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
- Hierachical 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 ofthen 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.

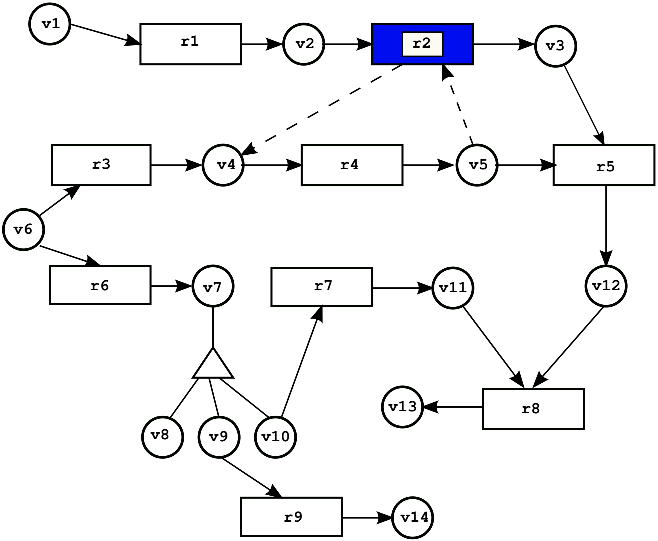
Proposed Solution

- *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.

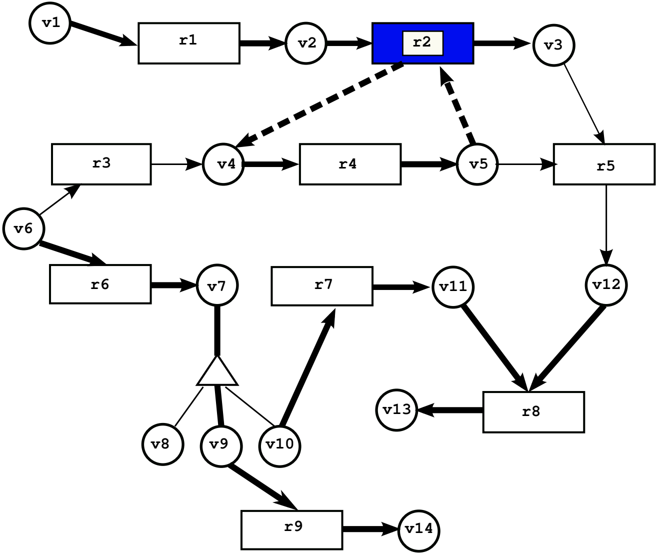
Service Model

- *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

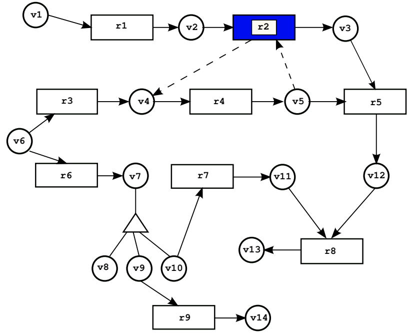


(Higher-Order) Workflows on Service Model



Services included into workflows: r1, r2(r4); r6, r9; r7, r8.

Logic for Representing Service Models



$V1 \supset V2\{r1\};$

$V3 \wedge V5 \supset V12\{r5\};$

$V6 \supset V4\{r3\};$

$V4 \supset V5\{r4\};$

$V6 \supset V7\{r6\};$

$V10 \supset V11\{r7\};$

$V9 \supset V14\{r9\};$

$V11 \wedge V12 \supset V13\{r8\};$

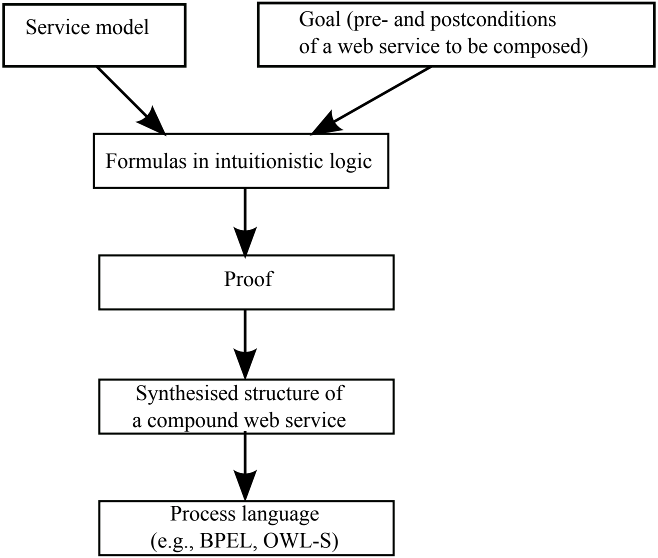
Higher-order component:

$(V4 \supset V5) \wedge V2 \supset V3\{r2\};$

Selector component:

$V7 \supset V8 \wedge V9 \wedge V10\{select\}.$

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)

Composition on Service Models

Rules of structural synthesis on programs:

$$\frac{A \supset B \wedge C\{f\} \quad B \wedge D \supset G\{g\}}{A \wedge D \supset C \wedge G\{f;g\}} \qquad \text{(SEQ)}$$

$$\frac{(A \supset B) \wedge X \supset Z\{f\} \quad A \wedge W \supset B\{g\}}{X \wedge W \supset Z\{f(g)\}} \qquad \text{(HOW)}$$

Composition on Service Models – Example

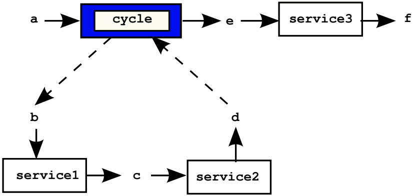
Let us have the following services:

$$B \supset C\{service1\};$$

$$C \supset D\{service2\};$$

$$E \supset F\{service3\};$$

$$A \wedge (B \supset D) \supset E\{cycle(\phi)\};$$



and a goal: $A \supset F$.

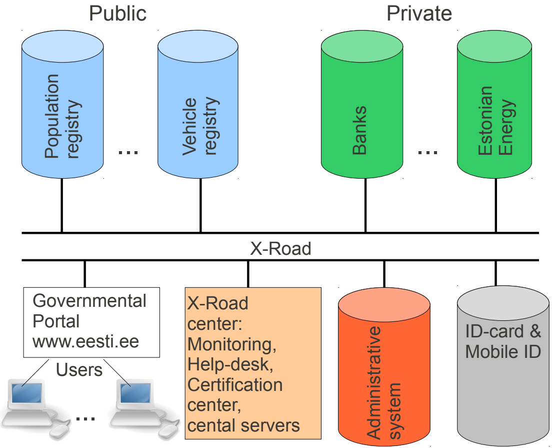
Dervivation of the proof:

$$\dfrac{\dfrac{A \wedge (B \supset D) \supset E\{cycle(\phi)\} \qquad \dfrac{B \supset C\{service1\} \quad C \supset D\{service2\}}{B \supset D\{service1; service2\}}\,SEQ}{A \supset E\{cycle(service1; service2)\}}\,HOW \qquad E \supset F\{service3\}}{A \supset F\{cycle(service1; service2); service3\}}\,SEQ$$

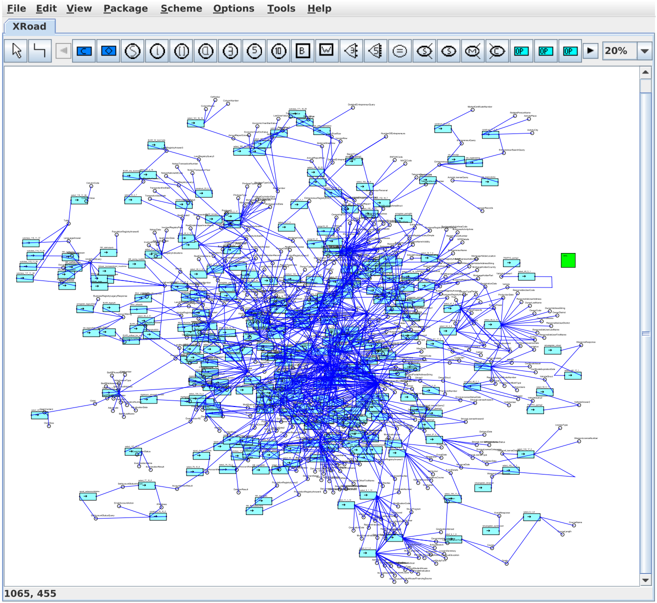
Web Service Composition Tool

- 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)



