ArcStyler MDA-Business Transformer Tutorial

For ArcStyler Version 3.x

Interactive Objects

Copyright

Interactive Objects Software GmbH.
Basler Strasse 65
79100 Freiburg, Germany
http://www.io-software.com

Tel:+49 761 40073 0
Fax: +49 761 40073 73

Table of Contents

Before You Begin

This *ArcStyler MDA-Business Transformer Tutorial* describes the sample transformation of an ARIS *extended Event-Driven Process Chain* (eEPC). The model follows the *ArcStyler MDA-Business Transformer Modeling Style Guide for ARIS* and uses methods according to ARIS Best Practices to model assigned processes.

What Is in this Tutorial

The document includes the following chapters:

- Chapter 1, “From eEPCs to UML”
  This chapter describes how to map an ARIS eEPC to an UML State Machine Model. It describes the individual steps involved in the transformation, how to complete the imported model, and how to handle the restrictions applying to the different target platforms (JAVA2, WLI, or WEB).

- Chapter 2, “From UML to Java2 FSM”
  This chapter describes the projection for Finite State Machines (JAVA2).

- Chapter 3, “From UML to WLI”
  This chapter describes the projection for WLI (BEA WebLogic Integration2.1).

- Chapter 4, “From UML to Web Application Front-Ends”
  This chapter describes the projection for JSPs (JavaServer Pages) for WebAccessors.

Who Should Read this Tutorial

This tutorial focuses on the model-to-model transformation of ARIS extended Event-Driven Process Chains (eEPCs) to UML Models. It also treats the steps involved in the transformation of the initial UML Model to a completed model that can be used for code generation for different implementation technologies. Please refer to the appropriate user’s guides and tutorials for additional information on working with ARIS and the ArcStyler.

Related ArcStyler Documents

Other directly related ArcStyler documents:

- *ArcStyler Modeling Style and User's Guide*
  Comprises a style guide for the technical modeling of components and their generative projection to a particular executable system infrastructure. The focus is on modeling and generating Enterprise Java Beans. In addition, the guide shows you how to work with the generated sources and describes the build support provided by the ArcStyler.

- *ArcStyler MDA-Cartridge Guide For BEA Weblogic Integration Server 2.1*
  Describes the features of the ArcStyler MDA-Cartridge for the BEA WebLogic Integration Server 2.1. It explains the ArcStyler support for model-driven development of Enterprise Java Beans systems for the BEA WebLogic Server.

- *ArcStyler Accessor Guide*
  Covers the general design and usage features of the patented ArcStyler Accessor Framework. This framework supports the model-based, object-oriented development of external interfaces for software systems, including multi-channel B2X Internet interfaces. In addition, it provides a detailed description of the JSP/Servlet technology projection provided by the ArcStyler Accessor Cartridge, which supports automatic generation and deployment of JSP/Servlet-based Web applications from UML models.

- *ArcStyler MDA-Business Transformer Modeling Style Guide for ARIS*
  Provides a general introduction to the ArcStyler MDA-Business Transformer for ARIS and gives a detailed description of the appropriate MDA-compliant modeling style for ARIS models.

1

From eEPCs to UML

This chapter describes how to use the ArcStyler MDA-Business Transformer for ARIS to import an ARIS *extended Event-Driven Process Chain* (eEPC) into the ArcStyler C-REF UML modeling environment.

In this chapter, the following topics will be discussed:

- "Loading the Manage.Vacation_Request Sample"
- "Mapping the eEPC to a UML State Machine"
- "Completing the UML State Machine"
- "Restrictions for Mapping"

Loading the Manage.Vacation_Request Sample

Start ARIS 6.x and load the ARIS database **BusinessTransformer_Tutorial** by importing the file **BusinessTransformer_Tutorial.adb** from **%ARCSTYLER_HOME%/Samples/BTA** into the ARIS Explorer. This eEPC complies with the modeling style as described in the *ArcStyler MDA-Business Transformer Modeling Guide for ARIS*.

Manage.Vacation_Request, the main eEPC, describes the business process of issuing a request for vacation and the handling of this request in the application by the different roles, modeled as positions.

The function **Create.Vacation_Request** contains an assigned process which is modeled in the assignment **Request.Vacation**. The content of this eEPC is implemented as the **Client** part of this application, i.e. starting from this model, you generate the corresponding Web application that the **Employee** will use to create a request for vacation (for more information, refer to Chapter 4, "From UML to Web Application Front-Ends").

If the assigned process is terminated successfully, the main process will continue. The roles involved in this eEPC are **Employee**, **Project_Manager** and **Human_Resource_Manager**. These roles are modeled as positions in this eEPC. This information will be used in the server application to be generated for BEA WebLogic Integration2.1 (WLI) in Chapter 3, "From UML to WLI".

