From the “Eyeball” Web to the Transaction Web

Katia Sycara
ISA – Intelligent Software Agents Lab
Carnegie Mellon University
katia@cs.cmu.edu
http://www.cs.cmu.edu/~softagents/
Talk Overview

- Eyeball Web and Transaction Web
- Requirements and Challenges
- Semantic Web
- Multi Agent Systems (MAS)
- Web Services
- Semantic Web Services
  - DAML-S/OWL-S
- Future Directions
The “Eyeball” Web

- More than 500 million users
- 3 billion pages and growing
- Distributed in the extreme
- Only human understandable
- Variable information quality
- It is difficult to find information
The “Transaction” Web

- Machine understandable Web
- Web filled with executables
- Enable automated information and knowledge management
- Enable automated e-commerce and e-science transactions
Requirements and Challenges

- Information integration across the Web (currently the user is the “glue”)
- System integration/interoperability Web-wide (within and across organizations)
- Consistency of behavior of long running transactions (both for e-commerce and e-science) in the face of partial, distributed failures

Tension between the “anarchy” of the Web and the requirements for reliable, consistent, trusted transactions of e-commerce and e-science
The Integration Challenge

Integrate multiple independent and heterogeneous
- Data repositories
- Processes
- Applications

Ensure
- Semantic equivalence of equivalent concepts
- Performance
- Communication of “semantic” agreements
- Dynamism: real time access works accurately
- Flexibility: systems enter and leave integration
The Integration Past and Future?

Numerous solutions to date (since the ’70s):
- Tool driven
- Data driven
- Process driven
- Model driven
- Web services

World Wide Annual Integration plus Data Quality Costs:
$1 Trillion / year

“The problem is not in the plumbing. It’s in the semantics”
(quotation from Michael Brodie’s invited talk at ISWC 2003)
Three Significant Technological Visions

- Semantic Web
- Multi Agent Systems (MAS)
- Web Services (Service Oriented Architectures)
Talk Overview

- Eyeball Web and Transaction Web
- Requirements and Challenges
- Semantic Web
- Multi Agent Systems (MAS)
- Web Services
- Semantic Web Services
  - DAML-S/OWL-S
- Future Directions
Semantics on the Web

Goal of Web Semantics: Development of common representation format, shared ontologies, reasoning mechanisms, and query engines to support improved utilisation of Web knowledge

Semantic Web technologies and tools
- Ontobroker
- iBrow
- SHOE
- Rdf
- DAML+OIL/OWL

Semantic Web: semantic metadata and ontologies for web content to enable information access, integration, interoperation and consistency.
Bringing Semantics to the Web

Semantic Web vision

Web wide autonomous information access and integration

Source: Terry Payne
Talk Overview

- Eyeball Web and Transaction Web
- Requirements and Challenges
- Semantic Web
- Multi Agent Systems (MAS)
- Web Services
- Semantic Web Services
  - DAML-S/OWL-S
- Future Directions
Agent Characteristics

A software agent is a computational entity that acts on behalf of a human, organization or other agent. An agent’s key characteristics are:

- **Situatedness**: agent receives sensory inputs from its environment and outputs actions (physical environments, Internet)

- **Autonomy**: agent is goal-directed and has control over its internal state (can reason and act without direct intervention of others)

- **Interactivity/Sociability**: agents interact in peer to peer fashion with humans and/or other agents in pursuit of their goals (individual or collective)

- **Adaptivity**: agents respond reactively or proactively to changes in their environment
MAS Interactions

Agents interactions are typically:

- Peer to peer (agent p2p networks are heterogeneous) where both parties can initiate interaction
- Mediated (Agents contact a 3rd party to mediate with location and/or various types of interaction with other agents (e.g., capabilities, trust, verification, ontology translation, protocol translation, language translation, qos, etc))

- Agents typically have persistent identity in the MAS
- Agents typically have roles in the MAS that dictate and constrain their scope of activity
- Agent interactions result in meaningful compositions of results to achieve overall system objectives or self interested agent objectives
Autonomous behavior is needed for Open Environments & Complex Interactions

Open Environments
- No predefined structure
- Agents leave and join the society dynamically
- Communication is not ensured all the time
- Information sources may appear and disappear

Complex Interactions
- Dynamic task decomposition & allocation
- Provider selection
- Planning for composition of MAS reasoning
Generic Tasks in Open Environments

Agents must be able to:

- **discover** each other. We distinguish the notion of agent identity from the notion of agent **functionality**.
  - Identity is found through Agent Name Services (ANS)
  - Functionality/capability is found through Middle Agents

- **interact/transact** with each other
- **compose** results of their reasoning
- **monitor** progress of delegated tasks
Multi-Agent Organization

