Semantics
for Web Service
Specification

Chris Bashioum
MITRE

David Martin
SRI International
Chair, OWL-S Coalition
Outline

• SOA - Setting The Context
• Web Service Specification Languages
  – Web Services: Basic Standards
  – The Need for More Semantics
  – What the Semantic Web Can Offer
• OWL-S: Semantics for Web Services
  – Approach
  – Status & Relationship with Industry Standards
  – Sample Applications
• Service Description Framework (SDF)
  – Motivations
  – Approach
  – Status
Motivation

• What does it take to make services understandable to both people and machines,

• Across the entire lifecycle of the service,

• And in the larger context of all system artifacts related to the service
Setting The Context

• We will use the OASIS Reference Model for Service Oriented Architecture
  – http://docs.oasis-open.org/soa-rm/v1.0/

• This is not a presentation on the RM
  – but the RM helps illustrate the need for semantic technologies
What is SOA: Formal Definition*

- Service Oriented Architecture is a **paradigm** for organizing and utilizing **distributed capabilities** that may be under the control of **different ownership domains**

* Taken from OASIS Reference Model for Service Oriented Architecture 1.0
SOA: Key Concepts

• Key concepts for describing SOA paradigm

  – Visibility
    ▪ possibility for matching needs and capabilities

  – Interaction
    ▪ activity of using a capability

  – Effect
    ▪ reason, or purpose, for using a capability
Fundamental to SOA is the Service*

- A Service Is ...
  - A mechanism to enable access to capabilities, where access is
    - Provided using a prescribed interface
    - Exercised consistent with constraints and policies specified in the service description
  - Provided for the use of others,
    - but eventual consumers may not be known to the service provider,
      - and may demonstrate uses of the service beyond the scope originally conceived
  - Accessed by means of a service interface
  - Opaque
    - Implementation is hidden from the service consumer
    - Information and behavior models exposed through interface

* Taken from OASIS Reference Model for Service Oriented Architecture 1.0
The concepts of visibility, interaction, and effect apply directly to services in the same manner as described for the SOA paradigm:

- **Visibility** is promoted through the Service description
  - Which is a collection point for all information related to a service

- Service description contains information necessary to interact with the service (e.g. service inputs and outputs, associated semantics, conditions for using the service)

- Service description conveys **real world effects** when one interacts with the service

_The service description allows prospective consumers to decide if the service is suitable for their current needs and establishes whether a consumer satisfies any requirements of the service provider_
How is SOA Different From Other IT Approaches?

• Central focus of SOA is the task or business function - getting something done

• SOA reflects the reality that ownership boundaries are a motivating consideration in the architecture and design of systems
  – Evident in the core concepts of visibility, interaction, and effect

• Within the context of SOA, additional conceptual frameworks and architectural elements are likely to be represented and referenced within service descriptions
  – e.g., trust, business transactions, authority, delegation, etc.
  – Facilitates reuse of externally developed frameworks
Visibility

- Visibility - capacity for those with needs and those with capabilities to be able to see each other
  - i.e., the possibility of matching needs to capabilities
  - Typically done by providing descriptions for such aspects as
    - functions and technical requirements,
    - related constraints and policies,
    - mechanisms for access or response
  - Descriptions need to be in a form in which syntax and semantics are
    - widely accessible
    - and understandable
Outline

• SOA - Setting The Context

• Web Service Specification Languages
  – Web Services: Basic Standards
  – The Need for More Semantics
  – What the Semantic Web Can Offer

• OWL-S: Semantics for Web Services
  – Approach
  – Status & Relationship with Industry Standards
  – Sample Applications

• Service Description Framework (SDF)
  – Motivations
  – Approach
  – Status
Web Services: The Essence

• “Loosely coupled software components that interact with one another dynamically via standard Internet technologies” (Gartner)

• Reliable, ubiquitous software interoperability
  – Across networks
  – Across organizations
  – Non-proprietary standards

• Focus on communications; content exchange
  – Basic infrastructure & tools
WS: The Broader ("dotcom") Vision

