Technical University of Crete, Greece
School of Electronic and Computer Engineering

Transforming ASEME Roles Models to Process Models

Diploma Thesis

Nektarios D. Mitakidis

Supervisor: Assistant Professor Georgios Chalkiadakis

Jury:

Assistant Professor Georgios Chalkiadakis, School of ECE, TUC

Associate Professor Michail G. Lagoudakis, School of ECE, TUC

Dr. Nikolaos Spanoudakis, Laboratory Teaching assistant School of PEM, TUC

Chania, October 2014

Πολυτεχνείο Κρήτης
Σχολή Ηλεκτρονικών Μηχανικών και Μηχανικών Υπολογιστών

Μετασχηματίζοντας Μοντέλα Ρόλων της Μεθοδολογίας ASEME σε Μοντέλα Διαδικασιών

Διπλωματική Εργασία

Νεκτάριος Δ. Μητακίδης

Υπεύθυνος διπλωματικής: Επίκουρος Καθηγητής Γεώργιος Χαλκιαδάκης

Επιτροπή:

Επίκουρος Καθηγητής Γεώργιος Χαλκιαδάκης, Σχολή Η.Μ.Μ.Υ., Π.Κ.

Αναπληρωτής Καθηγητής Μιχαήλ Γ. Λαγουδάκης, Σχολή Η.Μ.Μ.Υ., Π.Κ.

Δρ. Νικόλαος Σπανουδάκης, Ε.ΔΙ.Π., Σχολή Μ.Π.Δ., Π.Κ.

Χανιά, Οκτώβριος 2014

ACKNOWLEDGMENTS

I would like to express my gratitude to both my supervisors, Professor Georgios Chalkiadakis and Dr. Nikos Spanoudakis firstly for accepting me into their research field of autonomous agents and secondly for their guidance. They were not only tutors and supervisors but also friends. Without their help this thesis would not be completed.

I would like to thank Professor Michail Lagoudakis for accepting to be at the jury committee and for teaching me as an undergraduate student the basics of languages and a lot on the theory of computation.

I want to thank everyone in the Technical University of Crete, professors, staff and fellow students, who despite the national economic crisis and the little recognition they keep working, in order to offer high quality of studies and great achievements in the research. I hope for a brighter future for the university and Greece.

Special thanks to my family, for their patience and their endless love.

I would like to dedicate this thesis to my friends. They are the way to happiness and the best psychological support one can have after the endless lonely hours of studying and working at the office and in the house. Friends gave me all the power and the courage to carry on.

To My Friends

ABSTRACT

In this thesis we present how an engineer can transform a Gaia or ASEME role to a process model, compliant with the XML Process Definition Language (XPDL) portable standard. XPDL is a format standardized by the Workflow Management Coalition (WfMC) to interchange business process definitions between different workflow products. XPDL is currently the best file format for exchanging BPMN diagrams, because it has been designed specifically to store all aspects of a BPMN diagram.

An ASEME model is the Systems-Role Model (SRM). The thesis offers a transformation of the Systems-Role Model (SRM) to the XML Process Definition Language (XPDL), which is the XML of the Business Process Modeling Notation (BPMN). The transformation is achieved through Java source code. A tool is developed for aiding the modeler in the transformation process. The tool uses a recursive algorithm for automating the transformation process and guides the user to integrate two or more agent roles in a process model. The tool usage is demonstrated through a running example. Moreover, simulations of the transformed roles in an open source process management tool are offered in order to display the effectiveness and usage of the derived process model. The work fully demonstrates the transformation, the risks and the future of this effort.

ΠΕΡΙΛΗΨΗ

Στη συγγεκριμένη διπλωματική παρουσιάζεται ο τρόπος με τον οποίο κάποιος μηχανικός μπορεί να μετασχηματίσει έναν Gaia ή ASEME ρόλο σε ένα μοντέλο διαδικασιών, το οποίο είναι συμβατό με την XML γλώσσα ορισμού διαδικασιών (XPDL) που είναι και φορητό πρότυπο. Η XPDL είναι μορφή που έχει τυποποιηθεί από τη Workflow Management Coalition (WfMC) για την ανταλλαγή ορισμών επιχειρηματικών διαδικασιών μεταξύ διαφορετικών ροϊκών προϊόντων. Η XPDL είναι αυτή τη στιγμή η καλύτερη μορφή αρχείων για την ανταλλαγή BPMN διαγραμμάτων, επειδή έχει σχεδιαστεί συγγεκριμένα για να αποθηκεύει όλα τα στοιχεία ενός BPMN διαγράμματος.

Ένα μοντέλο ASEME είναι και το Μοντέλο Ρόλων Συστήματος ΜΡΣ. Η διπλωματική εργασία προσφέρει έναν μετασχηματισμό από το ΜΡΣ στην XPDL, η οποία είναι η XML της Σημειογραφίας για τη Μοντελοποίηση Επιχειρηματικών Διαδικασιών. Ο μετασχηματισμός επιτυγχάνεται μέσω πηγαίου κώδικα Java. Ένα εργαλείο αναπτύσσεται το οποίο βοηθάει τον χρήστη που μοντελοποιεί στη διαδικασία μετασχηματισμού. Το εργαλείο χρησιμοποιεί έναν αναδρομικό αλγόριθμο για την αυτοματοποίηση της διαδικασίας μετασχηματισμού και καθοδηγεί το χρήστη για να ενώσει δύο ή περισσότερους ρόλους πρακτόρων. Η χρήση του εργαλείου παρουσιάζεται μέσω ενός παραδείγματος που εκτελείται. Επιπλέον, προσομοιώσεις των μετασχηματισμένων ρόλων σε ανοιχτού λογισμικού εργαλείο προσφέρονται για να αποδειχθεί η αποτελεσματικότητα και η χρήση του παραγόμενου μοντέλου διαδικασιών. Η διπλωματική εργασία πλήρως παρουσιάζει το μετασχηματισμό, τους κινδύνους και το μέλλον της προσπάθειας.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

Chapter 1
Introduction

From the beginning of history humans envisioned different things in order to create machines that are able to think rationally and help them in their everyday life. These dreams led later the humanity to start thinking about robots and artificial intelligence. Although there is a gap between the artificial intelligence and the way the Homo sapiens thinks, great scientific research efforts were made in order to achieve better technology. The computerized era is a reality and this thesis hopes for a small stone on the great wall to be added.

The artificial intelligence programs that are trained to take decisions in order to complete a task are called agents. Agents can be simple, as far as their source code is concerned, but they can also be complex and difficult to the understanding even to the most experienced. Additionally, agents are supposed to interact with other agents. Interactions, such as the cooperation or the antagonism or even the control of other agents are something common and usual and therefore multi-agent systems have become a reality.

However, multi-agent systems can become really complicated as the number of agents increases. The answer to this problem is the modular design approach. With model driven engineering a simpler, easier and sometimes even more comprehensive approach to software development is provided. In order to achieve model driven software engineering the need of model transformations during the different development phases is of utmost importance.

One model driven engineering methodology is ASEME ([1], [2]). ASEME is an Agent-Oriented Software Engineering (AOSE) methodology for developing multi-agent systems. It uses the Agent Modeling Language (AMOLA, [3]), which provides the syntax and semantics for creating models of multi–agent systems covering the analysis and design phases of a software development process. In this thesis, on one hand there is the transformation process of the AMOLA analysis phase Systems-Roles Model (SRM), on the other hand there is the XML Process Definition Language (XPDL). With the transformation the thesis aims to bring agent technology close to the world of business modeling. It bridges the gap between software engineers and the business world by allowing a Multi-Agent System (MAS) analysis model to be represented as a business process model. Thus, on one hand, the software engineer can employ available tools to validate specific properties of the modeled system even before its final implementation, and, on the other hand, a business collaborator can understand the system being modeled. A first approach has been made on the paper of Nikolaos Spanoudakis and Pavlos Delias, [4]: “Simulating Multi-agent System Designs Using Business Process Modeling”. However, their approach of transforming the SRM to the BPMN met the obstacle of differences in the BPM Notations of different widely used and known BPMN tools. The only thing the tools had in common was the XPDL standard and not always was this the case. In order to help the transportation of the transformed SRM models between the different BPMN tools it was decided that the SRM should be transformed into XPDL.

Following there are the thesis goals, the progression of the thesis and an outline of the document.

1.1 Thesis Goals

The major goal of this thesis is to present all the work completed in order to have a transformed SRM compliant to the XPDL standard. The higher goal, is that agent software developers use this thesis work, in order to immediately have the XPDL of their Multi-agent system. Software developers need only the substance of the transformation, which is how can someone from a SRM file will get the XPDL representation. The development of a transformation tool that will guide in simple steps is of utmost importance.

Rana and Stout [5] highlighted the importance of combining performance engineering with agent oriented design methodologies in order to develop large agent based applications. To derive process performance measures, we need a quantitative process analysis technique. Process simulation appears to be a prominent technique that allows us to derive such measures (e.g. cycle time) given data about the activities (e.g. processing times) and data about the resources involved in the process. Process simulation is a versatile technique supported by a range of process modeling and analysis tools [6]. However, to run a process simulation, the engineer needs a process model. And this is where the thesis comes to the spotlight. XPDL is a process model suitable for these simulations.

Last but not least this thesis, tries to project the importance of the ASEME methodology, as it was not only the foundation of this thesis but also to other model transformations as well ([1], [2]). ASEME is efficient for validating and simulating MAS designs, for example with the use of the Rhapsody tool [7], the MARKET-MINER agent, which is a real world system has been successfully implemented [8].

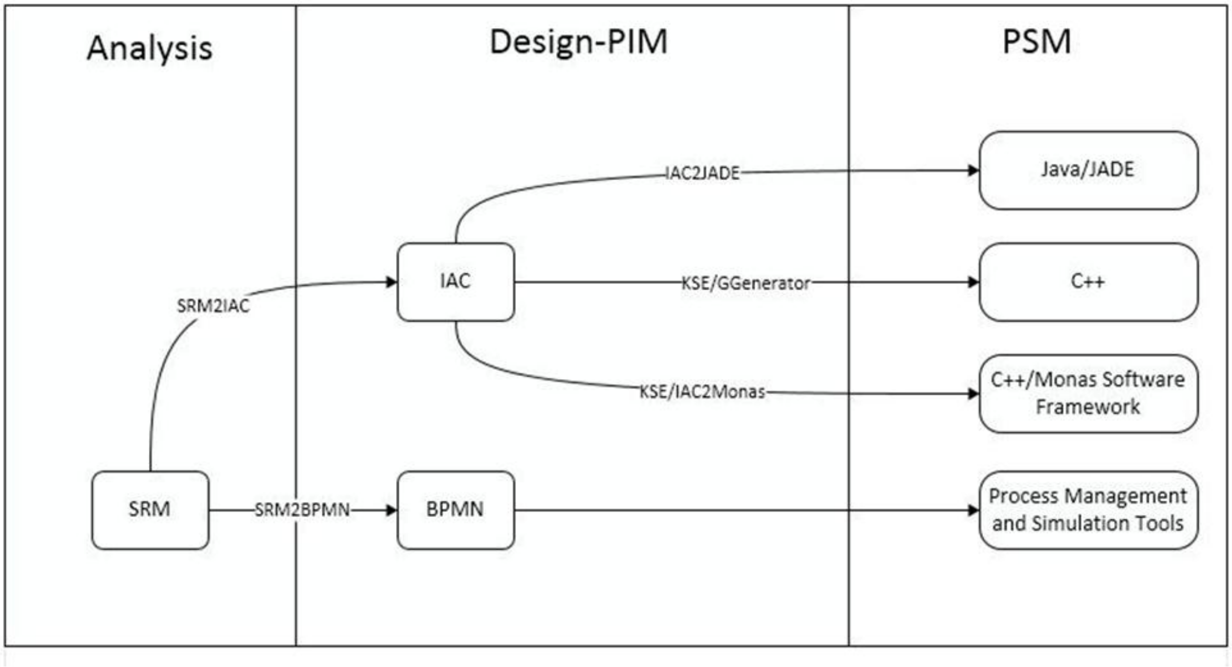


Figure 1:ASEME Process Tree from Analysis to Implementation

ASEME applies a model driven engineering approach to multi-agent systems development. It is compatible with the Model Driven Architecture (MDA) paradigm [9]. MDA's strong point is that it strives for portability, interoperability and reusability, three non-functional requirements that are deemed important for modern systems design. MDA defines three models:

- A computation independent model (CIM) is a view of a system that does not show details of the structure of the systems. It uses a vocabulary that is familiar to the practitioners of the domain in question as it is used for system specification.
- A platform independent model (PIM) is a view of a system that on one hand provides specific technical specification of the system, and on the other hand exhibits a specified degree of platform independence so as to be suitable for use with a number of different platforms. The system is described in platform independent format at the end of the design phase.
- A platform specific model (PSM) is a view of a system combining the specifications in the PIM with the details that specify how that system uses a particular type of platform.

Figure 1 presents how the MDA phases apply to ASEME. The ASEME Platform Independent Model (PIM), which is the output of the design phase, is a statechart that can be instantiated in a number of platforms using Process Management and Simulation tools and to an agent platform, the Java Agent Development Framework (JADE), or the C++ Monas Software Framework.

ASEME defines three levels of abstraction for each phase. The societal level, where the whole multi-agent system functionality is modeled. The agent level zooms in each part of the society. Finally, the details that compose each of the agents' parts are defined in the capability level.

1.2 Thesis Progression

This thesis starts at the end of the work of Nikolaos Spanoudakis and Pavlos Delias on the transformation of the Systems-Role Model to the Business Process Model Notation, [4]. The first attempt of this thesis was to transform the SRM to the BPMN 2.0. However, different BPMN 2.0 tools used different notations. It was important that the transformed models had to become portable between different tools. The only way to succeed in an acceptable transportation was through the XPDL.

And the route of the thesis changed. It now provides a complete XPDL schema from a simple liveness formula. Java was decided to be the programming language as there were open-source packages already available for the use of the different XPDL elements. The XPDL model is created through a recursion. The source code is compact and new aspects have been added such as the messaging interface.

Meanwhile, there was a necessity to find an open-source BPMN – XPDL tool that can provide an adequate simulator for the generated XPDL model. A critic is available on all the tested tools. The Signavio BPM Academic Initiative was used in order to simulate the generated XPDL model. The results of the simulation are quite interesting and are displayed for the reader to evaluate.

1.3 Document Outline

This thesis main contribution is the complete transformation of a liveness formula to a XPDL model. Also an effort is made to present the significant difficulties and also to project the complete methodology that led to the thesis completion.

In Chapter 2 the background that acts as a basis is set. It all starts with Model Driven Engineering, as it is expected, since the thesis tries to transform a model to another model. Then the definition of an API and a brief explanation of the Eclipse Modeling Framework are given. Then ASEME is presented and briefly described. The Systems Role Model (SRM), which is a basic milestone in the ASEME methodology, is later analytically outlined. On one hand there is the SRM and on the other there is the XML Process Definition Language, which is the XML Definition of the Business Process Modeling Notation. A complete listing and description of the BPMN elements is contained. XPDL needs the graphics of the BPMN in order to be represented. The previous work of the transformation of the SRM to the BPMN is put after the SRM and BPMN descriptions. Then the XPDL meta-model is presented. Chapter 2 ends with a listing of all the tools that offer the simulation option to the user, since simulations in process analysis are important and project the importance of the process development.

In Chapter 3 the XPDL importance is provided. XPDL is the basic format for interchanging information between different BPMN implementations of various tools. A review of different BPMN tools that the author tested is available.

Chapter 4 is the heart of the thesis. In the General Design Picture paragraph the complete design effort is portrayed. First, the SRM liveness formula grammar, in order to make clear the basic rules for the transformation. Second, the templates of the SRM to the XPDL transformation in order for the reader to distinguish how the specific part of the SRM is translated to the XPDL. The class diagram of the transformation effort explains how the different elements depend on each other and which class is responsible for which assignment. The recursive algorithm is available for a better comprehension of the effort. The importance of the package org.enhydra in this thesis is great and therefore a paragraph is offered to display which elements were necessary. After the design, the implementation of the transformation through Java Source code is offered. The basic element is the class Live2xpdl, which contains the methods for matching the SRM elements to their XPDL representations. If there is the need for a multi-agent system where the different roles have different responsibilities, then the Liveness2XPDL class comes into the spotlight which contains the methods for creating multi-agent XPDL. After, the classes Liveness2XPDLApp and the Inter_role_messages_definition are specified, which are graphical user interface classes that guide a user through all the procedure of the transformation. With Liveness2XPDLApp the user can create single or multi-agent systems from a liveness formula and with the Inter_role_messages_definition the user can create messages between the different roles.

Chapter 5 presents different examples of single and multi-agent XPDL files created as a result of the SRM2XPDL transformation. The efforts in importing this thesis created files are also presented. A simulation example is also available to project the importance of the work.

In Chapter 6 there is the conclusion of the thesis, the limitations, along with ideas for future research, a general evaluation of the diploma thesis and the contribution description of the thesis.

Chapter 2
Background

2.1 Model Driven Engineering

The Model Driven Engineering (MDE)[10] is a software development methodology, which becomes widely accepted in the software development field. Models are graphical representations of information that help the software developers program according to their design approach rather than the single dimensional source code programming.

Also, MDE focuses on creating and exploiting domain models, which are abstract representations of the knowledge and activities that govern a particular application domain, rather than on the computing (i.e. algorithmic) concepts. With MDE productivity is increased, because the compatibility between systems is maximized through the reuse of standardized models. Also the process of design becomes simpler because models of recurring design patterns are used. The communication between individuals and teams working on a system is more efficient, because of the standards of the terminology and also best practices are used in the application domain. A modeling paradigm for MDE is considered effective if its models make sense from the point of view of a user that is familiar with the domain and if they can serve as a basis for implementing systems. The models are developed through extensive communication among product managers, designers, developers and users of the application domain. As the models approach completion, they enable the development of software and systems.

Some of the better known initiatives are:

- The Object Management Group (OMG) initiative model-driven architecture (MDA), which is a registered trademark of OMG.(http://www.omg.org/)
- The Eclipse ecosystem of programming and modeling tools (Eclipse Modeling Framework).(http://www.eclipse.org/modeling/emf/)

2.2 EMF-API

An Application Programming Interface (API) is a programming language that allows two different applications to communicate with each other. With API features are enhanced and functionality is added either to one or to both applications. Its main purpose is to define a set of functionalities that are independent of their respective implementation, allowing both definition and implementation to vary without compromising each other. In most object-oriented languages, an object API is a prescription of how objects work in a language. In this thesis, the object-oriented language that will be used is JAVA. When related to a software framework, a framework can be based on several libraries implementing several APIs.

Eclipse Modeling Framework (EMF) is an Eclipse-based modeling framework and code generation facility for building tools and other applications based on a structured data model. From a model specification described in XMI[1], EMF provides tools and runtime support to produce a set of Java classes for the model, a set of adapter classes that enable viewing and command-based editing of the model, and a basic editor. Models can be specified using annotated Java[2], UML[3], XML[4] documents, or modeling tools, then imported into EMF. Most important of all, EMF provides the foundation for interoperability with other EMF-based tools and applications.

2.3 ASEME

According to [1], ASEME is an Agent-Oriented Software Engineering (AOSE) methodology for developing multi-agent systems. It applies a model driven engineering approach to multi-agent systems development, thus the models of the previous phase are transformed to models of the next phase. Different models are created for each development phase and the transition of one phase to another is assisted by automatic model transformation including model to model (M2M), text to model (T2M), and model to text (M2T) transformations leading from requirements to software development. In Figure 2 the whole ASEME MDE Process for Agent Development is described. In the beginning there is the System Actor Goal Model (SAGModel), which is an XMI model. Through the SAG2SUC transformation the System Use Case Model (SUC) is created. The developer can refine the SUCModel and then insert it in the SUC2SRM transformation, in order to get the SRMModelInitial. The System-Role Model (SRM) can be edited and then inserted to the SRM2IAC transformation in order to get the Intra-Agent Control (IAC) model that can be used

[1] The XML Metadata Interchange(XMI) is an Object Management Group (OMG) http://www.omg.org/ standard for exchanging metadata information via Extensible Markup Language (XML).

[2] Java is an object-oriented programming language

[3] UML is a modeling language in software engineering, which provides a standardized way of visualizing a design of a system.

[4] XML is a markup language that defines a set of rules for encoding documents in a format that is both readable by humans or machines.

to the IAC2JADE transformation in order to get the final .java file that describes the behavior of the agent.

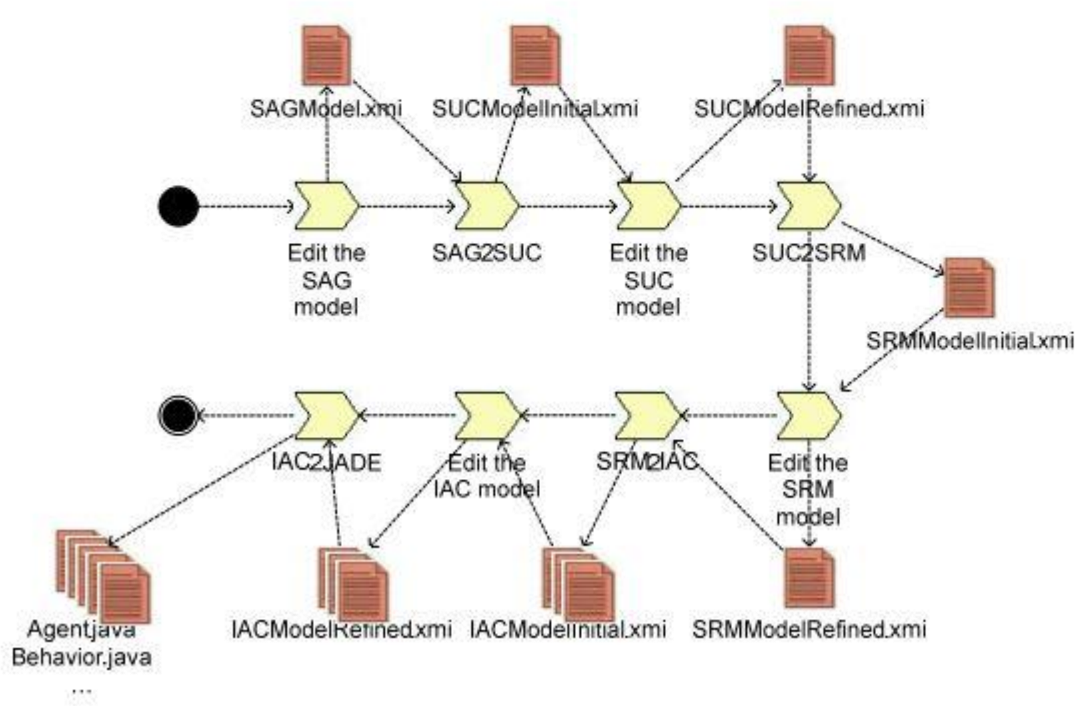


Figure 2: The ASEME MDE Process for Agent Development [1]

2.4 SRM2BPMN

2.4.1. The Systems Role Model (SRM)

The SRM is a model of AMOLA, [3]. In Figure 3, the metamodel of SRM is presented. An agent role aggregates capabilities and activities. Capabilities also aggregate activities. The liveness model has a formula at the first line (*root formula*) where activities or capabilities can be added. A *capability* must be decomposed to *activities* in a following formula.

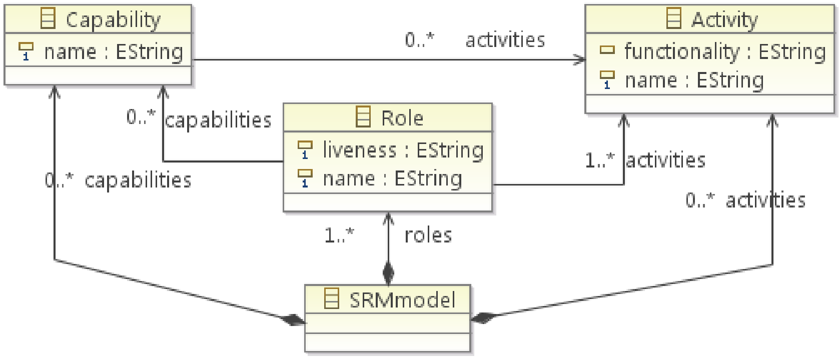


Figure 3: The ASEME Systems-Roles Model(SRM) metamodel

In the SRM, the Gaia operators are used ([1], [11]) for creating liveness formulas that define the dynamic aspect of the agent system. The Gaia operators are:

- A.B: means that activity B is executed after activity A,
- $A^{\sim}$: means that activity A is executed forever,
- A|B: means that either activity A or activity B is executed,
- A||B: means that activity A is executed in parallel with activity B,
- A+: means that activity A is executed one or more times,
- A*: means that activity A is executed zero or more times,
- [A]: means that activity A is optionally executed,
- $|A^{\sim}|^{n}$: means that activity A is executed forever n times parallel with A.

The liveness formula grammar is defined using the Extended Backus–Naur Form (EBNF), which is a metasyntax notation used to express context-free grammars EBNF was originally developed by Niklaus Wirth (1996). The EBNF syntax for the liveness formula is presented in Listing 2.1. With bold the gaia operators used are presented:

```
liveness       → {formula}
formula        → leftHandSide = expression
leftHandSide   → string
expression     → term | parallelExpr | orExpr | sequentialExpr
parallelExpr   → term || term { ||term }
orExpr         → term | term { |term }
sequentialExpr → term . term { . term}
term           → basicTerm | (expression) | [expression] | term* | term+ | term~
basicTerm      → string
```

Listing 2.1: The liveness formula grammar

2.4.2. Business Process Modeling Notation (BPMN)

Business Process Model and Notation (BPMN) is a standard for business process modeling that provides a graphical notation for specifying business processes in a Business Process Diagram (BPD), based on a flowcharting technique very similar to activity diagrams from Unified Modeling Language (UML)[5]. The objective of BPMN is to support business process management, for both technical users and business users, by providing a notation that is intuitive to business users, yet able to represent complex process semantics. The BPMN specification also provides a mapping between the graphics of the notation and the underlying constructs of execution languages.

The primary goal of BPMN is to provide a standard notation readily understandable by all business stakeholders. These include the business analysts who create and refine the processes, the technical developers responsible for implementing them, and the business managers who monitor and manage them. Consequently, BPMN serves as a common language, bridging the communication gap that frequently occurs between business process design and implementation.

BPMN is constrained to support only the concepts of modeling applicable to business processes. In addition, while BPMN shows the flow of data, and the association of data artifacts to activities, it is not a data flow diagram.

This thesis, uses the elements of the BPMN to demonstrate the graphic representation of XPDL.

[5] The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system.

It was created and developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software during 1994–95 with further development led by them through 1996.

In 1997 it was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2000 the Unified Modeling Language was also accepted by the International Organization for Standardization (ISO) as an approved ISO standard. Since then it has been periodically revised to cover the latest revision of UML.

ELEMENTS OF BPMN

BPMN models consist of simple diagrams constructed from a limited set of graphical elements. For both business users and developers, they simplify understanding business activities' flow and process.

