Model Interchange and Integration for Web Services

Robert J. Hall
bob-2modelxchg-@channels.research.att.com
AT&T Labs Research
Florham Park, NJ - USA

Andrea Zisman
a.zisman@soi.city.ac.uk
City University
London - UK
Outline

• Motivation
• OpenModel Modeling Language (OMML)
• OpenModel Validation Approach
• BookFind Example
  – Model translations
  – Model integration
  – Validation
• Demonstration
• Conclusion and Future Work
Motivation

• Web services are multi-stakeholder distributed systems (MSDS)
• Unrealistic to assume all stakeholders use the same behavioral specification (model) formalism
• This heterogeneity makes it difficult to share and integrate models to validate distributed systems
• Existing interface-based web service standards cannot support model-based validation

Therefore, model-based validation of web services fails for realistic web services
The Solution: OMML

- OpenModel Modeling Language (OMML)
- XML-based model interchange format
- Translation tools to/from other modeling languages
OMML General Design Goals

- *Function rich reactive system* language (functionality and control expressed procedurally in terms of function/object theories, not limited to finite state models)
- *Executable specifications* of single nodes & compositions
  - abstract state model with computable abstraction map
  - explicit connector declaration
  - instance data declaration tolerating missing data
- Support for
  - *systematic validation methods* (symbolic execution and term simplification, specification coverage measurement, theorem proving, model checking)
  - *automated reasoning over arbitrary function rich theories* within shared domain-specific ontologies
Design of OMML

• Based on ISAT’s Procedural Event-Based Formalism (P-EBF)

• Five different types of documents
  – **Theory**: declaration of domain-specific *types, functions* (signatures and body), and *reasoning axioms* governing their semantics
  – **Ontology**: group of related theories
  – **Specification**: synchronous executable behavior model
  – **Connection**: description of how outputs of one node relate to inputs of another node
  – **Model Instance**: instance information for one MSDS node (abstract state data, connections, specification)
Examples - Theory

