Simulation, Verification, & Automated Composition of Web Services

Srini Narayanan
ICSI & UC Berkeley
(formerly SRI)
snarayan@icsi.berkeley.edu

Sheila McIlraith
KSL, Dept. Comp. Science
Stanford University
sam@ksl.stanford.edu

WWW 2002
May 8, 2002
Web Services

Web-accessible programs and devices
Knowledge Rep'n for “Semantic Web Services”

DAML-L (Logic)  DAML-SL

DAML-S (Services)  DAML-L (Logic)

DAML+OIL (Ontology)

RDFS (RDF Schema)

XML Schema

RDF (Resource Description Framework)

XML (Extensible Markup Language)
DAML-S: Semantic Markup for Web Services

DAML-S: A DARPA Agent Markup Language for Services

- DAML+OIL ontology for Web services:
  - well-defined semantics
  - ontologies support reuse, mapping, succinct markup, ...

- Developed by a coalition of researchers from Stanford, SRI, CMU, BBN, and Nokia, Yale, under the auspices of DARPA.

- DAML-S version 0.6 posted October, 2001
  http://www.daml.org/services/daml-s

[DAML-S Coalition, 2001, 2002]
[McIlraith, Son & Zeng, 2001]
An Ontology for Service Descriptions

Service presents ServiceProfile
describedBy ServiceModel

ProcessModel

ProcessCtrl

Grounding
Process Model

**hasProcess**

- **service** → **process**
- **atomic process**
- **composite process** → **control constructs**
  - **sequence**
  - **If-then-else**
  - **fork**
  - **while**
  - **...**

**inputs**
- (conditional) outputs
- preconditions
- (conditional) effects
Problem: DAML+OIL has a well-defined semantics, but it is not sufficiently expressive to characterize all and only the intended interpretations of DAML-S.

Solution:
- Model-theoretic semantics defined by a translation to (mostly) first-order logic.
- Distributed operational semantics defined in terms of Petri Nets.
This Talk

✓ DAML-S
  - (conditional) side effects of services are critical for WS composition
  - Description logic not expressive enough for process modeling

➤ Model-Theoretic Semantics for DAML-S

• Distributed Operational Semantics of DAML-S

• Decision Procedures for Web Service Automation

• Implementation of DAML-S Decision Procedures

• Summary & Future Work
Model Theoretic Semantics

Task: Capture the intended interpretation of DAML-S by translating to a more expressive logic.

Approach: Translate DAML-S to Situation Calculus (SC), a first-order logical language for reasoning about action and change.

Key Idea: Preconditions = SC Preconditions
Effects = SC Effects
Inputs = SC Knowledge Preconditions
Outputs = SC Knowledge Effects

<Details of the logical translation are in the paper>
This Talk

✓ DAML-S

✓ Model-Theoretic Semantics for DAML-S
  - Intended semantics of DAML-S process model via translation to FOL action theory.

⇒ Distributed Operational Semantics of DAML-S

• Decision Procedures for Web Service Automation

• Implementation of DAML-S Decision Procedures

• Summary & Future Work
Distributed OPReational (DOPE) Semantics

**Task:** Map Situation Calculus Axiomatization to Petri Net based Formalism [Narayanan 99]

Features of High Level Stochastic Petri Nets

- Natural representation of change and concurrency
- Execution semantics
- Can deal with quantitative information & resources
- Variety of well established analysis and simulation techniques including mappings to other logics of change.
Model Review

Basic Mechanism

1. Precondition arc
2. Resource arc
3. Inhibition arc
Model Review

Firing Semantics

1 -> [3] -> 2

1 -> 1


Narayanan-McIlraith
WWW2002 – 05/08/2002
Model Review

Result of Firing

DEADLOCK!
DAML-S Atomic Processes as Petri Nets

SC Knowledge Preconditions

\[ \text{Poss}_k(a,s) \]

Input

Action \( a \)

Effect

\( Kref \)

\( Kref \)

\( \ldots \)

\( Kref \)
DAML-S Atomic Processes as Petri Nets

...and SC World Preconditions

\( \pi_1 \) \( \pi_2 \) \( \ldots \) \( \pi_n \)

\( Kref \) \( Kref \) \( \ldots \) \( Kref \)

Input

World

\( \wedge \)

\( Poss_w(a,s) \)

Action \( a \)

Effect

\( \wedge \)

\( Poss_k(a,s) \)
Modeling Composite Process Constructs
DAML-S Sequence: P1;P2
DAML-S Sequence: P1;P2

start

Ready → Atomic Process P1 → Done(P1) → Atomic Process P2 → Done(P1;P2)

finish
DAML-S Fork: P1 || P2

- **start**
  - Ready(P1)
  - Ready(P2)

- **Done(P1 || P2)**
  - Atomic Process P1
  - Atomic Process P2

- **finish**
DAML-S Concurrent-Sync

start

Ready(P1) -> Atomic Process P1 -> Done(P1)

Ready(P2) -> Atomic Process P2 -> Done(P2)

finish
This Talk

✓ DAML-S

✓ Model-Theoretic Semantics for DAML-S

✓ Distributed Operational Semantics of DAML-S
  - Situation Calculus translated to Petri Nets
  - Petri Net composition of atomic services

➔ Decision Procedures for Web Service Automation

• Implementation of DAML-S Decision Procedures

• Summary & Future Work
Web Service Automation Tasks

• **Simulation**: simulate the evolution of a Web service under different conditions.