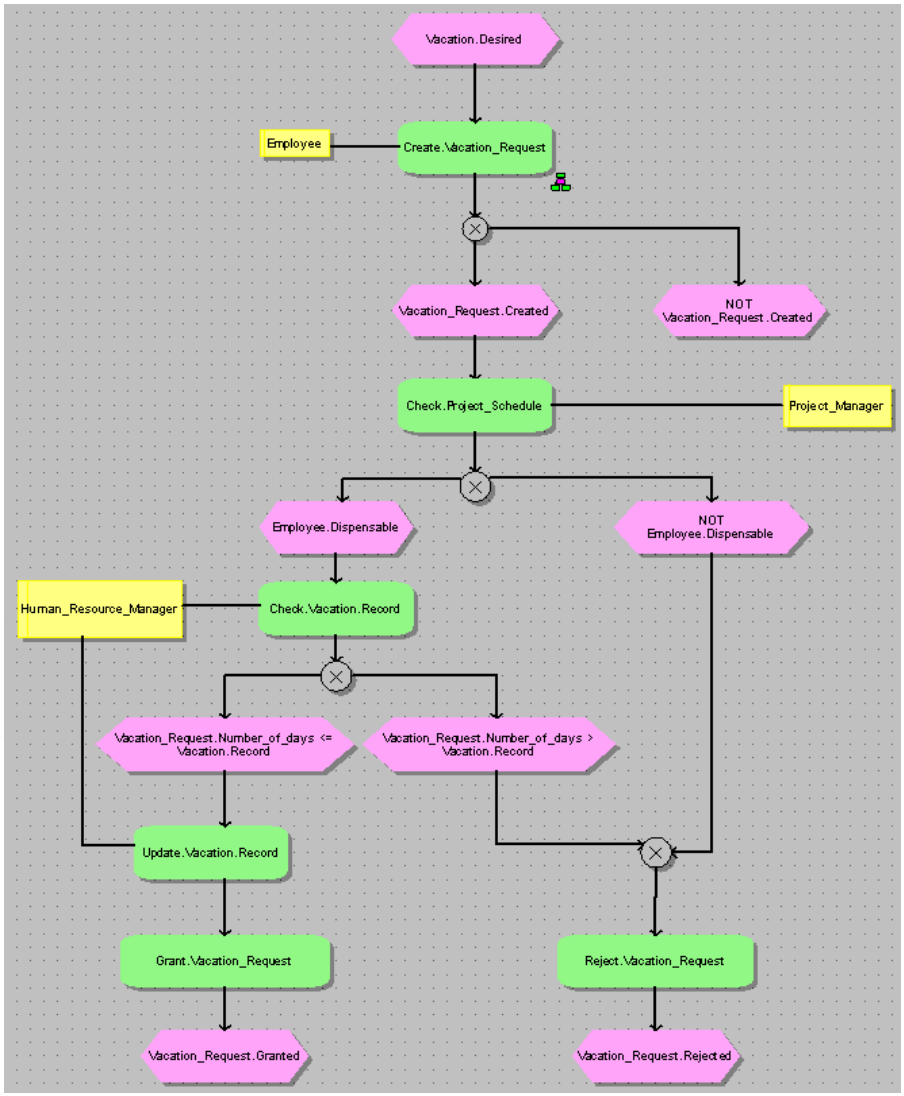

Figure 1: Manage.Vacation_Request [eEPC].

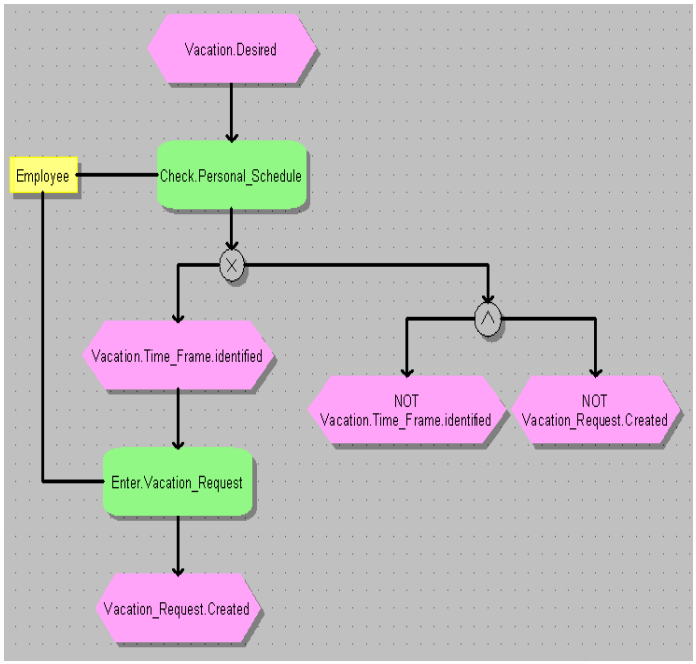

Figure 2: Create.Vacation_Request [eEPC].

The models shown in Figure 1 and Figure 2 follow standard ARIS modeling methods for assigned processes and the *ArcStyler MDA-Business Transformer Modeling Guide for ARIS*. Save the model and proceed with the ArcStyler UML Modeler.

Mapping the eEPC to a UML State Machine

Start the ArcStyler UML Modeler (ArcStyler 3.x), and map the ARIS eEPC to an UML State Machine.

To do so, proceed as follows:

- Build a new businessTransformer package in the Logical View, and then select ArcStyler → ARIS Explorer from the businessTransformer context menu.

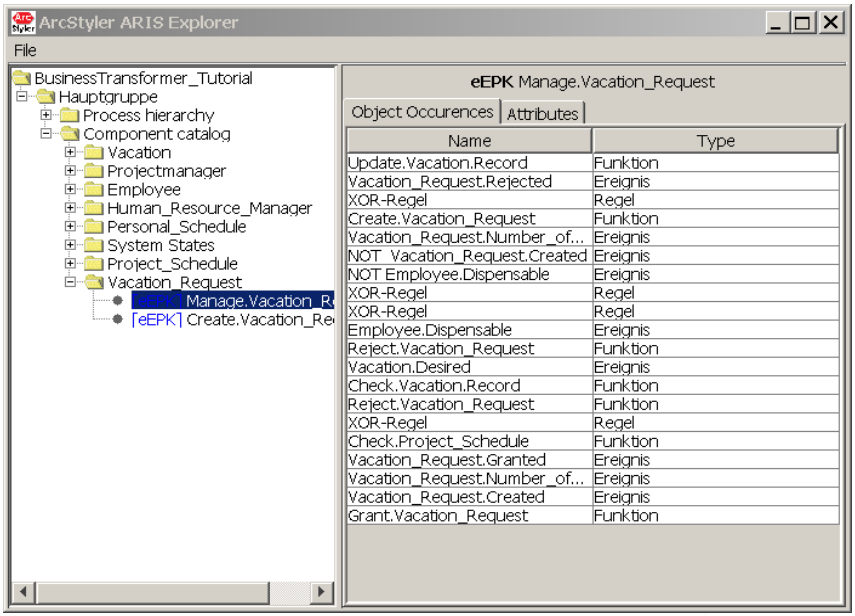

Figure 3: ARIS Explorer in the ArcStyler.