BPMN's four basic element categories are:

- *Flow objects*
    - events
    - activities
    - gateways
- *Connecting objects*
    - sequence flow
    - message flow
    - association
- *Swim lanes*
    - pool
    - lane
- *Artifacts*
    - data object
    - group
    - annotation

These four categories enable creation of simple business process diagrams (BPDs). BPDs also permit making new types of flow object or artifact, to make the diagram more understandable.

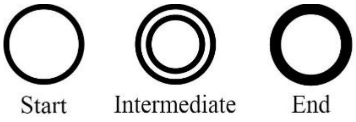


Figure 4: BPMN Events

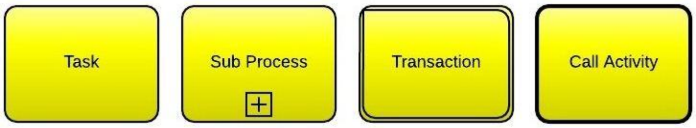


Figure 5: BPMN Activities

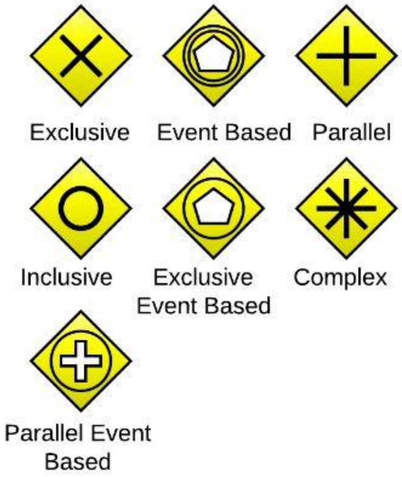


Figure 6: BPMN Gateways

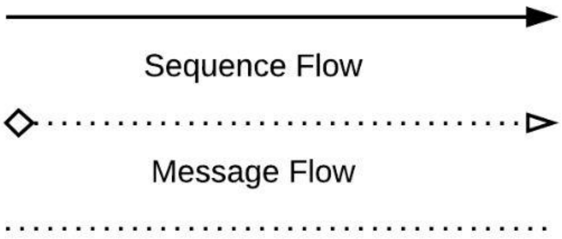


Figure 7: BPMN Connections

Flow objects are the main describing elements within BPMN, and consist of three core elements: events, activities, and gateways. In Figure 4 the BPMN Events are displayed.

Event

An Event is represented with a circle and denotes something that *happens* (compared with an activity, which is something that is *done*). Icons within the circle denote the type of event (e.g., an envelope representing a message, or a clock representing time). Events are also classified as **Catching** (for example, if catching an incoming message starts a process) or **Throwing** (such as throwing a completion message when a process ends).

Start event

Acts as a process trigger; indicated by a single narrow border, and can only be *Catch*, so is shown with an open (outline) icon.

Intermediate event

Represents something that happens between the start and end events; is indicated by a double border, and can *Throw* or *Catch* (using solid or open icons as appropriate). For example, a task could flow to an event that throws a message across to another pool, where a subsequent event waits to catch the response before continuing.

End event

Represents the result of a process; indicated by a single thick or bold border, and can only *Throw*, so is shown with a solid icon.

Activity

An activity is represented with a rounded-corner rectangle and describes the kind of work which must be done. In Figure 5 the BPMN activities are displayed.

Task

A task represents a single unit of work that is not or cannot be broken down to a further level of business process detail without diagramming the steps in a procedure (which is not the purpose of BPMN).

Sub-process

Used to hide or reveal additional levels of business process detail. When collapsed, a sub-process is indicated by a plus sign against the bottom line of the rectangle; when expanded, the rounded rectangle expands to show all flow objects, connecting objects, and artifacts. Has its own self-contained start and end events; sequence flows from the *parent* process must not cross the boundary.

Transaction

A form of sub-process in which all contained activities must be treated as a whole; i.e., they must all be completed to meet an objective, and if any one of them fails, they must all be compensated (undone). Transactions are differentiated from expanded sub-processes by being surrounded by a double border.

Call Activity

A point in the process where a global process or a global Task is reused. A call activity is differentiated from other activity types by a bolded border around the activity area.

Gateway

A gateway is represented with a diamond shape and determines forking and merging of paths, depending on the conditions expressed. In Figure 6 the BPMN Gateways are displayed.

Exclusive

Used to create alternative flows in a process. Because only one of the paths can be taken, it is called exclusive.

Event Based

The condition determining the path of a process is based on an evaluated event.

Parallel

Used to create parallel paths without evaluating any conditions.

Inclusive

Used to create alternative flows where all paths are evaluated.

Exclusive Event Based

An event is being evaluated to determine which of mutually exclusive paths will be taken.

Complex

Used to model complex synchronization behavior.

Parallel Event Based

Two parallel process are started based on an event, but there is no evaluation of the event.

Connections

Flow objects are connected to each other using **Connecting objects**, which are of three types: sequences, messages, and associations. In Figure 7 the BPMN Connections are displayed.

Sequence Flow

A Sequence Flow is represented with a solid line and arrowhead, and shows in which order the activities are performed. The *sequence flow* may also have a symbol at its start, a small diamond indicates one of a number of **conditional flow**s from an activity, while a diagonal slash indicates the **default flow** from a decision or activity with conditional flows.

Message Flow

A Message Flow is represented with a dashed line, an open circle at the start, and an open arrowhead at the end. It tells us what messages flow across organizational boundaries (i.e., between pools). A message flow can never be used to connect activities or events within the same pool.

Association

An Association is represented with a dotted line. It is used to associate an Artifact or text to a Flow Object, and can indicate some directionality using an open arrowhead (toward the artifact to represent a result, from the artifact to represent an input, and both to indicate it is read and updated). No directionality is used when the Artifact or text is associated with a sequence or message flow (as that flow already shows the direction).

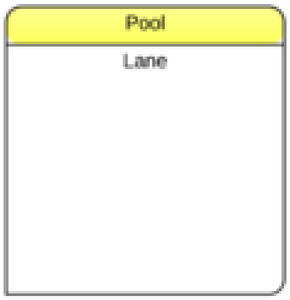


Figure 8: A BPMN Pool with a BPMN Lane inside

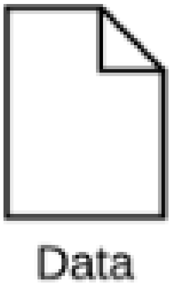


Figure 9: BPMN Data Object

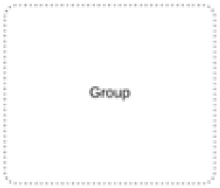


Figure 10: A BPMN Group

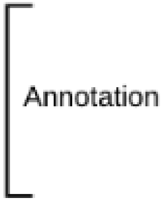


Figure 11: A BPMN Annotation

Swim Lanes

Swim lanes are a visual mechanism of organizing and categorizing activities, based on cross functional flowcharting, and in BPMN consist of two types. In Figure 8 a Pool with a Lane inside is displayed:

Pool

The Pool represents major participants in a process, typically separating different organizations. A pool contains one or more lanes (like a real swimming pool). A pool can be open (i.e., showing internal detail) when it is depicted as a large rectangle showing one or more lanes, or collapsed (i.e., hiding internal detail) when it is depicted as an empty rectangle stretching the width or height of the diagram.

Lane

The Lane is used to organize and categorize activities within a pool according to function or role, and depicted as a rectangle stretching the width or height of the pool. A lane contains the flow objects, connecting objects and artifacts.

Artifacts

The artifacts allow developers to bring some more information into the model/diagram. In this way the model/diagram becomes more readable. There are three pre-defined Artifacts and they are:

Data objects: Data objects show the reader which data is required or produced in an activity. A Data Object is displayed in Figure 9.

Group: A Group is represented with a rounded-corner rectangle and dashed lines. The group is used to group different activities but does not affect the flow in the diagram. A Group is displayed in Figure 10.

Annotation: An annotation is used to give the reader of the model/diagram an understandable impression. An annotation is displayed in Figure 11.

2.4.3. Transforming the SRM to the BPMN

Software Engineering (SE) and Business Process Management (BPM) are two disciplines with clear associations. A visible influence of SE to BPM concerns quality assessment, while SE aims its attention to BPM mainly to take advantage of its experiment design principles. For example, following the BPM paradigm, one can find solutions about how business people and software engineers are facilitated in communicating system requirements. Stakeholders are able to get involved in the system's design, and hence to assure the alignment of the produced software with the business objectives.

Simulation is employed to quantify the impact that a process design is likely to have on its performance, and to numerically indicate the best design alternatives.

Popular modeling languages in designing software systems, such as the object-oriented ones (e.g. UML), lack process views, an issue that has been early identified by [12]. On the other hand, process models do not usually map clearly to a programming environment. Both approaches have their relative advantages, so it is a hard decision to spare one. This is why there have been efforts to bridge object-oriented models and process models through model transformations ([13], [14]).

For transforming the SRM to a BPMN model [4], it is essential to transform the liveness formula to a valid BPMN graph. Delias and Spanoudakis defined the transformation templates shown in Figure 12 which are applied recursively to a Gaia formula from left to right. With these templates the transformation from the SRM liveness property to a BPMN model becomes a reality. The transformation is a text to model transformation (T2M) transformation that can be automated using existed techniques [15]

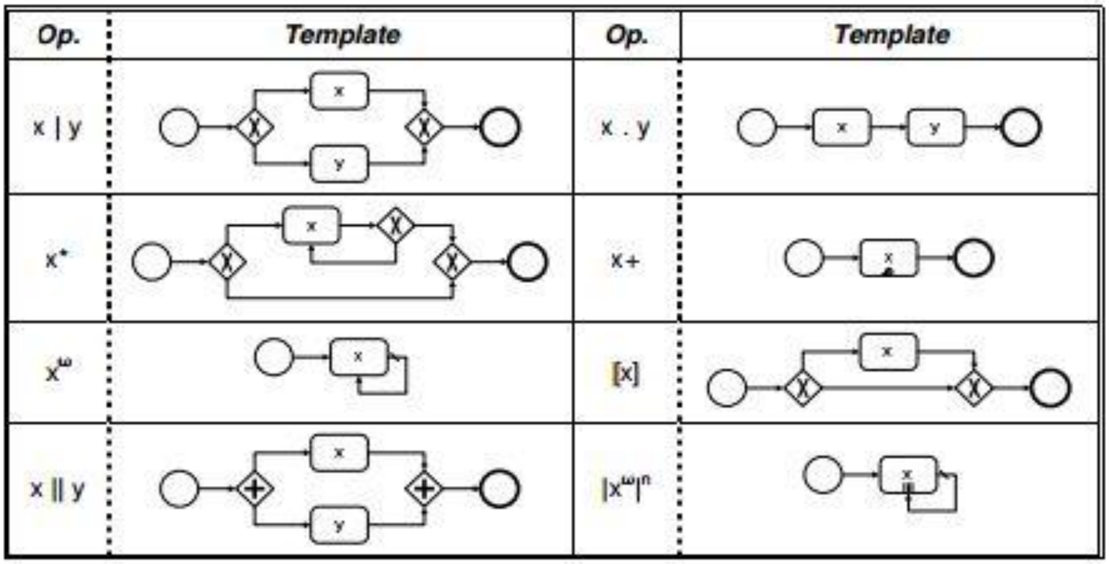


Figure 12: Templates of extended Gaia operators (Op.) for BPMN model generation [4]

After the process of transformation the software developer owns a BPMN model of the agent. This model can be used to simulate the system. For a single-agent system the transformed SRM can be used for simulation, verification and optimization. Considering a multi-agent system design, the individual process models must be combined into a functional ecosystem. This fact raises some additional transformation requirements.

In order to integrate multiple roles Delias and Spanoudakis set the followed requirements. A distinct participant (represented with a Pool in BPMN) was created for every role instance described in the SRM model. The pool derives from the outer level of the agents' process models (the sub-process element is transformed into a swimlane). This transformation brings also an additional action: the elimination of the outermost start and end events. However, in order to generate the society level, the major actions concern the messages' flows. In order to be more compliant with the business perspective of BPMN, the following rules are manually applied:

- All activities that stand for sending or receiving messages (the activities that the name starts with "send" or "receive" keywords) are labeled as message type activities.
- When a receive activity follows a start event, then the start event and the activity are merged into a start event triggered by a message.
- When a receive activity follows precedes an end event, then the two are merged into an end event triggered by a message.
- When a message can be sent to one or more out of many recipients, and this decision has to be evaluated during runtime, then before the "send message" activity a data-based exclusive gateway is added.

The last rule was introduced because BPMN does not provide a standard solution for this requirement. This deficiency is discussed in more detail in [16]. A BPMN modeling alternative, which responds to this requirement, is to use signal- broadcasting events. However, the later solution was not adopted since broadcasting does not rigorously match the message exchange logic.

2.5 The XPDL meta-model

Similarly to the definition of SRM, we use the Eclipse EMF technology to represent the metamodel for XPDL. The metamodel that we used for our project [17] is shown in Figure 13. The *Package* concept represents a set of processes and contains:

● *pools*, which represent major participant roles in a process, typically separating different organizations. A pool can contain:

○ *lanes*, which are used to organize and categorize activities within a pool ac-cording to function or role.

● *workflowProcesses*, which aggregate sets of activities and transitions

○ *activities* are represented by rounded rectangles and correspond to the execu-tion of a task or to the functionality of a gateway, which can be:

■ *XOR* gateway (one of the outgoing transitions will be followed), which is represented by a diamond shape with the “X” character in the middle

■ *parallel* gateway (all the outgoing transitions lead to activities that will be executed in parallel), which is represented by a diamond shape with the “+” character in the middle

○ *events* are represented by circles and are specific kinds of activities that cor-respond to something that happens. Common events are the start of a process lane and its ending

○ *transitions*, are represented with a solid line and arrowhead and have source and target (at the arrowhead) activities and define the control flow in the workflow process

● *associations*, are represented with a dotted line and arrowhead and have source and target (at the arrowhead) activities and define the message flow between different pools. Therefore, they also have source and target pools.

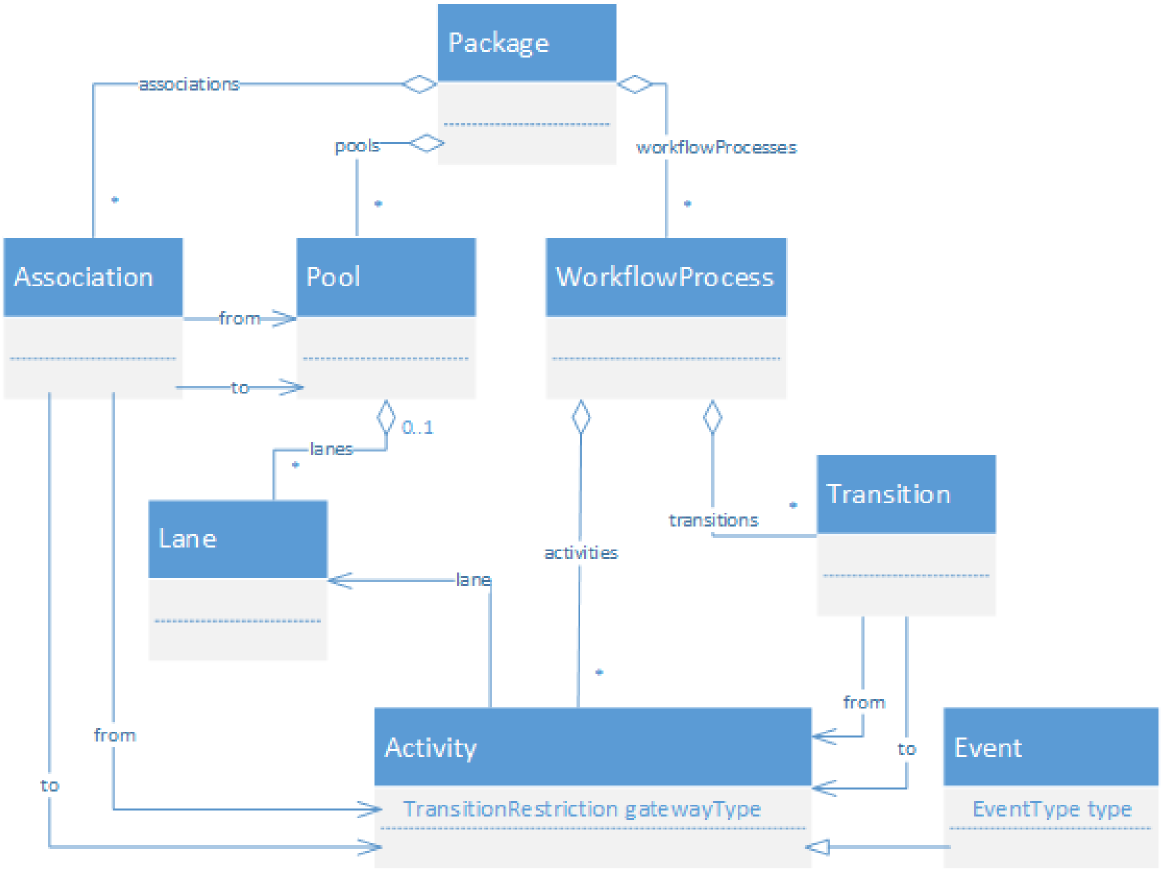


Figure 13: The thesis XPDL Meta-Model

2.6 Process Simulation Tools

A list of Business Process Modeling Notation Tools is offered:

(http://en.wikipedia.org/wiki/Comparison_of_Business_Process_Modeling_Notation_tools)

From the previous list the tools that offer simulation are presented below:

Table 1:List of BPMN tools that offer the simulation feature

Name	Platform/OS	XPDL Version	Software License
Activiti Modeler	Cross-platform	XPDL 2.1+	Apache License 2.0
ADONIS (Software)	Windows	XPDL 2.1+	Proprietary/Freeware
AuraPortal	Windows	XPDL 2.1+	Suite Proprietary or Cloud and Modeler Freeware
BPMN Visio Modeler	Windows	XPDL 2.1+	Proprietary, shareware, 1 month free trial
BPMN Web Modeler	Cloud	XPDL 2.1+	Proprietary, shareware, 1 month free trial
IBM Process Designer	eclipse based tool for creating executable processes	XPDL 2.1+	Proprietary
INNOVATOR for Business Analysts	Windows	XPDL 2.1+	Proprietary, free Personal Edition
Logizian	Windows, Linux, OS X, Solaris	XPDL 2.1+	Proprietary
Pega Systems	Windows	XPDL 2.1+	Proprietary
Process Modeler for Microsoft Visio	Windows	XPDL 2.1+	Proprietary
SemTalk	Windows SPoint Visio	XPDL 2.1+	Proprietary
Signavio Process Editor	Cloud or On-premise (Windows, Linux) server), Client-side browser	XPDL 2.1+	Proprietary
TIBCO ActiveMatrix	Linux, AIX, HP-UX, Solaris, Windows	XPDL 2.1+	Proprietary
Triaster	Windows		Proprieatary

From the above listing it is easy deducted that there are not so many tools that offer simulation. The most are designed for Windows. All but the last support the BPMN2.0 version. As it is deducted from the Listing only a few tools have a free License. These are Activity Modeler, ADONIS, AURAPortal, and Innovator with a free Personal Edition.

Chapter 3
Problem Statement

This thesis hopes to provide solutions to the following not solved yet problems. At first, the question about how can someone verify and validate a system analysis or a design. The only way to succeed this is by simulating experiments that resemble the reality objectives. Simulation is employed to quantify the impact that a process design is likely to have on its performance, and to numerically indicate the best design alternatives. Regarding business process simulation, various tools exist [18], which facilitate the adoption of business process modeling as a practical way for designing systems. However, a critical factor in selecting which tool is more appropriate is the modeling language used.

The modeling language that will be used should offer a practical way of communicating between the software developers and business analysts. On the one hand, agents, whose behavior is described in the SRM, are programs that complete specific tasks in a specific way in order to fullfil their purpose. On the other hand, there is the business process modeling, which represents processes of an enterprise, so that each process in a system may be analyzed and improved. The agent task and the business process have semantic resemblance, and therefore a way of representing both can be available, but is not offered.

Another issue that the chosen modeling language should overcome is that in modern society there are specific tasks completed by agents and specific tasks completed by humans. A logical question now rises, can the common displaying of the interactions between different kinds of roles whether they are human or artificial intelligence ones become possible. This can be solved, if the modeling language is simple but comprehensible by both kinds of roles.

The modeling language should be portable to agent platforms like WADE[6] in order for the agent developers to use the created models the moment they are created.

[6] http://jade.tilab.com/wadeproject/

The previous issues are solved by choosing the XPDL as the target transformation language for the SRM.

3.1 Problems in Simulations

At the beginning this thesis was supposed to offer a SRM to BPMN2.0 transformation, but since the procedure was to use liveness formula and generate XML elements, the thesis changed direction to transform SRM to XPDL. This was a correct decision since all the BPMN tools are based on the XML. It was of utmost importance that the generated files would be easily transported through different tools and there comes the XPDL into the account. A few tools were tested in order to take the former decision. The tools tested are presented below:

Table 2:List of tested tools

Name	Reviewer's month of testing	XPDLv2.0+
ADONIS (Software)	07-2014	✓
Bonita BPM	03-2014	✓
Signavio Process Editor	06-2014	✓
TIBCO ActiveMatrix	05-2014	✓

A small review is offered for each of the listed tools:

- **ADONIS:** is a Business Process Analysis (BPA) tool supporting business process management based on BPMS framework created at the University of Vienna. It is developed by the BOC Information Technologies Consulting GmbH. It allows business process modeling using BPMS notation and BPMN 2.0, process analysis, simulation, evaluation as well as publishing and process automation with BPMN 2.0 XML (BPMN DI) and XPDL. ADONIS is freeware that comes in handy for small enterprises. With ADONIS the user can model in a relative easy way and the models can be saved as HTML and can also be embedded in Word documents and presentations. As far as the analysis of the model is concerned ADONIS is great in finding bottlenecks or inefficiencies in the system. ADONIS offers also simulation which is not easily found on freeware BPM tools. Last but not least, with ADONIS the sharing of models becomes something trivial. The user can publish in HTML or also print in Word format.
- **Bonita BPM:** is an open-source business process management and workflow suite created in 2001. It was started in France National Institute for Research in Computer Science, and then had incubated several years inside of the French computer science company Groupe Bull. Since 2009, the development of Bonita is supported by a company dedicated to this activity: Bonitasoft. With Bonita BPM Studio the user can easily design the processes graphically. Bonita offers easy connectivity to IT systems by including a great amount of connectors, for example connectors for databases, ERP, CRM, ECM. Another advantage of Bonita is that through its portal the user can manage daily tasks and follow process

activity. Bonita’s engine is service-based and can support intensive workloads, is flexible and can be used to third party applications. Lastly, the Bonita BPM offers an open community edition.

- **Signavio ProcessEditor:** is a Berlin- and Silicon Valley-based software vendor in the Business Process Management (BPM) space. Its main product, the Signavio Process Editor is a web-based business process modeling tool, which was launched in May 2009. The product enables the creation of process diagrams using the Business Process Model and Notation and it is available as Software as a Service (SaaS) and for On-Premise installations. Signavio offers process modeling using graphical editor and QuickModel, and a spreadsheet-oriented process editing mode. There is also a simulation for process in order to identify bottlenecks or evaluating alternatives. With model-repository the user can build multi-level process architectures, can manage different versions and reuse the different objects. Sharing capability for process diagrams, for collecting feedback and enforcing approval workflows. The user can publish using reporting mechanisms and an integrated process portal. The cloud service is a multi-tenant installation that offers online workspaces to organizations with a subscription model. The Signavio Process Editor is also available for on-premise installations with a traditional license purchase and support/maintenance model.
- **TIBCO Active Matrix:** is a technology-neutral platform for composite business process management (BPM) and service-oriented architecture (SOA) applications. The platform includes products for service creation and integration, distributed service and data grids, packaged applications, BPM and governance. TIBCO is an On-premise BPM software solution. It is designed for communication between IT staff and business users. The user can easily create processes. TIBCO displays process performance via visual analytics. Also TIBCO can enforce compliance with company regulations. TIBCO has a proprietary software license.

Chapter 4
Our Approach

4.1. The General Design Picture

In order to succeed in the correct transformation of the SRM to the XPDL, some basics steps were followed. In the beginning the grammar of the SRM was defined. According to the grammar, the corresponding templates to the XPDL were carefully builded. The model that will be created is formed step by step recursively. Then a set of Java classes to create the XPDL files of different kinds of multi-agent systems were designed. Lastly, an open source package was found in order to acquire the XPDL elements descriptions in Java source code.

4.1.1. The Grammar of the SRM

The system roles model (SRM) is mainly inspired by the Gaia roles model (Wooldridge et al., 2000). A role model is defined for each agent role. The role model contains the following elements:

a) The interaction protocols that this agent will be able to participate in.
b) The liveness model that describes the role's behavior.

The liveness model has a formula at the first line (root formula) where activities or capabilities can be added. A capability must be decomposed to activities in a following formula. The Gaia operators have been enriched with a new operator, the $|x^{\sim}|^{n}$, with which a defined activity can be concurrently instantiated and executed more than one times (n times).

The liveness formula grammar has not been defined formally in the literature, thus it is defined here using the Extended Backus–Naur Form (EBNF), which is a metasyntax notation used to express context-free grammars. It is a formal way to describe computer programming languages and other formal languages. It is an extension of the basic Backus–Naur Form (BNF) metasyntax notation. EBNF was originally developed by Niklaus Wirth (1996). The EBNF syntax for the liveness formula is presented in Table 3, using the BNF style followed by Russel and Norvig(2003), i.e. terminal symbols are written in bold.

Table 3:The liveness formula of the SRM in EBNF Format

```
liveness                          → {formula}

formula                           → leftHandSide = expression

leftHandSide                      → string

expression                        → term

                                  |parallelExpression

                                  |orExpression

                                  |sequentialExpression

parallelExpression                → term||term{||term}

orExpression                      → term|term|term}

sequentialExpression              → term.term{.term}

term                              → basicTerm

                                  |(expression)

                                  |[expression]

                                  |term*

                                  |term+

                                  |term~

                                  ||basicTerm ~|number

basicTerm                         →string

number                            →digit | digit number

digit                             →1|2|3|…

string                            → letter|letter string

letter                            →a|b|c|…
```

Some examples of roles generated by the former grammar are listed below:

Liveness:

MeetingsManager = RequestNewMeeting~

RequestNewMeeting = ReceiveNewMessage.SelectMeetingDate.SendNewResponse

Liveness:

complex provider = |requestforservicesSP~|n

requestforservicesSP = receiverequestmessage.processrequest.sendresponsemessage

processrequest = (decideroutetype.requestforservicesSR.sortroutes) |

(decidePOItypes.request forservicesSR.decidePOIs.request for services SR)

requestforservicesSR = sendrequestmessage.receiveresponsemessage

Liveness:

PersonalAssistant = (ManageMeetings.LearnUserHabits)~||(NegotiateMeetingDate)~

ManageMeetings=GetUserRequest.(ReadSchedule|RequestChangeMeeting|

RequestNewMeeting).ShowResults

LearnUserHabits = LearnUserPreference.UpdateUserPreferences

RequestNewMeeting = SendNewRequest.ReceiveNewResults.UpdateSchedule

RequestChangeMeeting = SendChangeRequest.ReceiveChangeResults.UpdateSchedule

NegotiateMeetingDate = ReceiveProposedDate.

