was a key omission given the importance of preconditions and effects.

Conditions are now officially XML Literals, with the expectation that these will be SWRL expressions. (Although other possibilities are allowed, such as strings representing Kif, PDDL, or Common Logic formulas.)

Owl-S Process Model

“Bundled” effects and outputs –

If a bunch of effects and outputs depend on a condition, they are put into a single Result element:

<Result>

<InCondition> *c* </InCondition>

<hasEffect>*e*</hasEffect>

<withOutput>

<outputBinding> …</…>

</withOutput

</Result>

Owl-S Process Model

Atomic processes have inputs, outputs, and results.

Composites have that stuff + a *body* that spells out the subinteractions that occur when dealing with the web service

Owl-S Process Model

Several ways to specify values in Bindings

- valueSource: an output from a step of a composite
- valueData: constant
- valueFunction: arbitrary expression
- valueForm: Owl expression with pieces filled in
- valueType: (Just constrains type of value)

Owl-S Process Model

Conditional outputs in composites can depend on control flow

<If-then-else>
<ifCondition> … </ifCondition>
<then>

</then>
<else>

</else>
</If-then-else>

Outputs differ in these two places

Solution:

<Produce>
<producedBinding>
<OutputBinding>
…

Owl-S Process Model

```
<If-then-else>
        <then>
                <Produce> …. </Produce>
        </then>
        <else>
                <Produce> …. </Produce>
        </else>
  </If-then-else>
```

Security for OWL-S

OWL-S Coalition + UMBC

Presented by Grit Denker

November 2004

Overview

- Summary of Achievements
  - Specifying and matching security markup
  - Security Services
  - Semantic Firewall for Grid Services
- Latest Developments
  - Compliance checking of privacy policies
  - Enforcement of access control policies
  - Trust and communication
- Plans for 2005

Achievements

- **Ontologies**
  - Credential, security mechanisms (e.g., protocols, cryptographic technique), cryptographic characteristics of service parameters
  - Security markup extensions for services and agents
  - Authorization and privacy policies in Rei
- **Algorithms**
  - Matching algorithms for security annotations & Rei policies
  - Integrated into CMU’s Matchmaker
- **Examples in agent/service context**
- Ontologies available at new OWL-S security page

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

Achievements II

- Security Services
  - Reusable security capabilities: En/Decryption, Signature
  - OWL-S annotation and service deployment
  - http://www.csl.sri.com/~denker/owl-sec/SecurityServices/
  - Will be moved to OWL-S security page
- Semantic Firewall for Grid Services
  - Collaboration with Univ. Southampton (Terry Payne), IT Innovation (Mike Surridge), IHMC (Jeff Bradshaw)
  - Specification of service interaction protocol
    - Stateful model, multi-party, directed, msgs & internal events
  - Basis for enforcement at semantic firewall
  - See http://www.csl.sri.com/~denker/owl-sec/sfw

Latest Developments

- Compliance checking of privacy policies
  - Demo
- Enforcement of access control policies
  - Demo
- The impact of trust in the context of communication

Privacy Policy: Specification

- Privacy schema in OWL
  - Define privacy rules for services and agents
- Distinguishing three kinds of rules

  Authorization, Capability, Obligation

  - Subclasses:
    - Neg/pos authorization/capability/obligation
    - Neg/pos intent < pos capability
- Distinguishing various kinds of actions

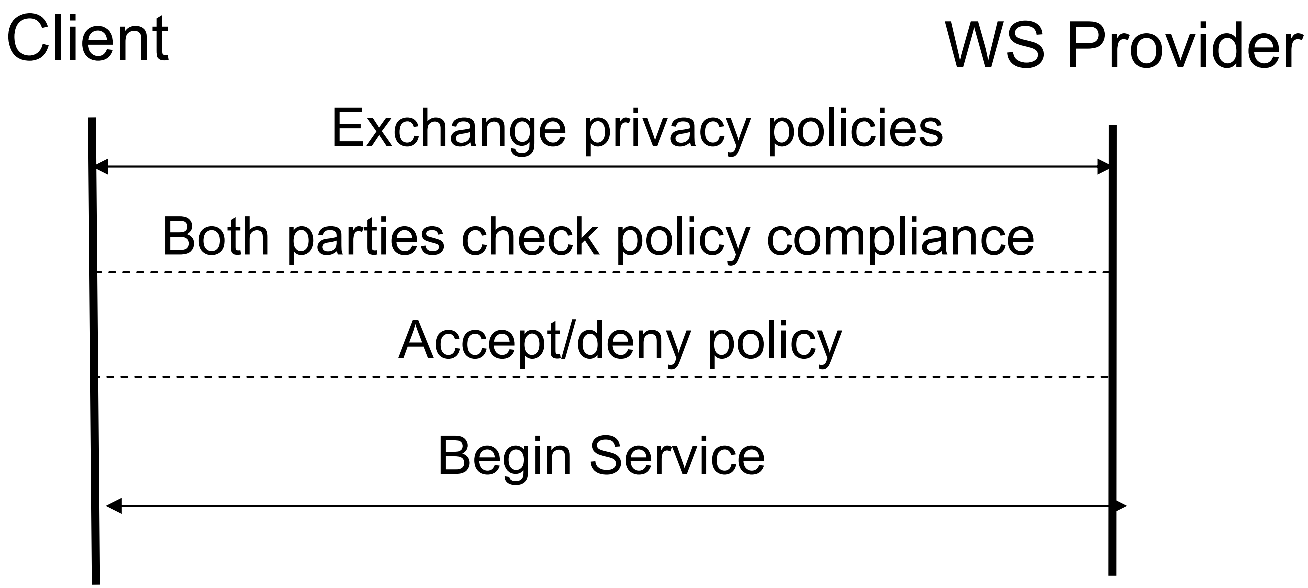
  Disclosure, Storage, Transmission

  - Subclasses:
    - Local, Third Party, Forum < Disclosure
    - Encrypted, Signed < Storage
    - Send, Receive, Encrypted, Signed, Plaintext < Transmission

Privacy Policy: Examples

- Client policies
- *“Server must store data encrypted”*
  - Rule: Obligation
  - Action: Encrypted Storage
  - Resource: data
- *“Allow server to collect user’s personal preferences to be disclosed locally”*
  - Rule: Positive Authorization
  - Action: DataCollection and LocalDisclosure
  - Resource: user’s personal preferences