- **Open** an **ARIS database** via the File menu (refer to Figure 3).
- Right-click the eEPC in the tree and choose **Map to UML state machine** from the context menu.
- Decide which context type you want implemented in the UML model (class, process, or accessor; refer to “Restrictions for Mapping” for more information).
- Decide whether to map the eEPC including assigned eEPCs or ignoring assigned eEPCs (refer to “Restrictions for Mapping” for more information).
- Decide whether to map the events to triggers or guards (refer to “Restrictions for Mapping” for more information).
- Decide whether to include or exclude the static parts of your model in the UML model (refer to “Restrictions for Mapping” for more information).
- Select the target location for the UML model of the new package **businessTransformer**.
- Import the model (refer to Figure 4 for the last four steps).

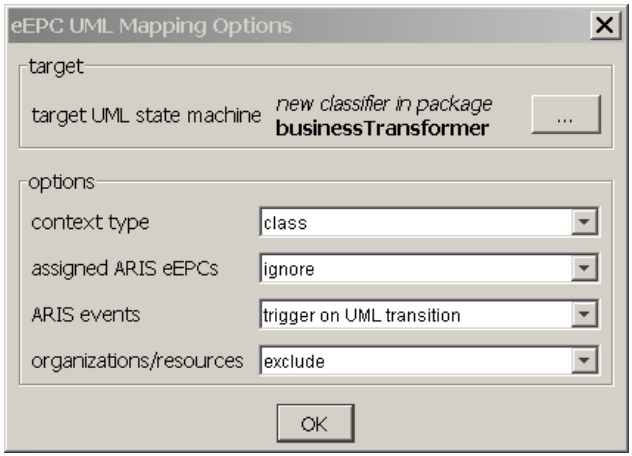

Figure 4: UML Mapping Options.

This example uses the context type *class*. Assigned ARIS eEPCs will be *ignored* for the server part, the ARIS events will be mapped to *triggers on UML transitions*, and organizations and resources will be *excluded*. As a result of this mapping, the state machine diagram is loaded.

Completing the UML State Machine

These steps will complete the UML state machine and prepare the model for code generation.

- Step 1: Insert a single State Machine **Start State** and connect it via State Transition with the starting point of the UML State Machine Diagram.
- Step 2: Connect the end states with **End States** in the state machine diagram (refer to Figure 5).

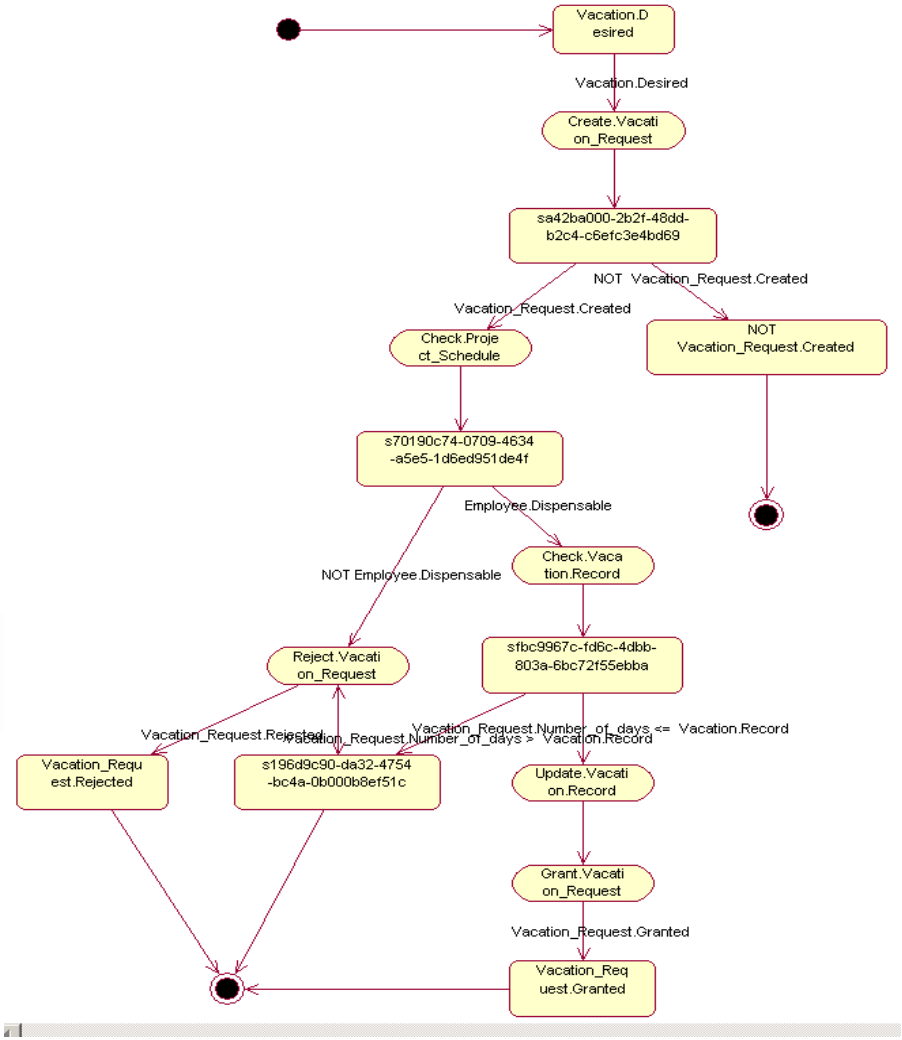

Figure 5: UML State Machine Diagram with Start and End States.