(DecideResponse.SendResults.ReceiveOutcome)+.UpdateSchedule

Figure 14: Examples of roles generated by SRM grammar

4.1.2. The templates of SRM2XPDL

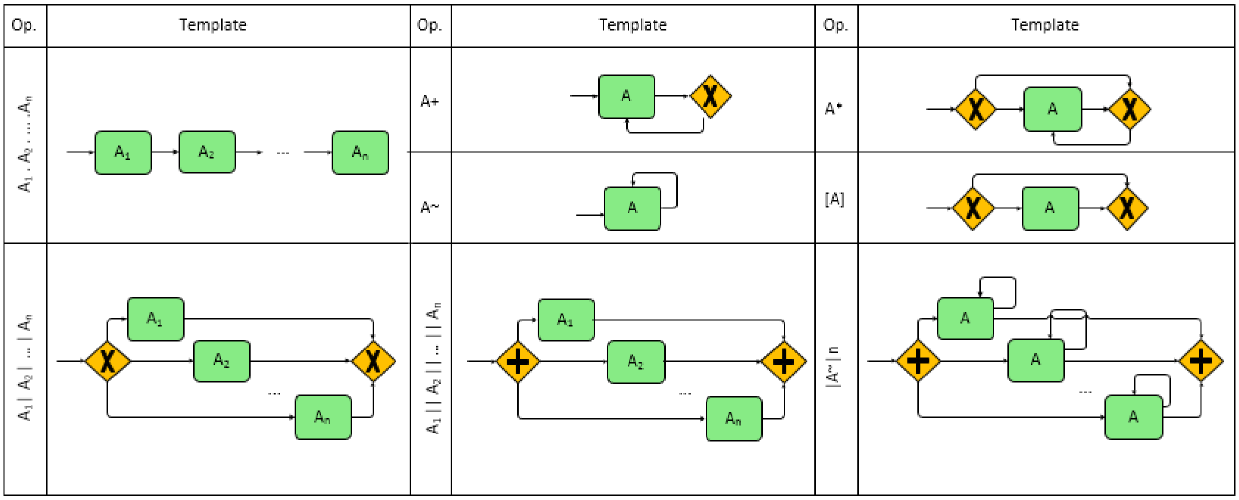


Figure 15: Templates of extended Gaia operators (Op.) for XPDL generation

4.1.3. Class Diagram of the SRM2XPDL Transformation.

In order for this thesis to present a sound transformation of the SRM to the XPDL some Java classes were developed. Below is the class diagram of these classes:

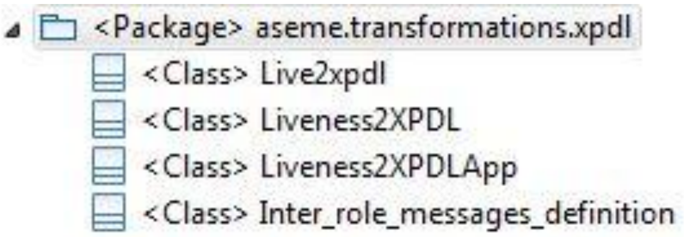

<Package> aseme.transformations.xpdl
<Class> Live2xpdl
<Class> Liveness2XPDL
<Class> Liveness2XPDLApp
<Class> Inter_role_messages_definition

Figure 16: The contents of Package aseme.transformations.xpdl

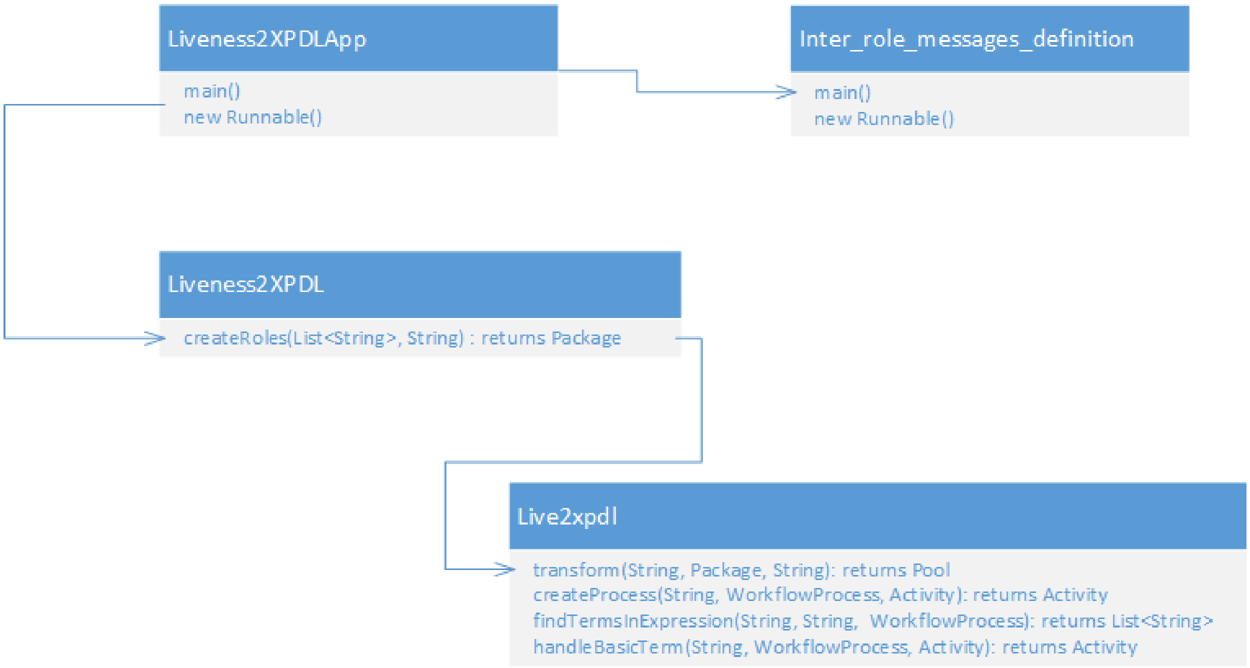

Liveness2XPDLApp
main()
new Runnable()
Inter_role_messages_definition
main()
new Runnable()
Liveness2XPDL
createRoles(List<String>, String) : returns Package
Live2xpdl
transform(String, Package, String): returns Pool
createProcess(String, WorkflowProcess, Activity): returns Activity
findTermsInExpression(String, String, WorkflowProcess): returns List<String>
handleBasicTerm(String, WorkflowProcess, Activity): returns Activity

Figure 17: The Class diagram for the SRM2XPDL transformation.

In just a few classes the transformation from SRM to the XPDL becomes a reality. The four classes are: *Liveness2XPDLApp*, *Inter_role_messages_definition*, *Liveness2XPDL*, *Live2xpdl*.

- **Liveness2XPDLApp**: This class creates a GUI that helps the user open SRM models, to write new formulas and to transform them into XPDL models. When the user decides to transform a liveness formula, the createRoles() method from Liveness2XPDL class is called. If there is a need to define messages between the roles in a multi-agent system, the main of the class Inter_role_messages_definition is called.
- **Liveness2XPDL:** This class contains the method createRoles(), where the basic XPDL elements are generated and stored in the XPDL package. The createRoles() method calls the transform() method from Live2Xpdl class in order to acquire all the pools of the package.
- **Live2XPDL:** This class is responsible for the creation of a pool, which represents a single role in the multi-agent system. The transform() method makes this possible, through the call of the createProcess() method. The createProcess() method is recursive and creates XPDL elements and connects them by matching terms in the liveness formula to their templates.
- **Inter_role_messages_definition:** This class creates a GUI that helps the user create message flows between the participant roles of the multi-agent system.

4.1.4. The recursive algorithm for creating pools.

The transformation algorithm uses elements from the liveness formulas grammar (Table 3), from the SRM metamodel (Figure 3) and the XPDL metamodel (Figure 13). It is a recursive algorithm that takes the liveness formula expression elements from left to right and applies the templates shown in Figure 13, gradually building the XPDL process. For all templates, the control flows from left to right, i.e. if a template follows another, then it is connected to its rightmost activity. Listing 4.2 presents the pseudocode of the tranformation algorithm. The different model elements are represented as classes and their properties as class properties, accessible using the dot operator, i.e. *<classname>.<property>*. For representing a list we use a *List* class that supports the operations *add* (to add an element to the list) and *size* (to return the number of its elements). The program takes as input an XPDL Package instance and the String liveness property of an SRM Role instance.

```
Program transform(String liveness, Package package)
      WorkflowProcess workflowProcess = new WorkflowProcess
      package.workflowProcesses.add(workflowProcess)
      Event startEvent = new Event
      startEvent.type = start
      workflowProcess.add(startEvent)
      Activity lastActivity = createProcess(liveness.formula1.expression,
      workflowProcess, startEvent)
      Event endEvent = new Event
      endEvent.type = end
      workflowProcess.add(endEvent)
      Transition transition = new Transition
      transition.from = lastActivity
      transition.to = endEvent
      workflowProcess.add(transition)
End Program
Function Activity createProcess(String expression, WorkflowProcess
workflowProcess, Activity activity)
      List terms = new List
            For Each $term_i$ In expression
                  terms.add($term_i$)
            End For
            If terms.size() > 1 Then
                  If expression Is sequentialExpr Then
                        For Each $term_i$ In expression
                              Activity newActivity = createProcess($term_i$,
                              workflowprocess, activity)
                              activity = newActivity
                        End for
                  Else If expression Is orExpr
                        Activity xorEntryGateway = new Activity
                        xorEntryGateway.gatewayType = XOR
                        workflowProcess.add(xorEntryGateway)
                        Transition transition = new Transition
                        transition.from = activity
                        transition.to = xorEntryGateway
                        workflowProcess.add(transition)
                        Activity xorExitGateway = new Activity
                        xorExitGateway.gatewayType = XOR
                        workflowProcess.add(xorExitGateway)
                        For Each $term_i$ In expression
                              Activity newActivity = createProcess($term_i$,
                              workflowprocess, xorEntryGateway)
                              transition = new Transition
                              transition.from = newActivity
                              transition.to = xorExitGateway
                              workflowProcess.add(transition)
                        End for
                        activity = xorExitGateway
                  Else If expression is parallelExpr
                        Activity parallelEntryGateWay = new Activity
                        parallelGateWay.gatewayType = Parallel
                        workflowProcess.add(parallelEntryGateway)
                        Transition transition = new Transition
                        transition.from = activity
                        transition.to = parallelEntryGateway
                        workflowProcess.add(transition)
                        Activity parallelExitGateway = new Activity
                        parallelExitGateway.gatewayType = Parallel
                        workflowProcess.add(parallelExitGateway)
                        For Each $term_i$ In expression
            Activity newActivity = createProcess($term_i$,
            workflowprocess,parallelEntryGateway)
            transition = new Transition
            transition.from = newActivity
            transition.to = xorExitGateway
            workflowProcess.add(transition)
        End For
        activity = parallelGateway
    End If
For Each $term_i$ In expression
    If $term_i$ Is basicTerm
        boolean foundLeftHandSideEqualsBasicTerm = false
        For Each $formula_i$ In liveness
            If $formula_i$.leftHandside = $term_i$ Then
                Activity newActivity =
                createProcess($formula_i$.expression,
                workflowprocess, activity)
                activity = newActivity
                foundLeftHandSideEqualsBasicTerm = true
            End If
            If foundLeftHandSideEqualsBasicTerm = false
                Activity newActivity = new Activity
                workflowProcess.add(newActivity)
                Transition transition = new Transition
                transition.from = activity
                transition.to = newActivity
                workflowProcess.add(transition)
                activity = newActivity
            End If
    Else If ($term_i$ is of type ‘(’ term ‘)’ ) Then
        Activity newActivity = createProcess(term,
        workflowprocess, activity)
        activity = newActivity
    Else If ($term_i$ is of type ‘[’ term ‘]’)Then
        Activity xorEntryGateway = new Activity
        xorEntryGateway.gatewayType = XOR
        workflowprocess.add(xorEntryGateway)
        Activity xorExitGateway = new Activity
        xorEntryGateway.gatewayType = XOR
        workflowprocess.add(xorEntryGateway)
        Transition transition = new Transition
        transition.from = activity
        transition.to = xorEntryGateway
        workflowprocess.add(transition)
        Activity newActivity = createProcess(term,
        workflowprocess, xorEntryGateway)
        Transition transition = new Transition
        transition.from = newActivity
        transition.to = xorExitGateway
        workflowprocess.add(transition)
        Transition transition = new Transition
        transition.from = xorEntryGateway
        transition.to = xorExitGateway
        activity = xorExitGateway
    Else If ($term_i$ is of type ‘*’) Then
        Activity xorEntryGateway = new Activity
```

```
                xorEntryGateway.gatewayType = XOR
                workflowprocess.add(xorEntryGateway)
                Activity xorExitGateway = new Activity
                xorEntryGateway.gatewayType = XOR
                workflowprocess.add(xorEntryGateway)
                Transition transition = new Transition
                transition.from = activity
                transition.to = xorEntryGateway
                workflowprocess.add(transition)
                Activity newActivity = createProcess(term,
                workflowprocess, xorEntryGateway)
                Transition transition = new Transition
                transition.from = newActivity
                transition.to = xorExitGateway
                workflowprocess.add(transition)
                Transition transition = new Transition
                transition.from = xorEntryGateway
                transition.to = xorExitGateway
                workflowprocess.add(transition)
                Transition transition = new Transition
                transition.from = xorExitGateway
                transition.to = startof(term)
                workflowprocess.add(transition)
                activity = xorExitGateway
            Else If ($term_i$ is of type ‘~’) Then
                Activity newActivity = createProcess(term,
                workflowprocess, activity)
                Transition transition = new Transition
                transition.from = newActivity
                transition.to = startof(term)
                workflowprocess.add(transition)
                activity = newActivity
            Else If ($term_i$ is of type ‘+’) Then
                Activity xorExitGateway = new Activity
                xorExitGateway.gatewayType = XOR
                workflowprocess.add(xorExitGateway)
                Activity newActivity = createProcess($term_i$,
                workflowprocess, activity)
                Transition transition = new Transition
                transition.from = newActivity
                transition.to = xorExitGateway
                workflowprocess.add(transition)
                Transition transition = new Transition
                transition.from = xorExitGateway
                transition.to = startof(term)
                workflowprocess.add(transition)
                activity = xorExitGateway
            End If
        End If
    End For
return activity
```

Figure 18: The recursive algorithm

The basic outline of the algorithm is:

- Lines 20-22: The findTermsInExpression() method is called to find the terms.
- Lines 24-29: A sequential expression is processed.
- Lines 30-49: An XOR expression is processed.
- Lines 50-70: A parallel expression is processed.
- Lines 72-90: A basic term is processed.
- Lines 74-81: The handle basic term method is described.
- Lines 91-94: A parenthesis term is processed.
- Lines 95-115: A brackets term is processed.
- Lines 116-142: A star term is processed.
- Lines 142-150: A tilda term is processed.
- Lines 150-164: A plus term is processed.

4.1.5. Class diagram of org.enhydra

The Together© Teamsolutions Co., Ltd. In Thailand has developed the org.enhydra and is distributed under the GNU Free Documentation License.

This thesis uses extensively the org.enhydra package in order to produce XPDL(v2.1) files. The complete class hierarchy is not described below, there are only packages present that have some impact on the transformation, the packages that are imported are underlined:

Hierarchy For All Packages

Package Hierarchies:

- org.enhydra.jxpdl,
- org.enhydra.jxpdl.elements,
- org.enhydra.jxpdl.utilities

Class Hierarchy

- java.lang.Object
    - org.enhydra.jxpdl.**XMLElement** (implements java.lang.Cloneable, java.io.Serializable)
        - org.enhydra.jxpdl.**XMLBaseForCollectionAndComplex**
            - org.enhydra.jxpdl.**XMLComplexElement**
                - org.enhydra.jxpdl.elements.**NodeGraphicsInfo**
                - org.enhydra.jxpdl.elements.**Package**
                - org.enhydra.jxpdl.elements.**TransitionRestriction**
                - org.enhydra.jxpdl.**XMLCollectionElement**
                    - org.enhydra.jxpdl.elements.**Activity**
                    - org.enhydra.jxpdl.elements.**Association**
                    - org.enhydra.jxpdl.elements.**Lane**

        - org.enhydra.jxpdl.elements.**Pool**
        - org.enhydra.jxpdl.elements.**Transition**
        - org.enhydra.jxpdl.elements.**WorkflowProcess**
    - org.enhydra.jxpdl.**XMLUtil**
    - org.enhydra.jxpdl.**XPDLRepositoryHandler**

A brief description of each package that is imported in the source code is available:

- org.enhydra.jxpdl.elements.**NodeGraphicsInfo**: Contains all the graphical information of the xpdl elements that need to be represented.
- org.enhydra.jxpdl.elements.**Package**: Contains everything a user needs in order to create a XPDL Package and to effectively use it. It refers to the Package presented in the XPDL metamodel.
- org.enhydra.jxpdl.elements.**TransitionRestriction**: Contains information about restrictions that the Transitions need to comply with in order to connect different kind of XPDL Elements.
- org.enhydra.jxpdl.elements.**Activity**: Contains everything a user needs in order to create an Activity and to effectively use it. It refers to the Activity presented in the XPDL metamodel.
- org.enhydra.jxpdl.elements.**Association**: Contains the information to create Associations between XPDL elements. It refers to the Association presented in the XPDL metamodel.
- org.enhydra.jxpdl.elements.**Lane**: Contains the information to create Lanes and to effectively use them. It refers to the Lane presented in the XPDL metamodel.
- org.enhydra.jxpdl.elements.**Pool**: Contains the information to create Pools and to effectively use them. It refers to the Pool presented in the XPDL metamodel.
- org.enhydra.jxpdl.elements.**Transition**: Contains the information to create Transitions between XPDL elements. It refers to the Transition presented in the XPDL metamodel.
- org.enhydra.jxpdl.elements.**WorkflowProcess**: Contains the information to create a WorkflowProcess and to effectively use it. It refers to the WorkflowProcess presented in the XPDL metamodel.
- org.enhydra.jxpdl.**XMLUtil**: Contains the information that is needed to describe a file as XML file.
- org.enhydra.jxpdl.**XPDLRepositoryHandler**: Contains the information to handle XPDL files in the repository system. It is used to write the package in a file as XPDL.

4.2 Implementation API

4.2.1 The class Live2xpdl

The class Live2xpdl contains the basic methods in order to transform a SRM liveness formula into a XPDL model. The SRM liveness formula corresponds to a single role in a system. The class Live2xpdl contains the following methods:

- ***Activity*** *createProcess(String expression, Workflowprocess workflowprocess, Activity actprevious):* The createProcess method is a recursive method responsible for filling the workflowProcess with XPDL elements. The process is completed by matching terms of the input expression, which is a liveness formula, to their respective templates. The method needs an activity as argument in order to store the activity before the call of the recursion. It returns the last activity to be used as a node for the continuation of the model. The createProcess method is described in lines 23-168 of the recursive algorithm.
- ***List<String>*** *findTermsInExpression(String expression, String connector, Workflowprocess workflowprocess):* This method is responsible for tokenizing a string expression in order to separate the terms between the connectors **(., |, ||)** and returns a list of string terms to be used by createProcess. The findTermsInExpression method is not analytically described, but it corresponds to the lines 20-22 of the recursive algorithm.
- ***Activity*** *handleBasicTerm(String term, Workflowprocess workflowprocess, Activity actprevious):* This method is used in order to substitute a term with a basic term so as for the recursion to find which is the current node. It returns an activity as the current node. The handleBasicTerm method is described in lines 74-81 of the recursive algorithm.
- ***String*** *preprocessing(String formula):* This method is used to substitute a $|x\sim|^n$ term to its semantic equal string, which is a parallel expression of x~ term n times with itself. It returns the semantic equal string.
- ***String*** *remover(String term):* The remover method was developed in order to remove invalid characters of a string that might be used as an id. In XPDL there can be no special characters or symbols in the ids. It returns the string without the removed elements.
- ***Pool*** *transform(String liveness, Package package, String outputfile):* The transform method is responsible for creating a lane to be stored in the XPDL package. It needs as arguments a String which should be a liveness formula of a role, the package of the XPDL and an output file for an optional write to file capability. It returns a pool.

A finding of this thesis is that all XPDL elements that will be created need unique XMI ids. Also the ids should not contain special characters or symbols. In order to avoid creating elements with the same id, the following convention was followed: Each time an element is created that needs id except of the package, the pools and the workflowprocess the time milliseconds of the program time run, the role that the element is created in and a counter that is increased, are put in this respective order in a string that will be the id of the created element. This way ids are created that are unique and are easily and effectively monitored in the debugging process.

4.2.2. The class Liveness2XPDL

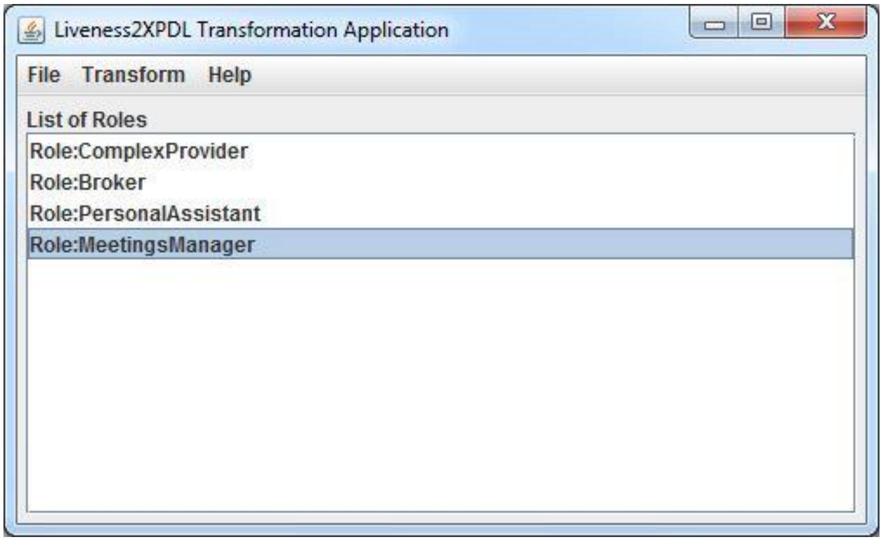
The class Liveness2XPDL is the API and contains the methods to generate a multi-agent XPDL model from some liveness formulas. The liveness formulas are added as lanes to the pool of the multi-agent system. The class Liveness2XPDL contains the following methods:

- ***Package*** *createRoles(List<String> roles, String filename):* The createRoles() is responsible for creating the XPDL package of the multi-agent XPDL and adding all the generated pools to it. Therefore it calls multiple times the Live2xpdl.transform() method. It returns the package.

4.3 GUI

4.3.1. The Liveness2XPDLApp class

This thesis offers a friendly user interface for opening an SRM model from an existing file, creating a liveness formula, and transforming one or more roles to a single or a multi-agent XPDL model. The Liveness2XPDLApp extends JFrame and implements MouseListener and ActionListener. Subsequently the procedure is presented:



Role:ComplexProvider
Role:Broker
Role:PersonalAssistant
Role:MeetingsManager

Figure 19: The Liveness2XPDL Transformation Application

The Liveness2XPDL Transformation Application contains three menus in the menu bar. The menus provide the following functionalities:

- **File**
    - Open SRM
    - Edit Gaia Formula
    - Delete Selected Role(s)
    - Exit
- **Transform**
    - Single Role Transformation
    - Multiple Role Transformation
- **Help**
    - About

Open a SRM Model:

To open a SRM model from the existing file, the user has to choose File then click Open SRM and find the location of the SRM in the file system.

Edit a formula:

To edit or write a new formula the user has to choose File then click Edit Formula and write the formula in the message box.

Delete formulas

To delete one or more roles that are presented in the list the user has to first select which roles to delete by ctrl + click and then choose File then click Delete Selected Role(s) or by pressing the delete button while the specific roles are selected.

Transform a single role model

To transform a single role model the user has to choose a role from the list of roles then click Transform then click Single Role Transformation.

Transforming a multi role model

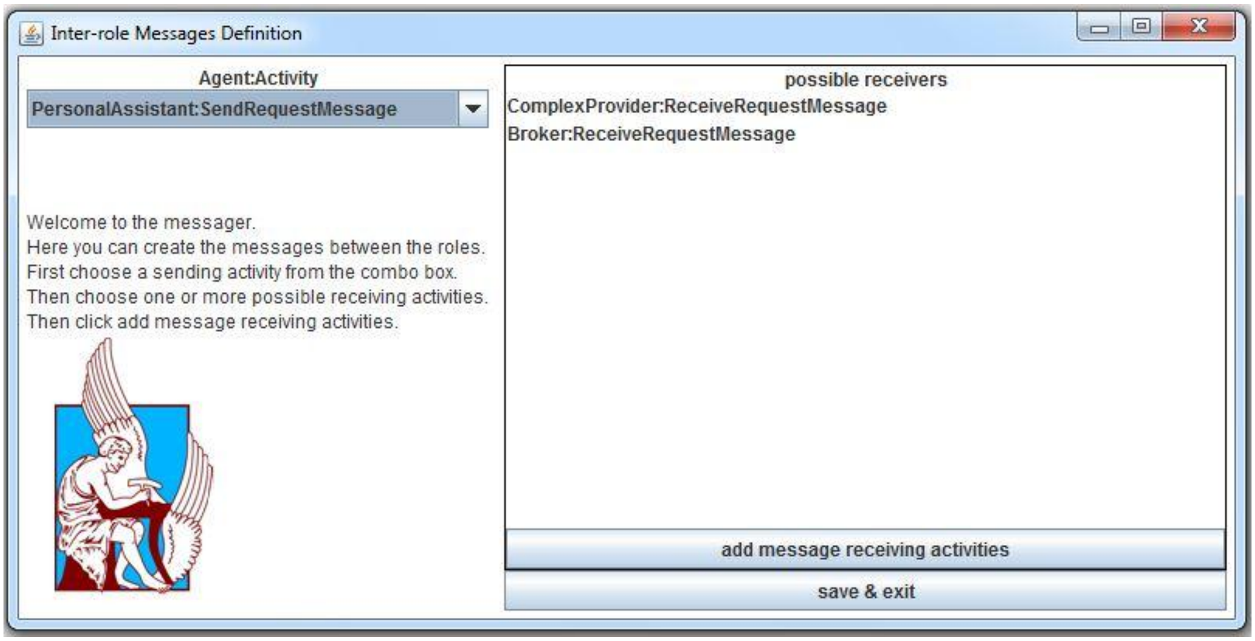
To transform a multi role model the user has to choose one or more roles from the list of roles then click Transform then click Multiple Role Transformation.