• Widely distributed, decentralized, reusable capabilities
• Accessible from a variety of platforms & devices
• The Internet as a global platform where organizations and individuals engage in cooperative activities & transactions
  – Highly dynamic, flexible “virtual organizations”
• Adaptive, composable workflows

“When new techniques improve the reaction times of organizations and people from weeks to seconds, they change the very structure of business. This is not a mere quantitative change, but a major qualitative change.” (Singh & Huhns)
WS: Basic Building Blocks

- UDDI Registry
  - Points to Description
  - Points to Service
  - Finds Service
  - Communicates with XML Messages
  - Describes Service
- Service Consumer
- WSDL
- SOAP
- Web Service
I can receive a message having this form ...

And I will reply with a message having this form ...

On port 5552, using HTTP transport, SOAP format
Limitations of WSDL

UDDI Registry

Finds Service

Points to Description

WSDL

Describes Service

Syntax Only

Service Consumer

Communicates with XML Messages

SOAP

Web Service

Dec. 6, 2006 Semantics for WS Specification Bashioum / Martin
Semantics Needed

To use this service you must be a member of AAA.

If you’ve been a member for 3 or more years, you get a 15% discount.
Semantics Needed

When you access this service, you may use TLS or WS-Security.

WS-Security is preferred.

Using TLS costs $9; using WS-Security $15.
Semantics Needed

If I fail to deliver this item within 7 days, I will pay a 30% penalty.
Semantics Needed

You can only access this information if you agree to make changes to it freely available.

Web Service

Describes Service

???
Semantics Needed

I will arrange for the requested book to be shipped to you

and

I will debit your credit card account for the listed price
How Can “More Semantics” Add Value to WS?

Many ways; for example –

• More effective service discovery
• (Semi-) Automatic service composition
• Automatic monitoring, recovery from failure
• Verification of service specifications & uses
• More powerful tools
• More effective documentation
So How Do We Get There?

Many possible paths

• This presentation is about the most comprehensive approach available
  – Semantic Web Services
• And its relationship to (basic) Web Service standards
What is the Semantic Web (1)

Problem:

Computers cannot process most of the information stored on web pages

Solution:

Augment the web to link machine-readable knowledge to web pages
- Extend RDF with Description Logic
- Use a frame-based language design
- Create the first fully distributed web-scale knowledge base out of networks of hyperlinked facts and data

Approach:

Design a family of new web languages
- Basic knowledge representation (OWL)
- Rules (SWRL, RIF)
- Process representation (OWL/S)

Build definition and markup tools

Link new knowledge to existing web page elements

Test design approach in the Intelligence Community and others

Standardize the new web languages

Computers require explicit knowledge to reason with web pages

People use implicit knowledge to reason with web pages

Courtesy of Mark Greaves, DARPA (now at Vulcan, Inc.)
What is the Semantic Web (2)?

- A Vision for the Evolution of the Web
  - An (envisioned) pervasive information infrastructure
  - A web for machines as well as people
- A Growing Collection of Web-enabled Ontologies and Knowledge Bases
- A Set of “Knowledge Representation” Languages
  - RDF, RDF-S, Web Ontology Language (OWL)
    - W3C Standards
    - “Mandated” by DoD IT Standards Registry
  - Rules Interchange Format (RIF, under development)
- Reasoners and Tools for working with the languages
- A Research Area
What are Ontologies & OWL?

- An **Ontology** is a set of concept definitions, defined formally in a way that machines can reason about.
- **OWL** is a language for defining ontologies:
  - Well suited for use on the Web *(or elsewhere)*
  - Well suited for arranging concepts in class hierarchies
  - XML-based
Objectives of Semantic Web Services

- Automation of service use by software agents
  - Ideal: full-fledged use of services never before encountered
- Enable reasoning / planning about services
  - e.g., On-the-fly composition
- Comprehensive framework for the entire lifecycle of service mgmt. tasks
  - Discovery, selection, composition, invocation, monitoring, ...
- Integrated use with information resources
- Ease of use (for users and developers)
Automation Envisioned by SWS