Restrictions for Mapping

Table 1 shows the possible combinations of assigned ARIS processes and transition types as well as the supported type of mapping from ARIS to UML. The WLI MDA-Cartridge supports only UML models without assigned processes and with transition type guards.

The properties for mapping are specified in the dialog displayed when the eEPC is imported (refer to Figure 4).

Table 1: Mapping Options.

Technology	JAVA2FSM	WLI	JSP WebApplication
Context Type	class	process	accessor
Assigned ARIS eEPCs	supported as UML composite state (by value)	supported as UML composite state (by value)	supported as UML composite state (by value)
ARIS events	triggers guards	guards	triggers guards
Organizations/ Resources	exclude	include	exclude

Note: Recursive assignments will be ignored in all cases.

1 FROM EEPCS TO UML

Restrictions for Mapping

2

From UML to Java2 FSM

This chapter describes how to simulate the imported finite state machine and how to generate Java code for the FSM.

For detailed information concerning the FSM, please refer to the *ArcStyler Modeling Style and User's Guide.*

The following topics will be discussed in this chapter:

- "Simulating the FSM"
- "Generating JAVA Code"

Simulating the FSM

To check the functionality of the imported finite state machine, use the FSM Simulator of ArcStyler 3.x (refer to Figure 6).

A detailed description of the FSM Simulator is provided in the *ArcStyler Modeling Style and User's Guide*.

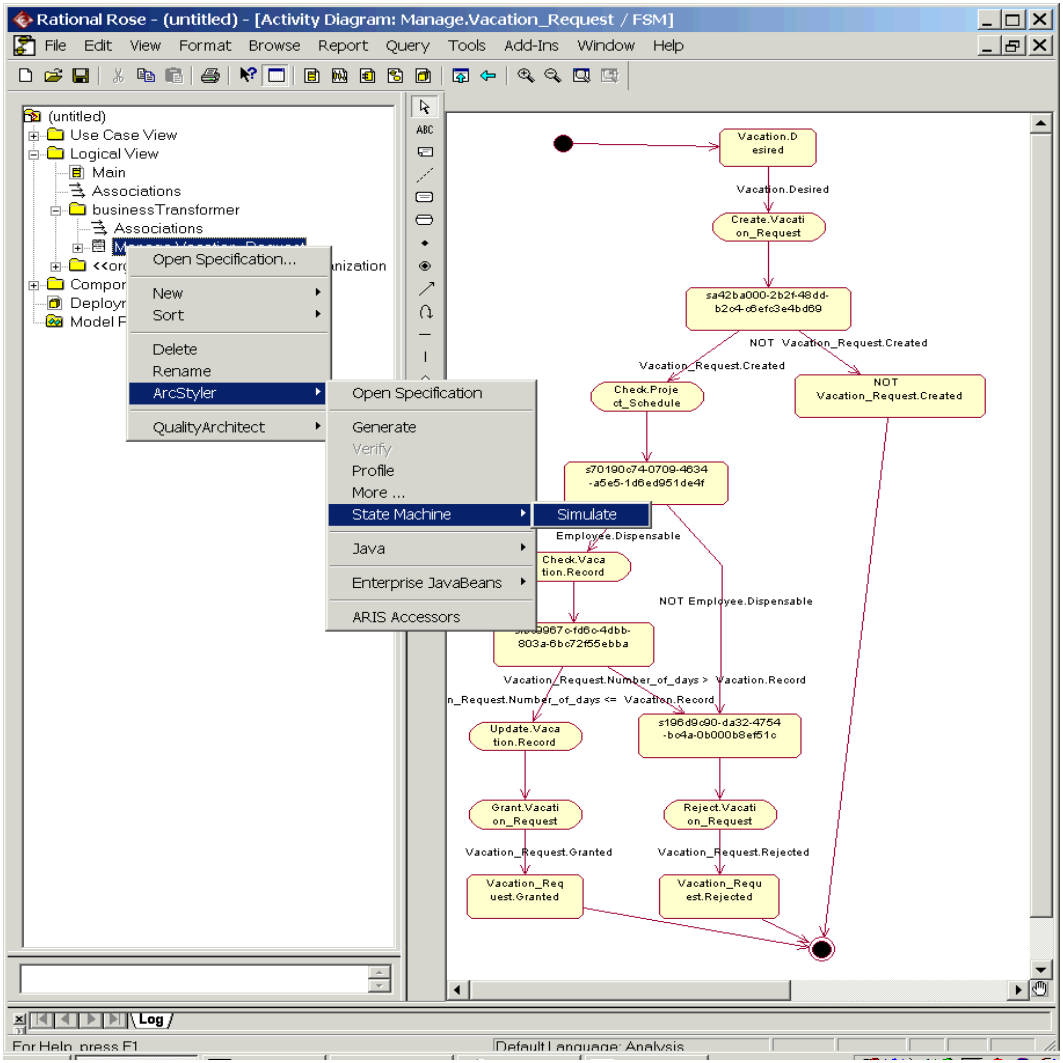

Figure 6: Starting the FSM Simulator.