Help

If the user needs help the user needs to click Help then click About.

4.3.2. The Inter_role_messages_definition class

In some roles there might be activities that need the sending of a message and perhaps there might be activities in other roles to receive this message. If such is the case, a new GUI will appear. This GUI is described in the Inter_role_messages_definition class which extends JFrame. Below an example of this GUI is presented:



Inter-role Messages Definition

Agent:Activity

PersonalAssistant:SendRequestMessage

Welcome to the messager.
Here you can create the messages between the roles.
First choose a sending activity from the combo box.
Then choose one or more possible receiving activities.
Then click add message receiving activities.

possible receivers

ComplexProvider:ReceiveRequestMessage
Broker:ReceiveRequestMessage

Figure 20: The Inter-role Messages Definition

In the left side of the window there is a combo box where all the sending activities are shown.

When the user chooses a sending activity the possible receivers that will receive the same kind of message will appear on the right list.

Then the user just needs to choose one or more possible receivers and click add message receiving activities.

When the users finishes creating messages, on save & exit there is a save to file dialog in order to update the existing package.

CHAPTER 5
Results

For the display of the XPDL files the Together Workflow editor (v.4.5-1) was used. The Together Workflow Editor was developed by Together© Teamsolutions Co., Ltd. in Thailand. The Together Workflow Editor can be downloaded by http://sourceforge.net/projects/jawe/files/jawe/4.5-1/. In that link there is also the manual of the tool to help decide if the tool satisfies the user's needs.

5.1 Transforming a single Agent

For the completion of this thesis a lot of single liveness formulas where transformed into XPDL. Some examples are presented below:

- **The Meetings Manager**:

  Liveness formula:

MeetingsManager = RequestNewMeeting~

RequestNewMeeting = ReceiveNewMessage.SelectMeetingDate.SendNewResponse

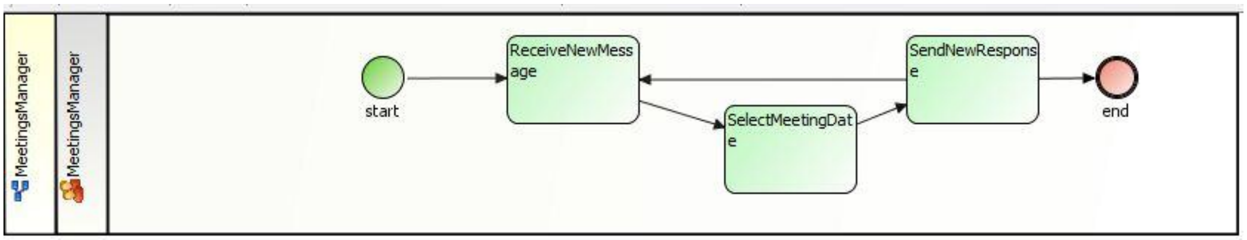


Figure 21: The Meetins Manager XPDL representation

The MeetingsManager contains only one activity, the RequestNewMeeting. RequestNewMeeting has a ~ operand which means that will be executed forever. RequestNewMeeting is then substituted with what it semantically means. RequestNewMeeting is a sequential expression between the activities ReceiveNewMessage which is followed by SelectMeetingDate and the SendNewResponse after.

- **The Complex Provider**
  Liveness formula:

complex provider = |requestforservicesSP~|n

requestforservicesSP = receiverequestmessage.processrequest.sendresponsemessage

processrequest = (decideroutetype.requestforservicesSR.sort routes)|(decidePOI types.request forservicesSR.decidePOIs.request for services SR)

requestforservicesSR = sendrequestmessage.receiveresponsemessage

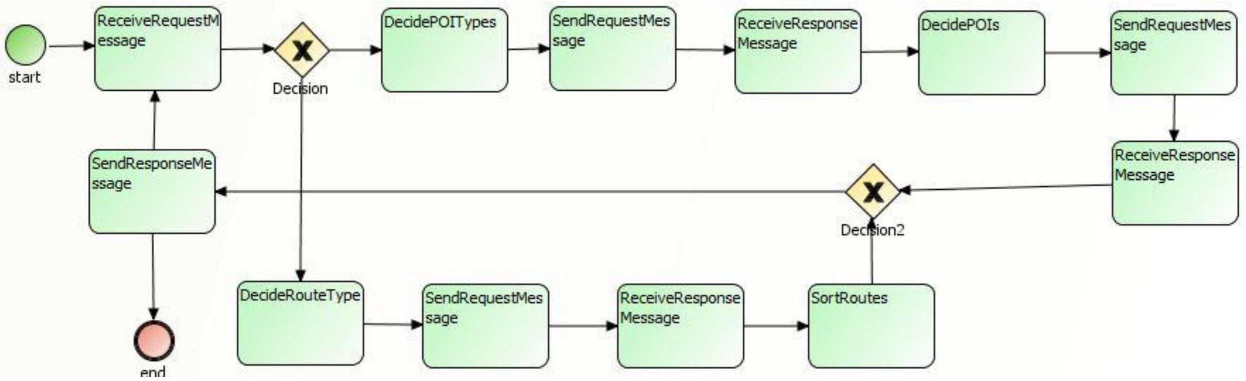


Figure 22: The Complex Provider XPDL representation

The Complex Provider runs the activity requestforservicesSP for ever.

The requestforservicesSP means to ReceiveRequestMessage then ProcessRequest and then SendResponseMessage.

The ProcessRequest means to either (DecideRouteType then requestforservicesSR then SortRoutes) or (DecidePOITypes then requestforservicesSR then DecidePOIS then requestforservicesSR)

The requestforservicesSR means to SendRequestMessage then to ReceiveResponseMessage.

- **The Broker**
  Liveness formula:
  Broker = |RequestforServicesSP~|n
  RequestforServicesSP=ReceiveRequestMessage.ProcessRequest.SendResponseMessage
  ProcessRequest = ServiceMatch.[(InvokeDataManagement | RequestForServicesSR)]
  RequestForServicesSR = SendRequestMessage.ReceiveResponseMessage

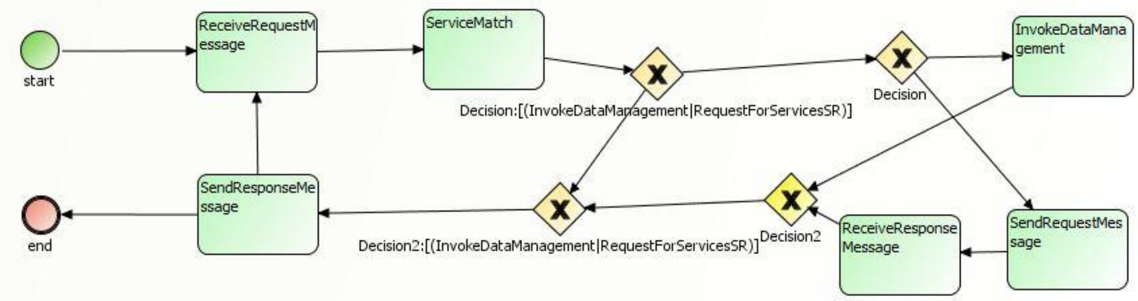


Figure 23: The Broker XPDL representation

The Broker runs the activity RequestForServicesSP for ever.

The RequestForServicesSP means to ReceiveRequestMessage then ProcessRequest and then SendResponseMessage

The ProcessRequest means first to ServiceMatch and then optionally InvokeDataManagement or RequestForServicesSR.

The RequestForServicesSR means to SendRequestMessage and then ReceiveResponseMessage.

- **The Personal Assistant**
  Liveness formula:
  PersonalAssistant = RequestForServicesSR
  RequestForServicesSR = SendRequestMessage.ReceiveResponseMessage

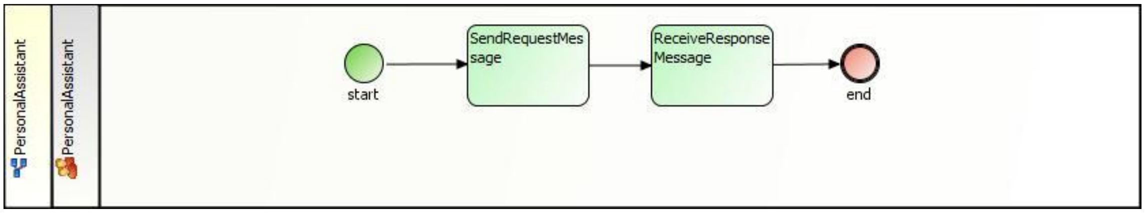


Figure 24: The Personal Assistant XPDL representation

The Personal Assistant is simple. This role just SendRequestMessage and then ReceiveResponseMessage.

5.2 Creating Processes from Multi-Agents.

One of the greatest challenges that this thesis needed to overcome was to create a XPDL that contains many agents and to represent the communication between them. With the use of the Liveness2XPDL Transformation Application and the Multiple Roles Transformation choice the multi-agent system between the Personal Assistant, the Broker and the Complex Provider of the previous paragraph becomes a XPDLv2.1 file ready to be imported to different BPMN tools.

The representation is displayed below. The associations that are represented depict the message flow between different activities of different roles and are created with the Inter-role Messages Definition GUI.

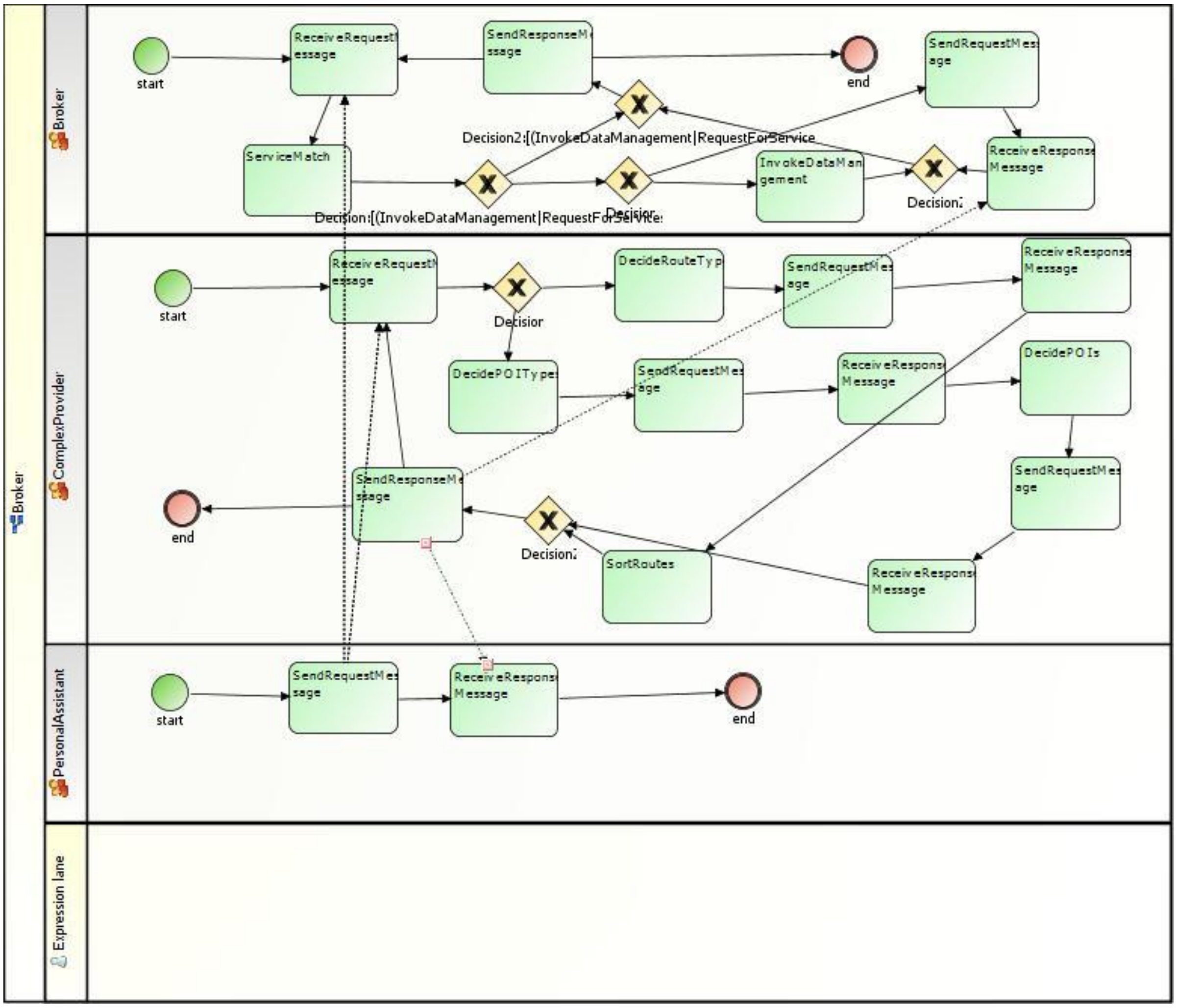


Figure 25: An example of a multiagent system's XPDL representation

5.3 Imports in different tools

As the thesis progressed different tools were tested in order to verify the xpdl portability and how different tools utilize the XPDL files. The Signavio BPM Academic Initiative and Adonis Community edition v3.0 had the most encouraging results. Signavio and Adonis have great research value, because they are free and also provide user-friendly simulators. Most BPMN tools trumpets that they encourage the transfer of models that were created in different tools, however not all the information is transferred and inconcistencies occur.

5.3.1 Importing xpdl files in Signavio

When importing the generated broker.xpdl to the Signavio BPM Academic Initiative directly through the import of XPDL2.1 choice in the menu bar the result after the import XPDL was disappointing. From the importer recognized the Pool, the Lanes and the Transitions. All the other elements were not recognized. The XPDL 2.1 importer is malfunctioning and can be deducted that is under development.

In order to get a better perspective an indirect method of import was adopted. First, the Broker.xpdl was online converted to Broker.bpmn.

After that the converted BPMN was imported to the Signavio BPM initiative and the result is given below:

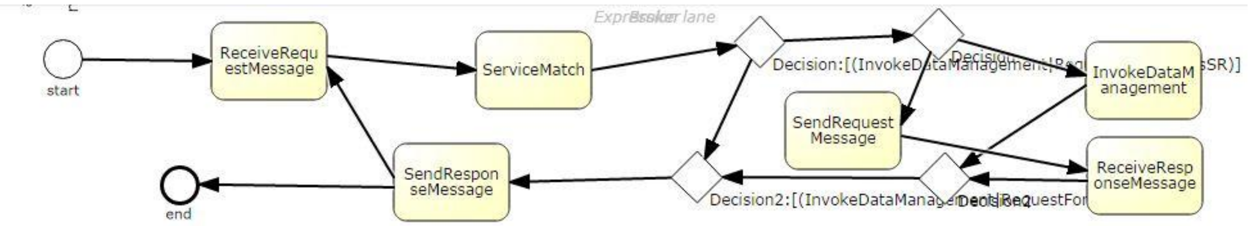


Figure 26: Converted BPMN imported to Signavio

The conversion to BPMN stored more information into the XML that was understandable by Signavio, however some apparent problems exist, for example in the top of the figure there is not clear distinguishing between the Pool and the Lanes.

5.3.2. Importing xpdl files to ADONIS Community Edition 3.0

When importing the generated broker.xpdl to the ADONIS Community Edition 3.0 directly through the import choice of menu there is a prompt to convert the imported xpdl file to an ADONIS accepted format. The conversion takes place online. But after the user presses OK a webpage appears informing the user that the XPDL converter is under construction. Since ADONIS is a free tool, malfunctioning is easily tolerated by the users. The author believes that the ADONIS online XPDL conversion will produce acceptable files.

So in order to get a better perspective the same indirect method was adopted. The same broker.bpmn file was imported into the ADONIS tool.

And when this XML is imported to ADONIS the result is:

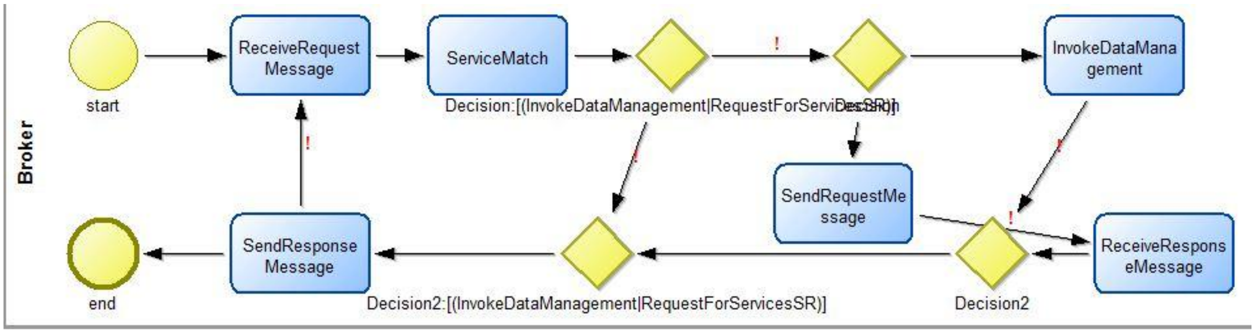


Figure 27: Converted BPMN imported to ADONIS

The ADONIS has a sufficient XML conversion, which keeps the information intact and therefore the resulting models are really close to the model before the conversions. The most apparent problem of the ADONIS is that it does not focus on the graphics infos of the Pools and the Lanes which results in a mess when the BPMN is depicted on the palette.

The former examples are results that illustrate, that if there will be a formal and widely accepted language in which all the information correctly is stored, then the different tools will have better results when getting imports from files that were generated in other tools. This is the problem for which the XPDL tries to offer a solution. After all, the same procedure is hidden. When a BPMN is about to travel between different tools, an XML conversion occurs and then from the XML conversion a new BPMN is created in order to get imported into a different tool.

5.4 Simulations

In this section, is demonstrated how simulation can aid the system modeler and project manager alike to make important decisions, mainly concerning non-functional requirements.

Firstly, the liveness formulas of the agents are defined, that when they are combined they will be elements of the system that will be simulated .

Role: PersonalAssistant

Liveness: PA = SendServiceRequest. ReceiveServiceResponse

Role: Broker

Liveness: Broker = ServicePAs || ServiceCP

ServicePAs = ReceiveServiceRequest. ProcessRequest. (InvokeDataManagement | SendComplexServiceRequest. ReceiveComplexServiceResponse). SendServiceResponse

ServiceCP = ReceiveSimpleServiceRequest. InvokeDataManagement. SendSimpleServiceResponse

Role: ComplexProvider

Liveness: CP = ReceiveComplexServiceRequest. DecideRouteType. SendSimpleServiceRequest. ReceiveSimpleServiceResponse. SortRoutes. SendComplexServiceResponse

Figure 28: The Agent roles that will be elements of the simulation

The XPDL of the multi-agent system was imported in Signavio BPM Academic in order to begin the simulation.

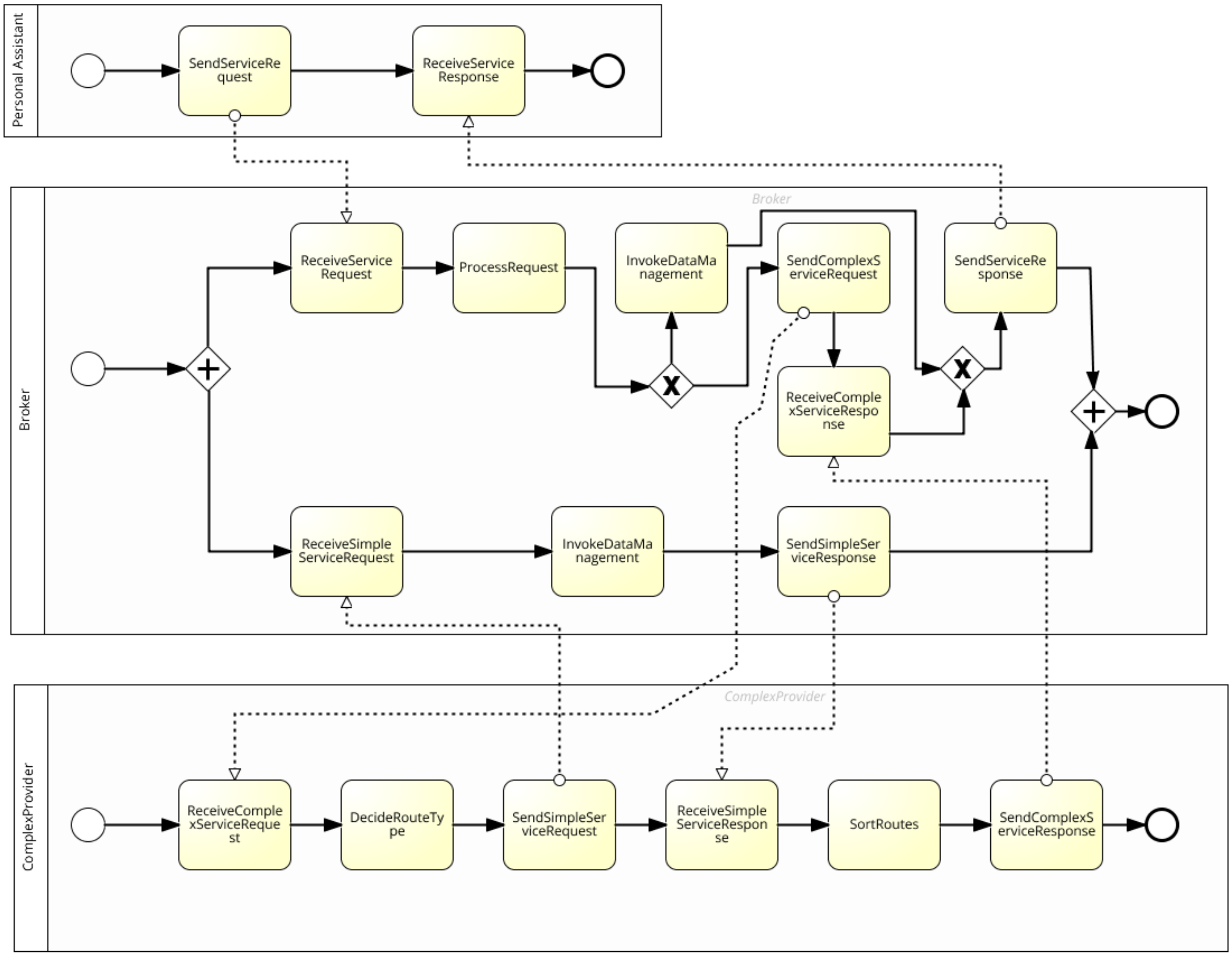


Figure 29: The multi-agent system of the simulation

In order for the reader to have a more specific view the settings are presented in table form:

Table 4:The settings of the simulation

ActivityName	Distribution	Mean	Standard Deviation	Performer
SendServiceRequest	Normal	0.024	0.063	PA
ReceiveServiceResponse	Normal	0	0	PA
ReceiveServiceRequest	Normal	0.002	0.002	BR
ProcessRequest	Normal	0254	0.112	BR
InvokeDataManagement	Normal	2.639	1.113	BR
SendComplexServiceRequest	Normal	0.007	0.006	BR
ReceiveComplexServiceRequest	Normal	0.024	0.063	BR
SendServiceResponse	Normal	0.002	0.002	BR
ReceiveSimpleServiceRequest	Normal	0.024	0.063	BR
InvokeDataManagement	Normal	2.639	1.113	BR
SendSimpleServiceResponse	Normal	0.007	0.006	BR
ReceiveComplexServiceResponse	Normal	0.024	0.063	CP
DecideRouteType	Normal	0.127	0.056	CP
SendSimpleServiceRequest	Normal	0.007	0.006	CP
ReceiveSimpleServiceResponse	Normal	0.024	0.063	CP
SortRoutes	Normal	0.127	0.056	CP
SendComplexServiceResponse	Normal	0.007	0.006	CP

Fixed Requests Every: 30, 20, 10, 5, 3, 2, 1, 0.5 seconds.

Gateway Probabilities: 50% - 50%

Number of Brokers: 1, 2, 3

Number of Complex Providers: 1, 2

The results after the simulation are presented below:

Table 5:Results of the simulation

Request Every	Number of Brokers	Number of Complex Providers	Complex Provider Utilization	Broker Utilization	Average Cycle Time	Max Cycle Time	Min Cycle Time
30	1	1	0.96%	15.87%	5.7	10.3	2
20	1	1	1.52%	21.33%	5.2	11.2	2.3
10	1	1	3.33%	44.64%	5.5	10.2	2.2
5	1	1	6.14%	84.54%	6.7	17.2	2.3
3	1	1	7.70%	99.49%	78	120	5.4
2	1	1	7.66%	99.54%	120	174	5.2
1	1	1	7.31%	99.53%	162	198	9.6

0.5	1	1	7.68%	99.53%	174	204	14.9
30	2	1	1.04%	7%	4	6.7	2
20	2	1	1.58%	10.70%	4.1	7.3	2
10	2	1	3.11%	21.53%	4	7.8	2
5	2	1	6.88%	42.54%	4.5	8.6	2.1
3	2	1	12.36%	64.08%	4.3	8.2	2
2	2	1	15.38%	97.15%	10.1	16.8	3
1	2	1	14.03%	98.20%	58.4	84	2.9
0.5	2	1	14.06%	98.93%	84	108	2.2
30	3	1	1.13%	5.66%	4.3	7.1	2.4
20	3	1	1.65%	6.96%	4.1	7.5	2
10	3	1	2.92%	13.14%	4	6.9	2.1
5	3	1	6.66%	29.29%	4.1	6.4	2.2
3	3	1	11.27%	50.77%	4.2	7.8	2.3
2	3	1	17.01%	68.95%	4.2	8.2	2
1	3	1	19.19%	96.07%	29.9	48.9	4.1
0.5	3	1	21.03%	97.31%	49	60	4.5
30	1	2	0.52%	14.81%	5.4	9	2.5
20	1	2	0.74%	22.46%	5.5	9	2.4
10	1	2	1.60%	42.01%	5.1	9	2.3
5	1	2	3.17%	78.10%	5.6	10.3	2.6
3	1	2	3.69%	98.07%	59	108	2.3
2	1	2	3.78%	99.54%	102	168	9.5
1	1	2	3.65%	99.53%	156	192	12.8
0.5	1	2	3.19%	99.57%	198	222	11.8
30	2	2	0.55%	7.69%	4.2	7.4	2.3
20	2	2	0.88%	10.21%	4.1	6.6	2.1
10	2	2	1.59%	23.28%	4.1	7.1	2.1
5	2	2	3.44%	38.93%	3.9	8.7	2.1
3	2	2	5.28%	67.26%	4.7	8.4	2
2	2	2	6.39%	97.74%	24.1	43.9	3.9
1	2	2	7.25%	98.71%	51	78	4.3
0.5	2	2	7.11%	98.51%	78	96	4.4
30	3	2	0.53%	5.02%	4.1	7	2.1
20	3	2	0.81%	7.08%	4.1	6.9	2
10	3	2	1.70%	15.55%	4.3	7.3	1.9
5	3	2	3%	31.49%	4.3	6.9	1.9
3	3	2	4.88%	51.39%	4.4	7.3	2.1
2	3	2	7.60%	66.31%	4.3	8.8	2
1	3	2	11.34%	97.10%	19.9	31.9	3.5
0.5	3	2	9.33%	96.01%	52.4	72	2.7

Initially, there were two reasons for simulating the ASK-IT system. The first was that the ASK-IT service providers needed to know if the system can satisfy non-functional user requirements, one of which was the delivery of the service within ten seconds. The frequency of service requests was calculated to be one request per 30 seconds. The second was to find out how would the system scale when service demand increased for use in preparing the project's exploitation plan.