Privacy: Compliance Checking

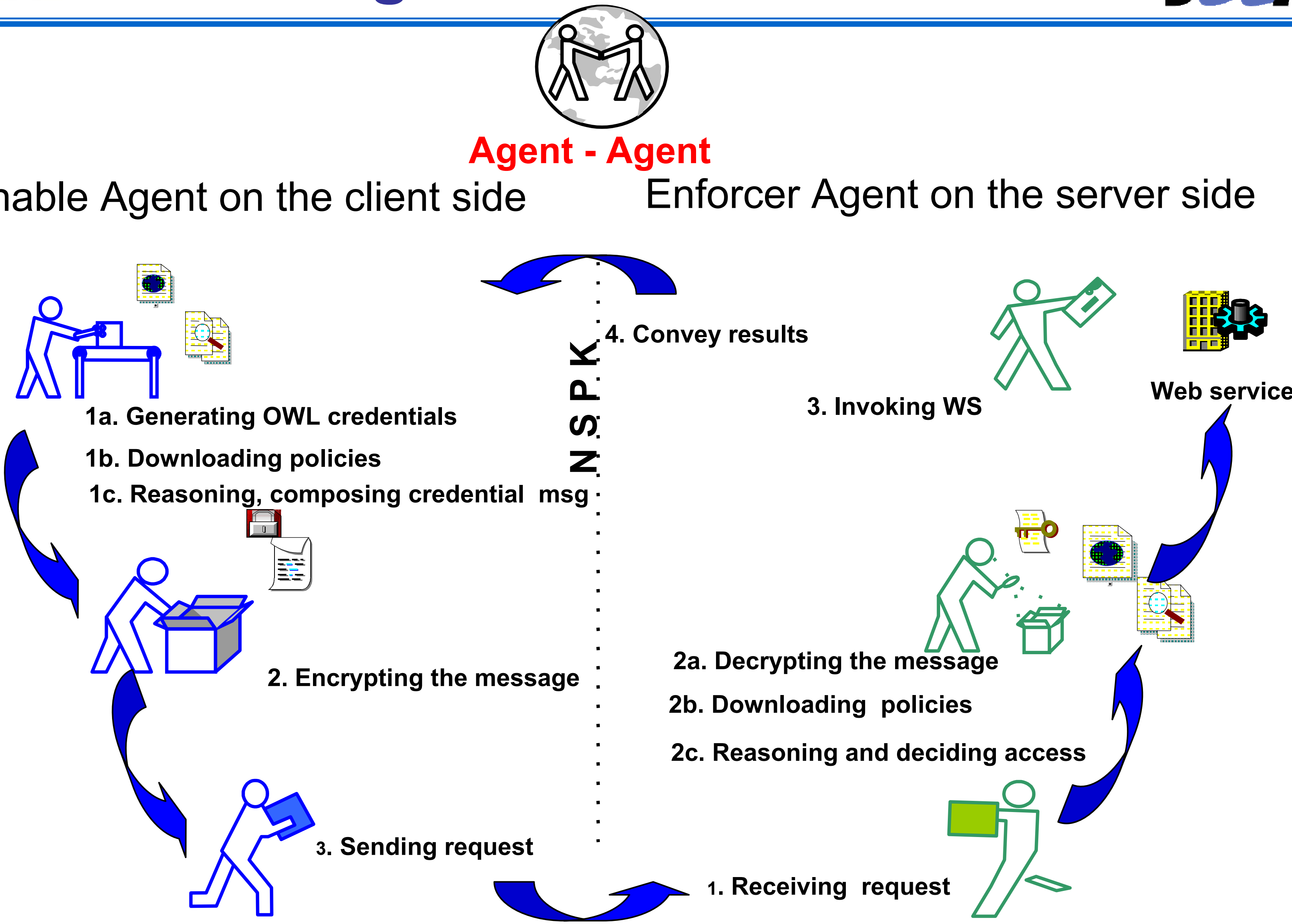


Privacy Policies in OWL

Jena for parsing and subsumption reasoning

Special-purpose algorithm for policy (rule/action) structure

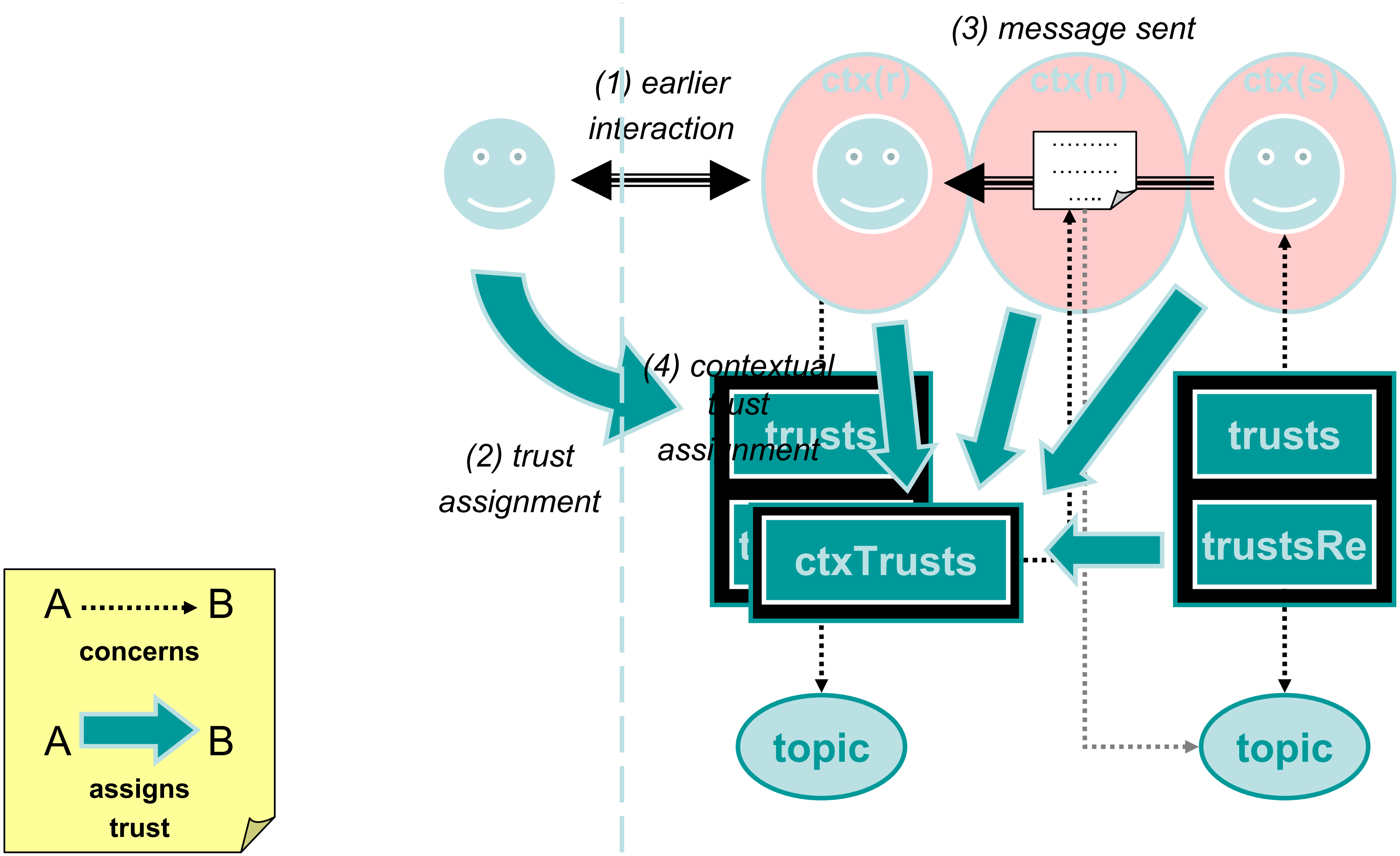
Enforcing Access Control Policies



Trust and Communication

- Goal: Trust with regard to **communication**
- Assumption: The **degree of trust** a message receiver assigns to the message she receives depends on the contextual details of
  - the message **sender**,
  - the mediating **network**, and
  - the message **receiver** herself
- Approach: **ontological**
  - Basis: Mindswap's trust ontology
  - Extended with concepts for grasping contexts and communication
  - http://www.csl.sri.com/users/denker/owl-sec/context/

Attaching Context-sensitive Trust to Messages



Plans for 2005

- Focus: Standardization of security extensions
  - Integration with OWL-S 1.2 release
  - Maintenance of web page
  - Making matching algorithms open-source
- Focus: SWS policies
  - Past: OWL-S position paper at W3C workshop on constraints and capabilities
    - Use of Rei for policies (see also Rules session “Rei and Security” by Tim Finin)
  - Future: Case studies (use of SWRL, Rei, etc.)
  - Tools for SWS composition and policies

Outreach, Tools, & Standards

Presented by Katia Sycara
on behalf of the OWL-S Coalition

Outline

Goal: Create the Semantic Web Services revolution.

In support of this goal, the OWL-S coalition has engaged in the following outreach and standardization activities:

- Standardization activities
- Semantic Web Services Initiative (SWSI)
- Tutorials, talks
- Tools
- Impact of OWL-S
- Related Activities

Standardization activities

Participation of OWL-S coalition members in various W3C and OASIS working Groups

- **OWL-S Note published at W3C** (www.org/Submission/2004/07)
- **Web Services Architecture** (W3C) (2003)
  - Resulted in a W3C Note
  - Mapping of WSA to OWL
- **WSDL: Web Services Description** (W3C) (ongoing)
  - Mapping of WSDL to RDF
- **Web Services Choreography** (W3C) (ongoing)
- **UDDI Technical Committee** (OASIS) (ongoing)
  - TC adopted OWL for expressing service taxonomies
  - Semantic search scheme of these taxonomies under development
- **OWL-S Position Paper** accepted at W3C workshop on constraints and capabilities to initiate activities toward a WS Policy standard (10/2004)
- **Semantic Web Services Interest Group** (SWS-IG) was created within the Web Services Activity (W3C) (10/2003)

Semantic Web Services Initiative

- OWL-S Coalition continued its participation in SWSI: a US EU initiative, comprised of EU and US researchers and industry members
- Results to date:
  - Semantic Web Services Language (SWSL)
    - Requirements document
    - Use cases www.daml.org/services/use-cases/language
    - Proposals for extending OWL-S: SWSL Rules, FLOWS
    - Plans to submit to W3C (or OASIS) in early 2005
  - Semantic Web Services Architecture (SWSA)
    - Use case repository www.daml.org/services/use-cases/architecture
    - Requirements document
    - Plans to submit to W3C (or OASIS) in early 2005
- More details in SWSI activity outbriefs