• **Verification**: automatically establish that the Web service upholds specified properties (e.g., that it maintains certain properties, that it ensures safety, etc.)

• **Composition**: automatically generate a composition of Web services that achieves a specified goal.

• **Performance Analysis**: evaluate the ability of a service to meet requirements with respect to throughput times, service levels, and resource utilization.
Key Decision Problem 1: Reachability

**Reachability**: A marking $M$ is *reachable* if it is the marking reached by some *occurrence sequence* (Definition 4, paper). Given a marking $M$ of $N$, the set of reachable markings of the net $(P; T; F; M)$ (i.e., the net obtained by replacing the initial marking $M_0$ by $M$) is denoted by $[M >$.

- Safety $= \text{not (reachable (unsafe state))}$
- Composition $= \text{reachable (goal state)}$
Key Decision Problem 2: Deadlock

**Deadlock:** A marking of a net is a *deadlock* if it enables no transitions. The deadlock problem for a net is the problem of deciding if any of its reachable markings is a deadlock.
Our Web Service Automation Task (Simulation, Verification, Automated Composition) can all be characterized in terms of Reachability and Deadlock
# Complexity of DAML-S Decision Procedures

**Key Idea:** Leverage expressiveness to gain tractability

<table>
<thead>
<tr>
<th>DAML-S Subset</th>
<th>Reachability</th>
<th>Deadlock</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAML-S 0.5</td>
<td>P-Space hard</td>
<td>P-Space hard</td>
</tr>
<tr>
<td>DAML-S \ Iterate</td>
<td>NP-Complete</td>
<td>Polynomial</td>
</tr>
<tr>
<td>DAML-S \ Iterate &amp; Cond</td>
<td>NP-Complete</td>
<td>Linear</td>
</tr>
<tr>
<td>DAML-S \ Iterate &amp; Choice</td>
<td>Polynomial</td>
<td>Polynomial</td>
</tr>
<tr>
<td>+ Resources</td>
<td>Exp-Space-Time hard</td>
<td>Exp-Space-Time hard</td>
</tr>
</tbody>
</table>
This Talk

✓ DAML-S

✓ Model-Theoretic Semantics for DAML-S

✓ Distributed Operational Semantics of DAML-S

✓ Decision Procedures for Web Service Automation
  - reachability & deadlock are key for simulation, verification and automated composition of Web Services.
  - tradeoff DAML-S expressiveness for tractability of decision procs.

→ Implementation of DAML-S Decision Procedures

• Summary & Future Work
Implementation

DAML-S translation to the modeling environment KarmaSIM [Narayanan, 97] (http://www.icsi.berkeley.edu/~snarayan)

Basic Program:

Input: DAML-S description of Web Service

Output: Network Description of Web Service in KarmaSIM

Procedure:

• Recursively construct a sub-network for each control construct. Bottom out at atomic process.
• Construct a net for each atomic process
• Return network
Example


Developed by the DAML-S coalition, publically available at http://www.daml.org/services/daml-s/2001/05/Congo.daml
Implemented Features of Tool

• Interactive Simulation

• Variety of qualitative analysis techniques
  – Reachability
  – Deadlock
  – S and T invariants

• Variety of Quantitative analysis techniques
  – Throughput
  – MAP estimation including Vitterbi paths
This Talk

- DAML-S
- Model-Theoretic Semantics for DAML-S
- Distributed Operational Semantics of DAML-S
- Decision Procedures for Web Service Automation
- Implementation of DAML-S Decision Procedures
  - DAML-S to automated WS tasks

➔ Summary & Future Work
Summary

Claim
- Automation of Web Service Tasks is a key benefit of the Semantic Web.
- Precise description of Web Services is a prerequisite to Web Service automation.

Our Contributions
- Semantics (model-theoretic & distributed operational)
- Decision procedures & expressiveness-tractability tradeoff
- Implementation (from DAML-S to an executing model)

Broader Impact
- results applicable to any WS process model formalism (e.g., XLANG, WSFL, etc.)
Current Status/Work

• Release tool for DAML-S interpretation
  – Awaiting DAML-S stability

• Extend the Model to handle
  – Execution Monitoring
  – Resource Ontologies

• Link to Web Service Entry Tool (KSL)

• Link to general inference tool (SRI)