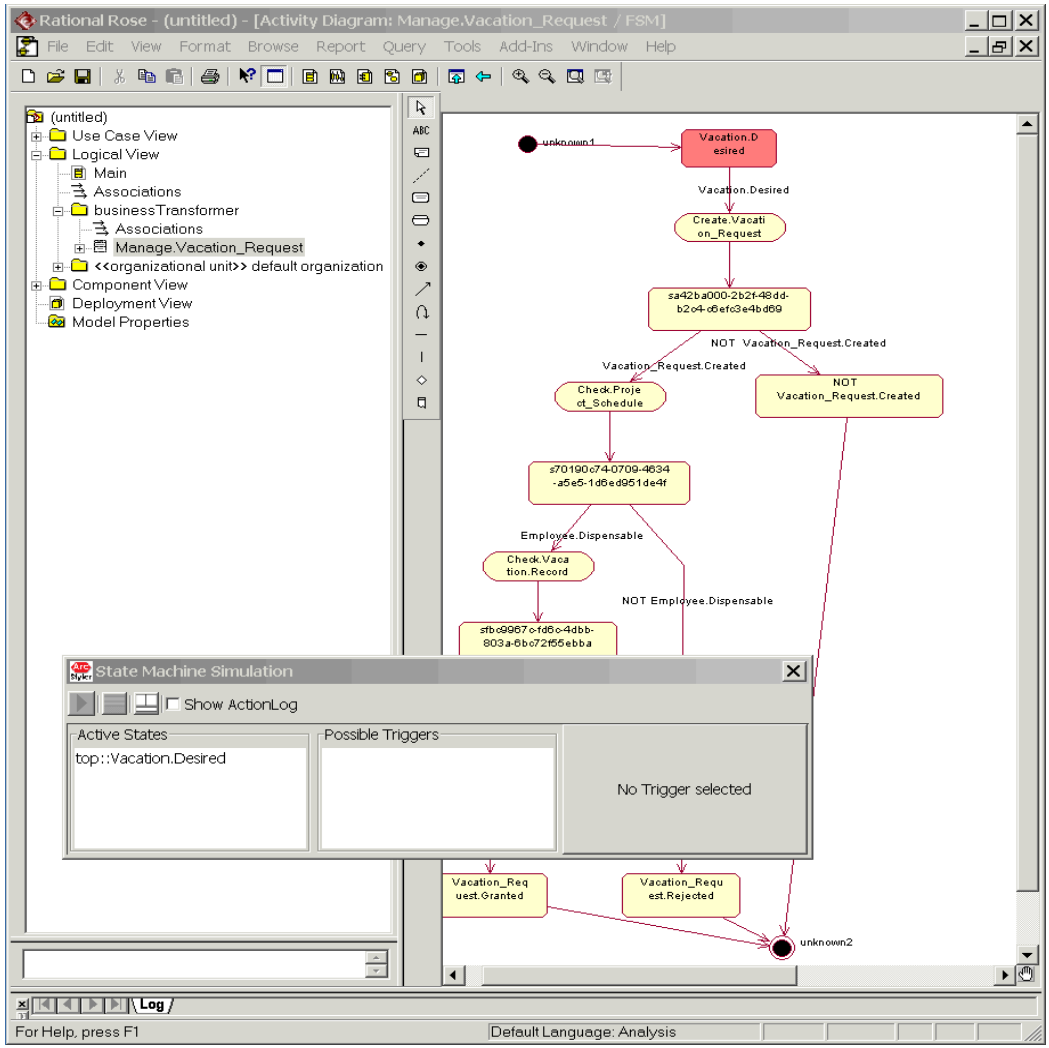

Figure 7: The FSM Simulator.

Generating JAVA Code

To generate the corresponding Java code, you have to create a new EJB Archive first. To do so, proceed as follows:

- Step 1: Create a new **businessProcess** package in the Component View context menu of the UML model.
- Step 2: Select ArcStyler → Java → New Java Archive from the businessProcess context menu.
- Step 3: Build a

3

From UML to WLI

This chapter describes how to generate a WebLogic Integration (WLI) application from the UML model. WLI is an add-on workflow engine for the EJB Container WebLogic Server.

For detailed information concerning the WLI MDA-Cartridge, please refer to the *ArcStyler Cartridge Guide For BEA Weblogic Integration Server 2.1*.

Generating JAVA Code

Note: The UML Mapping Options for WLI are different from those for JAVA2FSM!

To generate the corresponding Java code, you first have to create a new EJB Archive:

- Step 1: Create a new **businessProcess** package in the Component View context menu of the UML model.
- Step 2: Select ArcStyler → Enterprise JavaBeans→ New EJB Archive from the businessProcess context menu.
- Step 3: Build a **Realizes** relation for the process model object. Drag&drop the **Manage.Vacation_Request** object in the EJB Archive.
- Step 4: Set the appropriate ArcStyler configuration by selecting the WLI MDA-Cartridge for code generation.
- Step 5: Generate the **Manage.Vacation_Request** object. The resulting Java code can be customized in the protected areas.

You will find the complete build support for building, deploying and starting the WebLogic Server for your WLI application in the generated file environment **components/businessProcess/businessProcess/wls6**. Procedure:

- Step 1: Open a Command Prompt window for the **components/businessProcess/businessProcess/wls6** folder and type **build** to compile and build the application.
- Step 2: Type **build createTables** to build the database structure (this command is required only once per project to initialize the database tables).

- Step 3: Type **build initWLI** to initialize WLI (this command is required only once per project to initialize the WLI application).
- Step 4: Type **build startServer** to start the WLS Server.
- Step 5: When the WLS Server is running in development mode, type **build deployTemplates** (for technical reasons, this command may need to be entered twice).
- Step 6: Start the BEA WLI Worklist via the Windows Start menu to run the application. UserName and password for the WLI Worklist are *test/test*.

Refer to Figure 8, Figure 9 and Figure 10 for information on using WLI.

