

## Illustrative Example for Service Customization with the SHAPE Service Variability Tools

This document presents an example for creating a customized variant of a service with the tool-supported service variability modelling technique developed in the SHAPE project.

The following illustrates the service variability modelling techniques and tool-supported procedures for customizing an existing SAP Enterprise service for managing goods movement. We first explain the scenario setting, and then demonstrate the definition of a variability specification model as well as the creation of a service variant, using Service Variability Tools developed in SHAPE.

### Customizing the Goods Movement Enterprise Service

The example is based on the Enterprise Service "Goods Movement" that provides basic business facilities for managing the movement of goods and is publicly available via the SAP Developer Network (see [www.sdn.sap.com](http://www.sdn.sap.com)). It offers four operations: two for creating goods movement objects (one with references to related documents like purchase order, in- and outbound delivery, and one without), and one operation each for reading and updating goods movement objects. The message types are defined by business objects for goods movement along with standard objects like item, tax, transportation, and material.

In our example, we will create a variant of the service that only contains the operations and necessary message types for the simple creation and reading of goods movement objects. Figure 1 provides an overview with the original service on the left and the service variant on the right hand side, which obviously is simpler and thus easier to consume. Sufficient for demonstration purposes, we here work with already simplified data structures for the message types; the actual business objects used within SAP applications contain more than 100 nodes.

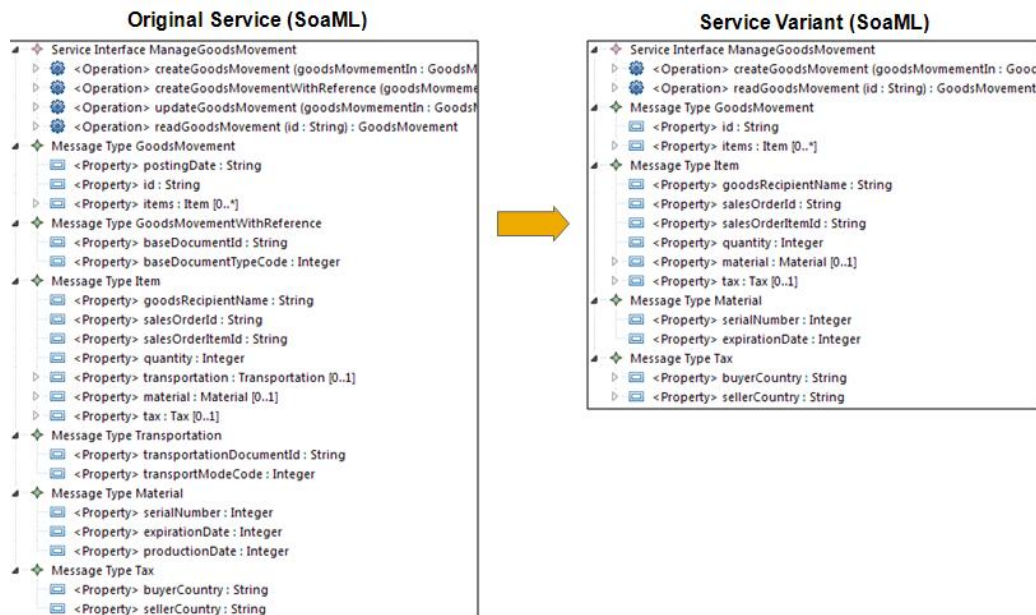


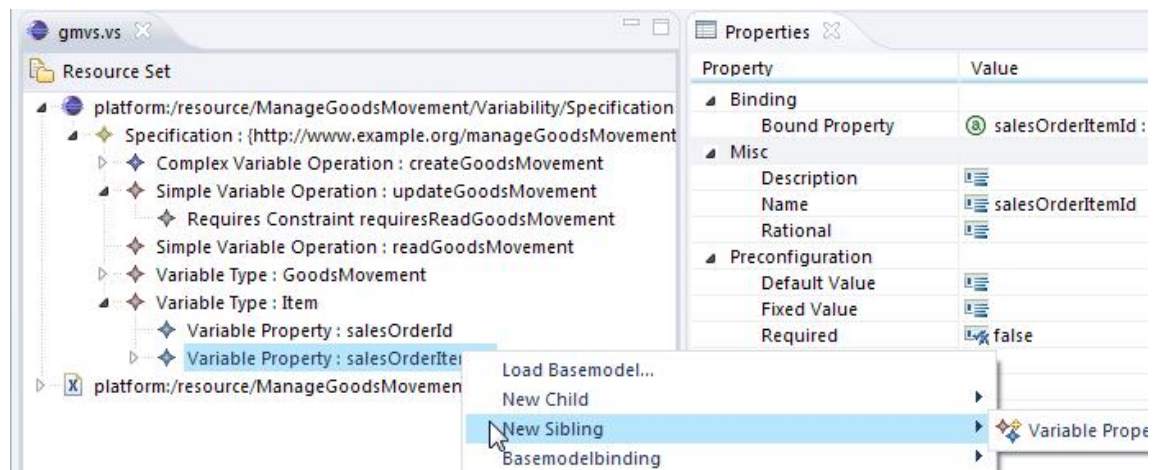
Figure 1: Goods Movement Service Scenario

## 1) Variability Specification Modelling

As explained above, the first step in the service customization process is the creation of the variability specification model.