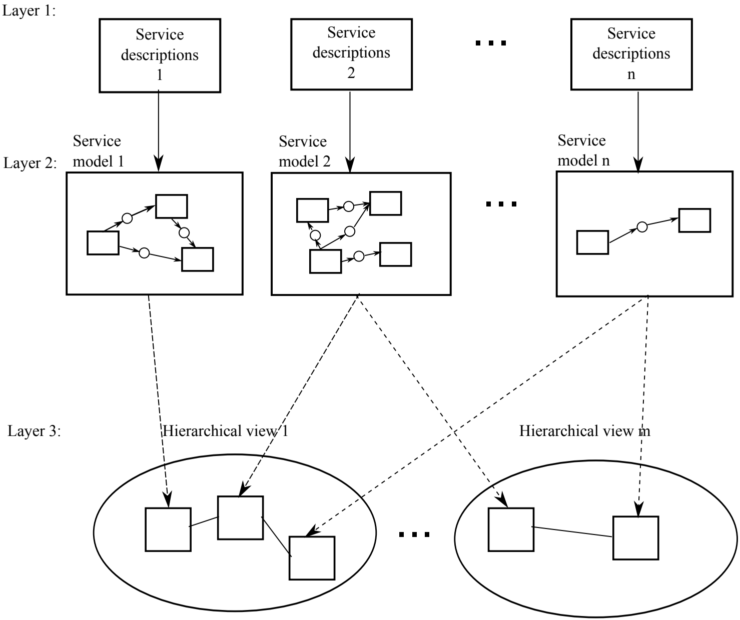
Large X-Road Service Model (1)



Large Service Model (2)

- This model contained about 900 components.
- Originally created by Peep Küngas in 2006.
- Synthesis on this large model showed the feasiblity of the proposed web service composition method.
- The service model was hard to use because of its size.

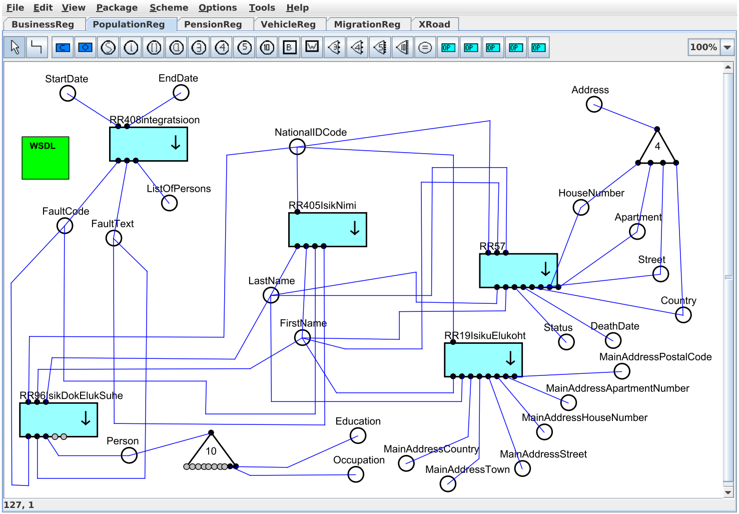
Hierarchical Service Models (1)



Hierarchical Service Models (2)

