



# RIDE

## “A Roadmap for Interoperability of eHealth Systems in Support of COM 356 with Special Emphasis on Semantic Interoperability”

### COORDINATION ACTION

**PRIORITY 2.4.11 Integrated biomedical information for better health”: eHealth**

## RIDE D.5.3.1 – Contribution to Standards: ebBP Editor v1.0.4 User Manual

*Due Date:*

*Actual Submission Date:*

Project Start Date: January 01, 2006  
Project End Date: December 31, 2007  
Project Duration: 24 months  
Leading Contractor: METU-SRDC  
Organization:

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
<b>PU</b>	Public	X
<b>PP</b>	Restricted to other programme participants (including the Commission Services)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission Services)	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)	

Document History:

<b>Version</b>	<b>Date</b>	<b>Changes</b>	<b>From</b>	<b>Review</b>
V0.1	July 14, 2006	Initial version created	METU	All partners

## RIDE Consortium Contacts:

N <sup>o</sup>	Organisation	Street name and number	Post Code	Town/ City	Country Code	Title	Family Name	First Name	Phone N <sup>o</sup>	Fax N <sup>o</sup>	E-Mail
1	METU-SRDC	Inonu Bulvari	06531	Ankara	Turkey	Prof. Dr.	Dogac	Asuman	+90-312-2105598	+90(312)2101004	asuman@srdc.metu.edu.tr
2	OFFIS	Escherweg 2	26121	Oldenburg	Germany	Dr.	Eichelberg	Marco	+49-441-9722-147	+49-441-9722-102	eichelberg@offis.de
3	IFOMIS	Campus Saarbrücken	66041	Saarbrücken	Germany	Prof.	Smith	Barry	+49(0)681/302-64777	+49(0)681/302-64772	phismith@buffalo.edu
4	EUROREC	co IDISS - Croix-Rouge Française route de Platon	42400	Saint Chamond	France	Prof.	DeMoor	Georges	+32-9-2403421	+32-9-2403439	georges.demoor@u Gent.be
5	CNR	Piazzale Aldo Moro 7	00100	Roma	Italy	Prof.	Rossi Mori	Angelo	+39 06 86 090 250	+39 06 86 090 340	angelo@ibt.rm.cnr.it
6	NTUA, ICCS	42, Patision street	10682	Athens	Greece	Prof.	Mentzas	Gregoris	+302107723895	+302107723550	gmentzas@softlab.ntua.gr
7	NUIG, DERI	University Road	na	Galway	Ireland	Prof.	Vitvar	Tomas	+353 91 495270	+353 91 495270	tomas.vitvar@deri.org
8	IHE-D	Stresemannallee 19	60596	Frankfurt	Germany	Prof.	Wein	Berthold B.	+49-241-559 559 1	+49-241-559 558 2	wein@radiologie-aachen.de
9	OLE	Hazenakkerstraat 20a	B9520	Zonnegem (Sint-Lievens-Houtem)	Belgium	Dr.	Ceusters	Werner	+32 475 486 587	-	werner.ceusters@ecor.uni-saarland.de

## Table of Contents

1	Introduction .....	5
2	Role of the ebBP Editor .....	6
3	User Guide.....	6
3.1	Installation Requirements.....	6
3.2	Getting Software .....	6
3.3	Starting the ebBP Editor.....	6
3.4	ebBP Editor components .....	9
3.5	XmlStylist.....	11
3.5.1	Abstract.....	11
3.5.2	Usage Areas .....	12
3.5.3	General View .....	12
3.5.4	General Usage.....	12
3.5.5	Validation.....	14
3.5.6	Importing Packages .....	15
3.6	Graphical Component Panel.....	15
3.6.1	Creating a Process Specification.....	16
3.6.2	Creating a Business Collaboration .....	16
3.7	Removing a Business Collaboration .....	17
3.8	Creating a Role .....	18
3.9	Creating a Business Transaction Activity .....	18
3.10	Creating a Collaboration Activity .....	21
3.11	Creating a Complex Business Transaction Activity .....	22
3.12	Creating a Success .....	23
3.13	Creating a Failure.....	23
3.14	Creating a Fork .....	23
3.15	Creating a Join .....	23
3.16	Creating a Decision.....	23
3.17	Specializing a Business Transaction .....	24
3.18	Creating a Business Transaction.....	27
3.19	Creating a Specialized Business Transaction .....	27
3.20	Editing the Process Specification.....	27
3.21	Saving the Process Specification .....	27
3.22	Importing another Process Specification (XInclude).....	27
4	Conclusion.....	28

# 1 Introduction

This document is a guide for the end-users of the ebBP (ebXML Business Process Specification Schema) Editor and provides brief information about the usage and capabilities of the ebBP Editor. Please note that, this document DOES NOT provide a background on ebBP.

The current version of ebBP Editor is based on ebXML Business Process Specification Schema Technical Specification v2.0.3. An adaptation for another specification document may need some changes in the editor implementation.

As used in the ebBP Technical Specification v2.0.3, ebBP technical specification refers to technical specification document, ebBP definition refers to ebBP business process definition, ebBP instance refers to ebXML BPSS (Business Process Specification Schema) instance and ebXML BPSS schema is referenced as ebBP schema.

The ebBP Editor is an open source tool designed to help the users in creating generic as well as domain specific Business Process Specifications based on ebBP Version 2.0 by using semantic mechanisms. The editor lets the user to create Process Specifications from scratch or use the existing ones.

As a part of ebBP Editor, users can create, modify and also validate ebBP instances in XML format by using XmlStylist. XmlStylist GUI displays process specifications, business transactions, packages and the corresponding specification document in tree structures. XmlStylist GUI dynamically updates itself by tree node selections allowing users to fill the attributes of the selected node, add new child nodes or remove the selected ones. To avoid generating invalid instances, after each modification a validation of the document is done. Problematic actions are not allowed and users are notified in these cases. Moreover, one can import packages to existing document through XmlStylist.

The ebBP Editor also provides a graphical user interface to define domain specific business transaction classification hierarchies based on ebBP Business Transaction and the associated patterns. A user can specialize any of the generic ebBP Business Transaction patterns by specifying properties, or the document to be exchanged, but not necessarily a complete transaction. For this purpose, basic ebBP ontology is dynamically extended. When the user finishes the specialization of a transaction template, the new transaction (template) is automatically annotated by a name given by the user.

When the user specializes a transaction annotated with a node of this ontology, a subclass is created with a name provided by the user, and the specialized transaction is annotated with it. Note that, the user has the ability to annotate this (template) transaction with the nodes of other domain-specific ontologies as well, since ebXML allows registry objects to be annotated with more than one classification hierarchy. The nodes of the ontologies introduced are then created in the ebXML registry as ClassificationNodes. Using these ClassificationNodes it is possible to search for specialized Transactions to be used as building blocks of Process Specifications.

## **2 Role of the ebBP Editor**

The ebBP Editor does not only facilitate creating a Process Specification conforming to the latest ebBP specification, but also allows domain-specific semantic annotation of Business Transactions. This semantic information provides building blocks for creating a Business Process Specification Schema.

## **3 User Guide**

### **3.1 *Installation Requirements***

Software explained through this document requires a computer system with the following features:

- Java 2 Standard Edition (see below)
- Relatively fast processor (300Mhz or higher)
- At least 32MB of RAM, 64MB or more recommended

ebBP editor requires a Java Virtual Machine (JVM) which implements Sun's Java 2 Standard Edition. Recommended versions are JDK or JRE version 1.5. ebBP has been tested to work with Sun's JDK build 1.5.0\_04 on Windows platform. Other platforms with a working JVM of these versions should work fine, but haven't been fully tested. Sun's Java 2 SE is available from <http://java.sun.com/j2se/>. You need either the Runtime Environment, or the SDK.

In order to use JDK or JRE versions previous than 1.5, the user should set the java endorsed mechanism properly. Classes implementing newer versions of endorsed standards should be placed in JAR files. The system property `java.endorsed.dirs` specifies one or more directories that the Java runtime environment will search for such JAR files. Thus, the user may set this property to overcome endorse mechanism problems.