OWL-S Information Dissemination

- **DAML-S/OWL-S publications**
  - Many and varied, tying in with several research areas & communities
  - See http://www.daml.org/services/owl-s/ for a partial listing
- **OWL-S presence** at
  - Semantic Web Conference Series (WWW, ISWC)
  - International Conference on Web Services (ICSW)
- **Tutorials on OWL-S** and Semantic Web services in academic and industrial conferences (e.g. AAMAS, ICWS, ISWC, NODE)
- **Workshops** at various conferences (e.g. AAMAS, IJCAI, ISWC, AAAI Spring Symposium)
- **Panels**
  - WWW04 Panel on "Semantic Web and Web Services: A Marriage Made in Heaven?"
  - ICWS 04 on "Quality of Service Management in Service Grids and Grid Services "
- **OWL-S presence at the Semantic Web for Military Users**
- **OWL-S presence at Semantic Web Applications for National Security**

OWL-S Tools & Components

- Major Effort in constructing authoring tools to support early adopters
  - **CODE: CMU OWL-S IDE** based on Eclipse supports programmers in end-to-end SWS development and deployment
  - **SRI OWL-S Editor** based on Protégé
- Web Service Discovery
  - CMU OWL-S/UDDI Matchmaker
  - KSL Semantic Discovery Service
  - CMU OWL-S for P2P
  - CMU OWL-S for bridging Communities of Interest
- Web Service Discovery and Mediation
  - CMU OWL-S Broker

Tools & Components

- Automatic WS Invocation
  - CMU OWL-S Virtual Machine
- Web Service Composition
  - Mind-Swap Composer
  - KSL Composition Tool
  - CMU Computer Buyer
- Libraries
  - CMU OWL-S API
  - MindSwap OWL-S API
- OWL-S is layered on OWL
  ➔ All the tools & technologies for OWL are relevant
- See also: http://www.daml.org/services/
  - Tools page
  - www.semwebcentral.com

CMU OWL-S IDE (CODE)

DEMO

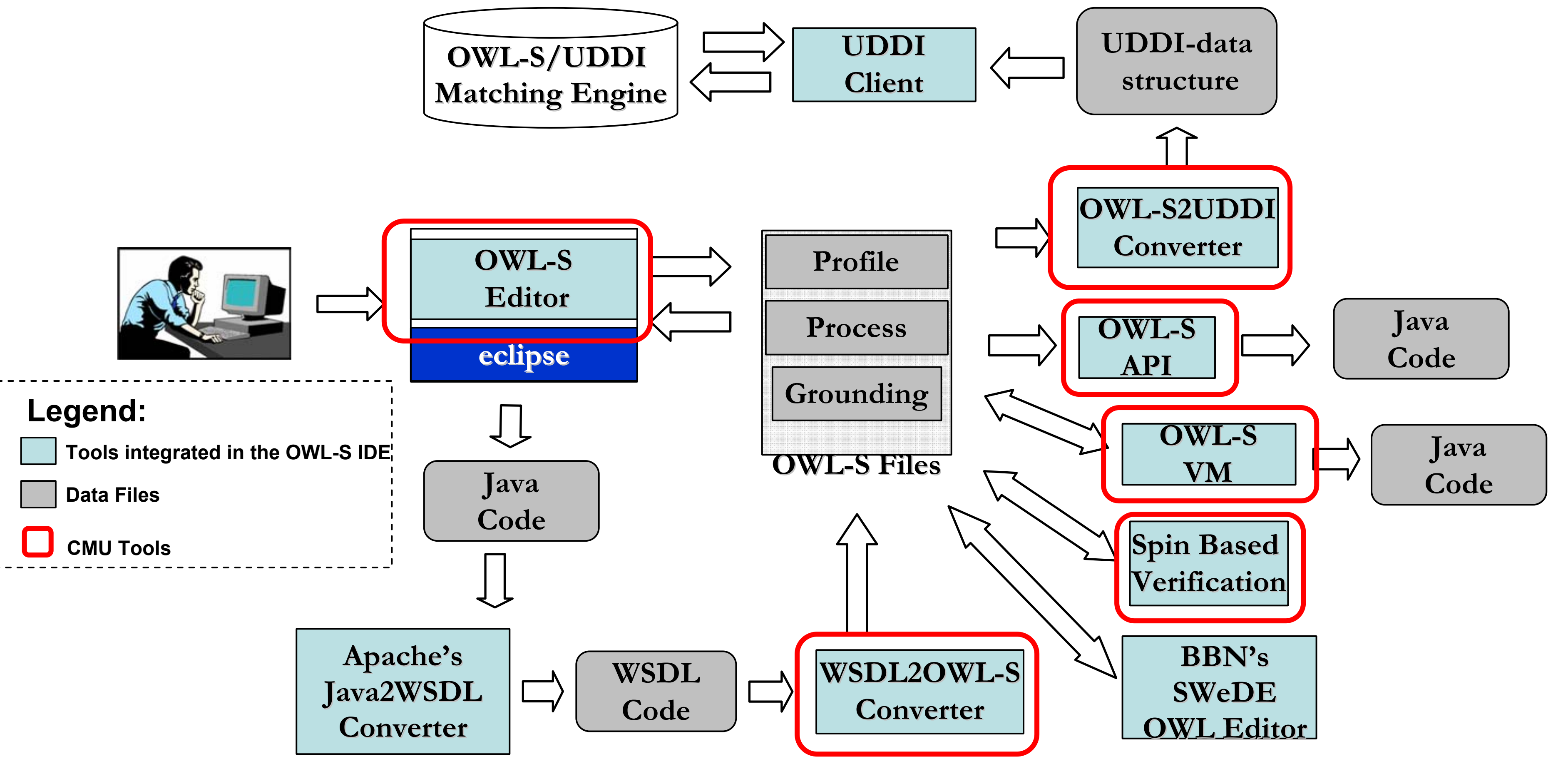
- CODE: **C**MU **O**WL-S I**DE** is an Eclipse based tool that allows end-to-end development and deployment of Semantic Web Services
- CODE integrates the generation of OWL-S representation with the generation of the WS Java code
- CODE provides tools for authoring, editing, visualization, deployment and client creation
- CODE is integrated with SWeDE OWL Editor
- CODE is targeted to Web service developers

Main idea is to allow developers to generate their web services' code and OWL-S descriptions within the same environment

http://projects.semwebcentral.org/projects/owl-s-ide/

CODE's Architecture

DEMO



OWL-S Editor for Protégé

DEMO

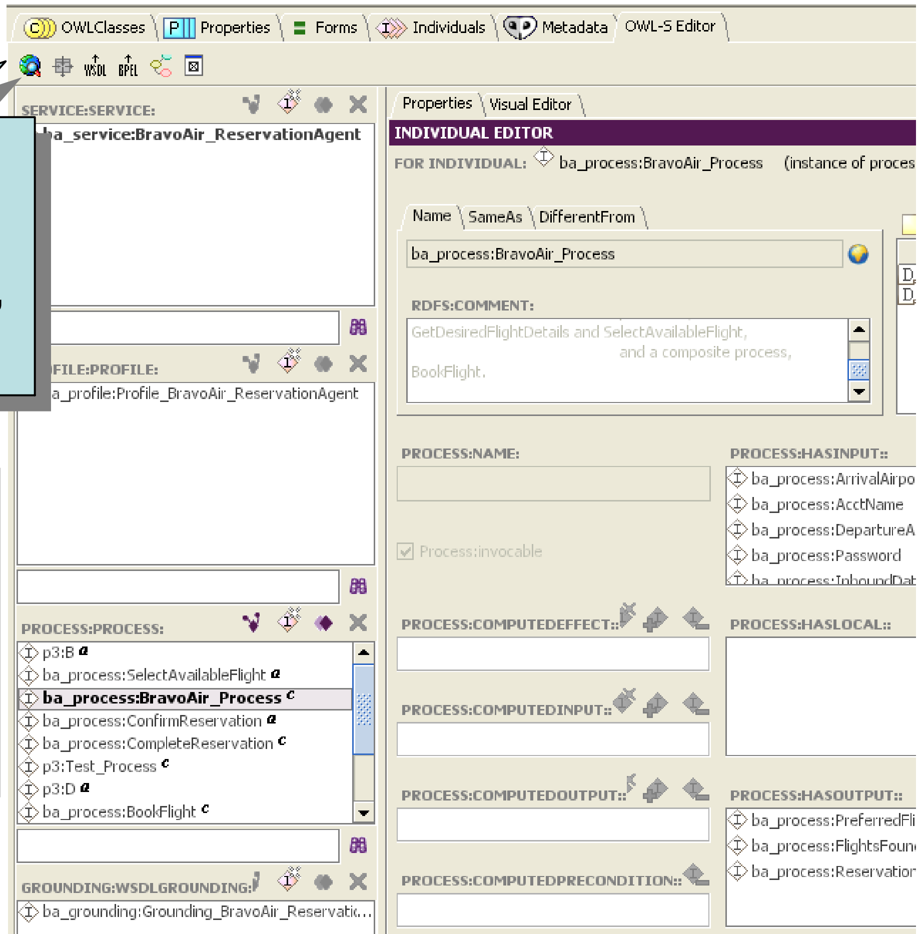
- Easy, intuitive OWL-S service development environment
- Based on popular Protégé/OWL ontology editor
- Open-source, with code available at
  http://owlseditor.projects.semwebcentral.org
- It provides
  - *IOPR* Manager
    - *I*nput/*O*utput/*P*recondition/*R*esult
    - Maintain IOPR correspondences between OWL-S sub-ontologies
    - Perform consistency checks
  - Graph Overview
  - Visualize & navigate relationships between OWL-S sub-ontologies
- Generate & import skeletal OWL-S from WSDL