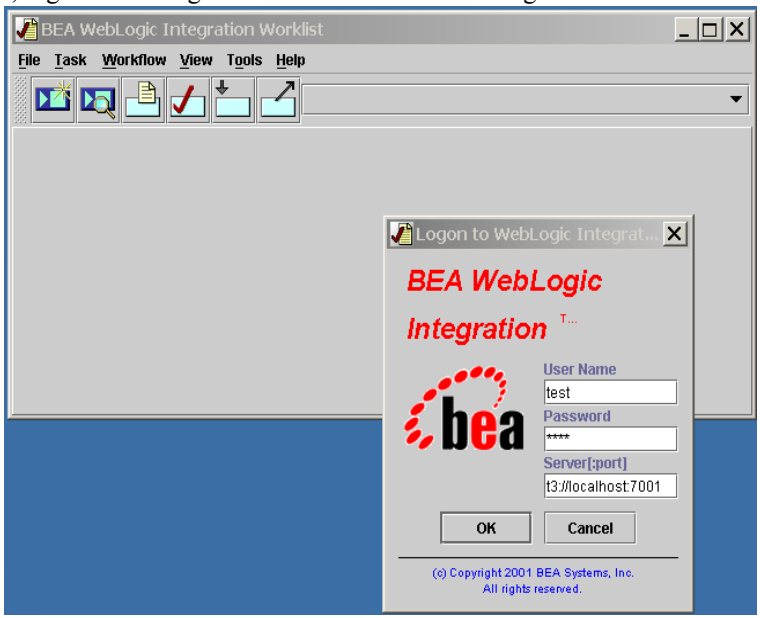

Figure 8: WLI Logon.

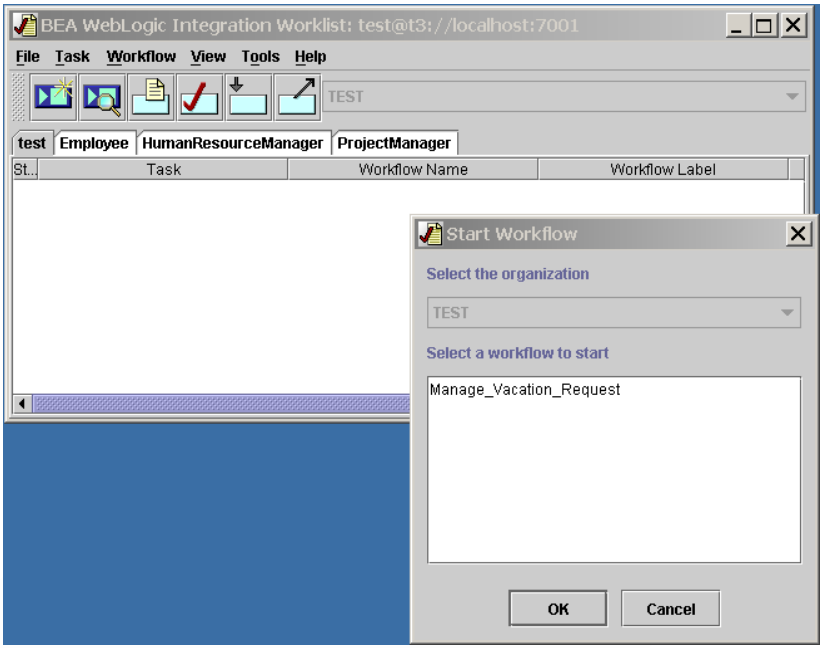

Figure 9: Workflow Selection.

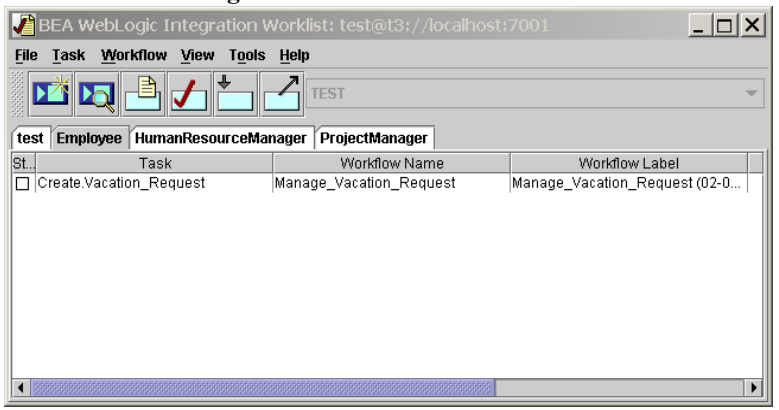

Figure 10: Running WLI Application.

3 FROM UML TO WLI

Generating JAVA Code

4

From UML to Web Application Front-Ends

This chapter describes how to transform the generated UML Activity Diagram into a Web Accessor model and how to generate the Web application.

For detailed information concerning the ArcStyler Accessor framework, please refer to the *Accessor Guide* and the *Accessor Tutorial*.

The following topics will be discussed in this chapter:

- "Generating the Accessor Model"
- "Generating the Web Application"

Generating the Accessor Model

The ARIS eEPC **Create.Vacation_Request** is to be implemented as a Web application. The first step is to import the eEPC from the ARIS database. Please refer to Chapter 1, "From eEPCs to UML" for details. Make sure to choose the appropriate UML mapping options. Proceed as follows:

- Create a new **businessProcess** package in the Logical View context menu of the UML model.
- Import the ARIS eEPC **Create.Vacation_Request** with the UML mapping options shown in Figure 11.