The Signavio tool allows simulating a process model involving several roles. For each simulation scenario, it allows to define:

- available resources for each role (how many instances of this role are available)
- the frequency in which a role can appear and start executing
- the percentage of times that a XOR gateway selects one or the other execution path
- activity duration (distribution type, mean and standard deviation values)
- number of simulations for each scenario

For the simulations several executions of function prototypes were used to define the activities durations. Moreover, the network latency in the message receiving activities was added. All the distributions are normal, since it is the most commonly used distribution and there must be specific circumstances to use others. Then, different scenarios were defined by varying the frequency of PAs appearing in the network and asking for services, the number of brokers serving the requests and the number of complex providers. The experiments are presented in Figure 30. It is validated that the system with one broker and one complex provider can respond within 10 seconds in the worst case when there is an incoming request every 30 seconds. Moreover, we can see what the expected quality of service will be, as the requests frequency rises. As far as system scaling is concerned we see that by adding more broker instances, the system performance has a better gain than by adding complex providers. Finally, we can claim that with three broker instances the system can offer the same quality of service (respond within ten seconds) even if we have a request every two seconds.

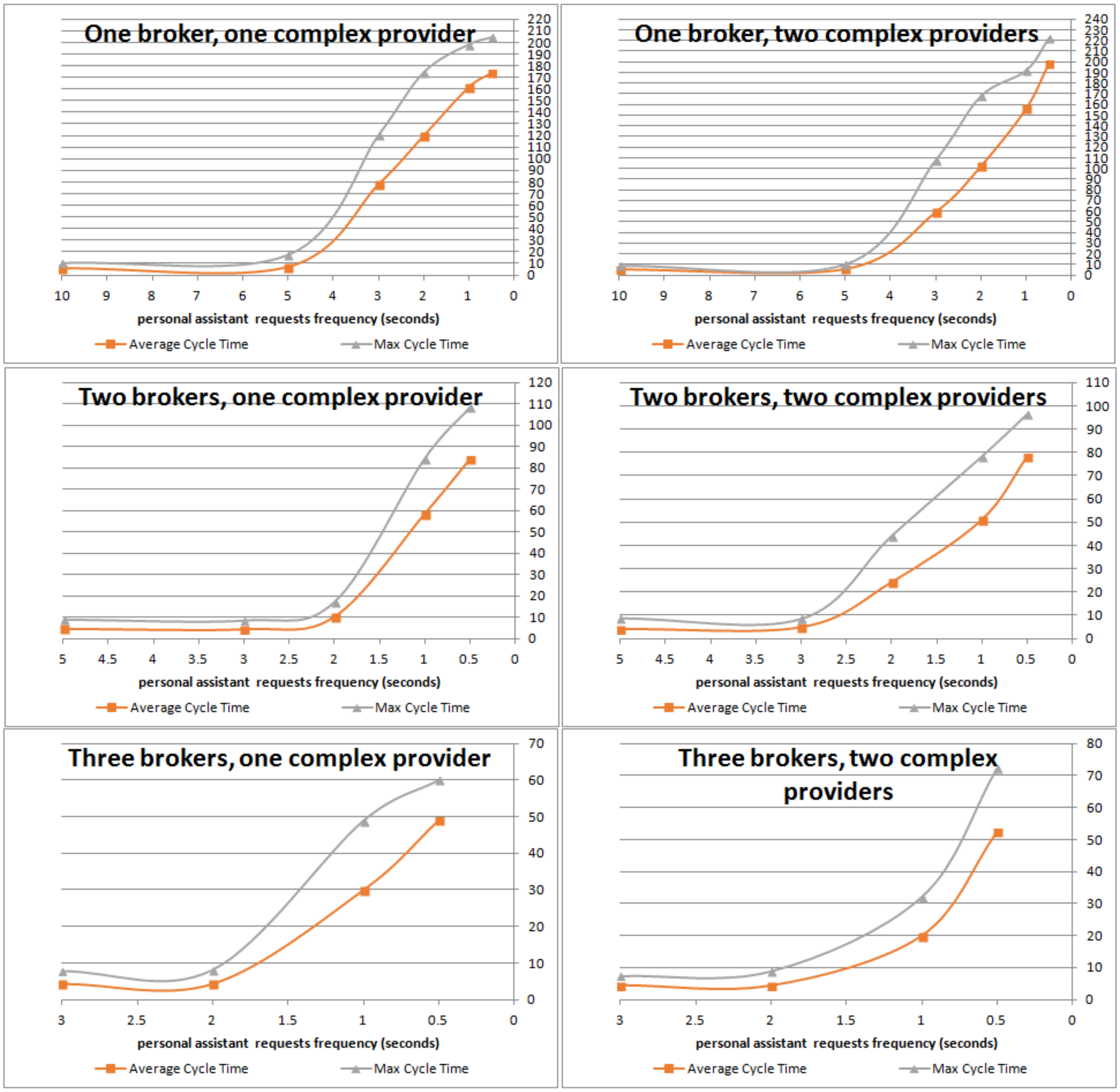


Figure 30: Average and maximum response times

Chapter 6
Conclusion

6.1 Discussion

In this thesis the whole transformation from SRM to XPDL is presented. In the beginning the ASEME and the BPMN background is set. The former work on the transformation of the SRM to the BPMN is the basis of this thesis. This basis is quite strong. Therefore, a new updated version of the ASEME process tree from the analysis to the implementation provides a more stable and more portable aspect in the whole after the SRM2XPDL transformation. The updated ASEME process tree is presented below:

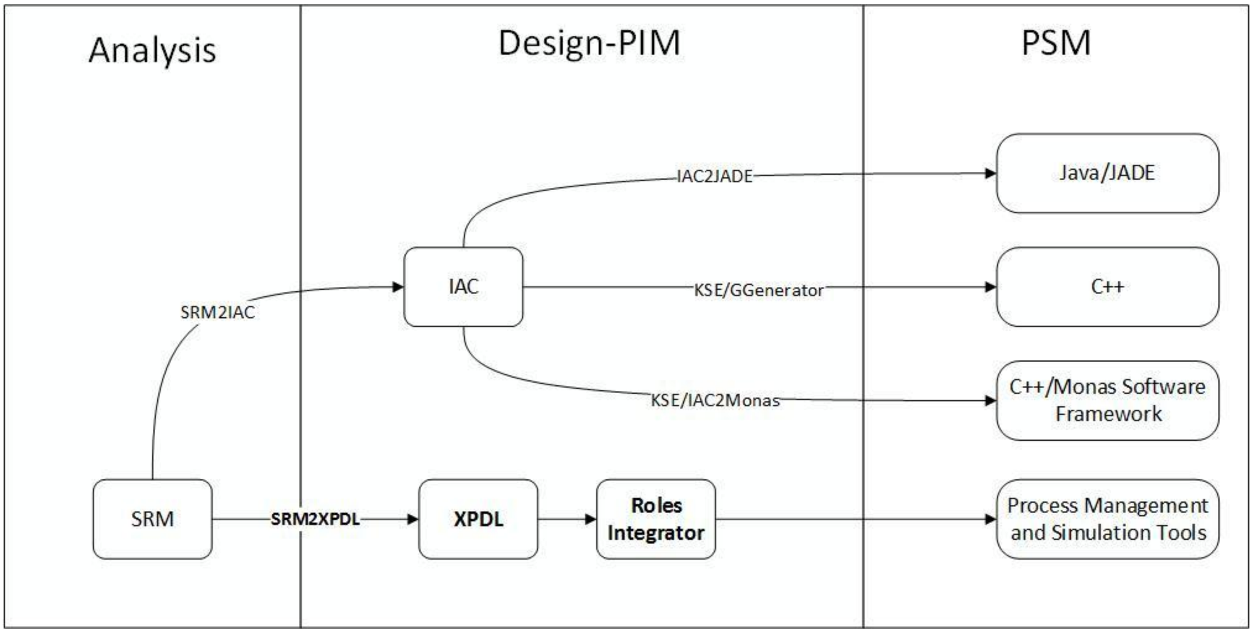


Figure 6.1 The updated ASEME Process tree from the analysis phase to implementation

There is also information on the different BPMN simulation tools that an interested user can use to the best of the needs. Some reviews on different tested tools are provided in order to help the reader avoid time and effort on trying to find the most suitable. BPMN is quite modern, powerful in the modeling information description and more and more business analysts and software developers begin to invest time and money. BPMN knowledge has become basic criterion in the industry and business environment.

The important part of the thesis is the transformation as a whole. Firstly, there is the Gaia formula grammar and from this the XPDL templates are created, opening the modeling perspective. Then the use of a complete and sound recursive algorithm for the creation of the process. The thesis reduction of the greater XPDL meta-model helps in succeeding in to the point transformation but also is useful in keeping the information minimal and without losing semantics.

The graphical user interfaces are basic aspects and help the user into distinguishing and better understanding the different parts of a transformation. Whether it is a single model transformation of one small agent role, or a transformation of a complex multi-agent system the user confronts only the important parts and is able to decide which information to store into the XPDL and which to discard.

Herein, the transformation algorithm is presented, the developed tool is demonstrated and is showed how it can be used to validate a system analysis for a real world application, which was created in the context of ASK-IT project. The code (and executable java .jar file) for the Liveness2XPDL tool is open and can be browsed by the interested reader at https://github.com/nektariosmitak/Liveness2XPDL.

The resulting XPDL files this thesis produces are important and can not be questioned, due to the liveness formulas mathematical and logic semantic match. The XPDL files created are without errors and therefore can become inputs to a big number of modern Process Simulation Tools that are based on the XPDL 2.1 for describing the BPMN 2.0.

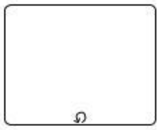
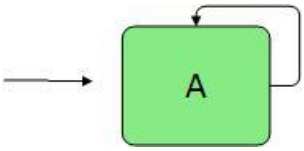
6.1.1. Limitations

This thesis met some major obstacles, some are partially overcome and some are not solved yet. Some limitations are presented below:

The messager issue

When the user decides to create multiple associations, that illustrate the message flows from an activity, which will be received by different activities in other pools there is not a way to distinguish which message flow was activated and what really the receiver has received. Therefore, the only way to distinguish is by creating associations between activities that have the same semantic in their name. For example the SendResponseMessageAbout activity should only send to the ReceiveResponseMessageAbout activity and not to multiple ReceiveResponseMessage activities.

The loop activity paradox

BPMN provides the user with the  activity that is executed many times until a specific requirement is met. However, in this thesis this kind of activity is represented by 

. There are two reasons that this is the case. This thesis recursion algorithm has as goal to produce XPDL files that do not contain subprocesses although this is supported in the BPMN. Through this assumption all the information of the liveness formula is clearly illustrated from the start event to the end event. The second, major reason is that different tools do not always recognize the BPMN representation. So the answer to this paradox was to keep the template simple and when a loop subprocess emerges, to substitute the A~ activity with the whole equivalent and send the loop Transition to the beginning of the equivalent.

The Gate or No Gate dilemma

In BPMN it is acceptable to create more transitions from an activity pointing to other activities. This rule reduces the complexity of the model as it is not mandatory to use XOR gates in order to represent this. However, the BPMN compilers do not accept this rule and most of the times they suggest that a gateway should be placed to avoid errors. Therefore the templates contain gateways to resolve this issue.

6.2 Future Work

This thesis hopes to add a small stone in the great wall of model driven engineering. The major problems encountered as well as the solutions that can be deduced by this work can produce new research material.

For example, one of the major obstacles that this thesis more than once encountered was the small number of tools that are available to the community. The majority of tools offer process simulators only in their commercial edition. Also most of the tools were developed by student teams that were later employed by or started a company. What is easily deducted is that perhaps, in the future a free process simulator can be made available due to the efforts of a brave team of post-doc or master thesis students.

Another important example that can be offered as a diploma thesis part to a student is that this thesis liveness formulas inputs are not checked for their correctness and therefore if the regex matcher can not make match it simply discards the formula. Therefore a formula ‘compiler’ would provide to the user the power to create correct liveness formulas and as a result different code and model transformations.

There is also the idea of an online SRM2XPDL transformer but also an online XPDL2BPMN, or perhaps a BPMN2XPDL transformer offered by the TUC community. These are quite challenging tasks that can make a team of electronic engineering students understand the depth of the subject that they study.

It is of utmost importance that model driven engineering becomes a widely accepted way of software development, so the syntax and semantics correctness of models are necessities. Models are graphical representations and simulations of the information and are usually the best way to prove the results.

ASEME has a strong basis and can support also other model to model, text to model, model to text transformations as long as there are sufficient and correct model specifications. Perhaps, it can provide an answer to a network model or a software model, perhaps the transformation into another model might help the research on even another extent.

6.3 Lessons Learned

This thesis not only is the presentation of the SRM2XPDL transformation, but also became the reason for a series and valuable life lessons. First, the most important part when someone has to complete a big in amount and difficult task is to be cooperative and listen. One should not try to overcome every obstacle alone, walking in subjective paths and perhaps even repeating the same mistakes over and over again. The guidance and knowledge of advisors can surprise and even retract treasures that stay in obscurity or vagueness.

Diploma thesis are a contract of trust first with one’s self and second with people that will refer to it. The procedure teaches to trust and be trusted. Also, when someone really gives at the little things the importance that the little things require a great outcome will eventually become a reality sooner or later. In fact, at the beginning of the thesis the author believed that this thesis would not even be successful in representing the amount of work or even the amount of work of people before it. The key to overcome this pessimistic fallacy was to love each part of the work.

At the beginning of the thesis the author believed that the only way to program was the hardcore way of just programming. At the end, the model driven engineering dynamic has made an impact and the ground shaking belief was established that in the future the best way to program is by creating models that will create models and so on.

Finally, the greatest lesson is that without personal sacrifice nothing of value can be achieved.

Annex 1.

References

[1] N. Spanoudakis, “The Agent Systems Engineering Methodology (ASEME),” Paris Descartes University, 2009.

[2] N. Spanoudakis and P. Moraitis, “Using ASEME Methodology for Model-Driven Agent Systems Development,” in *Agent-Oriented Software Engineering XI*, vol. 6788, D. Weyns and M.-P. Gleizes, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 106–127.

[3] N. Spanoudakis and P. Moraitis, “The agent modeling language AMOLA,” in *Dochev, D., Pistore, M., Traverso, P. (eds.) AIMSA. Lecture Notes in Computer Science*, Springer, 2008, pp. 32–44.

[4] P. Delias and N. Spanoudakis, “Simulating Multi-agent System Designs Using Business Process Modeling.,” in *Proceedings of the 8th European Workshop on Multi-Agent Systems (EUMAS 2010), Paris, France, December 16-17*, 2010.

[5] O. F. Rana and K. Stout, “What is scalability in multi-agent systems?,” in *International Conference on Autonomous Agents*, 2000, pp. 56–63 ACM.

[6] M. Dumas, M. La Rosa, J. Mendling, and H. Reijers, *Fundamentals of Business Process Management*. Berlin, Heidelberg, 2013.

[7] D. Harel and H. Kugler, “The RHAPSODY Semantics of Statecharts (Or on the Executable Core of the UML),” in *Integration of Software Specification Techniques for Applications in Engineering*, vol. 3147, Springer Berlin Heidelberg, 2004, pp. 325–354.

[8] N. Spanoudakis and P. Moraitis, “Automated Product Pricing Using Argumentation,” in *Proceedings of the 5th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI 2009)*, 2009.

[9] W. Kleppe, A.G, Warmer, J., Bast, “MDA Explained: The Model Driven Architecture: Practice and Promise.,” *Addison-Wesley, Boston, MA, USA*, 2003.

[10] S. Beydeda, M. Book, and V. Gruhn, *Model-Driven Software Development*. 2005, p. 464.

[11] F. Zambonelli, N. R. Jennings, and M. Wooldridge, “Developing multiagent systems: The gaia methodology.,” *ACM Trans. Softw. Eng. Methodol.*, vol. 12, no. 3, pp. 317–370, 2003.

[12] M. Snoeck, S. Poelmans, and G. Dedene, “An Architecture for bridging OO and business process modeling.,” in *33rd International Conference on Technology of Object-Oriented Languages and Systems*, 2000, pp. pp. 132–143.

[13] M. A. Cibran, “Translating BPMN models into UML Activities,” in *Ardagna, D. et al. (eds.) BPM 2008 International Workshops, Milano, Italy, September 1-4, 2008. Revised Papers.*, 2009, pp. 236–247.

[14] G. Redding, M. Dumas, A. Hof, H.M. ter Hofstede, and A. Iordachescu, “Generating Business Process Models from Object Behavior Models.,” in *Information Systems Management*, 2008, pp. 319–331.

[15] N. Spanoudakis and P. Moraitis, “Model-Driven Agents Development with ASEME.,” in *11th Int. Workshop on Agent Oriented Software Engineering (AOSE 2010), Toronto, Canada (2010)*, 2010.

[16] G. Decker and F. Puhlmann, “Extending BPMN for modeling complex choreographies.,” in *Proceedings of the COOPIS 2007, Vilamoura, Portugal,*, 2007, pp. 24–40.

[17] W. M. Coalition, *Workflow Standard Process Definition Interface - XML Process Definition Language*. WFMC-TC-1025., 2008.

[18] M. Jahangirian, E. Tillal, K. Naseer, L. Stergioulas, and T. Young, “Simulation in manufacturing and Business: A review,” *Eur. J. Oper. Res.*, vol. 203, no. 1, pp. 1–13, 2010.