Sample Functionalities

DEMO

Toolbar provides WSDL import, graphical overview, and more

Instance panes for Services, Profiles, Processes, and Groundings



Full control of OWL-S properties with customized widgets

Impact of OWL-S

IBM

- provide OWL-S API as part of Snowbase semantic web tool
- Use OWL-S for enhanced semantic UDDI

SAP

- Use OWL-S for automatic composition of services to manage border control

Toshiba

- Use OWL-S for publicly available UDDI at NTT (Main Japanese UDDI)

Fujitsu

- OWL-S used in their Task Computing Project that is expected to be in production in 2005

NIST

- Use OWL-S to describe capabilities of UAVs

MyGrid

- Use OWL-S to describe Web services on the Grid

AgentCities

- OWL-S used for discovery of new agents

Related Activities

- **DERI initiative**
  - WSMO/WSML/WSMX
  - OWL-S coalition initiated comparative work between OWL-S and WSMO
- Building the business case for semantics in Web Services
  - "Complete, do not compete" (e.g. OWL-S grounding layered on top of WSDL, OWL-S/UDDI matchmaker)
  - "A little semantics goes a long way" (e.g. WS Security, WS Management)
- OWL-S has been featured in a recent book on Semantic Web Services: "Developing Semantic Web Services" by Peter H. Alesso & Craig F. Smith

Outline

- OWL-S Status
  - Overview & Recent Milestones [David, 10]
  - Recent Evolution of Process Model [Drew, 10]
- Security Extensions [Grit, 15]
- Outreach, Tools, Standards [Katia, 15]
- **SWSI Status ←**
  - **SWSL Overview [David, 8, 9:50]**
  - **SWSL Ontology (FLOWS) [Rick,12, 9:58]**
  - **SWSL Rules work will be presented tomorrow**
  - **SWSA [Mark, 10:10]**
- Open Issues and Roadmap [David, 10:20]

SWSL Strategy

- Build out from OWL-S
  - to take advantage of more expressive languages
  - to extend the conceptual model
- Full-fledged use of FOL expressiveness
  - OWL-S can use SWRL and SWRL FOL in quoted contexts, in service descriptions (instances)
  - SWSL will use it throughout; both in ontology axioms and in all parts of service descriptions
- Leverage broad availability of LP-based languages, environments, tools, etc.
- Build on mature conceptual models
  - PSL, W3C architecture, Dublin core
- Maintain connections with the world of OWL
  - Layers of expressiveness

SWSL Components

- Conceptual Model
  - Build on OWL-S, PSL, [W3C WS Architecture]
- Language
  - SWSL FOL
  - SWSL Rules – LP with NAF; courteous, Hilog extensions
  - Shared presentation syntax; builds on F-Logic
  - Markup syntax – TBD probably with ruleML committee
- Ontology
  - Formal expression of conceptual model
  - Both in SWSL FOL and LP (as much as possible)
- Bridge (?)
  - What can we provide to enable coordinated use of FOL and LP reasoners
- Grounding
  - Like OWL-S Grounding, connects with WSDL

The SWSL Ontology

Presented by Rick Hull

November 2004

Key Contributors

• Steve Battle	(HP Labs, England)
• Daniela Berardi	(U Rome, “La Sapienza”)
• Michael Gruninger	(NIST, U Maryland)
• Rick Hull	(Bell Labs)
• Michael Kifer	(SUNY Stonybrook)
• Sheila McIlraith	(U Toronto)
• Jianwen Su	(UCSB)

SWSL Ontology
Overall Program Summary

- Challenge: OWL-S is based on Description Logics
  – OWL not expressive enough for key aspects of reasoning about services, e.g., to perform automatic compositions
  – To model “effects”, need axioms to support a multi-state world (e.g., a situation calculus)
- “FLOWS”: First-order Logic Ontology for Web Services
  – Requirements / Desiderata
  – Process Model
    - Based on Process Specification Language (PSL)
    - Extends OWL-S
    - Constructs specific to web services
  – Recent Progress and Status

Representational Desiderata for a Web Services Ontology

*	in SWSL requirements
**	refinement of reqs.
***	extensions to reqs.

- Leverages existing service ontologies (OWL-S) **
- Model-theoretic semantics **
- Taxonomic representation *
  - Captures activities, process preconditions and effects on world. *
  - Captures process execution history. **
  - Primitive and complex processes are first-class objects ***
- Can serve as common basis for different representational paradigms
  - Explicit representation of messages and dataflow (cf. BPEL, W3C choreography, behavioral message-based signatures, etc.) ***

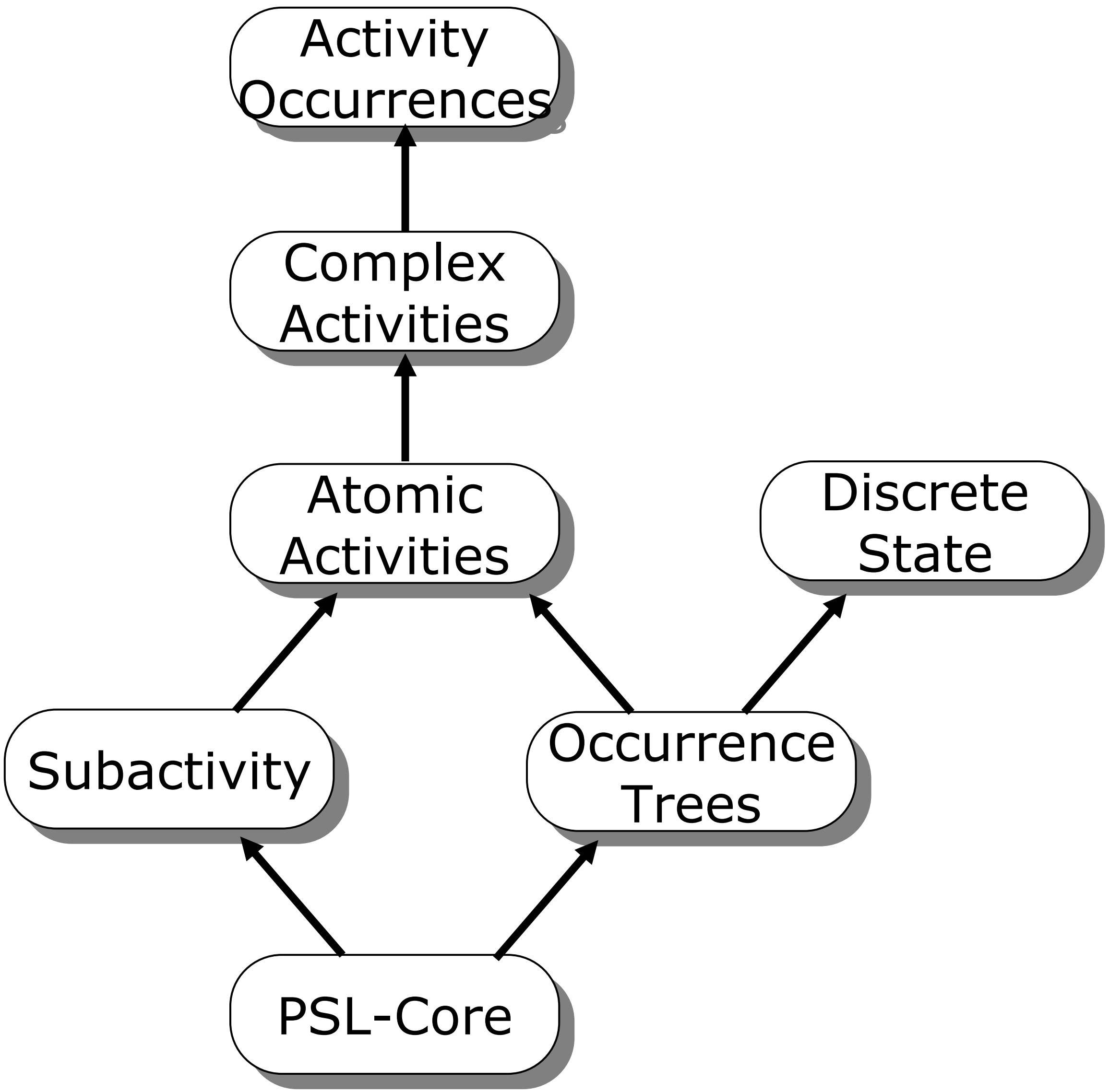
Guiding Case Studies

- Amazon example
  - Queries that we need to support
  - Compatibility, pre-conditions, ordering constraints
- Financial transaction example
  - Utility of building up named complex activities
  - E.g., *transfer($amount, $account1, $account2)*
- Travel service scenario
  - Different forms of service composition
    - Single-use vs. re-usable
    - Built from atomic vs. non-atomic services