Distributed adaptive collections of agents (autonomous services) that coordinate to do tasks on the user’s behalf.
Agent Discovery and Mediation

- Agents that provide services advertise their expertise/capabilities to middle agents
- Requester agents ask middle agents for agents with particular capabilities
- Middle agents match requests to advertisements and return results
- Communication protocols include formal semantics and ontologies for interoperation
- The discovery scheme enables system robustness through functional substitutability of agents
- Our research at CMU has identified 28 types of middle agents with different functionalities, protocols and performance tradeoffs

**Matchmaker**

- **Requester**
  - Request for service
  - Contact information of providers that match the request
  - Delegation of service
  - Results of service request

- **Matchmaker**
  - Advertisement of capabilities +para.

- **Provider 1** • • • **Provider n**
Facilitator

Delegation of service
+ preferences

Requester

Results of service

Facilitator

Delegation of service

Results of service

Advertisement of capabilities
+ para.

Provider 1 • • • Provider n
Talk Overview

- Eyeball Web and Transaction Web
- Requirements and Challenges
- Semantic Web
- Multi Agent Systems (MAS)
- Web Services
- Semantic Web Services
  - DAML-S/OWL-S
- Future Directions
**Web Services - A New Paradigm?**

*Web Services heralded as:*

- “… self-contained, self-describing, modular applications that can be published, located, and invoked across the Web…”

*Which will allow…*

- …large companies to shrink around their core competencies into small, flexible, and highly profitable units
- …on the fly composition of new functionality through the use of loosely coupled reusable software components
- …decomposition and distribution of large-scale processing tasks into component tasks executed simultaneously across many devices

“Web services are expected to revolutionize our life in much the same way as the Internet has during the past decade or so.” (Gartner)
So what is new about Web Services?

<table>
<thead>
<tr>
<th>Component-Based Model</th>
<th>Web Services Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tightly coupled software applications (high dependencies between systems)</td>
<td>Loosely coupled software applications (low dependencies between applications)</td>
</tr>
<tr>
<td>Mainly designed for processes within the enterprise</td>
<td>Mainly designed for processes across enterprises</td>
</tr>
<tr>
<td>Uses different protocols and technologies (e.g., Microsoft DCOM, CORBA)</td>
<td>Uses common protocols and technologies (e.g., XML, SOAP, WSDL, HTTP)</td>
</tr>
</tbody>
</table>
The Impact of Web Services?

- Improvement of operations
- Agile business relationships
- Reduced cost and increased flexibility
- Shorter time-to-market for new products and services
- Leverage existing infrastructure

Web Services will remedy many expensive and painful problems of today’s business uses of IT:

- Connecting business systems inside a firm is a nightmare
- Inter-enterprise process orchestration is impossible
- Inflexible systems impede business adjustments
- Fragmented personal data frustrates users
Web Services Standards

- **SOAP**: XML based web services communication protocol
  - Limitations:
    - Unbounded message format
    - Has no communicative speech acts (cannot determine intention of actors or type of the message)

- **WSDL**: Structured mechanism to describe a WS interface
  - Abstract operations that a Web Service can perform
  - Format of messages it can process
  - Protocols it can support
  - Physical bindings to URIs and protocols
  - Limitations:
    - No semantics for message sequencing and correlation
    - No semantics for message content
Web Services Standards

BPEL: Description of how Web Services are composed

- Flow Model describes the structure of the business process in terms of activities of process steps and data and control links

- Global Model
  - Describes interaction between provider and requester
  - Mappings between internal operations and WSDL port types

Limitations

- No IOPEs
- Allows execution of a manually constructed composition

UDDI: Directory Service for Web Services

Limitations: keyword searches, limited capability search
Web Services Reality

Web Services Usage in 2003 (1)
- XML: 87%
- SOAP: 31%
- WSDL: 3%
- UDDI: 14%

Web Service Plans in 2003 (2)
- 85% Web Services internally
- 57% Web Services for customers
- 44% adopt supplier Web Services
- 24% join standards activities

Risks
- Problems with what is in place (e.g. XML, SOAP, UDDI, WSDL)
- Incomplete technical solutions
- Standards take time and are competing
- Vendor cooperation breakdown

----------------------
1. Gartner, April 2003
2. Forrester, Web Services Reach the Big Time, September, 2003
Limitations of Current Web Service Standards

- Lack computer understandable semantics
- Provide limited semantics based discovery and Interoperation
- Lack ability for run time service discovery, negotiation, execution monitoring and composition
- Lack mechanisms for describing business relations, or agreements, e.g. contracts
Talk Overview