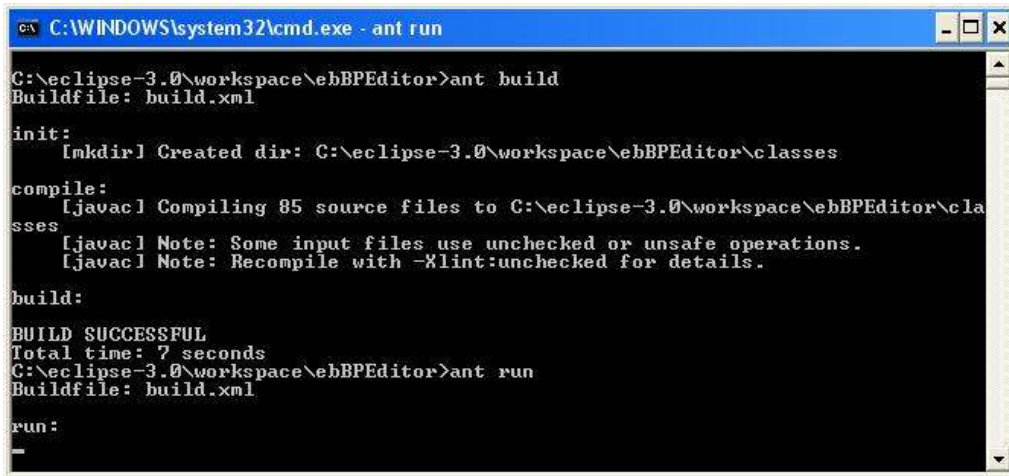
In order to avoid class loading problems, try using J2SE 1.5.

### **3.2 *Getting Software***

The latest release of the software will be available at [sourceforge.net](http://sourceforge.net) site. You will be able to download the current release for your system. The release includes the source code and libraries necessary to build the application.

### **3.3 *Starting the ebBP Editor***

The ebBP Editor can be compiled with the “ant build” command. After successfully compiling, the “ant run” command will start the editor (see Figure 1).



```
C:\WINDOWS\system32\cmd.exe - ant run

C:\eclipse-3.0\workspace\ebBPeditor>ant build
Buildfile: build.xml

init:
  [mkdir] Created dir: C:\eclipse-3.0\workspace\ebBPeditor\classes

compile:
  [javac] Compiling 85 source files to C:\eclipse-3.0\workspace\ebBPeditor\classes
  [javac] Note: Some input files use unchecked or unsafe operations.
  [javac] Note: Recompile with -Xlint:unchecked for details.

build:
BUILD SUCCESSFUL
Total time: 7 seconds
C:\eclipse-3.0\workspace\ebBPeditor>ant run
Buildfile: build.xml

run:
-
```

**Figure 1 - ant commands to start the ebBP Editor**

Initially, the ebBP Editor contains blank Process Specification document. From the menu bar of the editor GUI, users can create a new Process Specification document or open an existing one. After the decision is made by users, proper settings are loaded to XmlStylist and XmlStylist GUI shows up.

After XmlStylist is closed, business collaborations of the document are displayed by ebBP Editor in tabbed pane. For each business collaboration, a tab is created and the business activities are listed with the roles in this tab. The links between each element are also generated by ebBP Editor automatically to display the business activity choreography for that Business Collaboration.

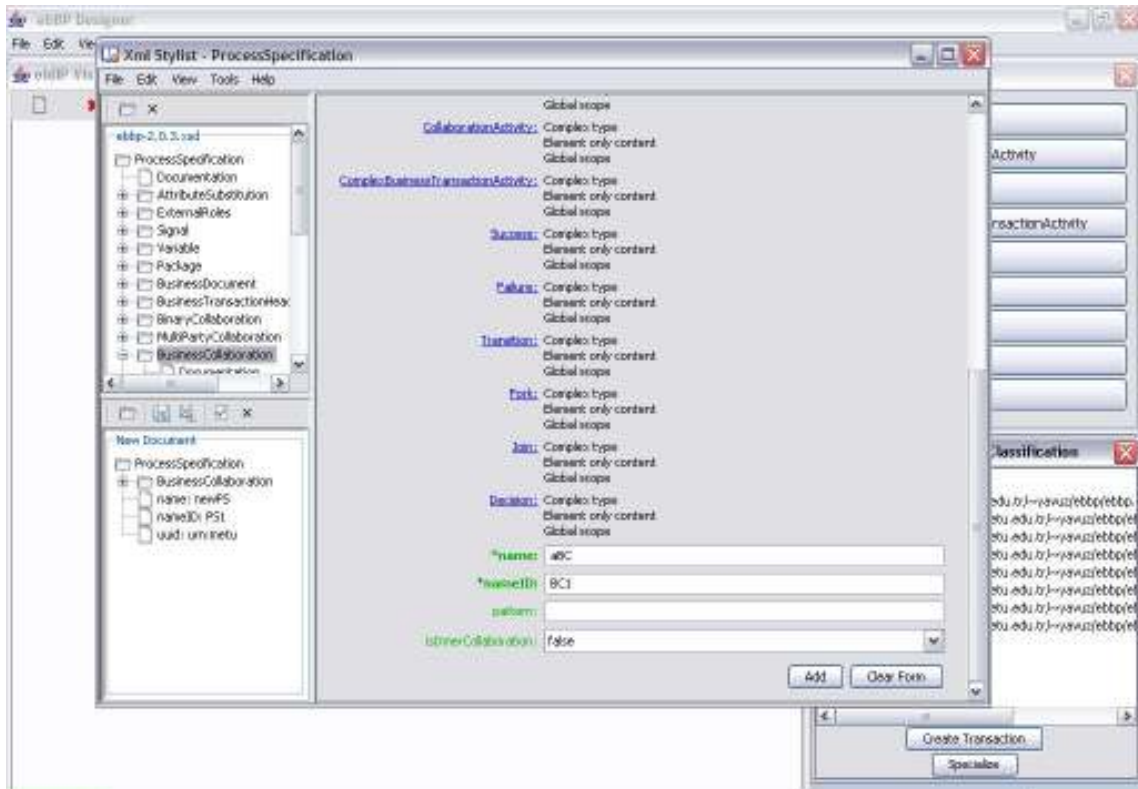


Figure 2 ebBP Editor with XmlStylist



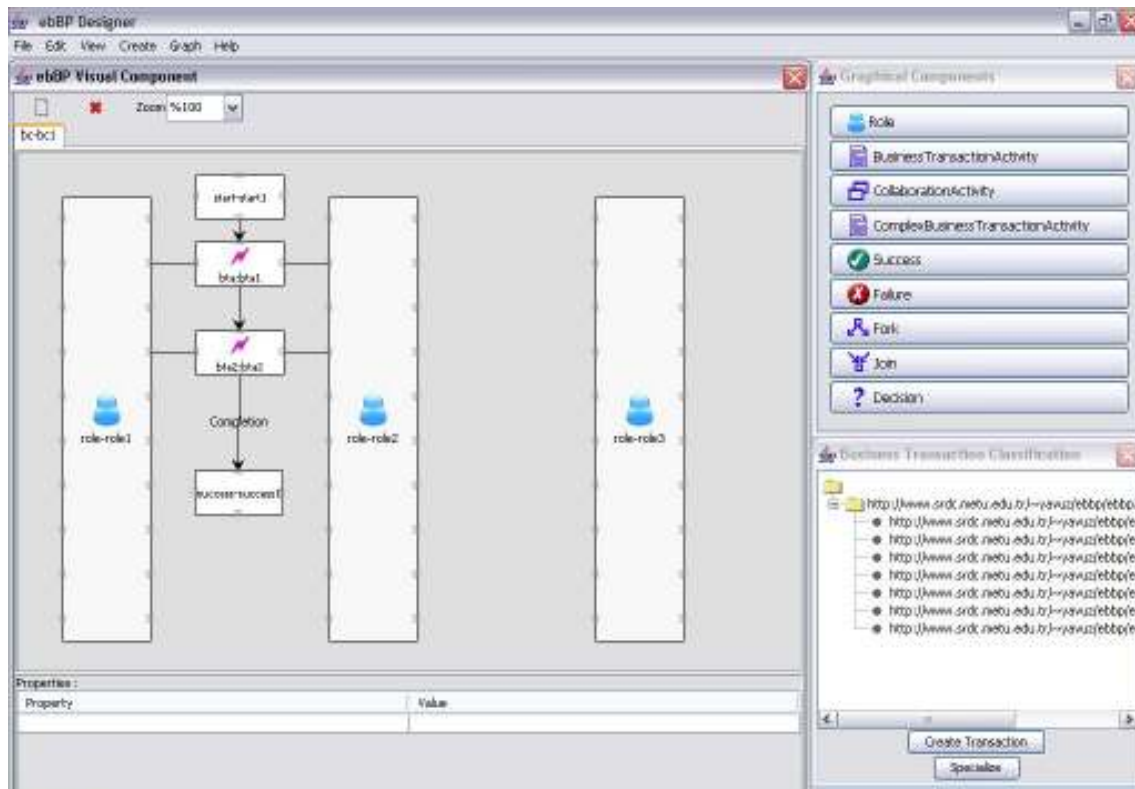
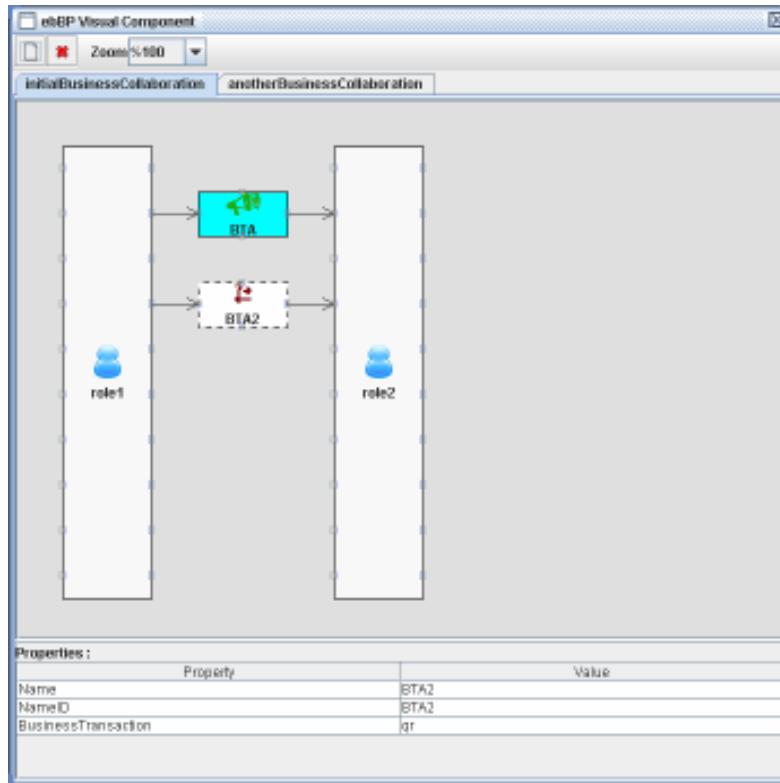