APPENDIX

Appendix A Java source code

The class Live2xpdl

```java
package aseme.transformations.xpdl;

import java.io.File;
import java.io.FileOutputStream;
import java.util.Hashtable;
import java.util.LinkedList;
import java.util.List;
import java.util.StringTokenizer;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.enhydra.jxpdl.XMLUtil;
import org.enhydra.jxpdl.XPDLRepositoryHandler;
import org.enhydra.jxpdl.elements.Activity;
import org.enhydra.jxpdl.elements.Lane;
import org.enhydra.jxpdl.elements.NodeGraphicsInfo;
import org.enhydra.jxpdl.elements.Package;
import org.enhydra.jxpdl.elements.Participant;
import org.enhydra.jxpdl.elements.Performer;
import org.enhydra.jxpdl.elements.Pool;
import org.enhydra.jxpdl.elements.Transition;
import org.enhydra.jxpdl.elements.TransitionRestriction;
import org.enhydra.jxpdl.elements.WorkflowProcess;
import org.w3c.dom.Document;

import java.util.Calendar;

public class Live2xpdl {
        Activity previous;
        int bracketsflag = 0;
        String lefthand = "";
        int activityid = 0;
```

```
    List<Activity> activitylist = new LinkedList<Activity>();
    int flag = 0;
    int numberoforexpressions = 0;
    String connector;
    int startchecker = 1;
    int term = 0;
    XMLUtil xml = new XMLUtil();
    int Xcoordinate = 0;
    int Ycoordinate = 0;
    int numberofpara = 0;
    int numberofseq = 0;
    int numberofors = 0;
    int previousterm = 0;
    int l = 0;
    String liveness = null;
    Hashtable<String, String> formulas = null;
    String orId = "";
    List<Activity> firstacts = new LinkedList<Activity>();
    int listpointer = 0;
    List<String> findTerms = new LinkedList<String>();
    List<String> complexTerms = new LinkedList<String>();
    List<String> expressionsimpleTerm = new LinkedList<>();
    List<String> expressionList = new LinkedList<>();
    List<String> foundTerms = new LinkedList<String>();
    public List<String> rolesnames = new LinkedList<String>();
    private String roleandname = "";
    List<Matcher> matcherlist = new LinkedList<Matcher>();
    private String sendername;
    private String bracketsterm;
    private String bracket1;
    private String term123;
    String insideTerm1 = "";
    private String rolename = "";

    public String pname = "";

    Pool p = null;
    WorkflowProcess wp = null;
    Lane l1 = null;

    public static void main(String[] args) throws Exception {
        // TODO Auto-generated method stub

        Live2xpdl live2xpdl = new Live2xpdl();

        if (args.length == 0) {
            live2xpdl.liveness = new String(
            // "ComplexProvider=SP~\n"
            //
+"SP=ReceiveRequestMessage.ProcessRequest.SendResponseMessage\n"
            //
+"ProcessRequest=(DecideRouteType.SR.SortRoutes)|[DecidePOITypes.SR.DecidePois.
SR]\n"
            // +"SR=SendRequestMessage.ReceiveResponseMessage"
            // "PersonalAssistant = SR\n"
            //
"SP=(ReceiveRequestMessage.A3)~||ProcessRequest~||SendResponseMessage~\n"
            // +"SR=SendRequestMessage.ReceiveResponseMessage"
```

```
                    // "x = A1|(A2)+|A3*|A~"
                    // "x=[A1.A2.A3]"
                                 "x =
[A1.A2]~|[A3|A4*|A6+]*|A7~|[A8~|A9*|A10~]~"
                    // "x = A*"
                    // "x = [A]+"
                    // "x=[A1|A2|A3+]+"
                    // "x=[A1+.A2~.A3~]+
                    // "x=A~"
                    // "x=(A1.A2~.A3)~.(A4~.A5.A6+)~"
                    // "x=(A1||A2~||A3)+.(A4|A5~|A6)~"
                    // "x=(A1|A2~|A3)~.A5.A6.A7+"
                    // "x=(A1|A2~|A3)~"
                    );
             } else {
                    live2xpdl.liveness = args[0];
             }
             live2xpdl.liveness = live2xpdl.liveness.replaceAll(" ", "");
             // live2xpdl.transform("C:/Users/Nektarios/Desktop/Myfirst.xpdl");

             // dimioyrgia toy file mas
             String outputFile = "C:/Users/nek/Desktop/Myfirst.xpdl";
             File f = new File(outputFile);
             outputFile = f.getCanonicalPath();
             String name = f.getName().substring(0,
                           ((File) f).getName().lastIndexOf("."));
             System.out.println("Creating XPDL Model.\n");

             String id = name;
             if (!XMLUtil.isIdValid(id)) {
                    id = "test";
             }

             System.out.println("...creating Package [Id=" + id + ",Name=" +
name
                           + ",Script-type=text/javascript]");
             Package pkg = new Package();
             pkg.setId(id);
             pkg.setName(name);
             pkg.getPackageHeader().setXPDLVersion("2.1");
             pkg.getPackageHeader().setVendor("TUC");

       pkg.getPackageHeader().setCreated(XMLUtil.getCurrentDateAndTime());

             pkg.getScript().setType("http://www.w3.org/1999/XPath");

             StringTokenizer line1 = new StringTokenizer(live2xpdl.liveness,
"=");
             String poolname = line1.nextToken();
             System.out.println("What is the pool name:" + poolname);
             // pool
             System.out.println("......creating Pool [Id=" + poolname +
",Name="
                           + poolname + ",Process=" + poolname + "]");
             Pool p = (Pool) pkg.getPools().generateNewElement();
             p.setId(poolname);
             p.setName(poolname);
```

```
		NodeGraphicsInfo ngip = (NodeGraphicsInfo)
p.getNodeGraphicsInfos()
				.generateNewElement();
		ngip.setWidth(350);
		ngip.setHeight(250);
		ngip.getCoordinates().setXCoordinate("0");
		ngip.getCoordinates().setYCoordinate("0");
		p.getNodeGraphicsInfos().add(ngip);

		System.out.println(".....creating WorkflowProcess [Id=" +
poolname
				+ ",Name=" + poolname + "]");
		WorkflowProcess wp = (WorkflowProcess) pkg.getWorkflowProcesses()
				.generateNewElement();
		wp.setId(poolname);
		wp.setName(poolname);

		// lane
		System.out.println("........creating Lane[Id=projectlane,Name="
				+ poolname + "]");
		Lane l1 = (Lane) p.getLanes().generateNewElement();
		l1.setId(poolname);
		l1.setName(poolname);
		Performer perf1 = (Performer)
l1.getPerformers().generateNewElement();
		perf1.setValue(poolname);
		l1.getPerformers().add(perf1);
		NodeGraphicsInfo ngia = (NodeGraphicsInfo)
l1.getNodeGraphicsInfos()
				.generateNewElement();
		System.out
				.println("........creating
NodeGraphicsInfo[LaneId=projectlane,Coordinates=20,30]");
		ngia.setLaneId(poolname);
		ngia.setWidth(300);
		ngia.setHeight(200);
		ngia.getCoordinates().setXCoordinate("20");
		ngia.getCoordinates().setYCoordinate("30");
		l1.getNodeGraphicsInfos().add(ngia);

		p.getLanes().add(l1);
		pkg.getWorkflowProcesses().add(wp);
		pkg.getPools().add(p);

		p = live2xpdl.transform(live2xpdl.liveness, pkg, outputFile);

		writeToFile(outputFile, pkg);
		System.out.println("\nWritting XPDL model into file \"" +
outputFile
				+ "\".");
	}

	public Pool transform(String liveness, Package pkg, String outputFile)
			throws Exception {

		firstacts = new LinkedList<Activity>();
		WorkflowProcess wp = (WorkflowProcess) pkg.getWorkflowProcesses()
				.get(0);
```

```
              formulas = new Hashtable<String, String>();
              StringTokenizer line = new StringTokenizer(liveness, "\n");

              while (line.hasMoreTokens()) {
                     String tmp = line.nextToken();
                     StringTokenizer formulaElements = new StringTokenizer(tmp,
"=");
                     String leftHandSide = formulaElements.nextToken();
                     String formula = formulaElements.nextToken();
                     String formula1 = preprocessing(formula);
                     System.out.println("What is the formula after
preprocessing:"
                                   + formula1);
                     formulas.put(leftHandSide, formula1);
              }

              System.out.println("What are the formulas?" + formulas);
              line = new StringTokenizer(liveness, "\n");

              StringTokenizer formulaElements = new
StringTokenizer(line.nextToken(),
                            "=");
              String leftHandSide = formulaElements.nextToken();

              Participant p1 = (Participant) pkg.getParticipants()
                            .generateNewElement();
              p1.setId(leftHandSide);
              p1.setName(leftHandSide);
              p1.getParticipantType().setTypeROLE();
              pkg.getParticipants().add(p1);
              rolename = leftHandSide;

              Calendar cal2 = Calendar.getInstance();
              long d = cal2.getTimeInMillis();
              // start
              activityid = activityid + 1;
              Activity start = (Activity)
wp.getActivities().generateNewElement();
              System.out.println("......creating start Event");
              start.setId("" + d + "" + rolename + "" + activityid);
              start.setName("start");
              start.getActivityTypes().setImplementation();
              start.getActivityTypes().setEvent();

       start.getActivityTypes().getEvent().getEventTypes().setStartEvent();
              start.setFirstPerformer(leftHandSide);
              NodeGraphicsInfo ngiact = (NodeGraphicsInfo) start
                            .getNodeGraphicsInfos().generateNewElement();
              ngiact.setLaneId(getPoolname());
              ngiact.setHeight(31);
              ngiact.setWidth(31);
              ngiact.getCoordinates().setXCoordinate("50");
              ngiact.getCoordinates().setYCoordinate("60");
              start.getNodeGraphicsInfos().add(ngiact);
              wp.getActivities().add(start);

              lefthand = leftHandSide;
              Activity act = this
```

```
                                .createProcess(formulas.get(leftHandSide), wp,
start);

                d = cal2.getTimeInMillis();
                activityid = activityid + 1;
                // end
                Activity endact = (Activity)
wp.getActivities().generateNewElement();
                System.out.println(".......creating end Event");
                endact.setId("" + d + "" + rolename + "" + activityid);
                endact.setName("end");
                endact.getActivityTypes().setImplementation();
                endact.getActivityTypes().setEvent();

        endact.getActivityTypes().getEvent().getEventTypes().setEndEvent();
                endact.setFirstPerformer(leftHandSide);
                NodeGraphicsInfo ngiact1 = (NodeGraphicsInfo) endact
                                .getNodeGraphicsInfos().generateNewElement();
                ngiact1.setLaneId(getPoolname());
                ngiact1.setShape("annotation");
                ngiact1.setHeight(31);
                ngiact1.setWidth(31);
                ngiact1.getCoordinates().setXCoordinate("50");
                ngiact1.getCoordinates().setYCoordinate("250");
                endact.getNodeGraphicsInfos().add(ngiact1);

                Transition endtra = (Transition) wp.getTransitions()
                                .generateNewElement();
                activityid = activityid + 1;
                endtra.setId("" + activityid);
                endtra.setFrom(act.getId());
                endtra.setTo(endact.getId());
                wp.getTransitions().add(endtra);

                System.out
                                .println("......creating ending Transition[Id =
endtra, From=acttemp, To=end, Type=null]");
                wp.getActivities().add(endact);

                return p;
        }

        protected void setsendername(String name) {
                sendername = name;
        }

        protected String getsendername() {
                // TODO Auto-generated method stub
                return sendername;
        }

        private String preprocessing(String formula) {
                String test = formula;

                Pattern patternParallelManyTimesTerm = Pattern
                                .compile("\\|[a-zA-Z_](\\w+)*~\\|\\d+");
                Matcher parallelManyTimesTermMatcher =
patternParallelManyTimesTerm
```

```
                .matcher(formula);
        if (parallelManyTimesTermMatcher.find()) {
            matcherlist.add(parallelManyTimesTermMatcher);

            int starting = parallelManyTimesTermMatcher.regionStart();
            System.out.println("What is the starting point" +
starting);
            int ending = parallelManyTimesTermMatcher.regionEnd();
            System.out.println("What is the ending point" + ending);
            int startof = parallelManyTimesTermMatcher.start();

            int end = parallelManyTimesTermMatcher.end();

            String inside = parallelManyTimesTermMatcher.group();
            String inside1 = inside.substring(1, inside.length() - 2);

            char endofstring = formula.charAt(formula.length() - 1);
            System.out.println("What is the endofstring" +
endofstring);
            int x = Character
                    .getNumericValue(inside.charAt(inside.length()
- 1));

            int i = 0;
            formula = "(";
            for (i = 0; i < x; i++) {
                formula = formula + inside1 + "||";
            }

            String betterformula = formula.substring(0,
formula.length() - 2);

            betterformula = betterformula + ")";

            String newstarttest = test.substring(0, startof);

            String newendtest = test.substring(end, test.length());

            String concat1 = newstarttest.concat(betterformula);
            String concat = concat1.concat(newendtest);

            test = preprocessing(concat);
        }

        return test;
    }

    public Activity createProcess(String expression, WorkflowProcess wp,
            Activity actprevious) {

        Calendar cal2 = Calendar.getInstance();
        long d = cal2.getTimeInMillis();
        String left = lefthand;
        Transition currentend = (Transition) wp.getTransitions()
                .generateNewElement();
        activityid = activityid + 1;
        currentend.setId("" + d + "" + left + "" + activityid);
        // TODO Auto-generated method stub
        int expressionType = 0;
        // pattern for parallelExpression
        Pattern patternParallelExpression = Pattern

    .compile("(((\\|.+~|(\\d))|(\\[.+\\])|(\\(.+\\))|([\\w&&[^()]])+)([~+*]?)
\\|\\|)+((\\|.+~\\|(\\d))|(\\[.+\\])|(\\(.+\\))|([\\w&&[^()]])+)([~+*]?)");
        Matcher parallelMatcher =
patternParallelExpression.matcher(expression);
        // pattern for orExpression : expressionType=2
        Pattern patternOrExpression = Pattern

    .compile("(((\\|.+~|(\\d))|(\\[.+\\])|(\\(.+\\))|([\\w&&[^()]])+)([~+*]?)
\\|)+((\\|.+~\\|(\\d))|(\\[.+\\])|(\\(.+\\))|([\\w&&[^()]])+)([~+*]?)");
        Matcher orMatcher = patternOrExpression.matcher(expression);
        // pattern for sequentialExpression : expressionType=3
        Pattern patternSequentialExpression = Pattern

    .compile("(((\\|.+~\\|\\d)|(\\[.+\\])|(\\(.+\\))|([\\w&&[^()]])+)([~+*]?)
\\.)+((\\|.+~\\|\\d)|(\\[.+\\])|(\\(.+\\))|([\\w&&[^()]])+)([~+*]?)");

        Matcher sequentialMatcher = patternSequentialExpression
                .matcher(expression);

        List<String> myTerms = null;

        switch (expressionType) {
        case 0:
            myTerms = new LinkedList<String>();
            myTerms.add(expression);
            break;
        }

        if (sequentialMatcher.find()
                && (sequentialMatcher.group().length() ==
expression.length())) {
            expressionType = 3;

            System.out.print("a sequential expression processed: " +
expression
                    + "\n");
            setconnector(".");
            List<String> seqterms =
this.findTermsInExpression(expression, ".",
                    wp);

            System.out.println("Seqterms are:" + seqterms);

            int listpointer2 = listpointer;
            int b = 0;
            for (b = 0; b < seqterms.size(); b++) {
                String currentterm = seqterms.get(b);
                System.out.println("The current term is:" +
currentterm);

                Activity seqact = createProcess(currentterm, wp,
actprevious);
                listpointer = listpointer + 1;
                actprevious = seqact;
            }

            int i = 0;
            for (i = 0; i < seqterms.size(); i++) {

                listpointer = listpointer - 1;
                if (listpointer == listpointer2) {
                    break;
                }

                try {
                    firstacts.remove(listpointer);
                } catch (Exception e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }

            }
        } else if (orMatcher.find()
                && (orMatcher.group().length() ==
expression.length())) {
            expressionType = 2;

            System.out
                    .print("an or expression processed: " +
expression + "\n");
            numberoforexpressions = numberoforexpressions + 1;
            System.out.println("The numberoforexpressions is:"
                    + numberoforexpressions);

            d = cal2.getTimeInMillis();
            activityid = activityid + 1;
            String rightor = remover(expression);
            Activity XORgate = (Activity) wp.getActivities()
                    .generateNewElement();
            XORgate.setName("X:" + rightor);
            XORgate.setId("" + d + "" + left + "" + activityid);
            XORgate.getActivityTypes().setImplementation();
            XORgate.getActivityTypes().getImplementation()
                    .getImplementationTypes().setValue("1");
            XORgate.getActivityTypes().getImplementation()
                    .getImplementationTypes().setNo();

    XORgate.getActivityTypes().getRoute().setGatewayTypeExclusive();
            NodeGraphicsInfo ngiact4 = (NodeGraphicsInfo) XORgate
                    .getNodeGraphicsInfos().generateNewElement();
            ngiact4.setShape("annotation");
            ngiact4.setLaneId(getPoolname());
            ngiact4.getCoordinates()
                    .setXCoordinate("" + Xcoordinate + 200 + "");
            ngiact4.getCoordinates().setYCoordinate("" + Ycoordinate +
40 + "");
            ngiact4.setHeight(43);
            ngiact4.setWidth(31);
            ngiact4.setBorderColor("0,0,0");
            ngiact4.setFillColor("255,229,124");
            XORgate.getNodeGraphicsInfos().add(ngiact4);
```

```
                    d = cal2.getTimeInMillis();
                    activityid = activityid + 1;
                    System.out
                                  .println("........creating
Activity[Id=actd,Name=Decision,Type=Route,Split-type=Exclusive]");
                    Activity actd = (Activity)
wp.getActivities().generateNewElement();
                    actd.setId("" + d + "" + left + "" + activityid);
                    actd.setName("Decision");
                    actd.getActivityTypes().setRoute();
                    System.out.println("...........creating
TransitionRestriction");
                    TransitionRestriction tre = (TransitionRestriction) XORgate

       .getTransitionRestrictions().generateNewElement();
                    tre.getSplit().setTypeExclusive();
                    actd.getTransitionRestrictions().add(tre);
                    System.out
                                  .println("...........creating
NodeGraphicsInfo[LaneId=lane1,Coordinates=200,60]");
                    NodeGraphicsInfo ngid = (NodeGraphicsInfo) actd
                                  .getNodeGraphicsInfos().generateNewElement();
                    ngid.setLaneId(getPoolname());
                    ngid.setWidth(40);
                    ngid.setHeight(40);
                    ngid.getCoordinates().setXCoordinate("200");
                    ngid.getCoordinates().setYCoordinate("60");
                    actd.getNodeGraphicsInfos().add(ngid);
                    wp.getActivities().add(actd);

                    d = cal2.getTimeInMillis();
                    activityid = activityid + 1;
                    Activity XORgate2 = (Activity) wp.getActivities()
                                  .generateNewElement();
                    XORgate2.setName("X2:" + rightor);
                    XORgate2.setId("" + d + "" + left + "" + activityid);
                    XORgate2.getActivityTypes().setImplementation();
                    XORgate2.getActivityTypes().getImplementation()
                                  .getImplementationTypes().setValue("1");
                    XORgate2.getActivityTypes().getImplementation()
                                  .getImplementationTypes().setNo();

       XORgate2.getActivityTypes().getRoute().setGatewayTypeExclusive();
                    NodeGraphicsInfo ngiact5 = (NodeGraphicsInfo) XORgate2
                                  .getNodeGraphicsInfos().generateNewElement();
                    ngiact5.setShape("annotation");
                    ngiact5.setLaneId(getPoolname());
                    ngiact5.getCoordinates()
                                  .setXCoordinate("" + Xcoordinate + 200 + "");
                    ngiact5.getCoordinates().setYCoordinate("" + Ycoordinate +
40 + "");
                    ngiact5.setHeight(43);
                    ngiact5.setWidth(31);
                    ngiact5.setBorderColor("0,0,0");
                    ngiact5.setFillColor("255,229,124");
                    XORgate2.getNodeGraphicsInfos().add(ngiact5);

                    d = cal2.getTimeInMillis();
                    activityid = activityid + 1;
                    System.out
                                  .println(".........creating
Activity[Id=actd,Name=Decision,Type=Route,Split-type=Exclusive]");
                    Activity actd1 = (Activity)
wp.getActivities().generateNewElement();
                    actd1.setId("" + d + "" + left + "" + activityid);
                    actd1.setName("Decision2");
                    actd1.getActivityTypes().setRoute();
                    System.out.println("...........creating
TransitionRestriction");
                    TransitionRestriction tre1 = (TransitionRestriction)
XORgate2

       .getTransitionRestrictions().generateNewElement();
                    tre1.getSplit().setTypeExclusive();
                    actd1.getTransitionRestrictions().add(tre1);
                    System.out
                                  .println("...........creating
NodeGraphicsInfo[LaneId=lane1,Coordinates=200,60]");
                    NodeGraphicsInfo ngid1 = (NodeGraphicsInfo) actd1
                                  .getNodeGraphicsInfos().generateNewElement();
                    ngid1.setLaneId(getPoolname());
                    ngid1.setWidth(40);
                    ngid1.setHeight(40);
                    ngid1.getCoordinates().setXCoordinate("200");
                    ngid1.getCoordinates().setYCoordinate("60");
                    actd1.getNodeGraphicsInfos().add(ngid1);
                    wp.getActivities().add(actd1);

                    d = cal2.getTimeInMillis();
                    Transition orfirst = (Transition) wp.getTransitions()
                                  .generateNewElement();
                    activityid = activityid + 1;
                    orfirst.setId("" + d + "" + left + "" + activityid);
                    orfirst.setFrom(actprevious.getId());
                    orfirst.setTo(actd.getId());
                    wp.getTransitions().add(orfirst);

                    int listpointer2 = listpointer;

                    listpointer = listpointer + 1;
                    firstacts.add(actd);

                    List<String> orterms =
this.findTermsInExpression(expression, "|",
                                  wp);
                    System.out.println("the orterms are:" + orterms);

                    int or = 0;
                    for (or = 0; or < orterms.size(); or++) {
                           String currentterm = orterms.get(or);
                           System.out.println("The current term is:" +
currentterm);
                           actprevious = actd;

                           Activity ortemp = createProcess(currentterm, wp,
actprevious);
```

```
                listpointer = listpointer + 1;
                d = cal2.getTimeInMillis();
                activityid = activityid + 1;
                Transition ortotemp = (Transition)
wp.getTransitions()
                        .generateNewElement();
                ortotemp.setId("" + d + "" + left + "" + activityid);
                ortotemp.setFrom(ortemp.getId());
                ortotemp.setTo(actd1.getId());
                wp.getTransitions().add(ortotemp);
            }

            int i = 0;
            for (i = 0; i < orterms.size() + 1; i++) {

                listpointer = listpointer - 1;
                if (listpointer == listpointer2) {
                    break;
                }
                firstacts.remove(listpointer);
            }

            actprevious = actd1;
        } else if (parallelMatcher.find()
                && (parallelMatcher.group().length() ==
expression.length())) {
            expressionType = 1;
            System.out.print("a parallel expression processed: " +
expression
                    + "\n");

            String rightexpression = remover(expression);
            d = cal2.getTimeInMillis();
            activityid = activityid + 1;
            Activity paralleljoin = (Activity) wp.getActivities()
                    .generateNewElement();
            paralleljoin.setId("" + d + "" + left + "" + activityid);
            paralleljoin.setName("para1");
            paralleljoin.getActivityTypes().setImplementation();
            paralleljoin.getActivityTypes().getImplementation()
                    .getImplementationTypes().setNo();

    paralleljoin.getActivityTypes().getRoute().setGatewayTypeParallel();
            NodeGraphicsInfo ngiact4 = (NodeGraphicsInfo) paralleljoin
                    .getNodeGraphicsInfos().generateNewElement();
            ngiact4.setLaneId(getPoolname());
            ngiact4.getCoordinates()
                    .setXCoordinate("" + Xcoordinate + 200 + "");
            ngiact4.getCoordinates().setYCoordinate("" + Ycoordinate +
40 + "");
            ngiact4.setHeight(43);
            ngiact4.setWidth(31);
            ngiact4.setBorderColor("0,0,0");
            ngiact4.setFillColor("255,229,124");
            paralleljoin.getNodeGraphicsInfos().add(ngiact4);

            d = cal2.getTimeInMillis();
            activityid = activityid + 1;
```

```
                    System.out
                                 .println("........creating
Activity[Id=actd,Name=Decision,Type=Route,Split-type=Parallel]");
                    Activity actd3 = (Activity)
wp.getActivities().generateNewElement();
                    actd3.setId("" + d + "" + left + "" + activityid);
                    actd3.setName("parallel");
                    actd3.getActivityTypes().setRoute();

       actd3.getActivityTypes().getRoute().setGatewayTypeParallel();
                    System.out.println("...........creating
TransitionRestriction");
                    TransitionRestriction tre2 = (TransitionRestriction)
paralleljoin

       .getTransitionRestrictions().generateNewElement();
                    tre2.getSplit().setTypeParallel();
                    actd3.getTransitionRestrictions().add(tre2);
                    System.out
                                 .println("...........creating
NodeGraphicsInfo[LaneId=lane1,Coordinates=200,60]");
                    NodeGraphicsInfo ngid2 = (NodeGraphicsInfo) actd3
                                 .getNodeGraphicsInfos().generateNewElement();
                    ngid2.setLaneId(getPoolname());
                    ngid2.setWidth(40);
                    ngid2.setHeight(40);
                    ngid2.getCoordinates().setXCoordinate("200");
                    ngid2.getCoordinates().setYCoordinate("60");
                    actd3.getNodeGraphicsInfos().add(ngid2);
                    wp.getActivities().add(actd3);

                    d = cal2.getTimeInMillis();
                    activityid = activityid + 1;
                    Activity paralleljoin2 = (Activity) wp.getActivities()
                                 .generateNewElement();
                    paralleljoin2.setName("para:2:" + rightexpression);
                    paralleljoin2.setId("" + d + "" + left + "" + activityid);
                    paralleljoin2.getActivityTypes().setImplementation();
                    paralleljoin2.getActivityTypes().getImplementation()
                                 .getImplementationTypes().setNo();
                    NodeGraphicsInfo ngiact5 = (NodeGraphicsInfo) paralleljoin2
                                 .getNodeGraphicsInfos().generateNewElement();
                    ngiact5.setShape("annotation");
                    ngiact5.setLaneId(getPoolname());
                    ngiact5.getCoordinates()
                                 .setXCoordinate("" + Xcoordinate + 400 + "");
                    ngiact5.getCoordinates().setYCoordinate("" + Ycoordinate +
40 + "");
                    ngiact5.setHeight(43);
                    ngiact5.setWidth(31);
                    ngiact5.setBorderColor("0,0,0");
                    ngiact5.setFillColor("255,229,124");
                    paralleljoin2.getNodeGraphicsInfos().add(ngiact5);

                    d = cal2.getTimeInMillis();
                    activityid = activityid + 1;
                    System.out
```

```
                                  .println("........creating
Activity[Id=actd,Name=Decision,Type=Route,Split-type=Exclusive]");
                    Activity actd1 = (Activity)
wp.getActivities().generateNewElement();
                    actd1.setId("" + d + "" + left + "" + activityid);
                    actd1.setName("parallel2");
                    actd1.getActivityTypes().setRoute();

       actd1.getActivityTypes().getRoute().setGatewayTypeParallel();
                    System.out.println("...........creating
TransitionRestriction");
                    TransitionRestriction tre1 = (TransitionRestriction)
paralleljoin2

       .getTransitionRestrictions().generateNewElement();
                    tre1.getSplit().setTypeParallel();
                    actd1.getTransitionRestrictions().add(tre1);
                    System.out
                                  .println("...........creating
NodeGraphicsInfo[LaneId=lane1,Coordinates=200,60]");
                    NodeGraphicsInfo ngid3 = (NodeGraphicsInfo) actd1
                                  .getNodeGraphicsInfos().generateNewElement();
                    ngid3.setLaneId(getPoolname());
                    ngid3.setWidth(40);
                    ngid3.setHeight(40);
                    ngid3.getCoordinates().setXCoordinate("200");
                    ngid3.getCoordinates().setYCoordinate("60");
                    actd1.getNodeGraphicsInfos().add(ngid3);
                    wp.getActivities().add(actd1);

                    d = cal2.getTimeInMillis();
                    Transition previoustopara = (Transition)
wp.getTransitions()
                                  .generateNewElement();
                    activityid = activityid + 1;
                    previoustopara.setId("" + d + "" + left + "" + activityid);
                    previoustopara.setFrom(actprevious.getId());
                    previoustopara.setTo(actd3.getId());
                    wp.getTransitions().add(previoustopara);

                    int listpointer2 = listpointer;
                    listpointer = listpointer + 1;
                    firstacts.add(actd3);

                    List<String> paraterms =
this.findTermsInExpression(expression,
                                  "||", wp);
                    System.out.println("the parallel terms are:" + paraterms);

                    int par = 0;
                    for (par = 0; par < paraterms.size(); par++) {
                           String currentterm = paraterms.get(par);
                           System.out.println("The current term is:" +
currentterm);
                           actprevious = actd3;

                           Activity paratemp = createProcess(currentterm, wp,
actprevious);
```

```
                           listpointer = listpointer + 1;

                           d = cal2.getTimeInMillis();
                           Transition parafirst = (Transition)
wp.getTransitions()
                                        .generateNewElement();
                           activityid = activityid + 1;
                           parafirst.setId("" + d + "" + left + "" +
activityid);
                           parafirst.setFrom(paratemp.getId());
                           parafirst.setTo(actd1.getId());
                           wp.getTransitions().add(parafirst);

                    }

                    int i = 0;
                    for (i = 0; i < paraterms.size() + 1; i++) {

                           listpointer = listpointer - 1;
                           if (listpointer == listpointer2) {
                                  break;
                           }
                           firstacts.remove(listpointer);
                    }

                    actprevious = actd1;

             } else {
                    String term1 = expression;
                    setTerm1(term1);
                    System.out.println("term1 is:" + term1);
                    Pattern patternBasicTerm = Pattern.compile("^\\w+$");
                    Matcher basicTermMatcher = patternBasicTerm.matcher(term1);
                    // pattern for (term)
                    Pattern patternComplexParenthesisTerm = Pattern
                                  .compile("^\\(.+\\)$");
                    Matcher complexParenthesisTermMatcher =
patternComplexParenthesisTerm
                                  .matcher(term1);
                    // pattern for [term]
                    Pattern patternComplexOptionalTerm =
Pattern.compile("^\\[.+\\]$");
                    Matcher complexOptionalTermMatcher =
patternComplexOptionalTerm
                                  .matcher(term1);
                    // pattern for term~
                    Pattern patternForeverTerm = Pattern.compile(".+~$");
                    Matcher foreverTermMatcher =
patternForeverTerm.matcher(term1);
                    // pattern for term+
                    Pattern patternOneOrMoreTimesTerm =
Pattern.compile(".+\\+$");
                    Matcher oneOrMoreTimesTermMatcher =
patternOneOrMoreTimesTerm
                                  .matcher(term1);
                    // pattern for term*
                    Pattern patternZeroOrMoreTimesTerm =
Pattern.compile(".+\\*$");
```

```
                    Matcher zeroOrMoreTimesTermMatcher =
patternZeroOrMoreTimesTerm
                                  .matcher(term1);

                    if (complexParenthesisTermMatcher.find()
                                  &&
(complexParenthesisTermMatcher.group().length() == term1
                                               .length())) {
                           System.out.println("A parenthesis term is being
processed");
                           String insideTerm = term1.substring(1, term1.length()
- 1);
                           System.out.println("the inside term is:" +
insideTerm);

                           Activity parenth = createProcess(insideTerm, wp,
actprevious);
                           actprevious = parenth;
                    } else if (complexOptionalTermMatcher.find()
                                  &&
(complexOptionalTermMatcher.group().length() == term1
                                               .length())) {
                           System.out.println("A brackets term is being
processed.");
                           // x optional [x]

                           d = cal2.getTimeInMillis();
                           activityid = activityid + 1;
                           Activity oroptional = (Activity) wp.getActivities()
                                        .generateNewElement();
                           oroptional.setName("Xbrackets:" + term1);
                           oroptional.setId("" + d + "" + left + "" +
activityid);
                           oroptional.getActivityTypes().setImplementation();
                           oroptional.getActivityTypes().getImplementation()
                                        .getImplementationTypes().setNo();
                           NodeGraphicsInfo ngiact5 = (NodeGraphicsInfo)
oroptional

       .getNodeGraphicsInfos().generateNewElement();
                           ngiact5.setShape("annotation");
                           ngiact5.setLaneId(getPoolname());
                           ngiact5.getCoordinates().setXCoordinate(
                                        "" + Xcoordinate + 200 + "");
                           ngiact5.getCoordinates().setYCoordinate(
                                        "" + Ycoordinate + 40 + "");
                           ngiact5.setHeight(43);
                           ngiact5.setWidth(31);
                           ngiact5.setShape("annotation");
                           ngiact5.setBorderColor("0,0,0");
                           ngiact5.setFillColor("255,255,51");
                           oroptional.getNodeGraphicsInfos().add(ngiact5);

                           d = cal2.getTimeInMillis();
                           activityid = activityid + 1;
                           System.out
                                        .println(".........creating
Activity[Id=actd,Name=Decision,Type=Route,Split-type=Exclusive]");
```

```
                            Activity actd = (Activity) wp.getActivities()
                                          .generateNewElement();
                            actd.setId("" + d + "" + left + "" + activityid);
                            actd.setName("Decision:" + term1);
                            actd.getActivityTypes().setRoute();
                            System.out
                                          .println("...........creating
TransitionRestriction");
                            TransitionRestriction tre = (TransitionRestriction)
oroptional

       .getTransitionRestrictions().generateNewElement();
                            tre.getSplit().setTypeExclusive();
                            actd.getTransitionRestrictions().add(tre);
                            System.out
                                          .println("...........creating
NodeGraphicsInfo[LaneId=lane1,Coordinates=200,60]");
                            NodeGraphicsInfo ngid = (NodeGraphicsInfo) actd

       .getNodeGraphicsInfos().generateNewElement();
                            ngid.setLaneId(getPoolname());
                            ngid.setWidth(40);
                            ngid.setHeight(40);
                            ngid.getCoordinates().setXCoordinate("200");
                            ngid.getCoordinates().setYCoordinate("60");
                            actd.getNodeGraphicsInfos().add(ngid);
                            wp.getActivities().add(actd);

                            d = cal2.getTimeInMillis();
                            activityid = activityid + 1;
                            Activity optionaltoend = (Activity)
wp.getActivities()
                                          .generateNewElement();
                            optionaltoend.setName("Xbrackets1:" + term1);
                            optionaltoend.setId("" + d + "" + left + "" +
activityid);
                            optionaltoend.getActivityTypes().setImplementation();
                            optionaltoend.getActivityTypes().getImplementation()
                                          .getImplementationTypes().setNo();
                            NodeGraphicsInfo ngiact6 = (NodeGraphicsInfo)
optionaltoend

       .getNodeGraphicsInfos().generateNewElement();
                            ngiact6.setShape("annotation");
                            ngiact6.setLaneId(getPoolname());
                            ngiact6.getCoordinates().setXCoordinate(
                                          "" + Xcoordinate + 200 + "");
                            ngiact6.getCoordinates().setYCoordinate(
                                          "" + Ycoordinate + 40 + "");
                            ngiact6.setHeight(43);
                            ngiact6.setWidth(31);
                            ngiact6.setShape("annotation");
                            ngiact6.setBorderColor("0,0,0");
                            ngiact6.setFillColor("255,255,51");
                            optionaltoend.getNodeGraphicsInfos().add(ngiact6);

                            d = cal2.getTimeInMillis();
                            activityid = activityid + 1;
```

```
                           System.out
                                        .println("........creating
Activity[Id=actd,Name=Decision,Type=Route,Split-type=Exclusive]");
                           Activity actd1 = (Activity) wp.getActivities()
                                        .generateNewElement();
                           actd1.setId("" + d + "" + left + "" + activityid);
                           actd1.setName("Decision2:" + term1);
                           actd1.getActivityTypes().setRoute();
                           System.out
                                        .println("...........creating
TransitionRestriction");
                           TransitionRestriction tre1 = (TransitionRestriction)
optionaltoend

       .getTransitionRestrictions().generateNewElement();
                           tre1.getSplit().setTypeExclusive();
                           actd1.getTransitionRestrictions().add(tre1);
                           System.out
                                        .println("...........creating
NodeGraphicsInfo[LaneId=lane1,Coordinates=200,60]");
                           NodeGraphicsInfo ngid1 = (NodeGraphicsInfo) actd1

       .getNodeGraphicsInfos().generateNewElement();
                           ngid1.setLaneId(getPoolname());
                           ngid1.setWidth(40);
                           ngid1.setHeight(40);
                           ngid1.getCoordinates().setXCoordinate("200");
                           ngid1.getCoordinates().setYCoordinate("60");
                           actd1.getNodeGraphicsInfos().add(ngid1);
                           wp.getActivities().add(actd1);

                           d = cal2.getTimeInMillis();
                           Transition tooroptional = (Transition)
wp.getTransitions()
                                        .generateNewElement();
                           activityid = activityid + 1;
                           tooroptional.setId("" + d + "" + left + "" +
activityid);
                           tooroptional.setFrom(actprevious.getId());
                           tooroptional.setTo(actd.getId());
                           wp.getTransitions().add(tooroptional);

                           d = cal2.getTimeInMillis();
                           Transition optionaltoend1 = (Transition)
wp.getTransitions()
                                        .generateNewElement();
                           activityid = activityid + 1;
                           optionaltoend1.setId("" + d + "" + left + "" +
activityid);
                           optionaltoend1.setFrom(actd.getId());
                           optionaltoend1.setTo(actd1.getId());
                           wp.getTransitions().add(optionaltoend1);

                           String insideTerm = term1.substring(1, term1.length()
- 1);
                           System.out.println("the inside term is:" +
insideTerm);
                           int listpointer2 = listpointer;
                           listpointer = listpointer + 1;
                           firstacts.add(actd);
                           Activity xbrackets = createProcess(insideTerm, wp,
actd);

                           d = cal2.getTimeInMillis();
                           Transition xbracketstooptional = (Transition) wp
                                         .getTransitions().generateNewElement();
                           activityid = activityid + 1;
                           xbracketstooptional.setId("" + d + "" + left + "" +
activityid);
                           xbracketstooptional.setFrom(xbrackets.getId());
                           xbracketstooptional.setTo(actd1.getId());
                           wp.getTransitions().add(xbracketstooptional);

                           firstacts.remove(listpointer);
                           listpointer = listpointer2;

                           actprevious = actd1;

                    } else if (foreverTermMatcher.find()
                                  && (foreverTermMatcher.group().length() ==
term1.length())) {
                           // activity x~
                           System.out.println("A tilda term is being
processed");
                           String insideTerm = term1.substring(0, term1.length()
- 1);
                           Activity tilda = createProcess(insideTerm, wp,
actprevious);
                           System.out.println();
                           Transition tildatostart = (Transition)
wp.getTransitions()
                                         .generateNewElement();
                           activityid = activityid + 1;
                           tildatostart.setId("" + d + "" + left + "" +
activityid);
                           tildatostart.setFrom(tilda.getId());

       tildatostart.setTo(firstacts.get(listpointer).getId());
                           wp.getTransitions().add(tildatostart);

                           actprevious = tilda;
                    } else if (oneOrMoreTimesTermMatcher.find()
                                  && (oneOrMoreTimesTermMatcher.group().length()
== term1
                                                .length())) {
                           // activity x+
                           System.out.println("A plus term is being processed");
                           String insideTerm = term1.substring(0, term1.length()
- 1);

                           d = cal2.getTimeInMillis();
                           activityid = activityid + 1;
                           Activity xplusgate = (Activity) wp.getActivities()
                                         .generateNewElement();
                           xplusgate.setId("" + d + "" + left + "" +
activityid);
```

```
                           xplusgate.setName("X1:" + term1);
                           xplusgate.getActivityTypes().setImplementation();
                           xplusgate.getActivityTypes().getImplementation()
                                        .getImplementationTypes().setNo();
                           NodeGraphicsInfo ngiact4 = (NodeGraphicsInfo)
xplusgate

       .getNodeGraphicsInfos().generateNewElement();
                           ngiact4.setShape("annotation");
                           ngiact4.setLaneId(getPoolname());
                           ngiact4.getCoordinates().setXCoordinate(
                                        "" + Xcoordinate + 200 + "");
                           ngiact4.getCoordinates().setYCoordinate(
                                        "" + Ycoordinate + 40 + "");
                           ngiact4.setHeight(43);
                           ngiact4.setWidth(31);
                           ngiact4.setShape("annotation");
                           ngiact4.setBorderColor("0,0,0");
                           ngiact4.setFillColor("255,229,124");
                           xplusgate.getNodeGraphicsInfos().add(ngiact4);

                           d = cal2.getTimeInMillis();
                           activityid = activityid + 1;
                           System.out
                                        .println(".........creating
Activity[Id=actd,Name=Decision,Type=Route,Split-type=Exclusive]");
                           Activity actd = (Activity) wp.getActivities()
                                        .generateNewElement();
                           actd.setId("" + d + "" + left + "" + activityid);
                           actd.setName("Decision:" + term1);
                           actd.getActivityTypes().setRoute();
                           System.out
                                        .println("...........creating
TransitionRestriction");
                           TransitionRestriction tre = (TransitionRestriction)
xplusgate

       .getTransitionRestrictions().generateNewElement();
                           tre.getSplit().setTypeExclusive();
                           actd.getTransitionRestrictions().add(tre);
                           System.out
                                        .println("...........creating
NodeGraphicsInfo[LaneId=lane1,Coordinates=200,60]");
                           NodeGraphicsInfo ngid = (NodeGraphicsInfo) actd

       .getNodeGraphicsInfos().generateNewElement();
                           ngid.setLaneId(getPoolname());
                           ngid.setWidth(40);
                           ngid.setHeight(40);
                           ngid.getCoordinates().setXCoordinate("200");
                           ngid.getCoordinates().setYCoordinate("60");
                           actd.getNodeGraphicsInfos().add(ngid);
                           wp.getActivities().add(actd);

                           Activity xplus = createProcess(insideTerm, wp,
actprevious);

                           d = cal2.getTimeInMillis();
```

```
                            Transition xplustogate = (Transition)
wp.getTransitions()
                                          .generateNewElement();
                            activityid = activityid + 1;
                            xplustogate.setId("" + d + "" + left + "" +
activityid);
                            xplustogate.setFrom(xplus.getId());
                            xplustogate.setTo(actd.getId());
                            wp.getTransitions().add(xplustogate);

                            d = cal2.getTimeInMillis();
                            Transition gatetostart = (Transition)
wp.getTransitions()
                                          .generateNewElement();
                            activityid = activityid + 1;
                            gatetostart.setId("" + d + "" + left + "" +
activityid);
                            gatetostart.setFrom(actd.getId());

       gatetostart.setTo(firstacts.get(listpointer).getId());
                            wp.getTransitions().add(gatetostart);

                            actprevious = actd;
                     } else if ((zeroOrMoreTimesTermMatcher.find() &&
(zeroOrMoreTimesTermMatcher
                                   .group().length() == term1.length()))) {
                            // activity x*
                            System.out.println("A star term is being processed");
                            String insideTerm = term1.substring(0, term1.length()
- 1);

                            d = cal2.getTimeInMillis();
                            activityid = activityid + 1;
                            Activity xstargate = (Activity) wp.getActivities()
                                          .generateNewElement();
                            xstargate.setId("" + d + "" + left + "" +
activityid);
                            xstargate.setName("X1:" + term1);
                            xstargate.getActivityTypes().setImplementation();
                            xstargate.getActivityTypes().getImplementation()
                                          .getImplementationTypes().setNo();
                            NodeGraphicsInfo ngiact4 = (NodeGraphicsInfo)
xstargate

       .getNodeGraphicsInfos().generateNewElement();
                            ngiact4.setShape("annotation");
                            ngiact4.setLaneId(getPoolname());
                            ngiact4.getCoordinates().setXCoordinate(
                                          "" + Xcoordinate + 200 + "");
                            ngiact4.getCoordinates().setYCoordinate(
                                          "" + Ycoordinate + 40 + "");
                            ngiact4.setHeight(43);
                            ngiact4.setWidth(31);
                            ngiact4.setShape("annotation");
                            ngiact4.setBorderColor("0,0,0");
                            ngiact4.setFillColor("255,229,124");
                            xstargate.getNodeGraphicsInfos().add(ngiact4);
```

```
                           d = cal2.getTimeInMillis();
                           activityid = activityid + 1;
                           System.out
                                         .println("........creating
Activity[Id=actd,Name=Decision,Type=Route,Split-type=Exclusive]");
                           Activity actd = (Activity) wp.getActivities()
                                         .generateNewElement();
                           actd.setId("" + d + "" + left + "" + activityid);
                           actd.setName("Decision:" + term1);
                           actd.getActivityTypes().setRoute();
                           System.out
                                         .println("...........creating
TransitionRestriction");
                           TransitionRestriction tre = (TransitionRestriction)
xstargate

       .getTransitionRestrictions().generateNewElement();
                           tre.getSplit().setTypeExclusive();
                           actd.getTransitionRestrictions().add(tre);
                           System.out
                                         .println("...........creating
NodeGraphicsInfo[LaneId=lane1,Coordinates=200,60]");
                           NodeGraphicsInfo ngid = (NodeGraphicsInfo) actd

       .getNodeGraphicsInfos().generateNewElement();
                           ngid.setLaneId(getPoolname());
                           ngid.setWidth(40);
                           ngid.setHeight(40);
                           ngid.getCoordinates().setXCoordinate("200");
                           ngid.getCoordinates().setYCoordinate("60");
                           actd.getNodeGraphicsInfos().add(ngid);
                           wp.getActivities().add(actd);

                           d = cal2.getTimeInMillis();
                           activityid = activityid + 1;
                           Activity xstargate2 = (Activity) wp.getActivities()
                                         .generateNewElement();
                           xstargate2.setId("" + d + "" + left + "" +
activityid);
                           xstargate2.setName("X1:" + term1);
                           xstargate2.getActivityTypes().setImplementation();
                           xstargate2.getActivityTypes().getImplementation()
                                         .getImplementationTypes().setNo();
                           NodeGraphicsInfo ngiact5 = (NodeGraphicsInfo)
xstargate2

       .getNodeGraphicsInfos().generateNewElement();
                           ngiact5.setShape("annotation");
                           ngiact5.setLaneId(getPoolname());
                           ngiact5.getCoordinates().setXCoordinate(
                                         "" + Xcoordinate + 200 + "");
                           ngiact5.getCoordinates().setYCoordinate(
                                         "" + Ycoordinate + 40 + "");
                           ngiact5.setHeight(43);
                           ngiact5.setWidth(31);
                           ngiact5.setShape("annotation");
                           ngiact5.setBorderColor("0,0,0");
                           ngiact5.setFillColor("255,229,124");
                           xstargate2.getNodeGraphicsInfos().add(ngiact5);

                           d = cal2.getTimeInMillis();
                           activityid = activityid + 1;
                           System.out
                                        .println(".........creating
Activity[Id=actd,Name=Decision,Type=Route,Split-type=Exclusive]");
                           Activity actd1 = (Activity) wp.getActivities()
                                        .generateNewElement();
                           actd1.setId("" + d + "" + left + "" + activityid);
                           actd1.setName("Decision:" + term1);
                           actd1.getActivityTypes().setRoute();
                           System.out
                                        .println("...........creating
TransitionRestriction");
                           TransitionRestriction tre1 = (TransitionRestriction)
xstargate2

       .getTransitionRestrictions().generateNewElement();
                           tre1.getSplit().setTypeExclusive();
                           actd1.getTransitionRestrictions().add(tre1);
                           System.out
                                        .println("...........creating
NodeGraphicsInfo[LaneId=lane1,Coordinates=200,60]");
                           NodeGraphicsInfo ngid1 = (NodeGraphicsInfo) actd1

       .getNodeGraphicsInfos().generateNewElement();
                           ngid1.setLaneId(getPoolname());
                           ngid1.setWidth(40);
                           ngid1.setHeight(40);
                           ngid1.getCoordinates().setXCoordinate("200");
                           ngid1.getCoordinates().setYCoordinate("60");
                           actd1.getNodeGraphicsInfos().add(ngid1);
                           wp.getActivities().add(actd1);

                           d = cal2.getTimeInMillis();
                           Transition previoustogate1 = (Transition)
wp.getTransitions()
                                        .generateNewElement();
                           activityid = activityid + 1;
                           previoustogate1.setId("" + d + "" + left + "" +
activityid);
                           previoustogate1.setFrom(actprevious.getId());
                           previoustogate1.setTo(actd.getId());
                           wp.getTransitions().add(previoustogate1);

                           d = cal2.getTimeInMillis();
                           Transition gate1togate2 = (Transition)
wp.getTransitions()
                                        .generateNewElement();
                           activityid = activityid + 1;
                           gate1togate2.setId("" + d + "" + left + "" +
activityid);
                           gate1togate2.setFrom(actd.getId());
                           gate1togate2.setTo(actd1.getId());
                           wp.getTransitions().add(gate1togate2);

                           Activity xstar = createProcess(insideTerm, wp, actd);
```

```
                d = cal2.getTimeInMillis();
                Transition startogate2 = (Transition)
wp.getTransitions()
                        .generateNewElement();
                activityid = activityid + 1;
                startogate2.setId("" + d + "" + left + "" +
activityid);
                startogate2.setFrom(xstar.getId());
                startogate2.setTo(actd1.getId());
                wp.getTransitions().add(startogate2);

                d = cal2.getTimeInMillis();
                Transition gate2tostar = (Transition)
wp.getTransitions()
                        .generateNewElement();
                activityid = activityid + 1;
                gate2tostar.setId("" + d + "" + left + "" +
activityid);
                gate2tostar.setFrom(actd1.getId());

    gate2tostar.setTo(firstacts.get(listpointer).getId());
                wp.getTransitions().add(gate2tostar);

                actprevious = actd1;
            } else if (basicTermMatcher.find()) {
                // basic activity x
                if (formulas.keySet().contains(term1)) {
                    Activity act = handleBasicTerm(term1, wp,
actprevious);
                    actprevious = act;
                } else {
                    d = cal2.getTimeInMillis();
                    activityid = activityid + 1;
                    Activity activitytemp = (Activity)
wp.getActivities()
                            .generateNewElement();
                    activitytemp.setId("" + d + "" + left + "" +
activityid);
                    activitytemp.setName(term1);

    activitytemp.getActivityTypes().setImplementation();

    activitytemp.getActivityTypes().getImplementation()

    .getImplementationTypes().setNo();
                    activitytemp.setFirstPerformer(lefthand);
                    NodeGraphicsInfo ngiact2 = (NodeGraphicsInfo)
activitytemp

    .getNodeGraphicsInfos().generateNewElement();
                    ngiact2.setLaneId(getPoolname());
                    ngiact2.setShape("annotation");
                    ngiact2.getCoordinates().setXCoordinate(
                            "" + Xcoordinate + 200 + "");
                    ngiact2.getCoordinates().setYCoordinate(
                            "" + Ycoordinate + 40 + "");

    activitytemp.getNodeGraphicsInfos().add(ngiact2);
```

```
                    wp.getActivities().add(activitytemp);
                    roleandname = rolename + ":" + term1;
                    rolesnames.add(roleandname);

                    firstacts.add(activitytemp);
                    d = cal2.getTimeInMillis();
                    Transition previoustobasic = (Transition) wp

    .getTransitions().generateNewElement();
                    activityid = activityid + 1;
                    previoustobasic.setId("" + d + "" + left + ""
+ activityid);
                    previoustobasic.setFrom(actprevious.getId());
                    previoustobasic.setTo(activitytemp.getId());
                    wp.getTransitions().add(previoustobasic);

                    actprevious = activitytemp;
                }
            }
        }
        return actprevious;
    }

    public List<String> findTermsInExpression(String expression,
            String connector, WorkflowProcess wp) {

        System.out.println("The connector is:" + connector);
        List<String> paraTerms = new LinkedList<String>();
        List<String> orTerms = new LinkedList<String>();
        List<String> seqTerms = new LinkedList<String>();

        if (connector == ".") {
            StringTokenizer t = new StringTokenizer(expression,
connector);
            String currentTerm = new String();
            while (t.hasMoreTokens()) {

                int parenthesis = 0;
                int brackets = 0;
                currentTerm = currentTerm + t.nextToken();
                System.out.println("The currentTerm is:" +
currentTerm);
                for (int i = 0; i < currentTerm.length(); i++) {
                    if (currentTerm.regionMatches(i, "(", 0, 1))
                        parenthesis++;
                    if (currentTerm.regionMatches(i, ")", 0, 1))
                        parenthesis--;
                    if (currentTerm.regionMatches(i, "[", 0, 1))
                        brackets++;
                    if (currentTerm.regionMatches(i, "]", 0, 1))
                        brackets--;
                }
                System.out.println("The currentTerm is:" +
currentTerm);
                if ((parenthesis == 0) && (brackets == 0)) {
                    foundTerms.add(currentTerm);
                    System.out.print("found term: " + currentTerm
+ "\n");
```

```
                    System.out.println("The found terms are:" +
foundTerms);
                    findTerms.add(currentTerm);
                    System.out.println("List is:" + findTerms);

                    if (connector == ".") {
                        numberofseq = numberofseq + 1;

    System.out.println("Numberofsequential:" + numberofseq);
                        seqTerms.add(currentTerm);
                        System.out.println("The seqterms are:"
+ seqTerms);
                        expressionList.add(currentTerm);
                        System.out.println("simpleTerm is:" +
expressionList);
                    } else if (connector == "||") {
                        numberofpara = numberofpara + 1;
                        System.out.println("Numberofparallel:"
+ numberofpara);
                        paraTerms.add(currentTerm);
                        expressionList.add(currentTerm);
                        System.out.println("simpleTerm is:" +
expressionList);
                    } else if (connector == "|") {
                        numberofors = numberofors + 1;
                        System.out.println("Numberofors:" +
numberofors);
                        orTerms.add(currentTerm);
                        expressionList.add(currentTerm);
                        System.out.println("simpleTerm is:" +
expressionList);
                    } else {
                        expressionList.add(currentTerm);
                    }
                    currentTerm = new String();
                } else
                    currentTerm = currentTerm + connector;
                System.out.println("The temp term is:" +
currentTerm);
            }
        } else if (connector == "|") {
            List<String> characterlist = new LinkedList<>();
            int parenthesis = 0;
            int brackets = 0;
            int i = 0;
            System.out.println("The expression is:" + expression);

            for (i = 0; i < expression.length(); i++) {
                String newterm = "";
                char character = expression.charAt(i);

                newterm = newterm + character;
                characterlist.add(newterm);
            }
            System.out.println("The characterlist is:" +
characterlist);

            String tempterm = "";
            String tempop = "";
            int k = 0;
            for (k = 0; k < characterlist.size(); k++) {
                tempterm = tempterm + characterlist.get(k);
                try {
                    tempop = characterlist.get(k + 1);
                } catch (Exception e) {
                    // TODO Auto-generated catch block
                }
                if (characterlist.get(k).equals("(")) {
                    parenthesis = parenthesis + 1;
                } else if (characterlist.get(k).equals(")")) {
                    parenthesis = parenthesis - 1;
                    if (parenthesis == 0) {
                        if (tempop.equals("~") |
tempop.equals("+")
                                | tempop.equals("*")) {
                            orTerms.add(tempterm + tempop);
                            tempterm = "";
                        } else {
                            orTerms.add(tempterm);
                            tempterm = "";
                        }
                    }
                } else if (characterlist.get(k).equals("[")) {
                    brackets = brackets + 1;
                } else if (characterlist.get(k).equals("]")) {
                    brackets = brackets - 1;
                    if (brackets == 0) {
                        if (tempop.equals("~") |
tempop.equals("+")
                                | tempop.equals("*")) {
                            orTerms.add(tempterm + tempop);
                            tempterm = "";
                        } else {
                            orTerms.add(tempterm);
                            tempterm = "";
                        }
                    }
                } else if (characterlist.get(k).equals("|")
                        && (parenthesis == 0 && brackets == 0))
{
                    if (tempterm.equals("|")) {
                        tempterm = "";
                    } else {
                        tempterm = tempterm.substring(0,
tempterm.length() - 1);
                        if (tempterm.equals("~") |
tempterm.equals("+")
                                | tempterm.equals("*")) {
                            tempterm = "";
                        } else {
                            orTerms.add(tempterm);
                            tempterm = "";
                        }
                    }

                } else if (k == characterlist.size() - 1) {
                    if (tempterm.equals("~") |
tempterm.equals("+")
                            | tempterm.equals("*")) {
                        tempterm = "";
                    } else {
                        orTerms.add(tempterm);
                    }
                }

            }
            // parallel
        } else if (connector == "||") {

            List<String> characterlist = new LinkedList<>();
            int parenthesis = 0;
            int brackets = 0;
            int i = 0;
            System.out.println("The expression is:" + expression);

            for (i = 0; i < expression.length(); i++) {
                String newterm = "";
                char character = expression.charAt(i);
                newterm = newterm + character;
                characterlist.add(newterm);
            }
            System.out.println("The characterlist is:" +
characterlist);

            String tempterm = "";
            int k = 0;
            String tempop = "";
            for (k = 0; k < characterlist.size(); k++) {

                tempterm = tempterm + characterlist.get(k);
                try {
                    tempop = characterlist.get(k + 1);
                } catch (Exception e) {
                    // TODO Auto-generated catch block
                    // e.printStackTrace();
                }
                if (characterlist.get(k).equals("(")) {
                    parenthesis = parenthesis + 1;
                } else if (characterlist.get(k).equals(")")) {
                    parenthesis = parenthesis - 1;
                    if (parenthesis == 0) {
                        if (tempterm.startsWith("|")) {
                            tempterm = tempterm.substring(1,
tempterm.length());

                            if (tempop.equals("~") |
tempop.equals("+")
                                    |
tempop.equals("*")) {
                                paraTerms.add(tempterm +
tempop);

                                tempterm = "";
                            } else {
                                paraTerms.add(tempterm);
                                tempterm = "";
                            }
                        }
                    }
                } else if (characterlist.get(k).equals("[")) {
                    brackets = brackets + 1;
                } else if (characterlist.get(k).equals("]")) {
                    brackets = brackets - 1;
                    if (brackets == 0) {
                        if (tempterm.startsWith("|")) {
                            tempterm = tempterm.substring(1,
tempterm.length());
                            if (tempop.equals("~") |
tempop.equals("+")
                                    |
tempop.equals("*")) {
                                paraTerms.add(tempterm +
tempop);
                                tempterm = "";
                            } else {
                                paraTerms.add(tempterm);
                                tempterm = "";
                            }
                        }
                    }
                } else if (characterlist.get(k).equals("|")
                        && characterlist.get(k + 1).equals("|")
                        && ((parenthesis == 0) && (brackets ==
0))) {
                    if (tempterm.equals("")) {
                        tempterm = "";
                    } else if (tempterm.equals("~") |
tempterm.equals("+")
                            | tempterm.equals("*")) {
                        tempterm = "";
                    } else {
                        if (tempterm.startsWith("|") &&
tempterm.endsWith("|")
                                && !(tempterm == "|")) {
                            if (tempterm.equals("~") |
tempterm.equals("+")
                                    |
tempterm.equals("*")) {
                                tempterm = "";
                            } else {
                                if (tempterm.equals("|"))
{
                                    tempterm =
tempterm.substring(0);
                                    tempterm = "";
                                } else {
                                    tempterm =
tempterm.substring(1,

    tempterm.length() - 1);
                                    if
(tempterm.equals("")) {
                                    } else {
```

```
       paraTerms.add(tempterm);
                                                                   tempterm =
"";
                                                            }
                                                      }
                                               }
                                        } else if (!(tempterm.startsWith("|"))
                                                      && tempterm.endsWith("|"))
{
                                               if (tempterm.equals("~") |
tempterm.equals("+")
                                                            |
tempterm.equals("*")) {
                                                      tempterm = "";
                                               } else {
                                                      tempterm =
tempterm.substring(0,

       tempterm.length() - 1);
                                                      if
(tempterm.endsWith("|")) {

                                                            tempterm =
tempterm.substring(0,

       tempterm.length() - 1);
                                                            if
(tempterm.equals("")) {
                                                            } else {

       paraTerms.add(tempterm);
                                                                   tempterm =
"";
                                                            }
                                                      } else {
                                                            if
(tempterm.equals("~")
                                                                          |
tempterm.equals("+")
                                                                          |
tempterm.equals("*")) {
                                                                   tempterm =
"";
                                                            }
                                                            if
(tempterm.endsWith("|")) {
                                                                   tempterm =
tempterm.substring(0,

       tempterm.length() - 1);
                                                                   if
(tempterm.equals("")) {
                                                                   } else {

       paraTerms.add(tempterm);

       tempterm = "";
                                                                   }
```

```
                                                               } else {
                                                                      if
(tempterm.equals("")) {
                                                                      } else {

       paraTerms.add(tempterm);

       tempterm = "";
                                                                      }
                                                               }
                                                        }
                                                 }
                                          } else {
                                                 if (tempterm.equals("~") |
tempterm.equals("+")
                                                               |

tempterm.equals("*")) {
                                                        tempterm = "";
                                                 } else {
                                                        paraTerms.add(tempterm);
                                                 }
                                          }
                                          tempterm = "";
                                   }
                            } else if (k == characterlist.size() - 1) {
                                   if (tempterm.equals("~") |
tempterm.equals("+")
                                                 | tempterm.equals("*")) {
                                          tempterm = "";
                                   } else {
                                          if (tempterm.startsWith("|")) {
                                                 tempterm = tempterm.substring(1,
tempterm.length());
                                                 if (tempterm.equals("")) {
                                                 } else {
                                                        paraTerms.add(tempterm);
                                                 }
                                          } else {
                                                 if (tempterm.equals("~") |
tempterm.equals("+")
                                                               |

tempterm.equals("*")) {
                                                        tempterm = "";
                                                 } else {
                                                        paraTerms.add(tempterm);
                                                        tempterm = "";
                                                 }
                                          }
                                   }
                            }
                     }
              }
              System.out.println("found terms" + foundTerms);
              System.out.println("expressionList" + expressionList);

              if (connector == ".") {
                     return seqTerms;
              } else if (connector == "|") {
```

```
        return orTerms;
    } else if (connector == "||") {
        return paraTerms;
    } else
        return foundTerms;

}

public void setinsideTerm(String insideTerm) {
    insideTerm1 = insideTerm;
}

public String getinsideTerm() {
    return insideTerm1;
}

private String remover(String term1) {

    String testterm1 = term1.replace("~", "");
    String testterm2 = testterm1.replace("*", "");
    String testterm3 = testterm2.replace("(", "");
    String testterm4 = testterm3.replace(")", "");
    String testterm5 = testterm4.replace("[", "");
    String testterm6 = testterm5.replace("]", "");
    String testterm7 = testterm6.replace("|", "");
    String testterm8 = testterm7.replace("+", "");

    return testterm8;
}

public void setconnector(String connector1) {
    connector = connector1;
}

public String getconnector() {
    return connector;
}

public static void writeToFile(String outputFile, Package pkg)
        throws Exception {
    // TODO Auto-generated method stub
    Document document = null;
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    DocumentBuilder dbuilder = dbf.newDocumentBuilder();
    document = dbuilder.newDocument();
    // output stream will either be the FileOutputStream in the
    // case of save as, or the ByteArrayOutputStream if we are
    // saving an existing file
    FileOutputStream os;
    // try to open random access file as rw, if it fails
    // the saving shouldn't occur
    os = new FileOutputStream(outputFile, false);

    // Here we get all document elements set
    XPDLRepositoryHandler repH = new XPDLRepositoryHandler();
    repH.setXPDLPrefixEnabled(true);
    repH.toXML(document, pkg);
```

```
        // Use a Transformer for output
        TransformerFactory tFactory = TransformerFactory.newInstance();
        Transformer transformer = tFactory.newTransformer();
        transformer.setOutputProperty("indent", "yes");
        transformer.setOutputProperty(
                     "{http://xml.apache.org/xslt}indent-amount", "4");
        transformer.setOutputProperty("encoding", "UTF8");
        DOMSource source = new DOMSource(document);
        StreamResult result = new StreamResult(os);
        transformer.transform(source, result);

        os.close();
}

public Activity handleBasicTerm(String term, WorkflowProcess wp,
             Activity actprevious) {

        Activity tmp = createProcess(formulas.get(term), wp, actprevious);
        return tmp;
}

public String getOrId() {
        return orId;
}

public void setOrId(String input) {
        orId = input;
}

public String getBracketsname() {
        return bracketsterm;
}

public void setBracketsname(String name) {
        bracketsterm = name;
}

public String getBrackets() {
        return bracket1;
}

public void setBrackets(String name) {
        bracket1 = name;
}

public String getTerm1() {
        return term123;
}

public void setTerm1(String name) {
        term123 = name;
}

public Transition setendTransition(Transition name) {
        return name;
}
```

```
    public void setPoolname(String poolname) {
        pname = poolname;
    }

    public String getPoolname() {
        return pname;
    }
}
```