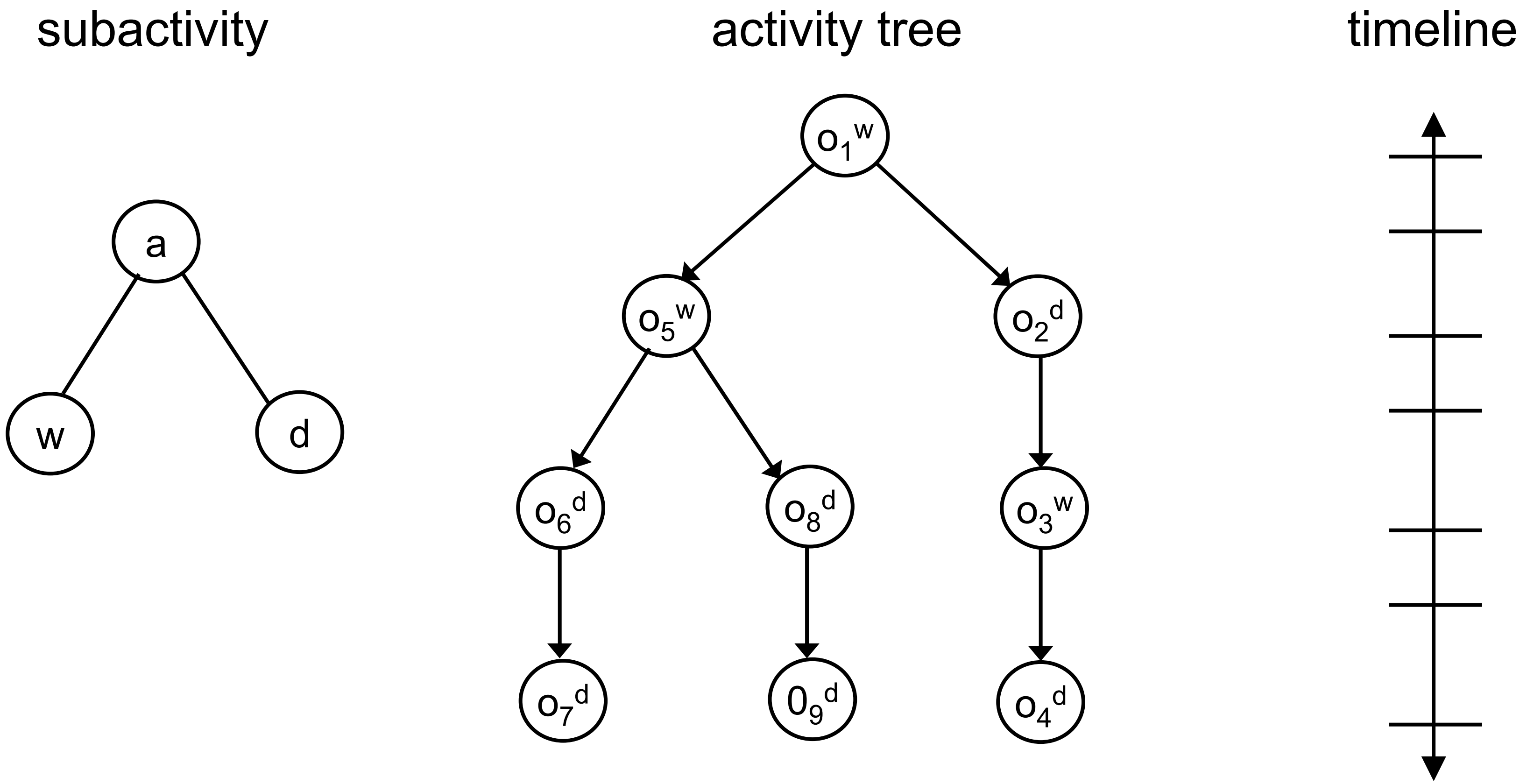
Process Specification Language (PSL)

- PSL is a modular, extensible first-order logic ontology capturing concepts required for manufacturing and business process specification
  - PSL is an International Standard (ISO 18629)
  - There are currently 300 concepts across 50 extensions of a common core theory (PSL-Core), each with a set of first-order axioms written in Common Logic (ISO 24707)
  - The core theories of the PSL Ontology extend situation calculus
  - PSL is a verified ontology -- all models of the axioms are isomorphic to models that specify the intended semantics
- PSL provides a mature foundation upon which to develop FLOWS

PSL Core Theories



Some Structures in Models of PSL



- ***Discrete State***: Adds fluents, and idea that activity occurrences change the values of fluents
- ***Complex Activities***: Allow groupings of atomic activities into complex ones

FLOWS Extensions of PSL

- Key constructs for the process model
  - Web services, including
    - Manipulation of fluents
    - Communication via messages (optional)
  - Explicit representation of state and state constraints
  - Flow-of-control *a la* OWL-S
  - Ordering and temporal constraints
  - Occurrence constraints
  - Composition
- Constraints as the guiding paradigm
  - Analogous to Golog, but several kinds of constraints
  - Can move gracefully between incomplete and complete specification
- "Views" as a mechanism to focus on sublanguages
  - E.g., for Profile only, Process model, Data-flow only

Relationship to OWL-S

- FLOWS provides a first-order axiomatization of the intended semantics of OWL-S.
  - OWL is too weak to completely axiomatize the intended semantics of OWL-S.
    - Fundamentally – issue of actions moving to new states
  - Any implementations of OWL-S must resort to extralogical mechanisms if they are to conform to the OWL-S semantics, whereas implementations of FLOWS will be able to use the axioms directly.
- *Note: FLOWS can provide a formal basis for comparing contrasting OWL-S with emerging WS standards (e.g., BPEL, WSDL).*

Status

- Status:
  - Mature version of
    - Embedding of OWL-S into PSL
  - Draft versions of
    - FLOWS process model
    - Some forms of constraints – flow-of-control, order
    - Axiomatization
    - FLOWS Query Language (FQL)
- Next steps
  - 12/8-9/2004: F2F at IBM Yorktown
  - 12/2004: Draft of SWSL white paper, including for Ontology:
    - Process Model (PSL extended to web services)
    - Presentation Syntax for various kinds of constraints
    - Axiomatization
- Challenges
  - Implementation strategies for key sublanguages/views
  - Formal underpinnings for rich flow-of-control models (e.g., FSMs)
  - More refined understanding of data flow within FLOWS ontology

Back-up: Tractability

- Use case scenarios show that in general we will need to solve intractable reasoning problems.
  - Reasoning problems for semantic web services are inherently intractable -- using a different language does not make them tractable.
  - If you restrict yourself to a language that is tractable, then there will exist reasoning problems that cannot be specified in the language.
  - FLOWS enables identification and exploitation of (pragmatically) tractable subclasses, while maintaining the virtues of the full FLOWS ontology.