Figure 3 Business Collaborations are displayed in tabbed pane

### 3.4 ebBP Editor components

The ebBP Editor has three internal windows representing its three components. This window design facilitates further extensions to the ebBP Editor. The current components are the “ebBP Visual Component” window, “Graphical Components” window, and “Business Transaction Classification” window (see Figure 3).

The ebBP Visual Component provides a visual representation of the Process Specification. However, instead of representing the whole Process Specification graphically, the Business Collaborations are presented in separate tabs (see Figure 4).



**Figure 4 ebBP Visual Component**

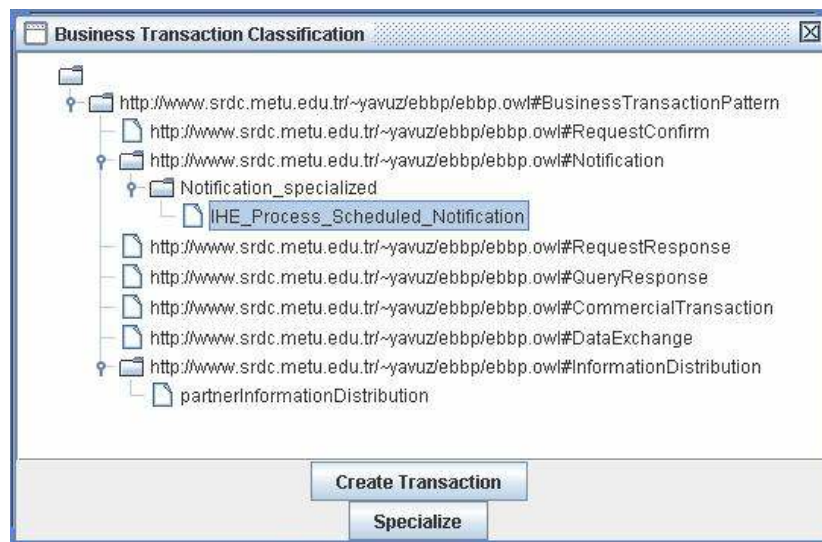
At the top-left there are “new Business Collaboration” and “remove Business Collaboration” icons. Users can add new business collaborations or remove the selected one from the document by using these buttons. In addition, at the bottom of the window, the properties of the selected item are displayed.

The Graphical Components Panel displays the constructs that can be created within a Business Collaboration. These constructs are: Role, Business Transaction Activity, Collaboration Activity, Complex Business Transaction Activity, Success, Failure, Fork, Join and Decision (see Figure 5). Dragging and dropping a button into the Business Collaboration tab, will create the corresponding ebBP construct via XmlStylist.



**Figure 5 Graphical Components**

Finally, the third component is the “Business Transaction Classification” Window (see Figure 6). This component provides a semantic infrastructure for the Business Transactions. The window presents a dynamic ontology representing the classification of Business Transaction patterns, and lets the user specialize any of the Business Transactions or a specialized Business Transaction. The Business Transaction instances are then annotated in the ebXML registry by using the selected classification nodes.



**Figure 6 Business Transaction Classification**

## 3.5 *XmlStylist*

### 3.5.1 Abstract

XmlStylist is a tool that generates ebBP instance documents from a given ebBP Schema file by providing a graphical user interface. XmlStylist parses the schema file and dynamically creates the graphical user interface of the corresponding elements defined in the schema. The user can then fill in the necessary fields by providing the required information. The editor also provides the user with information about necessity, cardinality and optionality of the fields.

Once the user has filled in the information through the GUI, a validation check is made, and the information is serialized into XML.

### 3.5.2 Usage Areas

XmlStylist is designed for creating new valid ebBP instances or modifying existing ones in accordance with the ebBP technical specification v2.0.3.

### 3.5.3 General View

The main screen of the XmlStylist has 3 parts. At the top left part of the window, ebBP technical specification file is visualized as a tree object. Below this, an ebBP instance is shown again as a tree object. Right part of the window contains the selected ebBP element's details as a form. Figure 7 shows the general view of XmlStylist.

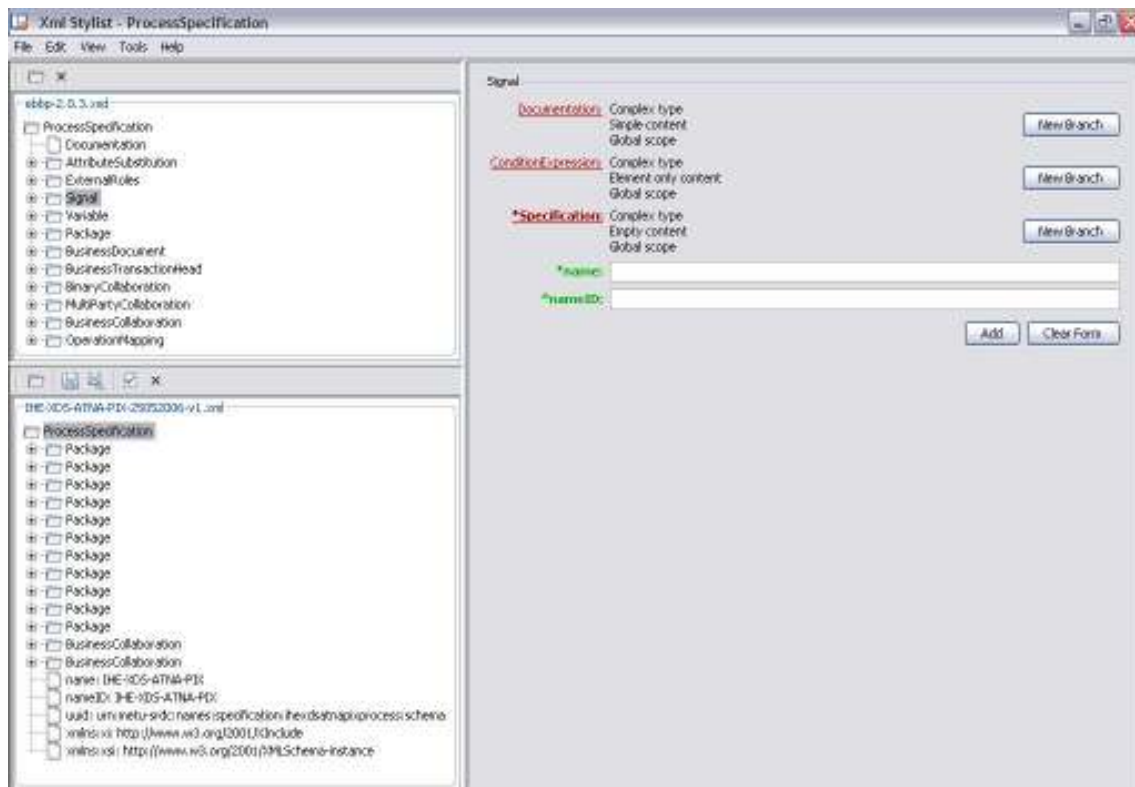
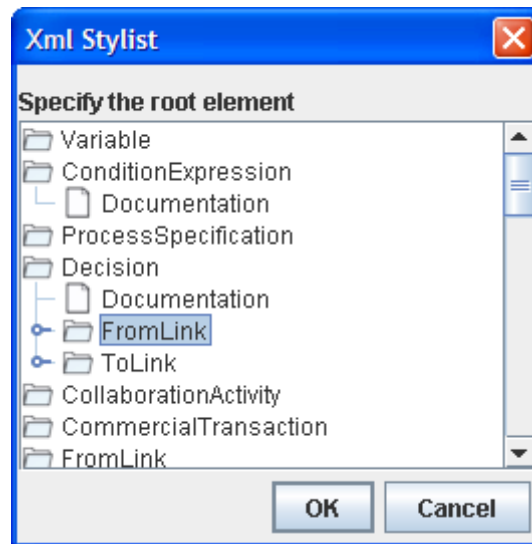