- Eyeball Web and Transaction Web
- Requirements and Challenges
- Semantic Web
- Multi Agent Systems (MAS)
- Web Services
- Semantic Web Services
  - DAML-S/OWL-S
- Future Directions
DAML-S

DAML-S: A DARPA Agent Markup Language for Services
- An upper ontology for describing properties & capabilities of Web services in an unambiguous, computer interpretable markup language.
- DAML+OIL/OWL Ontology for (Web) services

AI-inspired markup language:
- tailored to the representational needs of Services
- well-defined semantics
- ontologies support reuse, mapping, succinct markup, ...
- enables automation of service use by agents
- enables reasoning about service properties and capabilities

http://www.daml.org/services/owl-s/1.0
Acknowledgements to the DAML-S Web Services Coalition

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

**BBN:** Mark Burstein

**De Montfort University:** Monika Solanki

**ISI:** Jerry Hobbs

**U. Of Maryland:** Bijan Parsia

**Nokia:** Ora Lassila

**Southampton:** Terry Payne

**Stanford KSL:** Sheila McIlraith, Honglei Zeng

**SRI:** David Martin

**Vrije Universiteit:** Marta Sabou

**Yale:** Drew McDermott


Automation enabled by DAML-S

- Web Service Discovery & Selection
  - Find me an airline that can fly me to Edinburgh
- Web Service Invocation
  - Book flight tickets with British Midland to arrive on June 23rd
- Web Service Composition & Interoperation
  - Arrange travel in Edinburgh.
- Web Service Execution Monitoring
  - Has the hotel room been reserved?

Currently, DAML-S is being converted to OWL-S. OWL is on track to be a W3C recommendation.
Semantic Web Services Stack Diagram

Autonomous Semantic Services

Trusted Semantic Services
Eg Contracts; Info Integration; Business Process Automation; Tasking

Web Services Stack

Choreography eg BPEL
Process Specifications
Directories, e.g. UDDI
Messaging
WSDL
SOAP

Semantic Web Stack

XML

Namespaces
URI
Unicode
Encryption, Signature

Trust
Proof
Logic Framework
Rules eg RuleML
DAML+OIL
RDF Schema
RDF Model & Syntax

Copyright Semantic Web Services Initiative
## DAML-S Elements

<table>
<thead>
<tr>
<th>Discovery</th>
<th>DAML-S</th>
<th>Web Services Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Profile</td>
<td>UDDI API</td>
</tr>
<tr>
<td>Choreography</td>
<td>Process Model</td>
<td>WSCI BPEL4WS</td>
</tr>
<tr>
<td>Invocation</td>
<td>Grounding+ WSDL</td>
<td>WSDL</td>
</tr>
</tbody>
</table>

ODBase ‘03
Constructing a service

- Describe the process model
  - Atomic (functional) or composite (conversational)
  - Determining what to expose
    - Just interaction points
    - Additional process information for reasoning
- Describe the profile and advertise
  - Expose input & outputs, preconditions and effects
  - Determine relevant metadata
- Bind to a transport mechanism via the grounding
  - Provide (or augment existing) WSDL document and bind to it
- Define the Service concept to link the models together
  - presents profile
  - describedBy process
  - supports grounding
DAML-S Service Profile

Functionality Description

**Preconditions**
- Set of conditions that should hold prior to the service being invoked

**Inputs**
- Set of necessary inputs that the requester should provide to invoke the service

**Outputs**
- Results that the requester should expect after interaction with the service provider is completed

**Effects**
- Set of statements that should hold true if the service is invoked successfully.
- Often refer to real-world effects
DAML-S Service Profile

Parameters

- Provider’s contact information
- Security
- Range of operations of provider
- Quality ratings
- QoS
- ...

ODBase '03
DAML-S Matchmaker

DAML_S Matching Engine

Advertisements Data Base

Matching Engine

Ontologies Data Base

DAML+OIL Reasoner

Web Based DAML Ontologies

www.ri.damlsmm.cmu.edu
Architecture DAML-S/UDDI

DAML-S/UDDI Matchmaker

Communication Module

DAML-S/UDDI Translator

DAML-S Matching Engine

UDDI Registry

Web based DAML Ontologies
Requirements of Interaction

- **Shared Knowledge of interaction protocols:**
  - What information the provider needs
  - When does it need it

- **Shared understanding of content of messages**
  - Ontologies
  - Logic framework for correct interpretation

- **Agreement on ports and low level details**
Autonomous Invocation

Provider publishes
- Process Model
- Grounding, WSDL

Requester uses them to initiate the interaction with the provider
Process Model

- **Detailed view of the Web Service**
  - Provides description of how the Web Service accomplishes its tasks

- **Interaction protocol**
  - Specifies when Web Service requires inputs from requesters
  - Specifies what information (and when) Web Service sends to the requesters
Process Model

Processes are conceived as:
- Atomic
- Simple
- Composite

Associated with each service is a set of:
- Inputs
- Outputs
- Preconditions
- Effects

Invocable processes have an associated grounding:
- Includes WSDL description to model:
  - Operation
  - Message formats
  - Ports & Bindings
Process Model

Composite processes are compositions of simple or other composite processes in terms of constructs:
- Sequence
- if-then-else
- Fork
- Etc.