Back-up: Example queries* over (complex) activities

- *impacts*(*a*, *f*) can be defined to mean that at end of activity occurrence *a* fluent *f* has changed

- Services that change at least one fluent that *buy_product*("Alice") changes

$$\{ a \mid (\exists f)\ impacts(buy\_product(\text{"Alice"}), f) \wedge impacts(a, f) \}$$

- Services that change the same fluents that *buy_product*("Alice") changes

$$\{ a \mid (\forall f)\ impacts(buy\_product(\text{"Alice"}), f) \equiv impacts(a, f) \}$$

* Using database query language syntax (cf. relational calculus)

Back-up: Example Flow-of-Control Constraint

(Draft) Presentation syntax:

```
transfer(?Amount, ?Account1, ?Account2)   {
      Sequence
      ?occ1:occurrence   withdraw (?Amount, ?Account1)
      ?occ2:occurrence   deposit (?Amount, ?Account2)
      ?occ1 soo_precedes ?occ2
  }
```

Translation to Underlying FOL (behind the scenes)

```
(forall (?x ?y ?z)   (sequence (transfer ?x ?y ?z)))

(forall (?occ)
    (implies  (occurrence_of  ?occ (transfer  ?Amount  ?Account1  ?Account2))
              (exists (?occ1 ?occ2)
                   (and (occurrence_of ?occ1 (withdraw  ?Amount  ?Account1))
                        (occurrence_of ?occ2 (deposit   ?Amount   ?Account2))
                        (subactivity_occurrence  ?occ1  ?occ)
                        (subactivity_occurrence  ?occ2  ?occ)
                        (soo_precedes ?occ1 ?occ2  (transfer ?Amount  ?Account1  ?Account2))))))
```

Back-up: FLOWS Query Language (FQL)

- We are working on a query language proposal inspired by
  - PSL: activities and occurrences, testing based on fluents
  - OWL-S: permit additional structure for activities, including IOPE
  - OQL: functional query language for complex objects, extended and relativized to the structures and operators in web services
- Example (simple) query in *preliminary* version of FQL

```
hotel_reservation_service =
    select h
    from h in UDDI,
             hotel,person,d1,d2 in h.inputs
    where hotel.type subclass_of Hotels and
              person.type subclass_of String and
              d1.type, d2.type subclass_of Date and
              h.precond has_element_equiv 'val(d1) < val(d2)' and
              h.precond has_element_equiv 'vacancy(val(hotel), val(d1), val(d2))' and
              h.effect has_element_equiv  '+hotel_res(val(hotel),val(person), val(d1), val(d2))'
```

- Can exploit recursive structure of query components to create intricate but natural queries, including compositions
  - Can use quantifiers, but can express many things without them

SWSA Committee Update

Mark Burstein, BBN

November 2004

Contributors

- Mark Burstein, BBN, co-chair
- Chris Bussler, DERI, co-chair
- Mike Dean, BBN
- Carole Goble, Univ. of Manchester
- Tim Finin, UMBC
- Michael Huhns, Univ. of South Carolina
- Massimo Paolucci, CMU
- Norman Sadeh, CMU
- Amit Sheth, Univ of Georgia
- Stuart Williams, HP Labs, Bristol, UK
- Michal Zaremba, DERI

Recent Activities

- Requirements Document Released June 1
  - Announced on SWS-IG list.
- F2F at BBN Aug 16,7
  - Guest presentation by Patrick Gannon, OASIS
- Draft Note to be completed by Dec 20

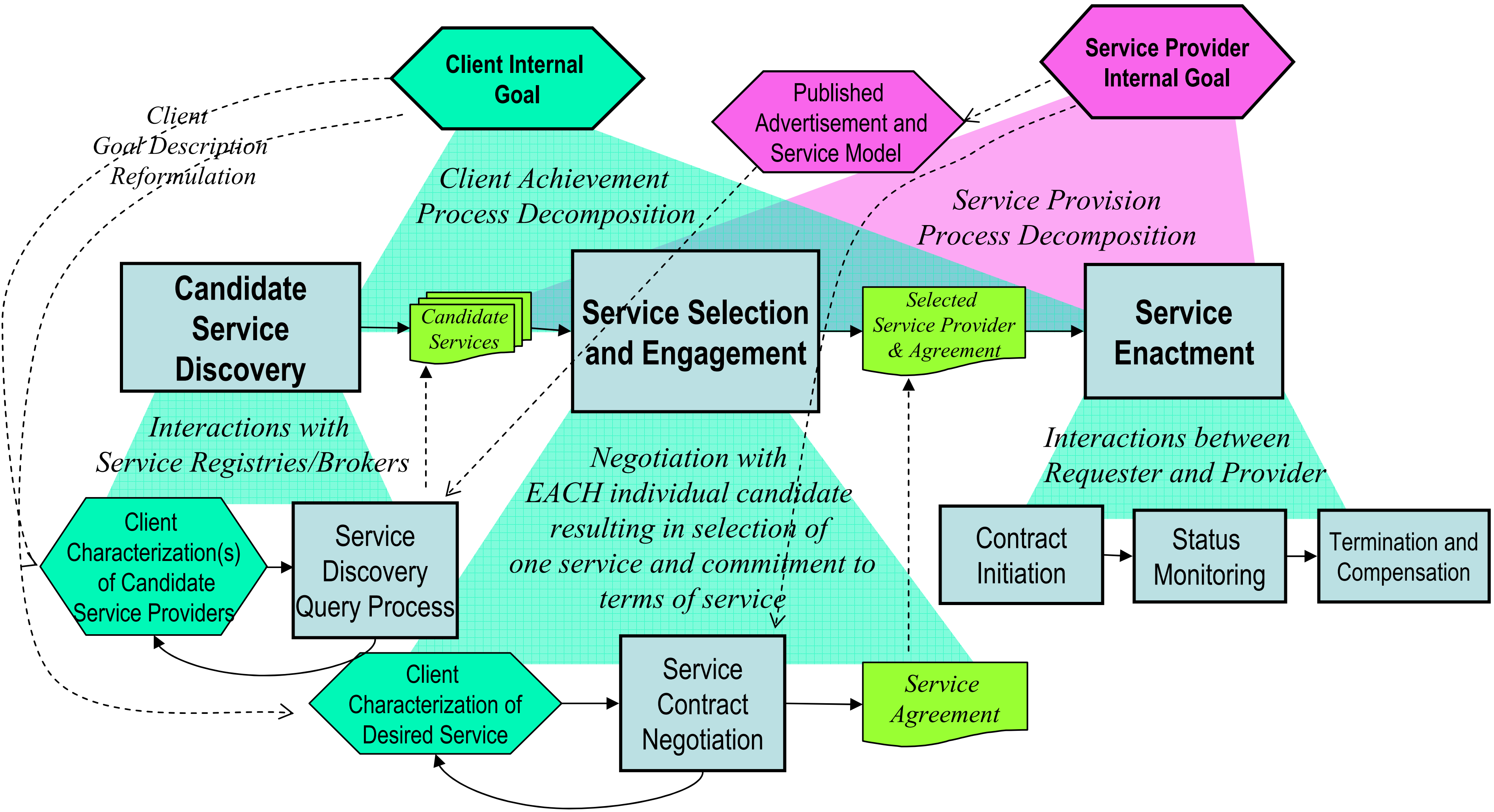
Approach

- Focus is on definition of abstract protocols and phases of service interaction.
  - Classes of activity specific to SWS interactions.
  - Individual services may define protocols that embody these elements in different ways.
- Builds on multiple previous efforts
  - W3C Architecture WG report
  - Conceptual Architecture for SWS (Preist, 2004, HP Labs)
  - Semantic Web Architecture Stack (Tim B-L, 2000)
  - OWL-S Use Model
  - Open Grid Services Architecture