Figure 7 Xml Stylist - General View

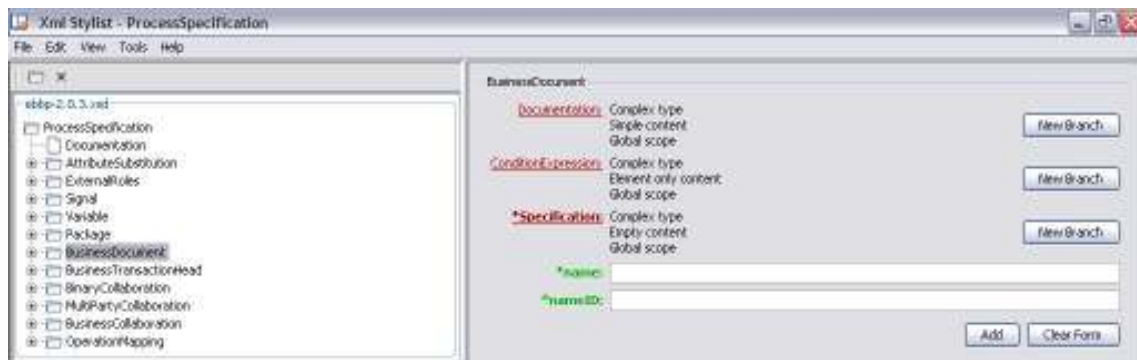
### 3.5.4 General Usage

First of all, XmlStylist forms the schema tree component by parsing a schema file. ebBP instance tree component can be visualized by loading an instance file or dynamically adding elements. When the user loads a schema file, XmlStylist checks if it has a single global element. If it contains multiple global elements, XmlStylist shows the whole global elements and their children as a tree as shown in Figure 8. The user must specify the root element of the instance by selecting the appropriate element.



**Figure 8 Xml Stylist - Select Root Dialog**

After that, the user can select the elements of the schema file by clicking the schema tree. A form, corresponding to the selected element is created dynamically at the right side of the window as shown in Figure 9. In this case, selected element's attributes are displayed in this form with blank text boxes. User can fill these attribute text boxes and then add this element to the document. Note that, for adding a child element, the right parent element is also selected from the instance tree.



**Figure 9 Xml Stylist – Schema Form**

The schema element's form is created as follows:

- Simple elements are shown as a text field, if their maximum occurrences are 1, or as a combo box, if their maximum occurrences are greater than one.
- Complex elements are shown as a link to their form. When you click their name on the form, it selects this element from the schema tree and its form is displayed in the panel. If this elements maximum occurrence is greater than 1, a new button appears next to the link named “New Branch” which adds a new element to the XML document.
- Attributes are also shown as a text field.

Field labels give valuable information. As Figure 10 shows, if you hold your mouse over a label, a tool tip that displays namespace, scope and cardinality information appears. If the field is a required field, a star (\*) precedes field label and the label is written in bold. Fields also have tool tips that show the type of that field.

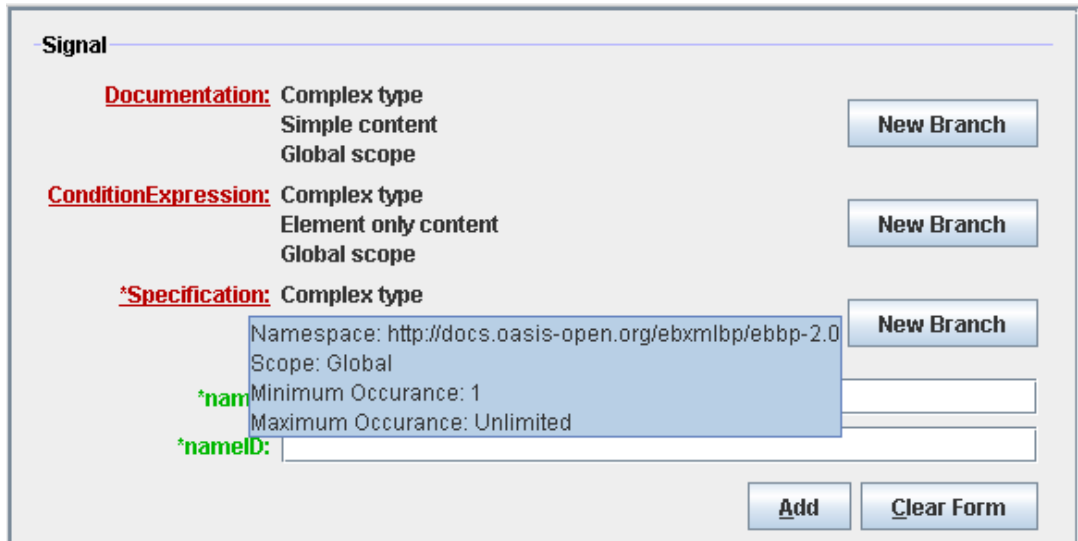


Figure 10 XmlStylist – Tool tips

When a tree node is selected from ebBP instance tree, the form and the schema tree are updated dynamically for the referenced ebBP instance element. Apart from schema form, instance form (Figure 11 XmlStylist – Instance Form) is filled with existing values of the element’s attributes and Edit / Remove buttons are displayed to modify the selected element. The schema tree changes its root and sets the root according to selected element.



Figure 11 XmlStylist – Instance Form

### 3.5.5 Validation

Validation can be achieved by many different ways. When adding a new element to the document, parent-child relationship is checked. XmlStylist does not allow users to misplace the document elements. XmlStylist also checks whether all required field of an element is

filled before adding that element to the document. The XML document is subjected to a full validation after a request is made by user or in each time XmlStylist is closed. Users are always informed about the problems after validation.

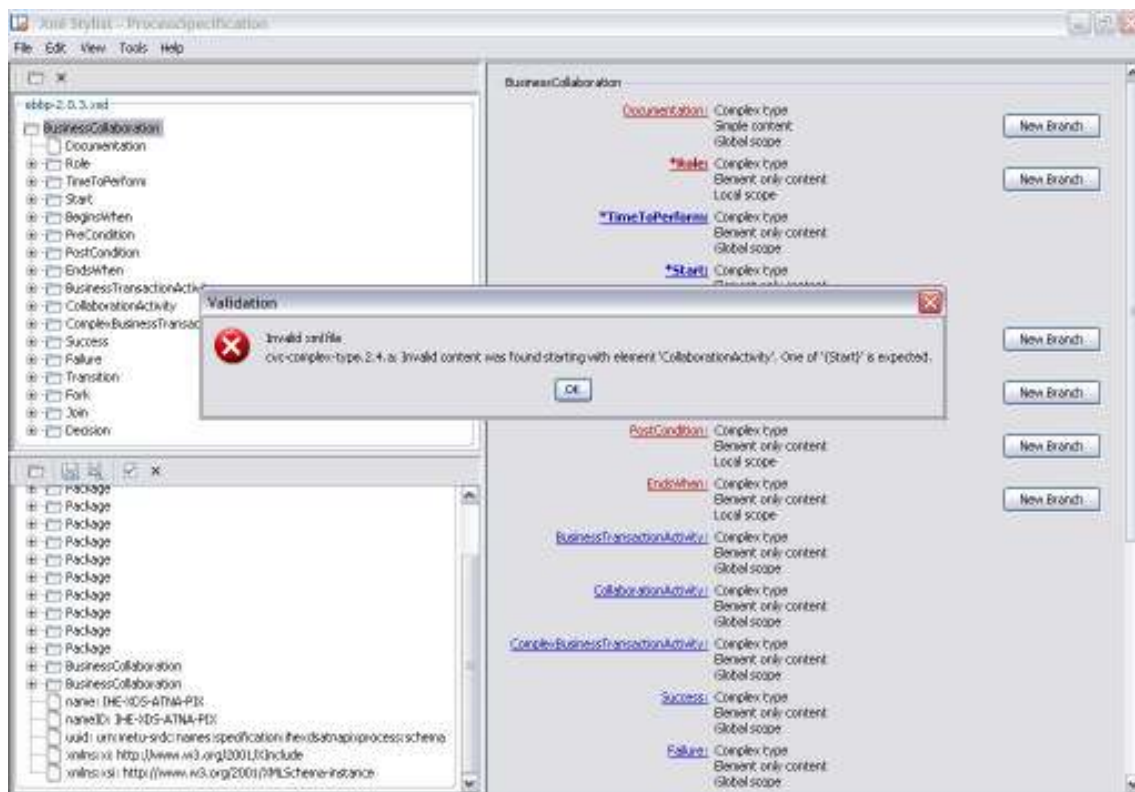


Figure 12 Xml Stylist – XmlStylist warn user about the invalid ebBP instance

### 3.5.6 Importing Packages

Packages are defined as reusable components for ebBP instances. One can import packages into a Process Specification document freely. Packages have recursive properties. They can be imported into existing ones. XmlStylist provides the importing functionality. “Import Package” menu item under the “File” menu allows users to select a package an import it into the ebBP instance.

