GSoaArchitect tutorial

Dániel Darvas, Gergő Horányi, Balázs Pál, Gábor Szárnyas

June 14, 2012

Contents

1 Introduction

The following document is the documentation for the project work done during the courses *Model Driven Software Development* and *System Integration*, held at the Budapest University of Technology of Economics by the Fault Tolerant Systems Research Group[1].

2 Usage

We present a possible use case scenario for our modeling and code generator framework. The example features a simple phonebook application that can store name-phone number pairs.

2.1 Modeling and validation

2.1.1 Prerequisites

- Eclipse Modeling 3.7 or newer – `http://www.eclipse.org/`
- IncQuery – `http://viatra.inf.mit.bme.hu/incquery/`

[1] `https://www.inf.mit.bme.hu/edu/courses/mdsd`

- An SVN client, e.g. Subversive SVN Team Provider with the SVN Kit Connector.

2.1.2 Steps

1. Checkout the code from our public Subversion repository at `https://code.google.com/p/gsoaarchitect/`.
2. Import the projects to Eclipse if necessary.
3. In the `hu.bme.mit.inf.gs.dsl` project open the `model/SoaModel.genmodel` file. Press right click on the root element of the SoaModel tree and click *Generate All*.
4. Run the `hu.bme.mit.inf.gs.dsl.validators.validation` project as an *Eclipse application*.
5. Create a new *Soamodel model* or open an existing SOA model file with the *Sample Reflective Ecore Model Editor*.
6. Press right click on the root element of the model, select *EMF-IncQuery / Initialize EMF-IncQuery Validators on Editor*.
7. Construct the model. There are a lot of element that you can use to construct your Service Oriented Architecture model, such as components (both for .NET and Java EE platforms), methods, method parameters, entities (for Entity Framework and JPA), attributes, etc. For more details, please see the Soamodel metamodel:
   `model/SoaModel.ecore` in the `hu.bme.mit.inf.gs.dsl` project.
8. Open Problems view: Window / Show view / Problems. If there are any problems with your model, you can see them in the Problems view. For example: duplicate names, duplicate IDs, inconsistent types etc.

2.1.3 Example model

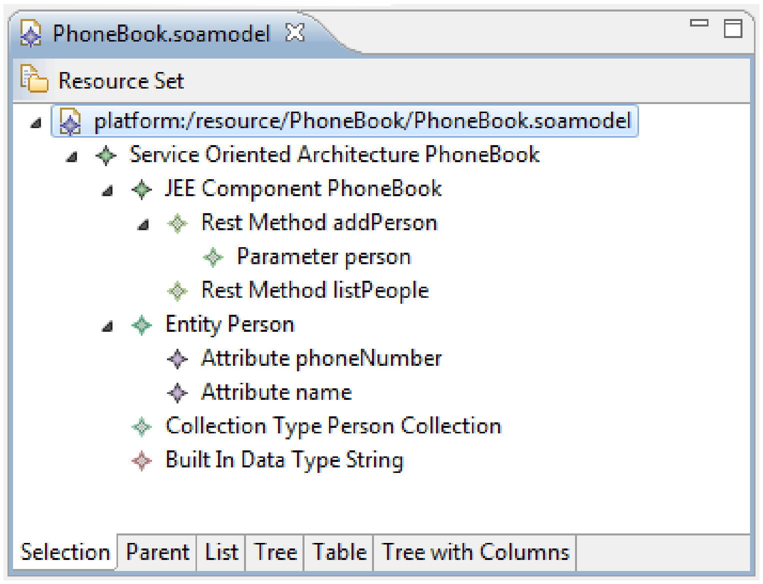
This section shows the sample model of the phonebook architecture example. In that model (Figure 1) there is a service component called PhoneBook. It has two REST methods: `addPerson` for adding a new person and `listPeople` for listing the stored people. There is a defined `Person` entity which has two attributes: `phoneNumber` and `name`. Also there is a technical collection type which represents some people and a built-in data type for representing the String type.

Listing 1 shows the full XML source of the model.

If there are any problems with your model, you can see them in the Problems view. Figure 2 shows an example where the model has problems (there are two components with the same name).