Data flow and Control flow should be described for each composite service

A black box and glass box view may be given of each composite service
Composite Process Example (Sequence)

```xml
<rdfs:Class rdf:ID="BookFlight">
    <rdfs:subClassOf rdf:resource="#CompositeProcess" />
    <rdfs:subClassOf rdf:resource="http://www.daml.org/Process#Sequence" />
    <daml:subClassOf>
        <daml:Restriction>
            <daml:onProperty rdf:resource="http://www.daml.org/Process#components" />
            <daml:toClass>
                <daml:subClassOf>
                    <daml:unionOf rdf:parseType="daml:collection">
                        <rdfs:Class rdfs:about="#GetFlightDetails" />
                        <rdfs:Class rdfs:about="#GetContactDetails" />
                        <rdfs:Class rdfs:about="#ReserveFlight" />
                        <rdfs:Class rdfs:about="#ConfirmReservation" />
                    </daml:unionOf>
                </daml:subClassOf>
            </daml:toClass>
        </daml:Restriction>
    </daml:subClassOf>
</rdfs:Class>
```
Grounding

- Specifies mapping to WSDL
  - Atomic Processes map to Operations
  - Inputs/Outputs described as messages
  - Specify XSLT transformations for mapping to and from DAML and XSD types
DAML-S VM

DAML-S processor that allows any Web service to interact with Web services using only DAML-S specifications
Features of DAML-S VM

- Uses DAML-S to represent service descriptions
- Uses DAML to represent information to exchange between Web services
- Actively adopts logic inference to reason about DAML-S and DAML ontologies
- Shows how to integrate DAML-S within Web services technology such as Axis
WSDL2DAML-S Converter

- Provides partial conversion from WSDL Web service descriptions to DAML-S descriptions
  - Generates complete specification of Grounding
  - Partial Specification of the Process Model
    - Including Atomic Processes
  - Partial Model of the Profile
- Resulting models require additional annotations to include semantic descriptions
- Combined with Java2WSDL to provide Java2DAML-S
- Web Based Interface

http://www.daml.ri.cmu.edu/wsdl2daml/s/
Composition of Services

Services may themselves be composed by a number of other services.

- Can be broken down into a hierarchy of subtasks
- Subtasks may be part of a larger service offered by a service provider
  - e.g. process of logging into an account
- May be offered by a different service provider
  - e.g. booking a hotel as part of a travel plan
DAMLzon: DAML-S for Amazon.com

- WSDL2DAML-S used to generate DAML-S for Amazon’s Web Service
- DAML-S VM used to interact with Amazon Web service

Process Model for Amazon.com
Performance

- DAML-S VM client on browsing + reserving task
- Analyzed data by computing:
  - Time required by DAML-S VM to execute Process Model
  - Time required for data transformation to fit Amazon requirements
  - Time required to invoke an operation on Amazon
- 98 runs total over 4 days in varying load conditions
- Results in milliseconds

<table>
<thead>
<tr>
<th></th>
<th>VM</th>
<th>Data Trsfm</th>
<th>Invocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average percentage</td>
<td>83</td>
<td>156</td>
<td>2797</td>
</tr>
<tr>
<td></td>
<td>3%</td>
<td>5%</td>
<td>92%</td>
</tr>
<tr>
<td>Strd dev</td>
<td>107</td>
<td>146</td>
<td>1314</td>
</tr>
</tbody>
</table>

ODBase ‘03
Contributions of the three technologies

- Semantic Web: ontologies, metadata annotations and knowledge based inference
- Multi Agent Systems: goal-directedness, semantically meaningful communication protocols, cooperative and self interested reasoning mechanisms
- Web Services: industry standards and industry buy in
Future Evolution of Web Services: Realizing the Transaction Web

- Semantic Web Services: services whose description is in a machine understandable language with formal semantics
- Agent-based or Autonomous Semantic Web Services:
  - goal-directed
  - autonomous choice of partners
  - based on own current internal goals, internal attitudes and their projection of their future needs.
  - semantics support ASWS

Semantic Web Services Interest Group created at W3C
public-sws-ig@w3.org