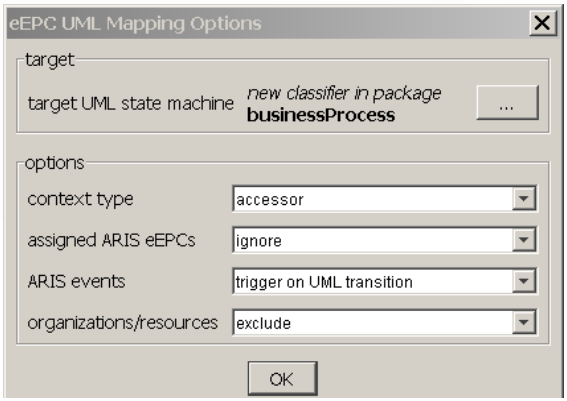

Figure 11: UML Mapping Options for Web Applications

To add the required interfaces to this UML model, proceed as follows:

To do so, proceed as follows:

- Import the ArcStyler type system: File → Import **%ARCSTYLER_HOME%/models/iOCATBase.ptl** into the Logical View.
- Import the representers specific to the ArcStyler MDA-Business Transformer for ARIS: File → Import **%ARCSTYLER_HOME%/model/p2a_accessors.cat** into the Logical View.

Note: Before you can use File → Import, you must open a Class Diagram!

These two imports are sufficient to generate the contain all the necessary interfaces. To continue, proceed as follows:

- Load the **WebAccessor** MDA-Cartridge in the ArcStyler Configuration dialog (Tools → ArcStyler → Configure) and add the following **Classpath extension** on the **ServerConfig** tab: **%ARCSTYLER_HOME%/rt/p2a/uiacc_jsp/p2a_accessors.jar** (use the absolute path name).
- Choose ArcStyler → ARIS Accessors from the context menu of the imported **Create.Vacation_Request** eEPC. You will note that the stereotypes of the states have changed to be in line with the Accessor metamodel, but the business logic of the control flow has remained unmodified (cf. Figure 12).

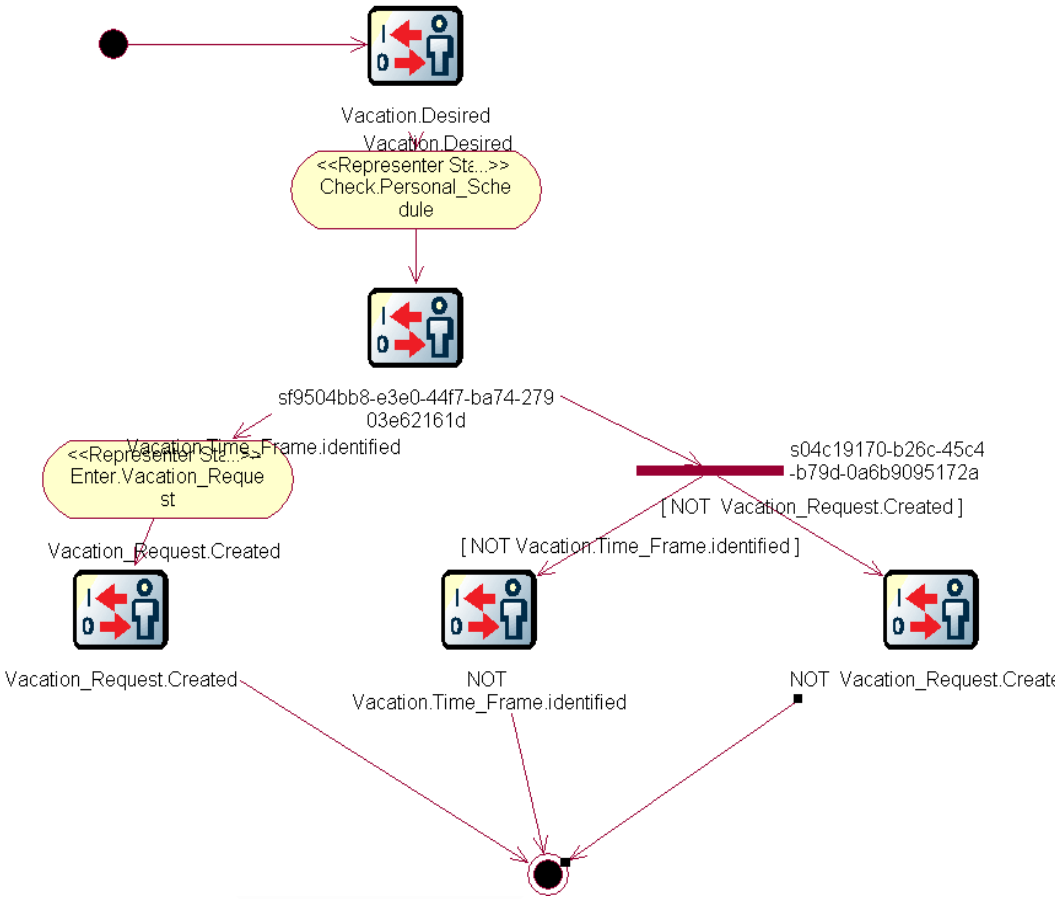

Figure 12: The Transformed UML Model for WebAccessors.

Generating the Web Application

To generate the Web application, you first have to create a deployable component in the Component View. To do so, proceed as follows:

- Step 1: Create a new **businessProcess** package in the Component View context menu of the UML model.
- Step 2: Select New → Component from the businessProcess context menu and name the component **businessProcess**.
- Step 3: Change the stereotype of the new component businessProcess to Webapplication.
- Step 4: Right-click the businessProcess component and select ArcStyler → Open Specification from the context menu.
- Step 5: Assign the **Create.Vacation_Request** accessor to the **WebApplication** businessProcess (cf. Figure 13) by clicking the appropriate button.