2.2 Code generation

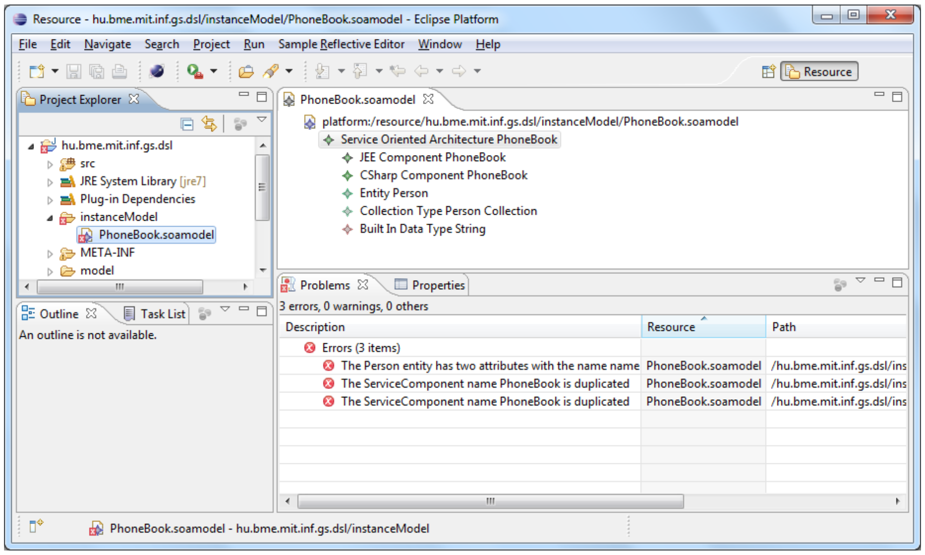
The code generator creates source code for the entites and the interfaces and class skeletons for the specified components.


PhoneBook.soamodel

Resource Set

```
platform:/resource/PhoneBook/PhoneBook.soamodel
  Service Oriented Architecture PhoneBook
    JEE Component PhoneBook
      Rest Method addPerson
        Parameter person
      Rest Method listPeople
    Entity Person
      Attribute phoneNumber
      Attribute name
    Collection Type Person Collection
    Built In Data Type String
```

Figure 1: Example model in the generated editor application


Resource - hu.bme.mit.inf.gs.dsl/instanceModel/PhoneBook.soamodel - Eclipse Platform

```
hu.bme.mit.inf.gs.dsl
  src
  JRE System Library [jre7]
  Plug-in Dependencies
  instanceModel
    PhoneBook.soamodel
  META-INF
  model
```

An outline is not available.

PhoneBook.soamodel

```
platform:/resource/hu.bme.mit.inf.gs.dsl/instanceModel/PhoneBook.soamodel
  Service Oriented Architecture PhoneBook
    JEE Component PhoneBook
    CSharp Component PhoneBook
    Entity Person
    Collection Type Person Collection
    Built In Data Type String
```

3 errors, 0 warnings, 0 others

Description	Resource	Path
Errors (3 items)		
The Person entity has two attributes with the name name	PhoneBook.soamodel	/hu.bme.mit.inf.gs.dsl/ins
The ServiceComponent name PhoneBook is duplicated	PhoneBook.soamodel	/hu.bme.mit.inf.gs.dsl/ins
The ServiceComponent name PhoneBook is duplicated	PhoneBook.soamodel	/hu.bme.mit.inf.gs.dsl/ins