• Web service discovery
  – Find a shipping service that transports goods to Dubai

• Web service enactment
  – 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?
Convergence of WS and SWS

SWS: Selecting, distilling, & applying mature semantic technologies

WS: Building up from syntactically well-formed documents (XML)

Greater expressiveness

Time
Evolution Towards an “Agent-Oriented Web”

- Semantic Web
- Semantic Web Services
- WWW
- Web Services

Directions & Challenges

Static Dynamic

Semantics Syntax
Summary: WS + SW = SWS

- WS provides the building blocks for
  - Constructing, publishing, finding, interoperating with behavioral building blocks --- mostly manually
  - Encapsulation, reuse, Web access
  - Lightweight, language- and vendor-neutral deployment
- WS makes available a vast global repository of interoperable services / procedures / devices
  - But labor-intensive, expertise-intensive
- SW provides infrastructure and technology for reasoning about this world of services
  - Using services more effectively
  - Providing better tools; building more robust services
  - Fuller automation of service use by software agents
  - Support for non-expert developers, end-users
- WS + SW allows us to create a powerful “Agent-Oriented Web”
Outline

• SOA - Setting The Context
• Web Service Specification Languages
  – Web Services: Basic Standards
  – The Need for More Semantics
  – What the Semantic Web Can Offer
• OWL-S: Semantics for Web Services
  – Approach
  – Status & Relationship with Industry Standards
  – Sample Applications
• Service Description Framework (SDF)
  – Motivations
  – Approach
  – Status
Contributors to OWL-S
(partial list)

**BBN**: Mark Burstein

**CMU**: Katia Sycara, Massimo Paolucci (now at Docomo Labs), Naveen Srinivasan (now at WebMethods)

**De Montfort University**: Monika Solanki

**Univ. of Maryland / College Park**: Bijan Parsia, Evren Sirin

**NIST**: Craig Schlenoff

**Nokia**: Ora Lassila

**SRI**: David Martin, Grit Denker, Daniel Elenius

**Stanford KSL**: Deb McGuinness

**Univ. of Southampton**: Terry Payne

**Univ. of Toronto**: Sheila McIlraith

**USC-ISI**: Jerry Hobbs

**Yale**: Drew McDermott
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/owl-s
Upper Ontology of Services

Ontology images compliments of Terry Payne, University of Southampton
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 (partial)
Class Hierarchies of Services

Tie in with UNSPSC, etc.
DL Basis for matchmaking
Multiple profiles; multiple hierarchies
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
Upper Ontology of Services

Ontology images compliments of Terry Payne, University of Southampton
Process Model: “How does it work?”

Process
– Potentially interpretable description of service provider’s behavior
– Tells service user how and when to interact (read/write messages)

• Used for:
  – Service invocation, planning/composition, interoperation, monitoring

• All processes have
  – Inputs, outputs, preconditions and effects

• Composite processes
  – Control flow
  – Data flow

• OWL standard serializations; presentation syntax
Function/Dataflow Perspective

Input:
- customer name
- flight number
- credit card
- ...

Output:
- confirmation no.
- ...
- flight available + valid credit card
- failure notification
- ...

www.acmeair.com book flight service

OWL-S / Process Model
**Action/Process Perspective**

**Input:**
- customer name
- flight number
- credit card
- ...

**Preconditions:**
- knowledge of the input
- ...

**Output:**
- confirmation no.
- ...

**Effect:**
- ticket purchased
- credit card debited
- ...

**Integration:**
- www.acmeair.com
- book flight service

**Decision:**
- flight available + valid credit card

**Output:**
- failure notification
- ...

**Effect:**
- <no effect>

Dec. 6, 2006  Semantics for WS Specification  Bashioum / Martin
Composite Process