The class Liveness2XPDL

```
package aseme.transformations.xpdl;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.LinkedList;
import java.util.List;
import java.util.StringTokenizer;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.eclipse.emf.common.util.URI;
import org.eclipse.emf.ecore.resource.Resource;
import org.eclipse.emf.ecore.resource.ResourceSet;
import org.eclipse.emf.ecore.resource.impl.ResourceSetImpl;
import org.eclipse.emf.ecore.xmi.impl.XMIResourceFactoryImpl;
import org.enhydra.jxpdl.XMLUtil;
import org.enhydra.jxpdl.XPDLRepositoryHandler;
import org.enhydra.jxpdl.elements.Lane;
import org.enhydra.jxpdl.elements.NodeGraphicsInfo;
import org.enhydra.jxpdl.elements.Package;
import org.enhydra.jxpdl.elements.Pool;
import org.enhydra.jxpdl.elements.WorkflowProcess;
import org.w3c.dom.Document;

import SRM.SRMPackage;
import SRM.SRMmodel;

public class Liveness2XPDL {

    Package pkg = null;
    Pool p = null;
    WorkflowProcess wp = null;
    Lane l1 = null;
    int xmiid = 0;

    public static void main(String[] args) throws Exception {
        // TODO Auto-generated method stub

        ResourceSet resourceSet = new ResourceSetImpl();
        resourceSet
```

```
                            .getResourceFactoryRegistry()
                            .getExtensionToFactoryMap()
                            .put(Resource.Factory.Registry.DEFAULT_EXTENSION,
                                          new XMIResourceFactoryImpl());

              // Register the package to ensure it is available during loading.

              resourceSet.getPackageRegistry().put(SRMPackage.eNS_URI,
                            SRMPackage.eINSTANCE);

              // load SRM model

              Resource r = null;
              if (args != null && args.length > 1) {
                     r = resourceSet.getResource(URI.createFileURI(args[0]),
true);
              } else {
                     try {
                            r = resourceSet.getResource(
                                          URI.createFileURI("thesis-Broker.srm"),
true);
                     } catch (Exception e) {
                            // TODO Auto-generated catch block
                            e.printStackTrace();
                     }
              }

              String liveness = null;
              // mms-initial-MeetingsManager.srm
              SRMmodel srm = (SRMmodel) r.getContents().get(0);
              // Liveness2XPDL.liveness = srm.getRoles().get(0).getLiveness();
              // Liveness2XPDL.liveness = Liveness2XPDL.liveness.replaceAll(" ",
"");
              liveness = srm.getRoles().get(0).getLiveness();
              liveness = liveness.replaceAll(" ", "");

              // System.out.println("What is the liveness?" +
Liveness2XPDL.liveness);
              System.out.println("What is the liveness?" + liveness);

              String agent1 = "ComplexProvider=SP~\n"
                            +
"SP=ReceiveRequestMessage.ProcessRequest.SendResponseMessage\n"
                            +
"ProcessRequest=(DecideRouteType.SR.SortRoutes)|(DecidePOITypes.SR.DecidePois.S
R)\n"
                            + "SR=SendRequestMessage.ReceiveResponseMessage";

              String agent2 =
"Testmessager1=ReceiveRequestMessage.ProcessRequest.SendResponseMessage\n"
                            +
"testreceiver2=SendRequestMessage.ReceiveResponseMessage";

              List<String> roles = new LinkedList<String>();
              roles.add(liveness);
              roles.add(agent1);
              roles.add(agent2);

              String filename = "C:/Users/nek/Desktop/Myfirst.xpdl";
```

```
                File f = new File(filename);
                try {
                        filename = f.getCanonicalPath();
                } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                String name = f.getName().substring(0,
                                ((File) f).getName().lastIndexOf("."));
                System.out.println("Creating XPDL Model.\n");

                String id = name;
                if (!XMLUtil.isIdValid(id)) {
                        id = "test";
                }

                System.out.println("...creating Package [Id=" + id + ",Name=" +
name
                                + ",Script-type=text/javascript]");
                Package pkg = new Package();
                pkg.setId(name);
                pkg.setName(name);
                pkg.getPackageHeader().setXPDLVersion("2.1");
                pkg.getPackageHeader().setVendor("TUC");

        pkg.getPackageHeader().setCreated(XMLUtil.getCurrentDateAndTime());

                // pkg.getScript().setType("text/javascript");
                pkg.getScript().setType("http://www.w3.org/1999/XPath");

                pkg = Liveness2XPDL.createRoles(roles, pkg, filename);

                writeToFile(filename, pkg);
                System.out.println("\nWritting XPDL model into file \"" + filename
                                + "\".");
        }

        public static Package createRoles(List<String> roles, Package pkg,
                        String filename) throws Exception {
                Live2xpdl live2xpdl = new Live2xpdl();

                System.out.println("What are the roles" + roles);

                StringTokenizer line2 = new StringTokenizer(roles.get(0), "=");
                String poolname = line2.nextToken();

                System.out.println("What is the pool name:" + poolname);

                System.out.println("......creating WorkflowProcess [Id=" +
poolname
                                + ",Name=" + poolname + "]");
                WorkflowProcess wp = (WorkflowProcess) pkg.getWorkflowProcesses()
                                .generateNewElement();
                wp.setId(poolname);
                wp.setName(poolname);

                pkg.getWorkflowProcesses().add(wp);
                int i = 0;
                for (i = 0; i < roles.size(); i++) {
                        System.out.println("What is the pool name:" + poolname);
                        // to pool mas
                        System.out.println("......creating Pool [Id=" + poolname +
",Name="
                                        + poolname + ",Process=" + poolname + "]");
                        Pool p = (Pool) pkg.getPools().generateNewElement();
                        p.setId(poolname + i);
                        p.setName(poolname);
                        NodeGraphicsInfo ngip = (NodeGraphicsInfo)
p.getNodeGraphicsInfos()
                                        .generateNewElement();
                        ngip.setWidth(350);
                        ngip.setHeight(250);
                        ngip.getCoordinates().setXCoordinate("0");
                        ngip.getCoordinates().setYCoordinate("0");
                        p.getNodeGraphicsInfos().add(ngip);

                        pkg.getPools().add(p);

                        p = live2xpdl.transform(roles.get(i), pkg, filename);
                }

                return pkg;
        }

        public static void writeToFile(String outputFile, Package pkg)
                        throws Exception {
                // TODO Auto-generated method stub
                // System.out.println("PKGEPS=" + pkg.getExternalPackageIds());
                Document document = null;
                DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
                DocumentBuilder dbuilder = dbf.newDocumentBuilder();
                document = dbuilder.newDocument();
                // output stream will either be the FileOutputStream in the
                // case of save as, or the ByteArrayOutputStream if we are
                // saving an existing file
                FileOutputStream os;
                // try to open random access file as rw, if it fails
                // the saving shouldn't occur
                os = new FileOutputStream(outputFile, false);

                // Here we get all document elements set
                XPDLRepositoryHandler repH = new XPDLRepositoryHandler();
                repH.setXPDLPrefixEnabled(true);
                repH.toXML(document, pkg);

                // Use a Transformer for output
                TransformerFactory tFactory = TransformerFactory.newInstance();
                Transformer transformer = tFactory.newTransformer();
                transformer.setOutputProperty("indent", "yes");
                transformer.setOutputProperty(
                                "{http://xml.apache.org/xslt}indent-amount", "4");
                transformer.setOutputProperty("encoding", "UTF8");
                DOMSource source = new DOMSource(document);
                StreamResult result = new StreamResult(os);
                transformer.transform(source, result);

                os.close();
        }
```

```
		public Package getPackage() {
			return pkg;
		}

	}
```