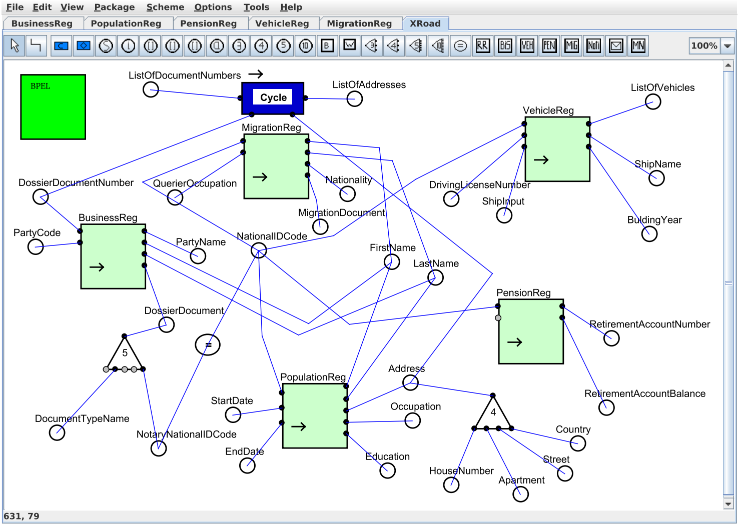
- Generated automatically from the (SAWSDL) descriptions of the X-Road web services.
- Service models are generated for each provider (registry) and saved as components to be used on the hierachical model.
- Different ontologies can be used to represent:
  - submodels
  - upper level model (hierachical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

Population Registry's Service Model



Hierarchical Service Model

ListOfDocumentNumbers.data->ListOfAddresses.data



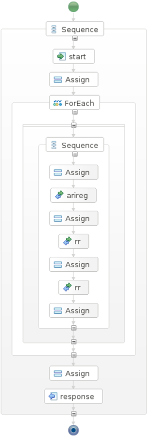
Output

As a result the following three services are cycled trough:

`arireg:toimiku_dokument` (**BusinessReg**)

`rr:RR405IsikNimi` (**PopulationReg**)

`rr:RR57` (**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 desciptions of the Estonian e-government information system.

Future Work

- Futher automation of service model generation:
  - complex data types,
  - higher-order components.
- Extentsions to service model:
  - quality of service information,
  - confidentiality and access rights.

Publications

- Riina Maigre, Enn Tyugu (2011). Composition of Services on Hierarchical Service Models. In: EJC 2011 : 21st European-Japanese Conference on Information Modelling and Knowledge Bases. [in print]
- Riina Maigre (2010). Survey of the Tools for Automating Service Composition. In: ICWS 2010 : 2010 IEEE Eighth International Conference on Web Services : Miami, Florida, 5-10 July 2010: IEEE Computer Society, 2010, 628 - 629.
- Riina Maigre, Peep Küngas, Mihhail Matskin, Enn Tyugu (2009). Dynamic Service Synthesis on a Large Service Models of a Federated Governmental Information System. International Journal on Advances in Intelligent Systems, 2(1), 181 - 191.
- Riina Maigre, Pavel Grigorenko, Peep Küngas, Enn Tyugu (2008). Stratified Composition of Web Services. In: Knowledge-based software engineering : Proceedings of the Eighth Joint Conference on Knowledge-Based Software Engineering: (Eds.) Virvou, Maria; Nakamura, Taichi. Amsterdam: IOS Press, 2008, (Frontiers in Artificial Intelligence and Applications; 180), 49 - 58.

- Riina Maigre, Pavel Grigorenko, Peep Küngas, Enn Tyugu (2008). Stratified Composition of Web Services. In: Knowledge-based software engineering : Proceedings of the Eighth Joint Conference on Knowledge-Based Software Engineering: (Eds.) Virvou, Maria; Nakamura, Taichi. Amsterdam: IOS Press, 2008, (Frontiers in Artificial Intelligence and Applications; 180), 49 - 58.

- Riina Maigre, Peep Küngas, Mihhail Matskin, Enn Tyugu (2008). Handling Large Web services models in a Federated Governmental Information System. In: The Third International Conference on Internet and Web Applications and Services, ICIW 2008 : 8-13 June 2008, Athens, Greece, proceedings: (Eds.) Mellouk, A.; Bi, J.; Ortiz, G. et al.. Los Alamitos: IEEE Computer Society, 2008, 626 - 631.
- Mihhail Matskin, Riina Maigre, Enn Tyugu (2007). Compositional Logical Semantics for Business Process Languages. In: Second International Conference on Internet and Web applications and services ICIW 2007, May 13-19, 2007, Morne, Mauritius: Los Alamitos, CA: IEEE Computer Society, 2007, 6 p.

Thank you!

Euroopa Liit
Euroopa Sotsiaalfond

Eesti tuleviku heaks

tiigriülikool