Input & Preconditions

www.acmetravel.com
book travel service

www.acmeair.com
book flight service

www.acmehotel.com
book hotel service

www.acmecar.com
book car service

Output & Effects

• customer name
• flight numbers
• dates
• credit card no.
•...•

• confirmation no.
•...•

• confirmation no.
•...•

• failure notification
•...•

• failure notification
•...•

• failure notification
•...•

OWL-S / Process Model
Upper Ontology of Services

Ontology images compliments of Terry Payne, University of Southampton
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
  - Atomic Process
    - Operation
    - Binding to SOAP, HTTP, etc.
- Resources/Concepts
  - Inputs / Outputs
    - Message

WSDL
Sample Applications Using OWL-S

• Many Application Domains, including:
  – Ubiquitous / Mobile Devices & Environments
  – eScience
  – Autonomous Vehicles
  – Software Interoperability
OWL-S: Summary & Status

- Describes “what it does”, “how it works”, “how to access it”
  - Profile, Process, Grounding subontologies
- Ties in naturally with WSDL, UDDI
- Additional semantics supports
  - Automation of a wide range of Web service tasks

- W3C member submission
  - http://www.w3.org/Submission/2004/07/
- 1.2 release upcoming this month
- Publications, tools, examples
  - See http://www.daml.org/services/owl-s/
  - See http://www.semwebcentral.org
Semantic Annotations for WSDL (SA-WSDL)

- Adds WSDL attributes allowing for semantic annotation of
  - Interfaces
  - Operations
  - Faults
  - Simple Types
  - Complex Types
  - Elements
- Agnostic about what semantic frameworks are referenced
- `liftingSchemaMapping` and `loweringSchemaMapping`

- W3C Working Group
- Last Call Working Draft
  - [http://www.w3.org/TR/sawSDL/](http://www.w3.org/TR/sawSDL/)
WSDL + SA-WSDL
WSDL + SA-WSDL + OWL-S
WSDL + SA-WSDL + OWL-S + SDF
Outline

• SOA - Setting The Context
• Web Service Specification Languages
  – Web Services: Basic Standards
  – The Need for More Semantics
  – What the Semantic Web Can Offer
• OWL-S: Semantics for Web Services
  – Approach
  – Status & Relationship with Industry Standards
  – Sample Applications
• Service Description Framework (SDF) ←
  – Motivations
  – Approach
  – Status
Motivation

• What Does it Take to Make Services understandable to both people and machines,

• Across the entire lifecycle of the service,

• And in the larger context of all system artifacts related to the service
Service Description Framework - What Is It?

• Proposed schema for describing services
  – Service descriptions are necessary to make services visible
  – *Sufficient* service descriptions are necessary to make services understandable
  – Consistent service descriptions are necessary to make service descriptions "findable" across enclaves

• Rich enough to be a specification for a service
  – More semantically rich than WSDL

• Builds on concepts identified in WSDL, WSDL-S, and OWL-S
Service Definition Framework

• Supports semantics across ownership boundaries
  – Organizations (or Communities of Interest in DoD terms) exposing services to "outsiders"

• Supports understandability across entire lifecycle of a service
  – Intended to be added to as service goes through lifecycle

• Supports all system artifacts related to a service
  – i.e., it is not just for the benefit of the consumer
Context - WSDL Document

- WSDL document uses the following elements in the definition of network services:
  - **Types** – The types element encloses data type definitions that are relevant for the exchanged messages
  - **Message** – Messages consist of one or more logical parts. Each part is associated with a type from some type system using a message-typing attribute
  - **Port Type** – A port type is a named set of abstract operations and the abstract messages involved.
    - **Operation** – an abstract description of an action supported by the service. Operations refer to the messages involved using the message attribute
  - **Binding** – A binding defines message format and protocol details for operations and messages defined by a particular portType. There may be any number of bindings for a given portType.
  - **Port** – A port defines an individual endpoint by specifying a single address for a binding.
  - **Service** – A service groups a set of related ports together.
- Abstract definition is separated from concrete network deployment or data format bindings.
Context - OWL-S Profile

• Who Provides The Service
  – Contact Information
  – Human Readable
  – E.g., maintenance operator, Customer service rep, etc.