### 3.6 Graphical Component Panel

Graphical Component Panel is another module of the ebBP Editor. This module is in coordination with the XmlStylist. They both work on the same document. For visualization purposes, Graphical Component Panel is very useful. Users can be aware of the status of an ebBP instance easily by looking at Graphical Component Panel. However, detailed and expert modifications on the document require XmlStylist-like modules. Therefore, whenever an element is added to the document from Graphical Component Panel, XmlStylist is displayed.

After closing XmlStylist successfully, Graphical Component Panel displays the elements and the links between each element automatically.



In this version, the document sharing between Graphical Component Panel and XmlStylist is limited with the Process Specification documents. Because, Graphical Component Panel is simply designed for visualizing the Business Collaborations in a Process Specification by displaying the choreography of Business Activities in these collaborations.

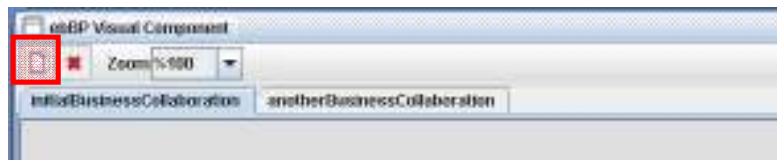
In subsections of this part, creating various ebBP elements from the Graphical Component Panel is explained.

### **3.6.1 Creating a Process Specification**

A Process Specification can be created by clicking on the “New” menu item which is under the “File” menu. By creating a new Process Specification or opening an existing one means that the ebBP Editor uses this Process Specification for further operations.

### **3.6.2 Creating a Business Collaboration**

A Business Collaboration can be created by clicking on the new Business Collaboration icon on the “ebBP Visual Component” (see Figure 13). This will create a new Tab in the ebBP Visual Component Window.



**Figure 13 New Business Collaboration**



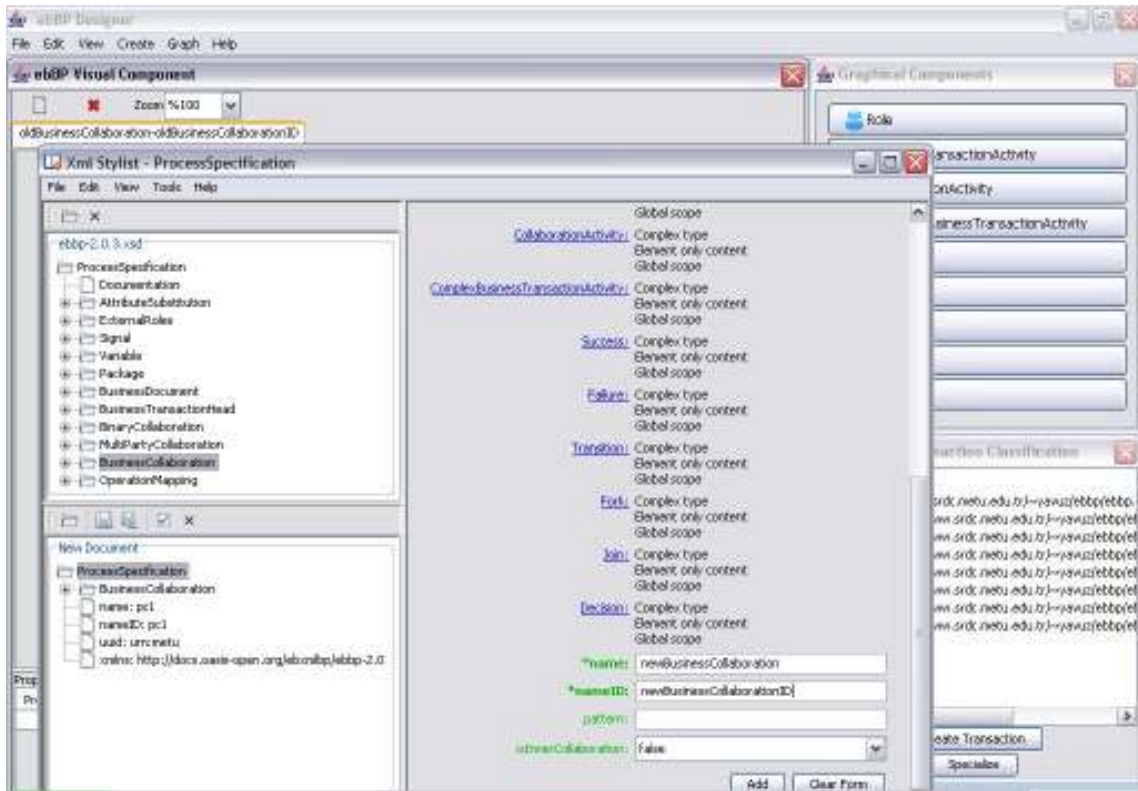


Figure 14 New Business Collaboration - XmlStylist

The XmlStylist will come up, so that the user can provide details about the Business Collaboration as described in Section 3.4. After the user provides the information and closes the XmlStylist, a new Business Collaboration tab will be created in the ebBP Visual Component (see Figure 15).

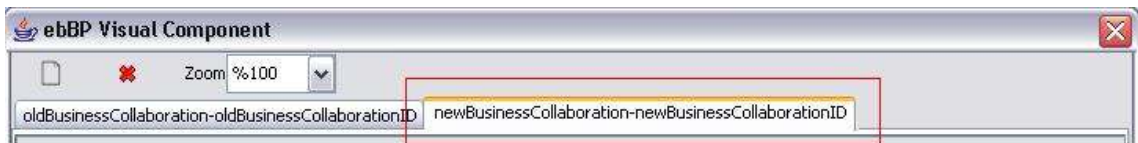


Figure 15 New Business Collaboration Tab

### 3.7 Removing a Business Collaboration

A Business Collaboration can be removed by clicking on the remove Business Collaboration icon on the “ebBP Visual Component” (see Figure 16). After removing the Business Collaboration, the tab representing this Business Collaboration will disappear.



Figure 16 Remove Business Collaboration

### 3.8 Creating a Role

A Role can be created by dragging and dropping the role button on the “Graphical Components” window into the ebBP Visual Component. The role will be created as a part of the selected Business Collaboration.

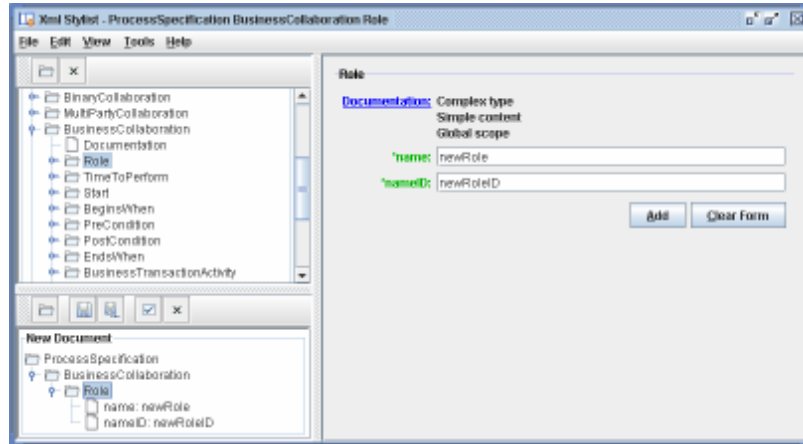


Figure 17 New Role

First, the XmlStylist will come up. The user is supposed to fill in the Role details (see Figure 17). After providing the information and closing the XmlStylist, the new Role will be placed into the Business Collaboration (see Figure 18).