Figure 2 shows a screenshot of our prototype tool for supporting domain experts in this task. The tool provides an editing facility for variability specification modeling with a tree-view representation and context-sensitive editing support for adding dependent variability modeling elements, constraints, and bindings to the base model. In order to minimize the manual editing effort, the build-in *extractor* feature creates a skeleton of the variability specification: this contains the variable elements for all elements of the base model; the domain expert then can add additional variability information, which is supported by the Variability Specification Editor.

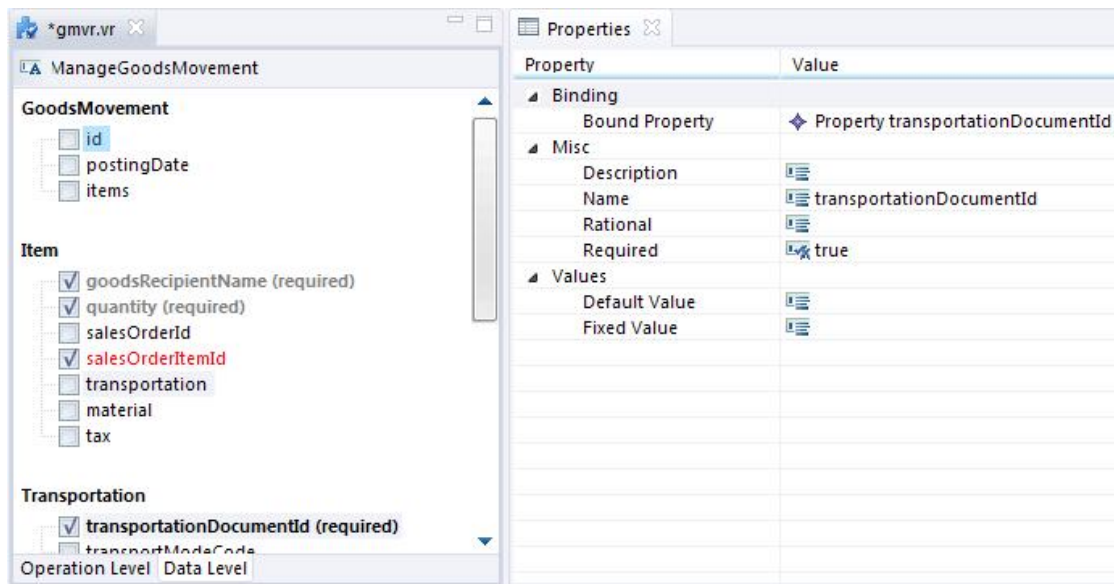
For our example, the domain expert defines the following explicit variability information. At first, the two operations for creating goods movement objects are grouped into a *ComplexVariableOperation*: conceptually, they present two versions of the same operation which differ in the input- and output parameters. However, they are non-exclusive, thus the *multiple*-property is set to *false*. Secondly, the *update*-operation requires the *read*-operation, which is modeled by a *RequiresConstraint*. Thirdly, the mandatory elements and their dependencies on the data level are defined. For instance, the usage of *salesOrderItem*, which is an attribute of the top-level type *item*, requires the provision of a value for the *salesOrderId* property. In addition, default values can be defined, e.g. setting country information to 'Germany' in order to pre-configure the service variant for German customers.



**Figure 2: Variability Specification Tool (Screenshot)**

## 2) Variability Resolution Modelling

We now turn to the creation of the actual service variant. As explained above, for this the consumer creates a resolution model by selecting the desired operations and data elements as well as defining default or fixed values for static data elements. From this a conventional service model is generated for the variant. In order to support consumers in this task, we provide a tool for selecting the desired features via a graphical user interface along with real-time validation of the dependencies and usage conditions defined in the variability specification model.



**Figure 3: Variability Resolution Tool (Screenshot)**

Figure 3 shows a screenshot of our prototype. It is organized by tabs that support variability resolution on different levels where variability can occur. Currently, the tool contains the operation and data level as the basic functional aspects covered by our metamodel; for the future, this can be extended with additional levels such as non-functional aspects or quality-of-service information.