<Theory TheoryName="LISTS" ... >
<IncludeDeclaration>
<IncludeTheoryFrom
  OntologyName="OntoBasis"
  OntologyLocation="www.att.com">
<IncludeTheory TheoryName="LOGICALS"/>
  ...
</IncludeDeclaration>
<TypeDeclaration>
<Type TypeName="LIST">
  <SubTypeOf OntologyName="OntoBasis"
    TheoryName="BASE"
    SubTypeOfName="THING"/>
  <FunctionDefault
    FunctionName="EMPTY-LIST"/>
</Type>
</TypeDeclaration>
<FunctionDeclaration>
<Function FunctionName="EMPTY-LIST">
  <FunctionReturnValue
    FunctionReturnValueType="LIST"/>
  ...
</Function>
<Function FunctionName="MEMBER">
  <FunctionParameter
    OntologyName="OntoBasis"
    TheoryName="BASE"
    FunctionParameterType="THING"/>
  <FunctionParameter
    FunctionParameterType="LIST"/>
  <FunctionReturnValue
    OntologyName="OntoBasis"
    TheoryName="LOGICALS"
    FunctionReturnValueType="BOOLEAN"/>
  <FunctionBody LanguageType="LISP">
    <![CDATA[
      DEFPURE LISTS.MEMBER (X Y)
      (DOLIST (YY Y (LOGICALS.FALSE)) (WHEN
        (IS-LOGICAL-TRUE (LOGICALS.EQUAL X YY))
        (RETURN (LOGICALS.TRUE)))))]]>
  </FunctionBody>
  ...
</FunctionDeclaration>
<IteratorDeclaration> ...
</IteratorDeclaration>
<RewriteRuleDeclaration>
<RewriteRule
  RewriteRuleType="DEFSIMPLIFIER"
  RewriteRuleName="MEMBER-CONS">
  ...
</RewriteRuleDeclaration>
</Theory>
Examples - Specification

```xml
<Model ModelName="WEBDELCIVERY">

<IncludeDeclaration> ...</IncludeDeclaration>
<IncludeTheory TheoryName="BOOK"/> ...</IncludeDeclaration>

<InputEventDeclaration>
<InputEvent InputEventName="DEL-BOOK">
<EventParameter>
  EventParameterType="LIST"/ > ...</EventParameter>

<StateRelationDeclaration>
<StateRelation>
  StateRelationName="BOOK-URL">
  <StateParameter>
    StateParameterType="STRING"/

  <StateValueParameter>
    StateValueParameterType="STRING"/ > ...</StateValueParameter>

<OutputEventDeclaration>
<OutputEvent OutputEventName="MAIL">
<EventParameter>
  EventParameterType="MESSAGE"/

  ...</EventParameter>

<LETSTAR> <Bind>
<Variable VariableName="TITLE"/> ...</LETSTAR>

<IF><FunctionRef FunctionRefName="A-STRING">. . .
<FunctionRef FunctionRefName="NTH">. . .
<Constant ConstantValue="3"/> . . .
<Variable VariableName="BOOKINFO" />

<STATELOOK><StateRelationLookRef>
  StateRelationLookRefName="VALID-TITLES"/ > ...
</STATELOOK>

<OUTPUTREF> <StateRelationLookRef>
  StateRelationLookRefName="BOOK-URL"/ > ...
</OUTPUTREF>
</IF>

<Variable VariableName="BOOKINFO"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/> <InputEventRef>
  InputEventRefName="DEL-BOOK"/>

<Variable VariableName="BOOKINFO"/>

<Variable VariableName="BOOKINFO"/>

<Variable VariableName="BOOKINFO"/> <StatementList>

<LETSTAR> <Bind>
<Variable VariableName="TITLE"/> ...</LETSTAR>

<IF><FunctionRef FunctionRefName="A-STRING">. . .
<FunctionRef FunctionRefName="NTH">. . .
<Constant ConstantValue="3"/> . . .
<Variable VariableName="BOOKINFO" />

<STATELOOK><StateRelationLookRef>
  StateRelationLookRefName="VALID-TITLES"/ > ...
</STATELOOK>

<OUTPUTREF> <StateRelationLookRef>
  StateRelationLookRefName="BOOK-URL"/ > ...
</OUTPUTREF>
</IF>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/>

<Variable VariableName="EMAIL"/> <InputEventRef>
  InputEventRefName="DEL-BOOK"/>

<Variable VariableName="BOOKINFO"/>

<Variable VariableName="BOOKINFO"/>

<Variable VariableName="BOOKINFO"/> <StatementList>

<LETSTAR> <Bind>
<Variable VariableName="TITLE"/> ...</LETSTAR>

<IF><FunctionRef FunctionRefName="A-STRING">. . .
<FunctionRef FunctionRefName="NTH">. . .
<Constant ConstantValue="3"/> . . .
<Variable VariableName="BOOKINFO" />

<STATELOOK><StateRelationLookRef>
  StateRelationLookRefName="VALID-TITLES"/ > ...
</STATELOOK>

<OUTPUTREF> <StateRelationLookRef>
  StateRelationLookRefName="BOOK-URL"/ > ...
</OUTPUTREF>
</IF>
```

© Hall & Zisman, 2004
OMMLTranslator

• OMML Translator (OMT) maps models in different languages to/from OMML
  – OMML to P-EBF
  – P-EBF to OMML
    \{ \text{total translation} \}
  – SCR to OMML
  – OMML to SCR: partial translation (infinite state spaces and function rich theories)

• Status
  – OMT is currently implemented partially in Java and partially in LISP (ISAT)
  – Goal: standalone Java-based tool
OpenModel Overview
OpenModel Project

- OpenModel Modeling Language - **OMML**
  - each node of an MSDS publishes behavioural model of itself based on shared domain specific ontologies at a certain level of abstraction

- OpenModel Validator - **OMV**
  - incremental generation of behavior trees (GST)
  - dynamic deduction of MSDS nodes relevant to the task/requirements of interest
  - tolerance of partial information by using symbolic simulation

- OpenModel Distribution Infrastructure - **OMDI**
  - protocol/API for model retrieval (planned)
OpenModel Validation (GSTView)

Graphical interactive tool built on top of OMV to support validation of personal requirements via assisted symbolic behavior browsing methodology

– visualisation and inspection of symbolic behavior
  • graphical representation of message passing
  • detailed examination of events, state changes

– automated reasoning support for
  • exact property checkers
  • approximate property checkers \{ ReqMons
  • noticers of situations \}
BookFind Example
# BookFind Example

<table>
<thead>
<tr>
<th></th>
<th>BS1</th>
<th>BS2</th>
<th>BS3</th>
<th>BS-S</th>
<th>DS1</th>
<th>DS2</th>
<th>DS3</th>
<th>DS-S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web delivery</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Spam messages (all)</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Spam messages (related)</td>
<td>-</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hold personal info</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hold cc info</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Overnight delivery</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>no</td>
<td>yes (EU)</td>
<td>no</td>
</tr>
<tr>
<td>P.O. Box address</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Signature required</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