PhoneBook.soamodel - hu.bme.mit.inf.gs.dsl/instanceModel

Figure 2: The generated editor with built-in validation

Listing 1: The XML source of the example model

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soamodel:ServiceOrientedArchitecture xmi:version="2.0"
  xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi=
  "http://www.w3.org/2001/XMLSchema-instance"
  xmlns:soamodel="http://soamodel/1.0" name="PhoneBook">
  <components xsi:type="soamodel:JEEComponent"
    name="PhoneBook" namespace="PhoneBook">
    <methods xsi:type="soamodel:RestMethod" name="addPerson"
      httpMethod="POST" path="add/{person}">
      <parameters parameterType="//@dataTypes.0"
      name="person"/>
    </methods>
    <methods xsi:type="soamodel:RestMethod"
      methodType="//@dataTypes.1" name="listPeople"
      path="list"/>
  </components>
  <dataTypes xsi:type="soamodel:Entity" name="Person"
    ID="//@dataTypes.0/@attributes.1">
    <attributes attributeType="//@dataTypes.2"
    name="phoneNumber"/>
    <attributes attributeType="//@dataTypes.2"
    name="name"/>
  </dataTypes>
  <dataTypes xsi:type="soamodel:CollectionType"
    name="Person Collection" itemType="//@dataTypes.0"/>
  <dataTypes xsi:type="soamodel:BuiltInDataType"
    name="String"/>
</soamodel:ServiceOrientedArchitecture>
```

2.2.1 Prerequisites

- Eclipse Modeling 3.7 or newer – `http://www.eclipse.org/`
- The following components from `http://download.eclipse.org/modeling/tmf/xtext/updates/composite/releases/`:
  - EMFT MWE-2.2.0
  - EMFT MWE2 Runtime-2.2.0
  - EMFT MWE2 Language-2.2.0
  - Xtend2-2.2.1
  - Xtext-2.2.1
  - M2T Xpand 1.0 / Xtend XSD typesystem

Please make sure you use separate Eclipse installations as IncQuery needs a newer version of Xtext / Xtend.

1. Change the value of `xmlns:soamodel` to `platform:/resource/hu.bme.mit.inf.gs.dsl/model/SoaModel.ecore`.
2. You have to modify the `xmlns:soamodel` value in the root element of the instance model. This value must be the full resource path of the metamodel (called `SoaModel.ecore`) of the instance model. For example you have to change this value from `http://soamodel/1.0` to `platform:/resource/hu.bme.mit.inf.gs.dsl/model/SoaModel.ecore`.
3. Open the `src/workflow/generator_soa.mwe` from `hu.bme.mit.inf.gs.codegen` project.
4. Modify the `model` property value to the previously created model's path. For example change to `<property name="model" value="hu.bme.mit.inf.gs.dsl/instanceModel/PhoneBook.soamodel" />`
5. Press right click on the `src/workflow/generator_soa.mwe` file and choose *Run as / MWE workflow*.
6. The generated code will be placed in the `hu.bme.mit.inf.gs.codegen/src-gen` folder.

2.2.2 Example

This section shows some examples of the generated source code from the model defined in Section 2.1.3.

There are different types of generated codes.

- *entities*, such as the PersonEntity.java (see Listing 2) which describes a Person entity

- *interfaces*, such as IPhoneBookComponent.java (see Listing 3) which describes the interface of the PhoneBook component (service)
- *implementation skeletons* (dummy implementations for interfaces)
- other descriptors, configuration items, etc.

Listing 2: PersonEntity.java

```java
// This entity is for Java.
package hu.bme.mit.inf.gs.AppStore.PhoneBook.model;
import javax.persistence.*;
import javax.xml.bind.annotation.XmlElement;
import javax.xml.bind.annotation.XmlRootElement;

@Entity
@XmlRootElement
public class PersonEntity {

  private java.lang.String phoneNumber;

  @Id
  private java.lang.String name;

  private void name;

  @XmlElement
  public java.lang.String getPhoneNumber() {
    return phoneNumber;
  }

  public void setPhoneNumber(java.lang.String arg) {
    phoneNumber = arg;
  }

  @XmlElement
  public java.lang.String getName() {
    return name;
  }

  public void setName(java.lang.String arg) {
    name = arg;
  }

  @XmlElement
  public void getName() {
    return name;
  }

  public void setName(void arg) {
    name = arg;
  }

}
```

Listing 3: IPhoneBookComponent.java

```java
// Generated file
// for PhoneBook component.
// This component is for JEE.

package hu.bme.mit.inf.gs.AppStore.PhoneBook;

import hu.bme.mit.inf.gs.AppStore.PhoneBook.model.*;
import javax.ws.rs.*;

public interface IPhoneBookComponent {
  /**
   *
   *
   * @param PersonEntity person
   * @return String
   */
  @POST
  @Path("add/{person}")
  @Consumes("text/plain")
  @Produces("text/plain")
  String addPerson(@PathParam("person") PersonEntity person)
     ;
  /**
   *
   *
   * @return java.util.List<PersonEntity>
   */
  @GET
  @Path("list")
  @Consumes("text/plain")
  @Produces("text/plain")
  java.util.List<PersonEntity> listPeople();

}
```