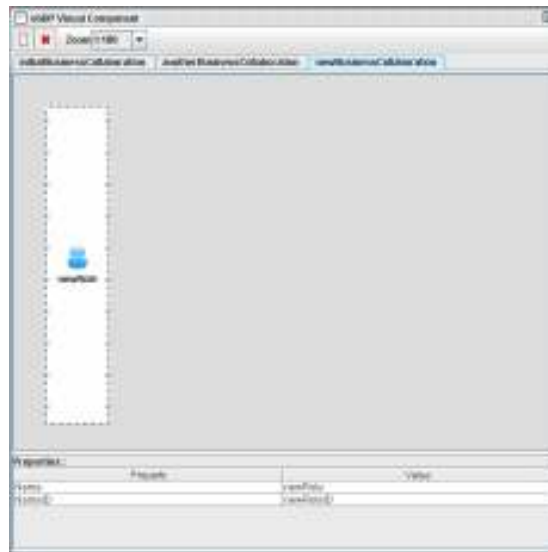
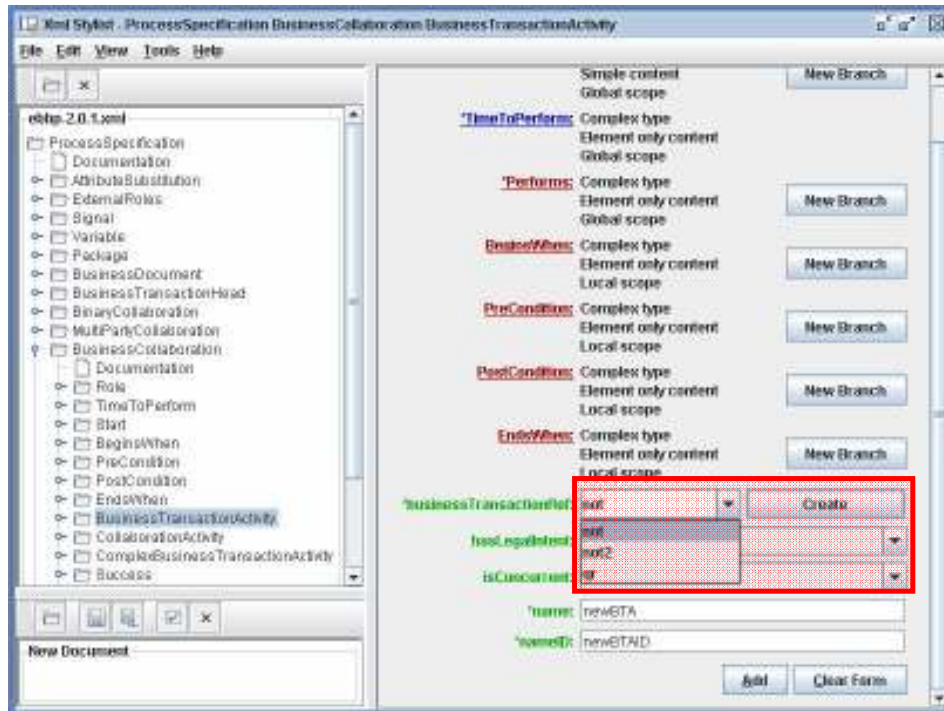


Figure 18 New Role Representation

### 3.9 Creating a Business Transaction Activity

A Business Transaction Activity can be created by dragging and dropping the Business Transaction Activity button on the “Graphical Components” window into the ebBP Visual Component. The Business Transaction Activity will be created as a part of the selected Business Collaboration.



**Figure 19 New Business Transaction Activity**

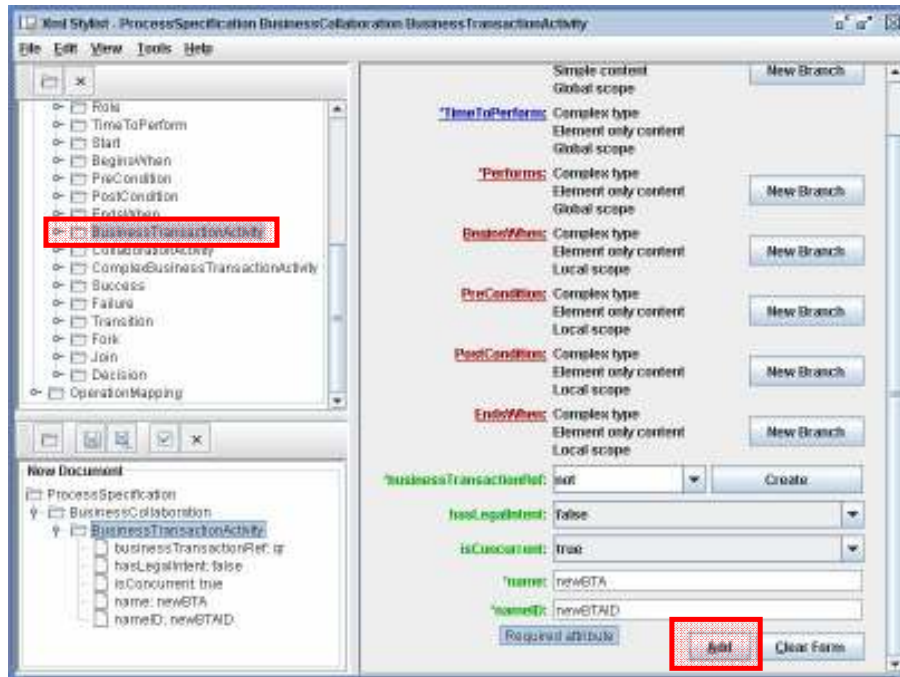
As shown in Figure 19, the user should first select the businessTransactionRef field. The combo box displays all Business Transaction patterns (Notification, Commercial Transaction, Query Response, Request Response, Request Confirm, Information Distribution, and Data Exchange) previously created. The user may choose one of these, or decide to create a new Business Transaction.



**Figure 20 Creating a New Business Transaction**

If the user decides to create a new Business Transaction and presses the Create button, a dialog will come up presenting the possible Business Transaction patterns (Figure 19). The user can create any of these Business Transactions. The XmlStylist will come up to create the new Business Transaction.

Upon closing the XmlStylist, the Business Transaction will be saved, and accessible from the combo box. (Note: At first the item may not appear in the combo box: either refresh the form, reselect the BusinessTransactionActivity node at the left (see Figure 21); or directly type in the nameID of the newly created Business Transaction into the combo box)



**Figure 21 New Business Transaction Activity**

After filling in the details of the Business Transaction Activity and pressing the Add button (see Figure 21), the user should add the required child elements of this Business Transaction Activity.

In Figure 22 the representation of the newly created Business Transaction Activities can be seen. Note that both Business Transaction Activities have different icons. This is because of the Business Transaction types they reference. The first one (newBTA) references a Query-Response Business Transaction, whereas the other references a Notification. Other types of Business Transactions also result in different icons.

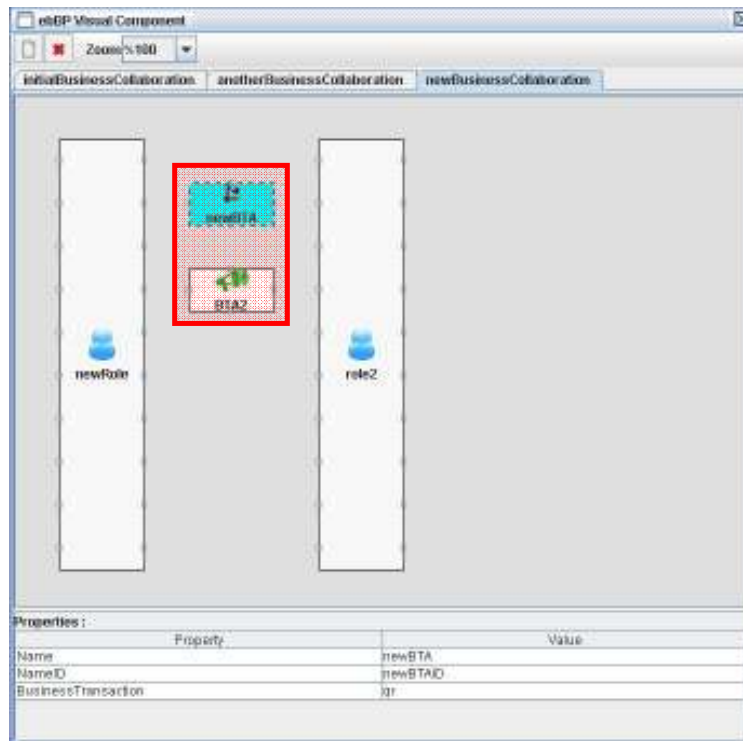


Figure 22 Business Transaction Activity Representation

### 3.10 Creating a Collaboration Activity

A Collaboration Activity can be created by dragging and dropping the Collaboration Activity button on the “Graphical Components” window into the ebBP Visual Component. The Collaboration Activity will be created as a part of the selected Business Collaboration.

The Collaboration Activity references a Business Collaboration. When a Collaboration Activity is created, the XmlStylist presents the user available Business Collaborations. The user may select one of the available Business Collaborations, or decide to create a new one (Figure 23). When a new one is created (see section 3.6.2) a new tab is added accordingly.