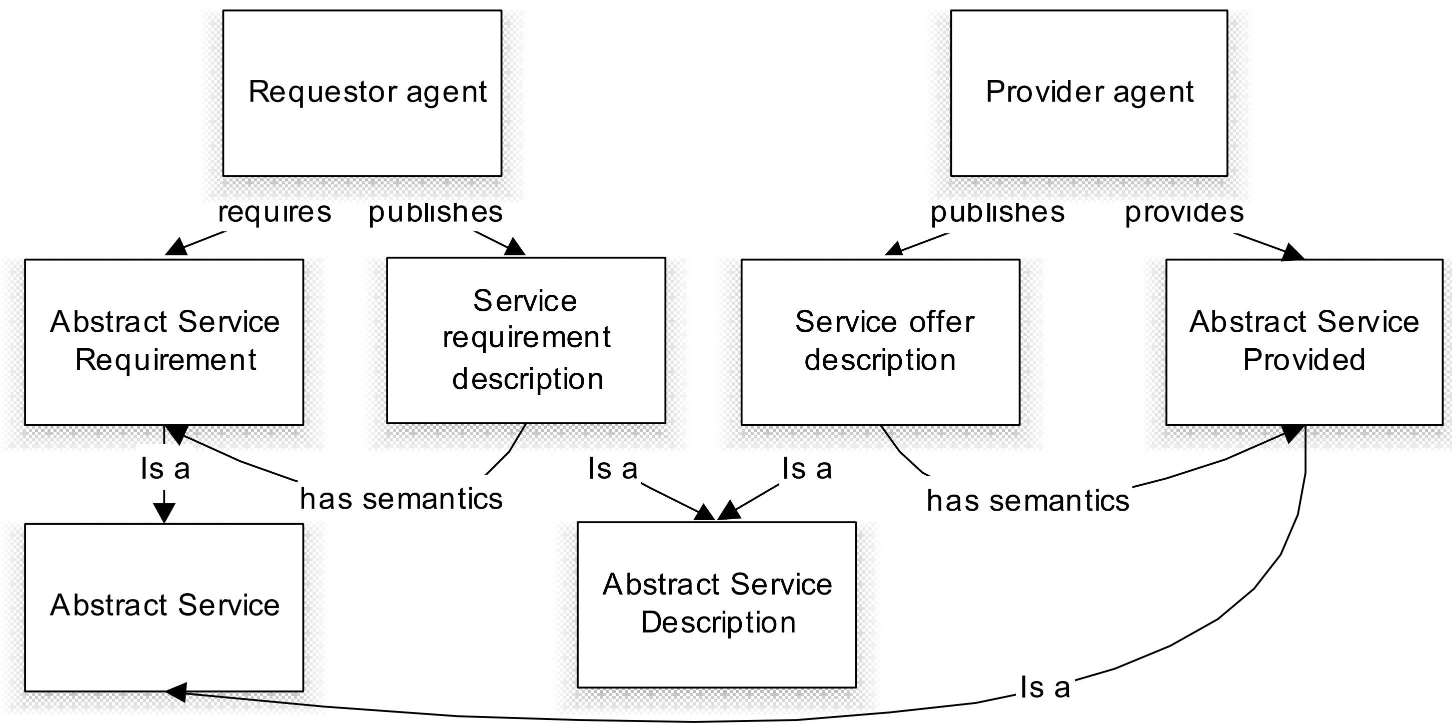
Abstract Protocols and Processes

- Discovery
  - Protocols: Advertisement, Capability Query
  - Processes: Advert formulation, Matching, Selection
- Engagement
  - Protocols: Service Model Query, Authentication, Contract Refinement Negotiation, Commitment
  - Processes: Service Description Publication & Interpretation, Contract Refinement, ID confirmation and authorization
- Enactment
  - Protocols: Initiation, Status Query, Finalization and Compensation
  - Processes: Service Execution, Monitoring, Compensation determination