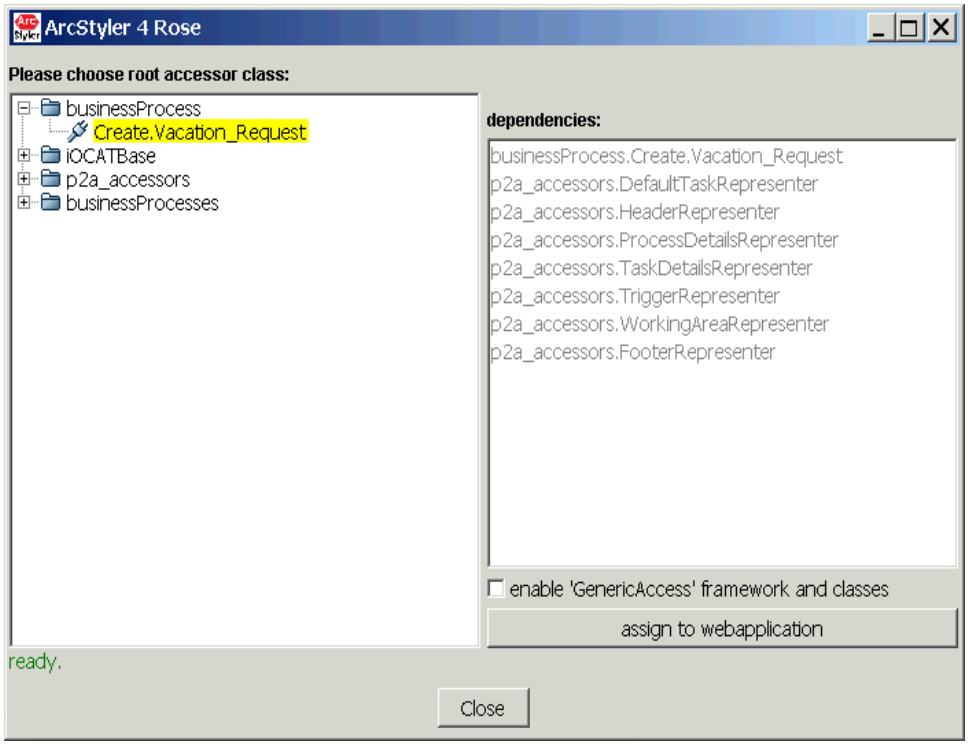

Figure 13: Assigning the Accessor to the Web Application.

- Generate the application.
- Add the following custom library location information to the generated **build.xml**:

```
<!-- /* START OF PROTECTED AREA  <<clientLibs>> */ -->
  <target name="clientLibs" depends="prepare" description="copy required project
     libraries to the webapp directory">
       <copy file="${as.home}/rt/p2a/uiacc_jsp/p2a_accessors.jar"
       tofile="${webapp.dir}/WEB-INF/lib/p2a_accessors.jar"/>
  </target>
<!-- /* END OF PROTECTED AREA 109db2c50000008d */ -->
```

The resulting Java code can be customized in the protected areas.

Open a Command Prompt window for the **components/businessProcess/businessProcess/webacc** folder and type **build run** to start the Web application.

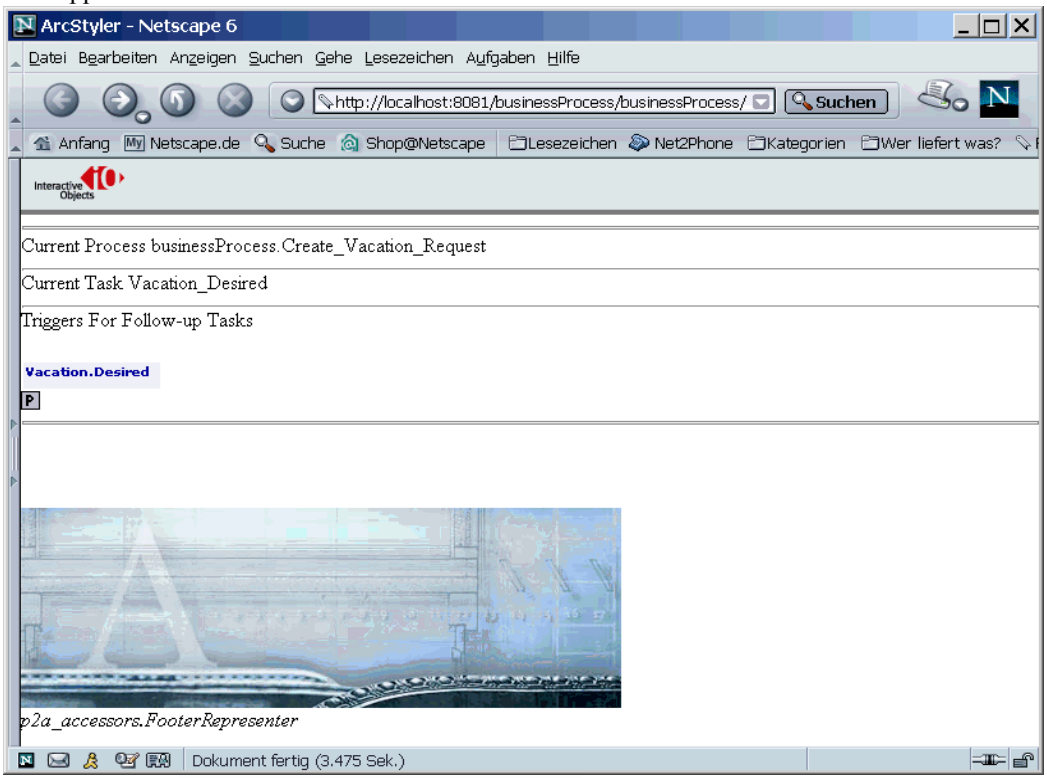

Figure 14: Generated Web Application.

Use the interaction elements listed below Triggers For Follow-Up Tasks on the generated Web page to run through the process. The image shown in the lower half of the page can be replaced by modeled and generated ArcStyler representers specific to the current task.