• Functional Description
  – Inputs required (e.g., Credit Card #, Expiration Date)
  – Outputs generated (e.g., receipt)
  – Preconditions required (e.g., valid credit card)
  – Effects expected (e.g., credit card is charged)

• Other Characteristics
  – Category
  – Quality rating
  – Unbounded list of service parameters
    ▪ Can contain any type of information

(See additional information in notes)
Information describing the service, focused on semantics

Identifies mechanisms, criteria and restrictions to access the service, as well as providing official IA information security metadata markings

Defines level(s) of service provided, and nominal network parameters under which levels of service hold

Provides example design patterns for consumers of the service, schedule dates for the service implementation, and URI to operational info for the service (to be used by operational manager or hosting manager)

Information on Points of Contact for this service – such as owner, operational manager, help desk, chief engineer, etc.

Information required to actually bind to the service implementation (or implementations)
SDF Expanded
Security Ontology

- Security Coalition in DAML program
  Various members from SRI, UMBC, CMU, etc.
- Ontologies and matching algorithms are documented in http://www.daml.org/services/owl-s/security.html
- Basis for NRL Security Ontology

- Ontologies:
  - Credential (Simple, Composed, Cookie, Login, Certificate, …)
  - SecurityMechanism (Authentication, Authorization, Access Control, …)
  - Service Security Extensions (securityRequirement, securityCapability)
  - Agent Security Extensions (securityRequirement, securityCapability)
  - Privacy (Policy has rules, each rule has an action and is applied on a resource; three types of rules: authorization, obligation, capability)
# SDF Info vs. Service Development Life Cycle (notional)

<table>
<thead>
<tr>
<th>Lifecycle SDF Cat.</th>
<th>Concept</th>
<th>Req'ts &amp; Arch</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Deploy</th>
<th>Operate</th>
<th>Deprecate</th>
<th>Retire</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Level Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementation Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POC Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAP Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Where We Are Going

• Actively coordinate OWL-S, SA-WSDL, and SDF
  – So they work together "seamlessly"
• Benefits include:
  – Facilitate reuse/composability of service descriptions
  – Improved search & discovery
  – Facilitate reasoning
  – Facilitate automation
Questions?
The End
Extra Slides
Why the Distinction Between Service and Capability?

- SOA is a means of organizing solutions that promotes reuse, growth and interoperability
  - Not itself a solution to domain problems
  - rather an organizing and delivery paradigm that enables one to get more value from use of capabilities
    - which are locally “owned”
    - and those under the control of others
  - Enables one to express solutions in a way that makes it easier to modify or evolve the identified solution or to try alternate solutions

- SOA does not provide any domain elements of a solution that do not exist without SOA
  - i.e., if you "remove" the SOA, you will still have the capabilities
  - Therefore - SOA Service exists apart from the capability
Business Service vs. SOA Service

- Often confusion in using the term “service”
  - when the context is not clear as to what variation is intended.
  - In everyday commerce, we make use of businesses that develop capabilities to address needs.
  - The SOA-RM perspective is that the underlying capability provides the real world effect that someone wants.
  - If the real world effect is not a physical product, we tend to say the business provides a service.

- Business service = underlying capability being provided
  - an action (task or business function) that a business (or organization) does for "someone else".

- SOA services = IT artifact that makes possible the efficient connectivity between consumer needs and provider capabilities.
Semantics = Meaning = Relationships

• Humans (and our machines) only ever understand anything in so far as it is related to other things

Courtesy of Chuck Mosher, Metamatrix
Semantics = Meaning = Relationships

- Ontologies help establish relationships between things
Semantics = Meaning = Relationships

- Humans (and our machines) only ever understand anything in so far as it is related to other things.
Semantics = Meaning = Relationships

- Humans (and our machines) only ever understand anything in so far as it is related to other things.