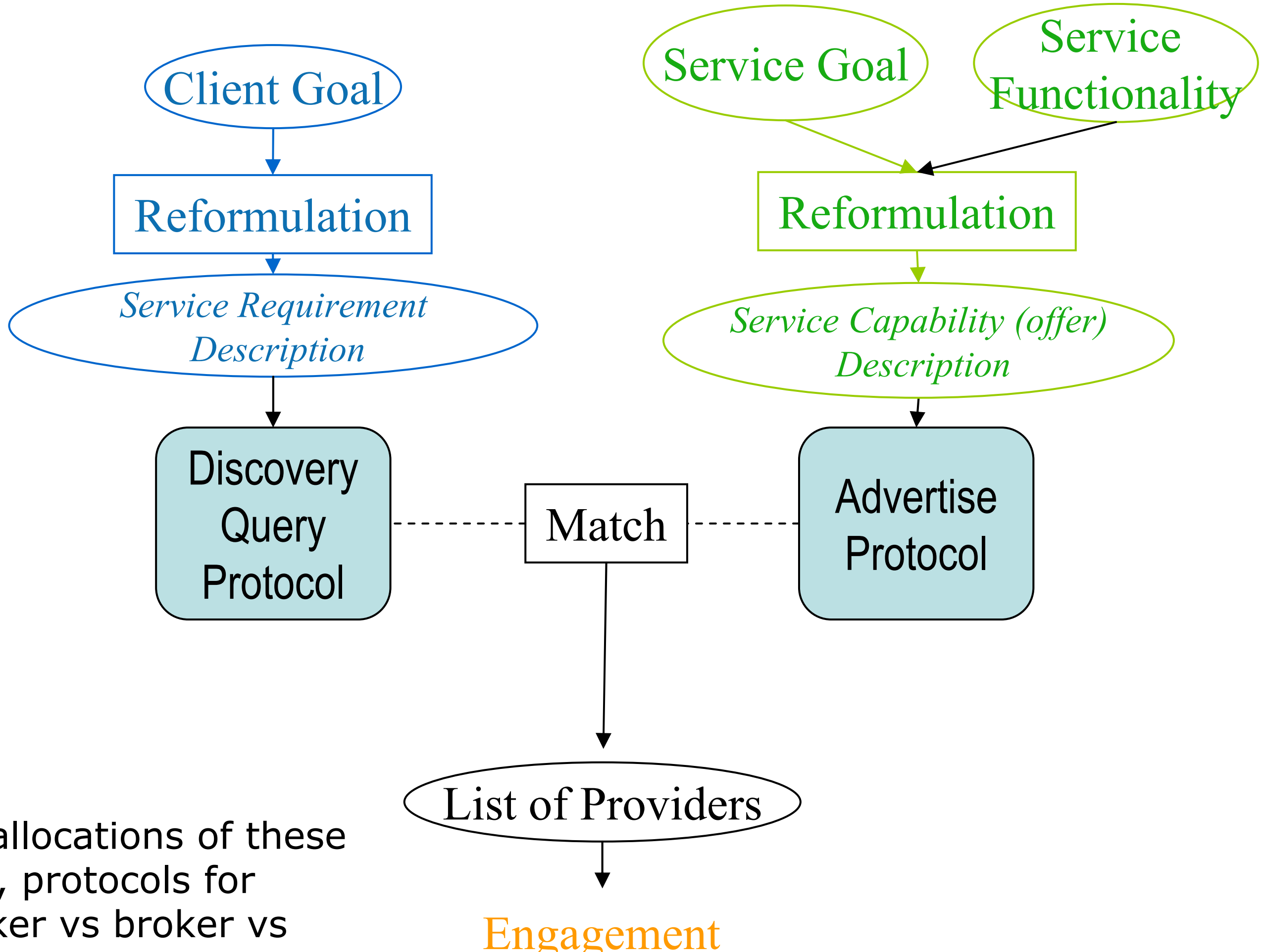
SWS Processes Overview



Conceptual Architecture – Discovery Model

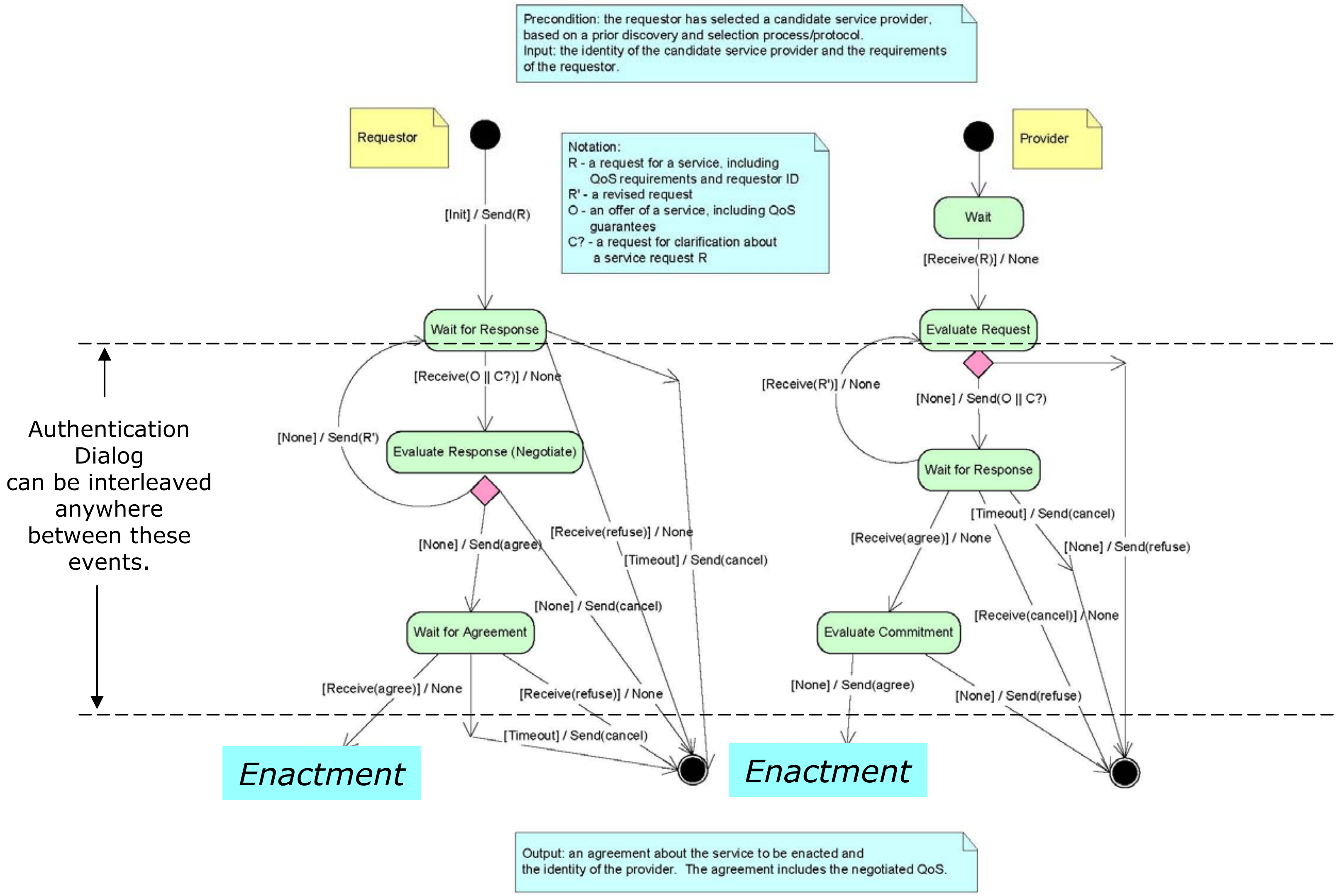


Discovery Process

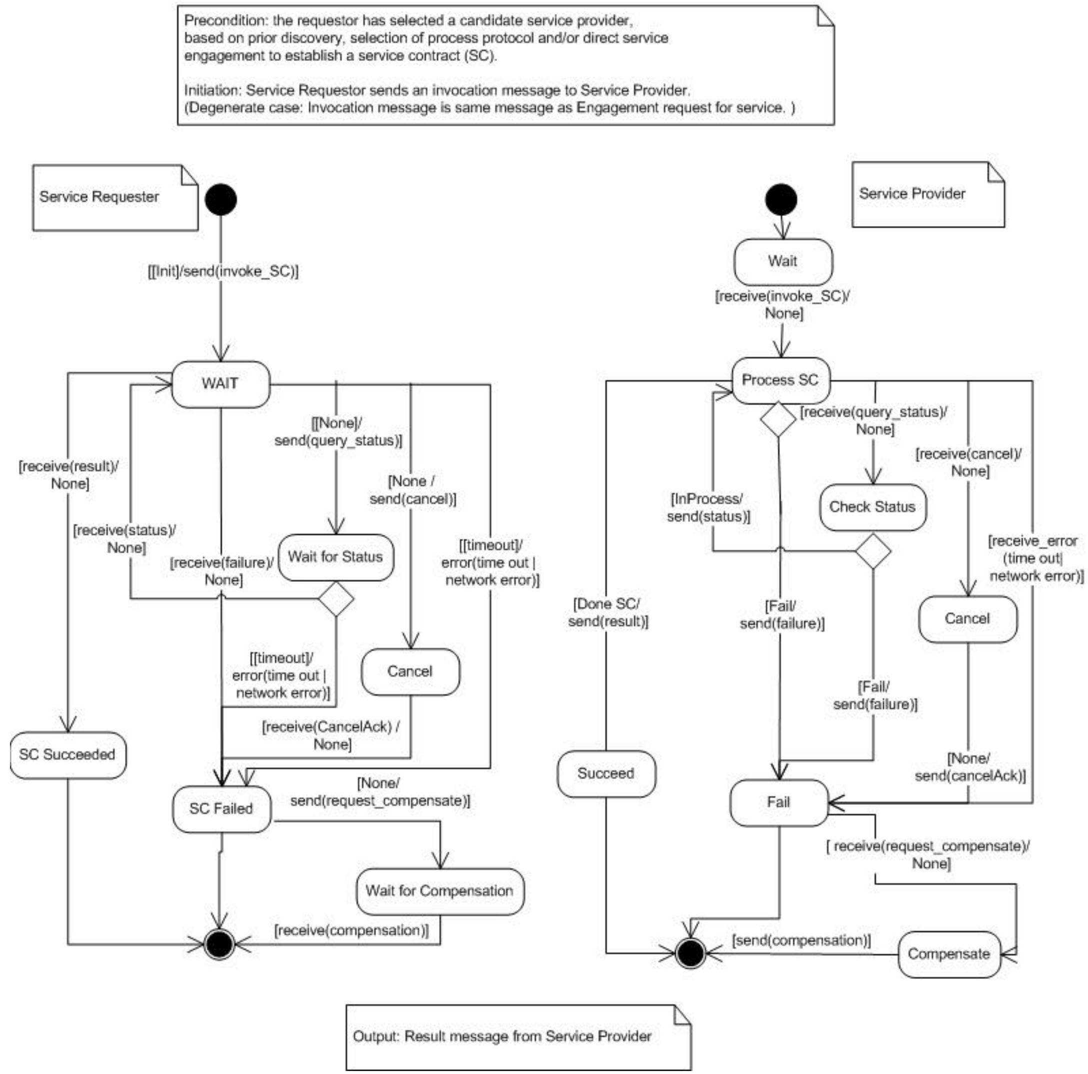


Different allocations of these processes, protocols for matchmaker vs broker vs peer-to-peer discovery mechanisms.

Engagement –
Negotiate and Commit Protocol



Enactment – Asynchronous Protocol



Some of the Remaining Issues

- Architectural elements and protocol support for semantic mediation and translation
  - Protocols for interactions with ontology servers, translators, semantic mapping servers
- Architectural support for publication and enforcement for semantically described
  - Authentication policies
  - Security and privacy policies
  - QoS guarantees

Current Plans

- Tentative Plan for next F2F at DERI, Galway in February, 2005 (hopefully *with* SWSL)
- Submission planned for end of 1st Qtr 2005
  - Still discussing if it should be a single SWSI document representing both committees.

Open Issues and Roadmap

David Martin

Outstanding tasks following Release 1.1 & submission

- Responses to remaining IG questions
- Finalize tools, finish updating examples
- Parser/Generator for surface syntax
- Tutorial materials for bleeding edge developers
- Submission update

Open Technical Issues

- Relationship between OWL-S and WSDL 2.0
  - Improved semantic support in new WSDL may require some realignment of at least the grounding ontology.
  - Mapping of OWL-S process *participants* to WSDL is not defined, as WSDL does not talk about the parties involved.
- Relationship to languages for Rules, Constraints
  - As SW Rule languages stabilize, we will have to revisit the integration of OWL-S and rules.
- Relationship to Choreography WG and WS-CDL (Choreography Description Language)
- Explicit process model support for exception handling and compensation
- Relationship between OWL-S and mediation/translation functions.
- Relationship to upper ontologies for activities (eg PSL), transactions…
- Review of arity relationships between Profile, Process models and groundings
- Better integrated support for security, privacy.

Directions and Roadmap OWL-S

- W3C follow-through.
  - The public commentary on the OWL-S Submisison ends with the statement

    "We intend to hold a Semantic Web Services workshop in the first half of 2005. One possible outcome would be a Recommendation track on a Semantic Web Services framework. "

  We need to push for this to occur and to be successful.
- Support users
- Examples and sample code
  - Transitioning to SemWebCentral with clear, working, tutorial examples that can be used as models by developers
  - Proof-of-concept showing value-added over commercial WS
  - Better illustrate the mapping to commercial standards
  - Tool tutorials should have suggestions for effective use of third party software where necessary.
- 'Maintenance Release' 1.2

Directions for OWL-S wrt. SWSI

- SWSI as a transition mechanism
  - Incorporate SWRL FOL into OWL-S
  - Rationalize OWL-S Process Model and SWSL Activity Ontology
  - Greater use of SWRL Rules for conditions and constraints
- Clarify role of OWL-S in SWSL development
  - Mapping of OWL-S into SWSL

SWSL Strategy

- Build out from OWL-S
  - to take advantage of more expressive languages
  - to extend the conceptual model
- Full-fledged use of FOL expressiveness
  - OWL-S uses SWRL and SWRL FOL in quoted contexts, in service descriptions (instances)
  - SWSL will use it throughout; both in ontology axioms and in all parts of service descriptions
- Leverage broad availability of LP-based languages, environments, tools, etc.
- Build on mature conceptual models
  - PSL, W3C architecture, Dublin core
- Maintain connections with the world of OWL
  - Layers of expressiveness

SWSL Tasks

- Resolve the remaining technical details in the definition of SWSL Rules, SWSL FOL, and the service ontology
- Finish specification of examples to be included in the report
- Refine specification of SWSL relationship to OWL, and how it may be used in an OWL framework
- Determine standardization strategy (e.g., W3C vs. OASIS and related issues)
- Meet with DERI to discuss their role in SWSL, language compatibility and plans for merger of SWSL and WSML approaches.

The End