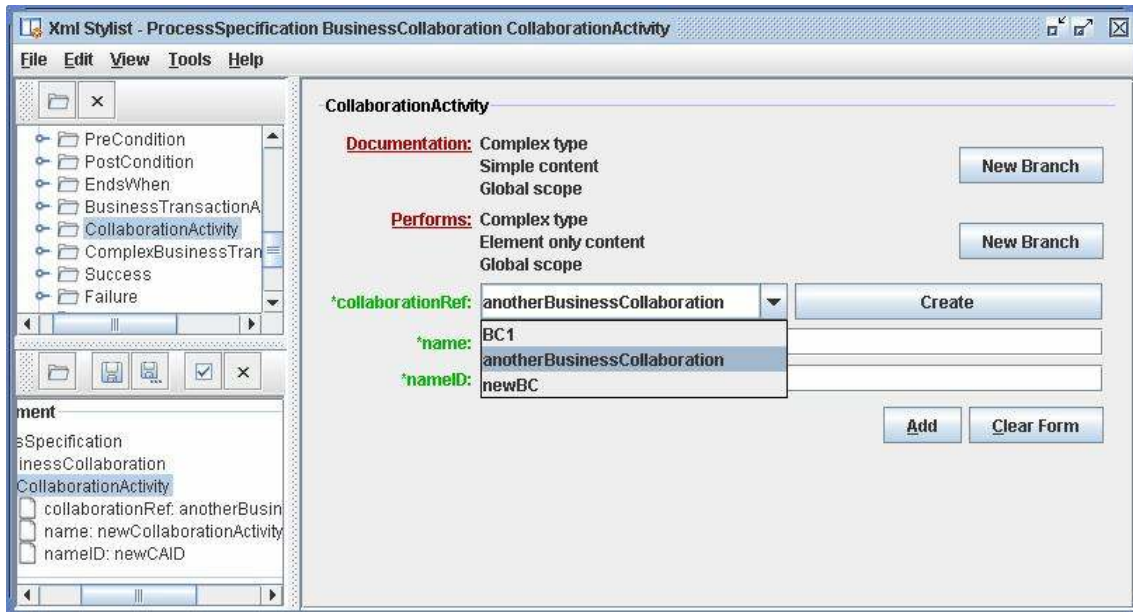


Figure 23 New Collaboration Activity

After closing the XmlStylist, the collaboration will be placed into the Business Collaboration as shown in Figure 24.

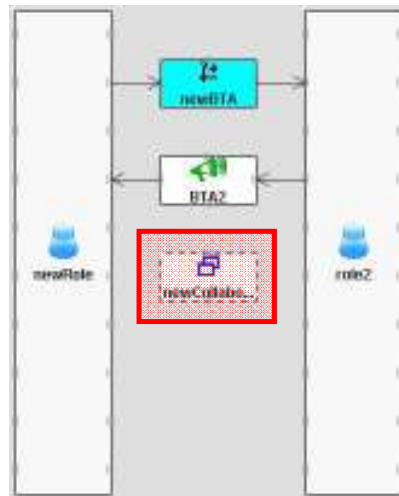


Figure 24 Business Collaboration Representation

### 3.11 Creating a Complex Business Transaction Activity

A Complex Business Transaction Activity can be created by dragging and dropping the Complex Business Transaction Activity button on the “Graphical Components” window into the ebBP Visual Component. The Complex Business Transaction Activity will be created part of the Business Collaboration within the Tab selected.



### 3.12 Creating a Success

A Success can be created by dragging and dropping the Success button on the “Graphical Components” window into the ebBP Visual Component. The Success will be created as a part of the Business Collaboration within the Tab selected.

### 3.13 Creating a Failure

A Failure can be created by dragging and dropping the Failure button on the “Graphical Components” window into the ebBP Visual Component. The Failure will be created as a part of the Business Collaboration within the Tab selected.

### 3.14 Creating a Fork

A Fork can be created by dragging and dropping the Fork button on the “Graphical Components” window into the ebBP Visual Component. The Fork will be created as a part of the Business Collaboration within the Tab selected.

### 3.15 Creating a Join

A Join can be created by dragging and dropping the Join button on the “Graphical Components” window into the ebBP Visual Component. The Join will be created as a part of the Business Collaboration within the Tab selected.

### 3.16 Creating a Decision

A Decision can be created by dragging and dropping the Decision button on the “Graphical Components” window into the ebBP Visual Component (see Figure 25). The Decision will be created part of the Business Collaboration within the Tab selected.

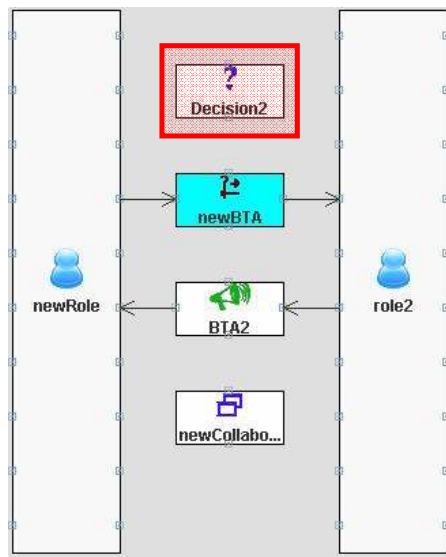


Figure 25 New Decision

After creating the Decision, the user can create the FromLink and ToLink constructs. Dragging from a Business Transaction Activity port to a Decision port will create a FromLink. On the other hand, a ToLink will be created dragging vice versa (see Figure 26).

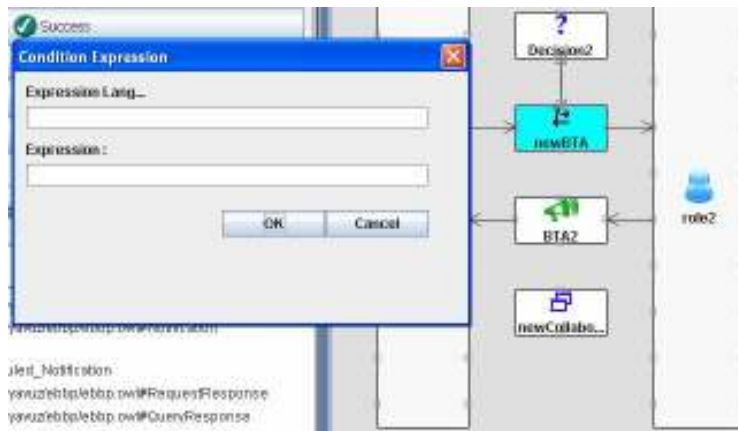


Figure 26 Decision Connections (ToLink, FromLink)

When a ToLink is created the user will be prompted to provide the condition expression. Multiple ToLink's can be created.

### 3.17 Specializing a Business Transaction

Business Transactions can be specialized by selecting a node from the ontology, which represents a Business Transaction pattern, and then pressing the Specialize button (see Figure 27). The XmlStylist will come up with specialized fields, if any. When the non-specialized (raw) Business Transactions are selected for specialization, the fields will all be empty. The user can fill in any detail about the Business Transaction.

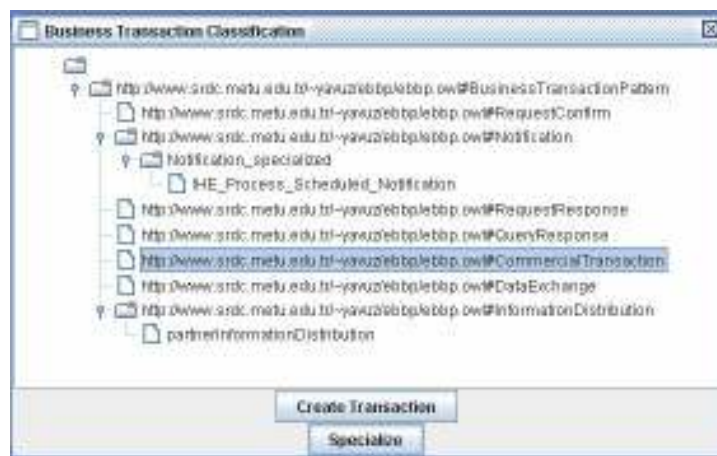
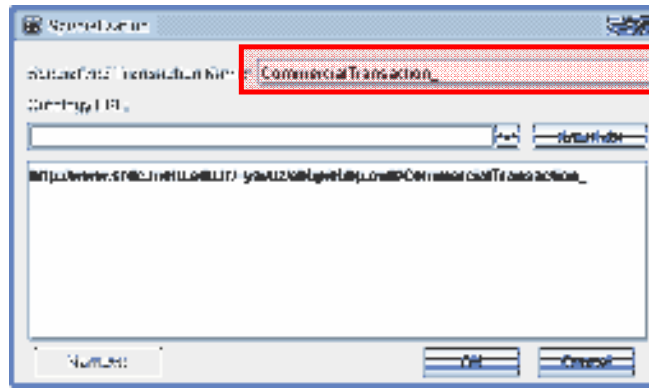


Figure 27 Business Transaction Specialization

When finished (please remember to press the Add button to save your entries), the user should close the XmlStylist. Upon closing the XmlStylist, a Specialization window is opened (see Figure 28). This window specifies the annotation for the newly specialized Business Transaction.