![Diagram showing relationships between CARD, ID, LICENSE, and BADGE. Courtesy of Chuck Mosher, Metamatrix.]
Security Ontology

• Security Coalition in DAML program
  Various members from SRI, UMBC, CMU, etc.
• Ontologies and matching algorithms are documented in http://www.daml.org/services/owl-s/security.html
• Basis for NRL Security Ontology

• Ontologies:
  – Credential (Simple, Composed, Cookie, Login, Certificate, …)
  – SecurityMechanism (Authentication, Authorization, Access Control, …)
  – Service Security Extensions (securityRequirement, securityCapability)
  – Agent Security Extensions (securityRequirement, securityCapability)
  – Privacy (Policy has rules, each rule has an action and is applied on a resource; three types of rules: authorization, obligation, capability)
NRL Security Ontology – Security Mechanism

Source of Figure: [1]
NRL Security Ontology (cont) – Security Concept and Security Objective

securityObjective
- confidentiality
- availability
- userAuthentication
- messageAuthentication
- authorization
- messageIntegrity
- keyManagement
- trust
- hostTrust
- replayPrevention
- covertChannelPrevention
- separation
- trafficHiding
- anonymity

securityConcept

securityProtocol
- hasAssurance
  - range: &securityAssurance;
  - assurance
  - hasAlgorithm
    - range: &securityAlgorithms;
    - algorithm
  - reqCredential
    - range: &credentials;credential

securityMechanism
- hasAssurance
  - range: &securityAssurance;
  - assurance

securityPolicy
- RBAC

very restrictive view

needs work!

not much semantics

source of figure: [1]
NRL Security Ontology (cont) - Objective

Source of Figure: [1]
Algorithm

<table>
<thead>
<tr>
<th>KeyExchangeAlgorithm</th>
<th>EncryptionAlgorithm</th>
<th>ChecksumAlgorithm</th>
<th>SignatureAlgorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oakley</td>
<td></td>
<td>CRC-8</td>
<td></td>
</tr>
<tr>
<td>DiffieHellman</td>
<td></td>
<td>CRC-16</td>
<td></td>
</tr>
<tr>
<td>KEA</td>
<td></td>
<td>CRC-32</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SymmetricAlgorithm</th>
<th>AsymmetricAlgorithm</th>
<th>HashAlgorithm</th>
<th>MACAlgorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES (keylength = 64)</td>
<td>RSA</td>
<td>SHA-1</td>
<td>HMAC</td>
</tr>
<tr>
<td>AES</td>
<td>ECC</td>
<td>SHA-256</td>
<td>CRC-8</td>
</tr>
<tr>
<td>Blowfish</td>
<td></td>
<td>RIPEMD</td>
<td>CRC-16</td>
</tr>
<tr>
<td>TripleDES (hasNSALevel = &amp;assurance; type3)</td>
<td></td>
<td>MD4</td>
<td>CRC-32</td>
</tr>
<tr>
<td>CAST</td>
<td></td>
<td>MD5</td>
<td></td>
</tr>
<tr>
<td>Skipjack (hasNSALevel = &amp;assurance; type2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRAYON (hasNSALevel = &amp;assurance; type1)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source of Figure: [1]
NRL Security Ontology (cont) - Assurance

byOrganization (Range:string)

Assurance

Standard

Accreditation

Evaluation

Certification

FIPS
- FIPS140-2
- FIPS46-3
- FIPS180-2

NSA
- Type1
- Type2
- Type3
- Type4

TCSEC (comment: Orange Book)

CommonCriteria

DITSCAP
- EAL1
- EAL2
- EAL3
- EAL4
- EAL5
- EAL6
- EAL7

DivisionA
- A1

DivisionB
- B1
- B2
- B3

DivisionC
- C1
- C2

DivisionD

Source of Figure: [1]
NRL Security Ontology (cont) – OWL-S Connection

Mechanisms to “hook” security specification into OWL-S specification

Source of Figure: [1]
NRL Security Ontology (cont) – Agent and InformationObject

- `securityRequirement` (range: &SecurityMain;SecurityConcept or &SecurityMain;SecurityObjective)
- `securityCapability` (range: &SecurityMain;SecurityConcept or &SecurityMain;SecurityObjective)

Source of Figure: [1]
NRL Security Ontology (cont) – Security Protocol

Source of Figure: [1]