Composition of Web Services on Large Service Models

Riina Maigre

Institute of Cybernetics at Tallinn University of Technology

17 May 2011
Outline

Introduction

Service Model

Logic for Composition on Service Models

Experiments
  Large Service models
  Hierarchical Service Models

Results and Future Work
Introduction

- Web services – software components, accessible over the web using the machine processable interface descriptions.
- Syntactic web service descriptions: WSDL.
- Semantic web service descriptions: OWL-S, SAWSDL, WSDL-S, WSML.
- Message exchange: RPC, SOAP, REST.
Automated Web Service Composition

- The goal of web service composition is to construct new web services from existing web services.
- Compound web services can also contain control constructs, such as, cycle and condition.
Why is Composition Complicated?

- Domain can be very knowledge intensive – domain expert is needed.
- The set of web services to choose from is large – automation is needed.
- Web service providers use different ontologies.
Current Solutions

- Automated web service mapping into service template.
- Automated workflow generation from a given goal.
- Combined approaches.
Automated Web Service Mapping into Service Template

▸ Entire workflow has to be created manually.
▸ Visual tools are often used to create workflows.
▸ Templates are created for suitable component services to map web services automatically into compound web service workflow.
▸ If suitable web services do not exist, then service template needs to be redesigned manually.
▸ Examples: eFlow, METEOR-S framework, Zeng et al., WSMO Studio.
Automated Workflow Generation

- Aim is to fully automate the web service composition.
- Easier with atomic services and sequential workflows.
- Synthesis of control constructs or composition of web services that are themselves compound web services needs a very detailed and complex goal specification language and a very detailed goal from the user.
- Examples: Sword; Rao, Küngas and Matskin; Haav et al.; Kona et al.; ASTRO toolset; QSynth.
Combined Approaches

- JOpera – Visual composition of workflow, data flow creation can be automated by mapping inputs and outputs.
- Web Service Composer – the user starts the composition by selecting the last web service. Web services that provide input are automatically proposed.
- Synthy – automated web service matching into user created workflow. Automated service finding and control flow creation if there are no exact matches.
Proposed Solution

- **Service model** – can be used to represent available web services, control constructs and concepts from ontology.
- Service model has a visual representation.
- Service model can be described in logic that allows to automatically synthesise new compound web services.
Service Model

- Service model is a description of a collection of interoperable services that includes information necessary for automatic composition of compound services and uses one ontology.

- A service model is abstractly represented as a bipartite graph with two sets of nodes $R$ and $V$. The set $R$ is a set of services and data dependency relations that can also be represented by atomic services. The set $V$ is a set of variables representing data that can be inputs and outputs of services, and logical variables that are pre- and postconditions of services. Elements of $V$ have names from the ontology used.
Example Service Model
Services included into workflows: r1, r2(r4); r6, r9; r7, r8.
Higher-order component:

\[(V4 \supset V5) \land V2 \supset V3\{r2\};\]

Selector component:

\[V7 \supset V8 \land V9 \land V10\{select\}.\]
Synthesis of Compound Services

- Service model
- Goal (pre- and postconditions of a web service to be composed)
- Formulas in intuitionistic logic
- Proof
- Synthesised structure of a compound web service
- Process language (e.g., BPEL, OWL-S)
Rules of structural synthesis on programs:

\[
\frac{A \supset B \land C\{f\} \quad B \land D \supset G\{g\}}{A \land D \supset C \land G\{f; g\}} \quad \text{(SEQ)}
\]

\[
\frac{(A \supset B) \land X \supset Z\{f\} \quad A \land W \supset B\{g\}}{X \land W \supset Z\{f(g)\}} \quad \text{(HOW)}
\]
Composition on Service Models – Example

Let us have the following services:

\[ B \supset C\{\text{service1}\}; \]
\[ C \supset D\{\text{service2}\}; \]
\[ E \supset F\{\text{service3}\}; \]
\[ A \land (B \supset D) \supset E\{\text{cycle}(\phi)\}; \]

and a goal: \( A \supset F \).

Dervivation of the proof:

\[
\begin{align*}
B \supset C\{\text{service1}\} & \quad C \supset D\{\text{service2}\} \\
A \land (B \supset D) & \supset E\{\text{cycle}(\phi)\} & \quad B \supset D\{\text{service1}; \text{service2}\} \\
\hline
A \supset E\{\text{cycle}(\text{service1}; \text{service2})\} & \quad \text{SEQ} & \quad \text{HOW} & \quad E \supset F\{\text{service3}\} \\
A \supset F\{\text{cycle}(\text{service1}; \text{service2}); \text{service3}\} & \quad \text{SEQ}
\end{align*}
\]
Web Service Composition Tool

- Proof of concept tool has been implemented on CoCoViLa model-based software development platform.
- CoCoViLa supports planning based on structural synthesis of programs and is able to handle specifications in the visual or textual form.
Estonian e-government Web Services (X-Road)
This model contained about 900 components.
Originally created by Peep Küngas in 2006.
Synthesis on this large model showed the feasibility of the proposed web service composition method.
The service model was hard to use because of its size.
Hierarchical Service Models (1)

Layer 1:

Service descriptions 1

Service descriptions 2

Service descriptions n

Layer 2:

Service model 1

Service model 2

Service model n

Layer 3:

Hierarchical view 1

Hierarchical view n

...
Hierarchical Service Models (2)

- Generated automatically from the (SAWSDL) descriptions of the X-Road web services.
- Service models are generated for each provider (registry) and saved as components to be used on the hierachical model.
- Different ontologies can be used to represent:
  - submodels
  - upper level model (hierarchical view)
Population Registry’s Service Model
Hierarchical Service Model

ListOfDocumentNumbers.data -> ListOfAddresses.data
As a result the following three services are cycled trough:

- arireg:toimiku_dokument \((\text{BusinessReg})\)
- rr:RR405IsikNimi \((\text{PopulationReg})\)
- rr:RR57 \((\text{PopulationReg})\)
Results

▶ A logic-based web service composition method suitable for automating web service composition in the knowledge intensive domain with a large number of services was proposed in this work.

▶ The service model concept was proposed as a way to represent available web services, control constructs and data dependency relations.

▶ A prototype tool, based on the method proposed, was implemented on the CoCoViLa model-based software development platform and tested on web service descriptions of the Estonian e-government information system.
Future Work

➡️ Further automation of service model generation:
   ➡️ complex data types,
   ➡️ higher-order components.

➡️ Extensions to service model:
   ➡️ quality of service information,
   ➡️ confidentiality and access rights.
Publications


Thank you!