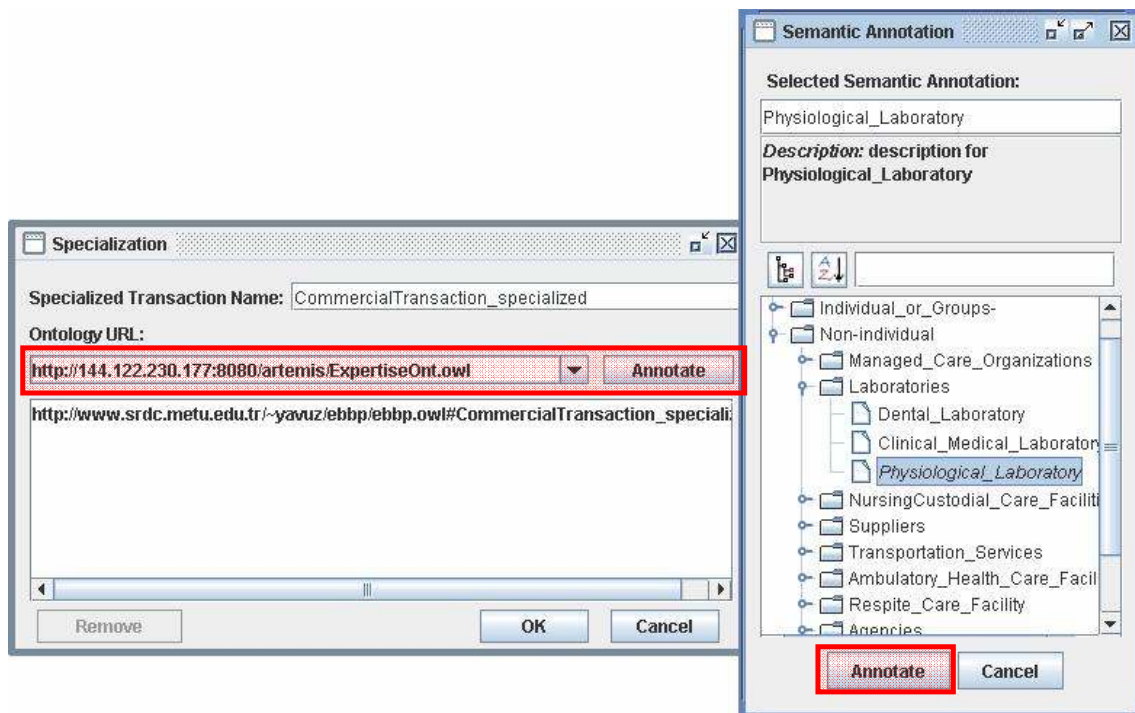




**Figure 28 Specialization Annotation**

The user should provide a name for the newly specialized Business Transaction. With this name a new subclass will be created in the dynamic ontology, and the transaction will be annotated with the Classification Node in the ebXML registry, that will be created dynamically. Note that, the user cannot remove this annotation from the list.

Note that, the user may annotate the transaction with a node within a domain-specific ontology as well, by entering the ontology URL into the field, and then pressing the Annotate button (see Figure 29). This will bring up the Annotation Window. The user can explore the ontology and select a node from the ontology, either using the tree representation or the alphabetically ordered list. Both representations have also a filter associated that filters the nodes beginning with the entered text (see Figure 29). When the user presses the Annotate button on this window (see Figure 29), the selected node is added to the annotation list. The user may remove this node later, after selecting the node and pressing the remove button.



**Figure 29 Domain-specific Annotation**

After adding all the nodes for annotation, the user should press the OK button to create the specialized Business Transaction (a template), and annotate it with the selected classification nodes in the ebXML registry (see Figure 30). If Cancel is pressed no action will be taken.



Figure 30 Annotations

The newly specialized Business Transaction can be seen in Figure 31.

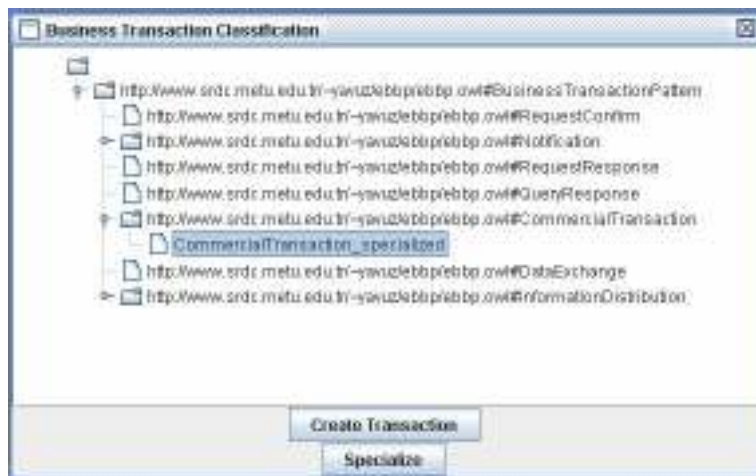


Figure 31 Specialized Commercial Transaction

To exemplify this specialization and annotation process, suppose the user wishes to specialize a Notification as a “Procedure Scheduled” transaction of IHE. The user may set the BusinessDocument in the DocumentEnvelope of the RequestingBusinessActivity to be sent by the requesting role to a proper value. After setting the domain specific fields, the user annotates the “Procedure Scheduled Notification”. The user may name this notification “IHE Procedure Scheduled Notification” and this procedure may be specialized under the “Scheduled Notification” node which could be a subclass of the “Notification” transaction. In this case, the ebBP (template) definition corresponding to “IHE Procedure Scheduled Notification” will first be annotated with [http://www.srdc.metu.edu.tr/ebbp.owl#IHE Procedure\\_Scheduled\\_Notification](http://www.srdc.metu.edu.tr/ebbp.owl#IHE_Procedure_Scheduled_Notification), which is created as a subclass of [http://www.srdc.metu.edu.tr/ebbp.owl#Scheduled\\_Notification](http://www.srdc.metu.edu.tr/ebbp.owl#Scheduled_Notification). In addition, the user may specify that this notification should also be annotated with [http://www.srdc.metu.edu.tr/ihe.owl#Procedure\\_Scheduled](http://www.srdc.metu.edu.tr/ihe.owl#Procedure_Scheduled). Then, the ebBP editor, will place

the ebBP (template) definition into the ebXML repository and annotate it with the classification nodes specified by the user.

### **3.18 *Creating a Business Transaction***

A Business Transaction can be created with two approaches: top-down, bottom-up. This is mainly due to the fact that the Business Transactions are not directly represented in the ebBP Visual Component. The Business Transactions should be part of a Business Transaction Activity. Therefore, the top-down approach is as follows: When adding a Business Transaction Activity (see Section 3.7), a user can provide the businessTransactionRef reference. This field is presented as a combo box, which presents the user with all Business Transactions created so far (as well as those in other Process Specifications with XInclude, see Section 3.22). The user can select (reuse) a previously created Business Transaction, or decide to create a new one, by pressing the Create button. When the user presses the Create button, a new XmlStylist window will be opened. The user can create a new Business Transaction with this instance of the XmlStylist.

The bottom-up approach is explained in the next section.

### **3.19 *Creating a Specialized Business Transaction***

The bottom-up approach is as follows: The user may drag and drop a Business Transaction or a specialized Business Transaction from the Business Transaction Classification Window. The user will be presented with an instance of the XmlStylist. The XmlStylist editor will come up empty for the top level Business Transactions (non-specialized), or some fields pre-filled for the specialized Business Transactions. The user is supposed to fill in at least the rest of the required fields and close the XmlStylist window. Upon closing the XmlStylist, a default Business Transaction Activity will be created, that references the newly created Business Transaction.

### **3.20 *Editing the Process Specification***

The user may edit the Process Specification from the Edit→Process Specification menu. The XmlStylist will come up with the Process Specification created so far. The user can make changes which will be saved after the XmlStylist is closed.

### **3.21 *Saving the Process Specification***

The Process Specification can be saved from the File→Save menu. A Save Dialog is opened to determine the File location and File name. If the Process Specification was previously saved it is directly overwritten.

### **3.22 *Importing another Process Specification (XInclude)***

The ebBP Editor also supports XInclude. The user can Import a file through the File→Import menu. An open dialog will come up to determine the file to be imported. After importing the file, XInclude structures can be used. However, this is not reflected to the user. XInclude operations are possible only where fields are represented with combo boxes. Selecting a value

existing in another file will result in the addition of the XInclude when the Process Specification is saved.

## **4 Conclusion**

This document provides a basic User Guide presenting the functionalities and capabilities of the ebBP Editor. Although the implementation provides a user interface for creating the Process Specification complying with the latest ebBP Specification, there are still some functionalities that need to be implemented. The editor also reflects some development based on ongoing research. In this respect, later versions may have extended functionalities, or changes in the implementation.