The tool supports the variability resolution by standard metaphors for graphical user interface design: checkboxes for selecting the desired elements, graphical accentuation of required elements, and colored display of constraint violations. Detailed information, which provides useful guidance for correcting are accessible via the context menu.

### 3) Generation of Service Interface for Variant

The final step for creating a service variant is the creation of a service interface description for the consumer variant that has been defined within the variability resolution model. For this, the tool automatically generates the SoaML descriptions for the service variant automatically.

This feature is accessible by invoking the transformation from the SHAPE context menu of the respective variability resolution model as shown in Figure 4. The generated SoaML description of the service variant is stored folder 'Variants' of the EMF engineering project, and can be used for invoking the original service as well as within subsequent service modeling and engineering techniques.

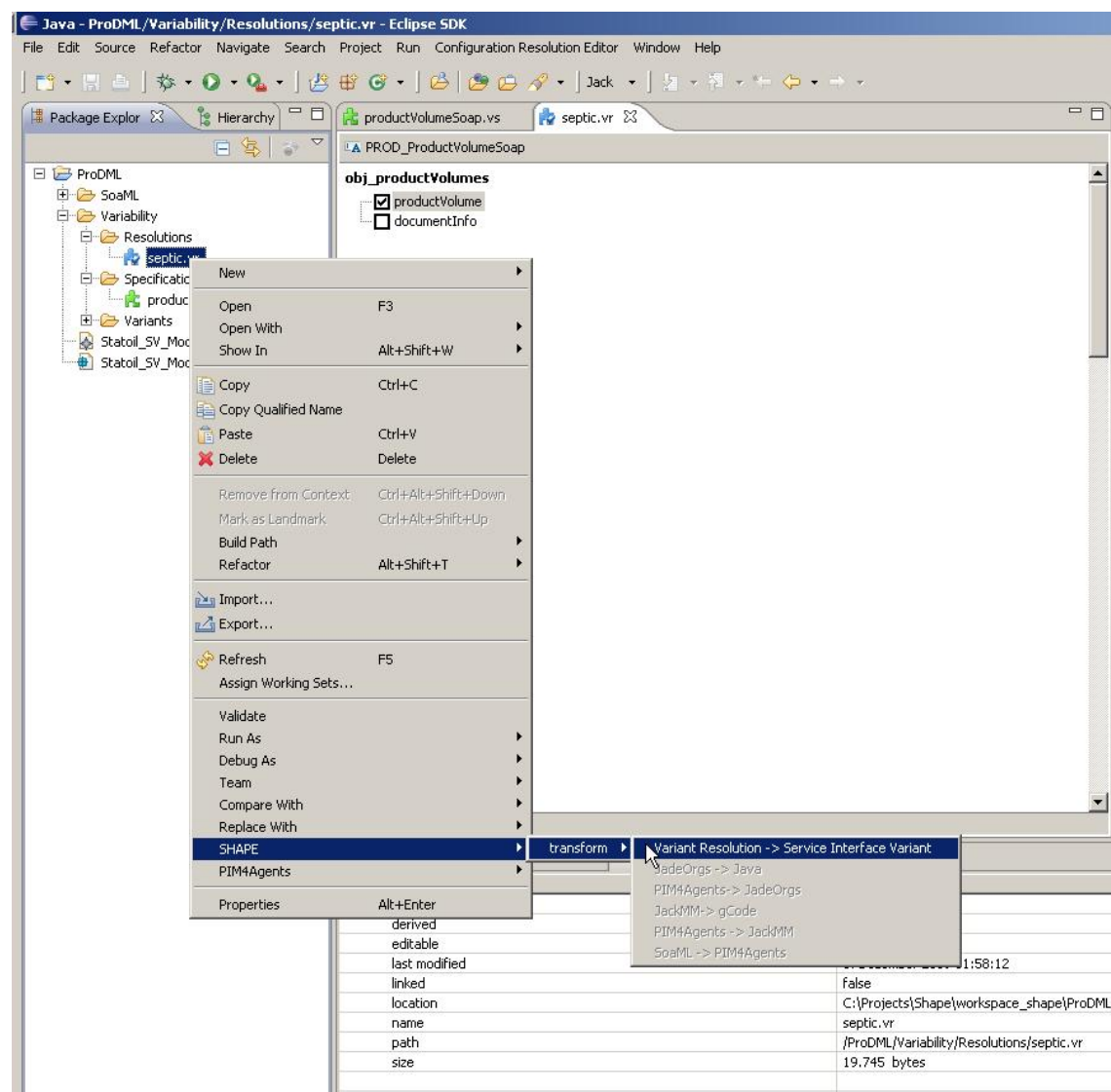


Figure 4: Generate a Service Variant