© Hall & Zisman, 2004
Case Study Steps

- BookFind servers modeled as OMML, P-EBF, and SCR
- SCR and OMML models were translated into P-EBF; P-EBF models were translated into OMML and re-translated into P-EBF
- Eight scenarios were simulated, visualised and validated using GSTView

Demo will be presented soon
## OMML Translation - Results

<table>
<thead>
<tr>
<th>Documents</th>
<th>P-EBF size</th>
<th>SCR size</th>
<th>OMML size</th>
<th>Into OMML time</th>
<th>OMML-PEBF time</th>
</tr>
</thead>
<tbody>
<tr>
<td>BookFinder.spec</td>
<td>9.11</td>
<td>-</td>
<td>38.6</td>
<td>3.7</td>
<td>&lt;1.0</td>
</tr>
<tr>
<td>BookService1.spec</td>
<td>8.81</td>
<td>-</td>
<td>39.9</td>
<td>3.3</td>
<td>&lt;1.0</td>
</tr>
<tr>
<td>BookServiceS.scr</td>
<td>27.4</td>
<td>11.0</td>
<td>121</td>
<td>30.4</td>
<td>&lt;1.0</td>
</tr>
<tr>
<td>WebDelService.spec</td>
<td>1.64</td>
<td>-</td>
<td>8.62</td>
<td>&lt;1.0</td>
<td>&lt;1.0</td>
</tr>
<tr>
<td>DeliveryService1.spec</td>
<td>2.22</td>
<td>-</td>
<td>10.2</td>
<td>1.0</td>
<td>&lt;1.0</td>
</tr>
<tr>
<td>DeliveryServiceS.scr</td>
<td>4.57</td>
<td>1.55</td>
<td>23.0</td>
<td>2.07</td>
<td>&lt;1.0</td>
</tr>
<tr>
<td>Lists.theory</td>
<td>5.24</td>
<td>-</td>
<td>17.9</td>
<td>2.0</td>
<td>&lt;1.0</td>
</tr>
<tr>
<td>DeliveryConfirm.conn</td>
<td>0.44</td>
<td>-</td>
<td>0.91</td>
<td>&lt;1.0</td>
<td>&lt;1.0</td>
</tr>
</tbody>
</table>

OMML = 4.5*P-EBF
Requirements

- **R1**: service node will send spam email to customers
- **R2**: unacceptably high delivery cost
- **R3**: vendor retains customer information after transaction
- **R4**: signature is required on delivery
- **R5**: undesirable web delivery (hard-copy only)
- **R6**: urgent delivery requested to p.o.box address

Input Constraints

- **I1**: USA address, or outside
- **I2**: p.o.box address, or not
- **I3**: urgent delivery, or not

* approximate
Requirements

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Nodes</th>
<th>Violations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1, !I2, !I3</td>
<td>65</td>
<td>R1, R3–R5</td>
<td>0.7</td>
</tr>
<tr>
<td>I1, I2, !I3</td>
<td>63</td>
<td>R1, R3, R5</td>
<td>0.7</td>
</tr>
<tr>
<td>!I1, !I2, !I3</td>
<td>65</td>
<td>R1–R5</td>
<td>2.2</td>
</tr>
<tr>
<td>!I1, I2, !I3</td>
<td>63</td>
<td>R1, R3, R5</td>
<td>2.1</td>
</tr>
<tr>
<td>I1, !I2, I3</td>
<td>76</td>
<td>R1, R3–R5</td>
<td>2.2</td>
</tr>
<tr>
<td>I1, I2, I3</td>
<td>94</td>
<td>R1, R3, R5, R6</td>
<td>3.9</td>
</tr>
<tr>
<td>!I1, !I2, I3</td>
<td>76</td>
<td>R1–R5</td>
<td>2.7</td>
</tr>
<tr>
<td>!I1, I2, I3</td>
<td>94</td>
<td>R1, R3, R5, R6</td>
<td>3.6</td>
</tr>
<tr>
<td>Indefinite</td>
<td>144</td>
<td>R1–R6</td>
<td>5.6</td>
</tr>
</tbody>
</table>

I1: USA address, or outside
I2: p.o.box address, or not
I3: urgent delivery, or not

* approximate
Conclusion and Future Work

• OMML supports integration of heterogeneous models, allowing validation of realistic web services

• Future Work:
  – extending OMT to allow translations between OMML and other languages (e.g. Statecharts, Action Language)
  – investigating size reduction of OMML documents
  – implementing OMDI using UDDI

www.research.att.com/~hall/openmodel-project.html