The class Liveness2XPDLApp

```
package aseme.transformations.xpdl;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.DefaultListModel;
import javax.swing.JFileChooser;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JMenu;
import javax.swing.JList;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.SwingConstants;

import java.io.File;
import java.util.LinkedList;
import java.util.List;
import java.util.StringTokenizer;

import org.eclipse.emf.common.util.URI;
import org.eclipse.emf.ecore.resource.Resource;
import org.eclipse.emf.ecore.resource.ResourceSet;
import org.eclipse.emf.ecore.resource.impl.ResourceSetImpl;
import org.eclipse.emf.ecore.xmi.impl.XMIResourceFactoryImpl;
import org.enhydra.jxpdl.elements.Activity;
import org.enhydra.jxpdl.elements.Package;
import org.enhydra.jxpdl.elements.WorkflowProcess;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

import SRM.SRMPackage;
import SRM.SRMmodel;

public class Liveness2XPDLApp extends JFrame implements MouseListener,
		ActionListener {

	/**
	 *
	 */
```

```
private static final long serialVersionUID = -7396635470457189928L;
private static JFrame frame2;
private static String[] arguments;
private JPanel contentPane;
private String input;
static ResourceSet resourceSet = new ResourceSetImpl();
static List<String> listofroles = new LinkedList<String>();
static DefaultListModel<String> model = new DefaultListModel<String>();
public static JList<String> rolelist = null;
private static List<String> miniroles = new LinkedList<String>();
private static List<String> formulalist = new LinkedList<String>();
private static Package Applicationpkg = null;
static String filename = "C:/Users/nek/Desktop/Myfirst.xpdl";

/**
 * Launch the application.
 */
public static void main(String[] args) {
	setArgs(args);
	EventQueue.invokeLater(new Runnable() {
		public void run() {
			try {
				Liveness2XPDLApp frame = new Liveness2XPDLApp();
				setFrame(frame);

				frame.setVisible(true);
			} catch (Exception e) {
				e.printStackTrace();
			}
		}
	});
}

/**
 * Create the frame.
 */
public Liveness2XPDLApp() {
	setTitle("Liveness2XPDL Transformation Application");
	setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
	setBounds(100, 100, 505, 307);

	JMenuBar menuBar = new JMenuBar();
	setJMenuBar(menuBar);

	JMenu mnFile = new JMenu("File");
	menuBar.add(mnFile);

	JMenuItem mntmNewMenuItem_1 = new JMenuItem("Open SRM");
	mntmNewMenuItem_1.setActionCommand("open");
	mntmNewMenuItem_1.addActionListener(this);
	mnFile.add(mntmNewMenuItem_1);

	JMenuItem mntmEditFormula = new JMenuItem("Edit Gaia Formula");
	mntmEditFormula.setActionCommand("edit");
	mntmEditFormula.addActionListener(this);
	mnFile.add(mntmEditFormula);

	JMenuItem mntmDeleteSelectedRoles = new JMenuItem(
			"Delete Selected Role(s)");
	mntmDeleteSelectedRoles.setActionCommand("delete");
mntmDeleteSelectedRoles.addActionListener(this);
mnFile.add(mntmDeleteSelectedRoles);

JMenuItem mntmExit = new JMenuItem("Exit");
mntmExit.setActionCommand("exit");
mntmExit.addActionListener(this);
mnFile.add(mntmExit);

JMenu mnTransform = new JMenu("Transform");
menuBar.add(mnTransform);

JMenuItem mntmSingleRoleTransformation = new JMenuItem(
              "Single Role Transformation");
mntmSingleRoleTransformation.setActionCommand("single");
mntmSingleRoleTransformation.addActionListener(this);
mnTransform.add(mntmSingleRoleTransformation);

JMenuItem mntmMultipleRoleTransformation = new JMenuItem(
              "Multiple Role Transformation");
mntmMultipleRoleTransformation.setActionCommand("multi");
mntmMultipleRoleTransformation.addActionListener(this);
mnTransform.add(mntmMultipleRoleTransformation);

JMenu mnHelp = new JMenu("Help");
menuBar.add(mnHelp);

JMenuItem mntmAbout = new JMenuItem("About");
mntmAbout.setActionCommand("about");
mntmAbout.addActionListener(this);
mnHelp.add(mntmAbout);
contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
contentPane.setLayout(new BorderLayout(0, 0));

JLabel lblListOfRoles = new JLabel("List of Roles");
contentPane.add(lblListOfRoles, BorderLayout.NORTH);
lblListOfRoles.setVerticalAlignment(SwingConstants.TOP);

JScrollPane scrollPane = new JScrollPane();
contentPane.add(scrollPane, BorderLayout.CENTER);

rolelist = new JList<String>();
rolelist.setModel(model);
scrollPane.setViewportView(rolelist);

rolelist.addListSelectionListener(new ListSelectionListener() {
       public void valueChanged(ListSelectionEvent evt) {
              if (evt.getValueIsAdjusting())
                     return;

              miniroles = rolelist.getSelectedValuesList();

              System.out.println("The minirolelist is:" + miniroles);
       }
});

rolelist.addKeyListener(new KeyListener() {

       @Override
```

```
            public void keyPressed(KeyEvent e) {
                // TODO Auto-generated method stub
                if (e.getKeyCode() == KeyEvent.VK_DELETE) {
                    System.out.println("Delete pressed");
                    int count = rolelist.getSelectedIndices().length;
                    for (int i = 0; i < count; i++) {

    model.removeElementAt(rolelist.getSelectedIndex());
                    }
                }
            }

            @Override
            public void keyReleased(KeyEvent arg0) {
                // TODO Auto-generated method stub

            }

            @Override
            public void keyTyped(KeyEvent arg0) {
                // TODO Auto-generated method stub
            }
        });
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        if ("open".equals(e.getActionCommand())) {

            JFileChooser fc = new JFileChooser();
            int returnVal = fc.showOpenDialog(getFrame());
            if (returnVal == JFileChooser.APPROVE_OPTION) {
                File file = fc.getSelectedFile();
                input = file.toString();
            }

            String agent = null;
            try {

                resourceSet
                        .getResourceFactoryRegistry()
                        .getExtensionToFactoryMap()

    .put(Resource.Factory.Registry.DEFAULT_EXTENSION,
                                new XMIResourceFactoryImpl());

                resourceSet.getPackageRegistry().put(SRMPackage.eNS_URI,
                        SRMPackage.eINSTANCE);
                // load SRM model
                Resource r = null;
                if (arguments != null && arguments.length > 1) {
                    r = resourceSet.getResource(
                            URI.createFileURI(arguments[0]), true);
                } else {
                    r = resourceSet.getResource(URI.createFileURI(input),
true);
                }

                SRMmodel srm = (SRMmodel) r.getContents().get(0);
```

```
                           String liveness = srm.getRoles().get(0).getLiveness();
                           liveness = liveness.replace(" ", "");
                           listofroles.add(liveness);

                           StringTokenizer temp = new StringTokenizer(liveness, "=");
                           String leftHandSide = temp.nextToken();
                           agent = "Role:" + leftHandSide;
                           model.addElement(agent);
                           rolelist.setModel(model);
                    } catch (Exception e1) {
                           // TODO Auto-generated catch block
                    }
             } else if ("edit".equals(e.getActionCommand())) {

                    String input = JOptionPane.showInputDialog(null,
                                 "Please insert your liveness formula.");

                    listofroles.add(input);

                    try {
                           StringTokenizer temp = new StringTokenizer(input, "=");
                           String leftHandSide = temp.nextToken();
                           String agent = "Role:" + leftHandSide;

                           model.addElement(agent);
                           rolelist.setModel(model);
                    } catch (Exception e1) {
                           // TODO Auto-generated catch block
                    }

             } else if ("exit".equals(e.getActionCommand())) {
                    getFrame().dispose();
             } else if ("about".equals(e.getActionCommand())) {
                    JOptionPane
                                 .showMessageDialog(
                                               null,
                                               "Welcome to the Liveness2XPDL
transformation application\n"
                                                             + "You can open a SRM file
to be transformed from File-->Open\n"
                                                             + "You can write your own
liveness formula from File-->Edit Gaia Formula\n"
                                                             + "You can delete one or
more roles from File--> Delete Selected Roles\n"
                                                             + "Or by selecting one or
more roles in the list and pressing the delete key\n"
                                                             + "If you want to
transform one role Transform-->Single Role Transformation\n"
                                                             + "If you want to
transform multiple roles Transform -->Multiple Roles Transformation\n"
                                                             + "If you want to write
the xpdl model to a certain file press the 'write to target file' button.");
             } else if ("single".equals(e.getActionCommand())) {

                    JOptionPane.showMessageDialog(null,
                                 "You will create a single role model.");
                    Applicationpkg = new Package();

                    if (miniroles.isEmpty()) {
                           JOptionPane.showMessageDialog(getFrame(),
                                     "Please select a role.");
                    } else {
                           String temp = miniroles.get(0);

                           String temp1 = temp.substring(5);

                           for (int k = 0; k < listofroles.size(); k++) {
                                 if (listofroles.get(k).startsWith(temp1)) {
                                       formulalist.add(listofroles.get(k));

                                 }
                           }
                           try {
                                 Applicationpkg =
Liveness2XPDL.createRoles(formulalist,
                                               Applicationpkg, filename);
                           } catch (Exception e2) {
                                 // TODO Auto-generated catch block
                           }

                           String newfile = "";
                           try {
                                 JFileChooser fc = new JFileChooser();
                                 int returnval = fc.showSaveDialog(getFrame());
                                 if (returnval == JFileChooser.APPROVE_OPTION) {
                                       File file = fc.getSelectedFile();
                                       newfile = file.toString();
                                 }
                                 Liveness2XPDL.writeToFile(newfile, Applicationpkg);
                           } catch (Exception e1) {
                                 // TODO Auto-generated catch block
                                 e1.printStackTrace();
                           }
                           System.out.println("\nWritting XPDL model into file this
\""
                                       + newfile + "\".");

                           miniroles = new LinkedList<String>();
                           formulalist = new LinkedList<String>();
                    }
             } else if ("multi".equals(e.getActionCommand())) {
                    JOptionPane.showMessageDialog(null,
                                 "You will create a multi role model.");

                    Applicationpkg = new Package();

                    if (miniroles.isEmpty()) {
                           JOptionPane.showMessageDialog(getFrame(),
                                       "Please select one or more roles.");
                    } else {
                           System.out.println("What is the miniroles list?" +
miniroles);
                           for (int i = 0; i < miniroles.size(); i++) {
                                 String temp1 = "";
                                 try {
                                       String temp = miniroles.get(i);
                                       temp1 = temp.substring(5);
                                 } catch (Exception e1) {
                                       // TODO Auto-generated catch block
                                       System.out.println("Listener exception");
```

```
                    }

                    for (int k = 0; k < listofroles.size(); k++) {
                        if (listofroles.get(k).startsWith(temp1)) {
                            formulalist.add(listofroles.get(k));
                        }
                    }
                }
                try {
                    System.out
                            .println("What is the formulalist:" +
formulalist);
                    Applicationpkg =
Liveness2XPDL.createRoles(formulalist,
                            Applicationpkg, filename);
                } catch (Exception e2) {
                    // TODO Auto-generated catch block
                }

                WorkflowProcess wp = (WorkflowProcess) Applicationpkg
                        .getWorkflowProcesses().get(0);

                for (int i = 0; i < wp.getActivities().size(); i++) {
                    try {

                        Activity temp = (Activity)
wp.getActivities().get(i);
                        if (temp.getName().startsWith("Send")) {
                            System.out.println("Execution of
messager");
                            Inter_role_messages_definition

    .setthePackage(Applicationpkg);

    Inter_role_messages_definition.main(getArgs());
                            Applicationpkg =
Inter_role_messages_definition
                                    .getthePackage();
                            break;
                        }
                    } catch (Exception e1) {
                        // TODO Auto-generated catch block
                    }
                }
                String newfile = "";
                try {
                    JFileChooser fc = new JFileChooser();
                    int returnval = fc.showSaveDialog(getFrame());
                    if (returnval == JFileChooser.APPROVE_OPTION) {
                        File file = fc.getSelectedFile();
                        newfile = file.toString();
                    }
                    Liveness2XPDL.writeToFile(newfile, Applicationpkg);
                } catch (Exception e1) {
                    // TODO Auto-generated catch block
                }
                System.out.println("\nWritting XPDL model into file \""
                        + newfile + "\".");

                miniroles = new LinkedList<String>();
```

```
                                formulalist = new LinkedList<String>();
                        }
                } else if ("delete".equals(e.getActionCommand())) {
                        int count = rolelist.getSelectedIndices().length;
                        for (int i = 0; i < count; i++) {
                                model.removeElementAt(rolelist.getSelectedIndex());
                        }
                }
        }

        @Override
        public void mouseClicked(MouseEvent arg0) {
                // TODO Auto-generated method stub

        }

        @Override
        public void mouseEntered(MouseEvent arg0) {
                // TODO Auto-generated method stub

        }

        @Override
        public void mouseExited(MouseEvent arg0) {
                // TODO Auto-generated method stub

        }

        @Override
        public void mousePressed(MouseEvent arg0) {
                // TODO Auto-generated method stub

        }

        @Override
        public void mouseReleased(MouseEvent arg0) {
                // TODO Auto-generated method stub

        }

        public static void setFrame(JFrame frame1) {

                frame2 = frame1;
        }

        public static JFrame getFrame() {
                return frame2;
        }

        public static void setArgs(String[] args) {
                // TODO Auto-generated method stub
                arguments = args;
        }

        public static String[] getArgs() {
                return arguments;
        }

}
```

The class Inter_role_messages_definition

```
package aseme.transformations.xpdl;

import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.ItemSelectable;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.DefaultListModel;
import javax.swing.JLabel;
import javax.swing.SwingConstants;
import javax.swing.JComboBox;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JTextArea;
import javax.swing.ImageIcon;
import javax.swing.JScrollPane;
import javax.swing.JList;
import javax.swing.JButton;
import javax.swing.border.LineBorder;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;

import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.io.File;
import java.util.LinkedList;
import java.util.List;
import java.util.StringTokenizer;

import javax.swing.UIManager;

import org.enhydra.jxpdl.elements.Activity;
import org.enhydra.jxpdl.elements.Association;
import org.enhydra.jxpdl.elements.Package;
import org.enhydra.jxpdl.elements.WorkflowProcess;

public class Inter_role_messages_definition extends JFrame {

        /**
         *
         */
        private static final long serialVersionUID = 5657667412211602793L;
        private static Package pkg1;
        private static String sendername;
        private JPanel contentPane;
        private JList<String> list;

        List<String> senderList = new LinkedList<String>();
        List<String> senderid = new LinkedList<String>();
        List<String> receiverList = new LinkedList<String>();
        List<String> receiverid = new LinkedList<String>();
        List<String> tempsenderlist = new LinkedList<String>();
```

```
    private static JFrame frame2;
    static DefaultListModel<String> model = new DefaultListModel<String>();

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Inter_role_messages_definition frame = new
Inter_role_messages_definition();
                    frame.setBounds(0, 0, 800, 400);
                    frame.setVisible(true);
                    setFrame(frame);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
     */
    public Inter_role_messages_definition() {

        setBackground(Color.white);
        setFont(UIManager.getFont("List.font"));
        setTitle("Inter-role Messages Definition");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        contentPane.setBackground(Color.white);
        contentPane.setLayout(new BorderLayout(10, 0));
        setContentPane(contentPane);

        JPanel panel = new JPanel();
        panel.setBackground(Color.WHITE);
        contentPane.add(panel, BorderLayout.WEST);
        panel.setLayout(new BorderLayout(0, 0));

        JPanel panel_2 = new JPanel();
        panel_2.setBackground(Color.WHITE);
        panel.add(panel_2, BorderLayout.CENTER);
        panel_2.setLayout(new BorderLayout(0, 0));

        JLabel lblAgentactivity = new JLabel("Agent:Activity");
        panel_2.add(lblAgentactivity, BorderLayout.NORTH);
        lblAgentactivity.setHorizontalAlignment(SwingConstants.CENTER);

        JPanel panel_6 = new JPanel();
        panel_6.setBackground(Color.WHITE);
        panel_2.add(panel_6, BorderLayout.CENTER);
        panel_6.setLayout(new BorderLayout(0, 0));

        JComboBox<String> comboBox = new JComboBox<String>();
```

```
              panel_6.add(comboBox, BorderLayout.NORTH);

              JPanel panel_3 = new JPanel();
              panel_3.setBackground(Color.WHITE);
              panel.add(panel_3, BorderLayout.SOUTH);
              panel_3.setLayout(new BorderLayout(0, 0));

              JTextArea txtrWelcomeToThe = new JTextArea();
              panel_3.add(txtrWelcomeToThe, BorderLayout.CENTER);
              txtrWelcomeToThe
                             .setText("Welcome to the messager.\r\nHere you can
create the messages between the roles.\r\nFirst choose a sending activity from
the combo box.\r\nThen choose one or more possible receiving
activities.\r\nThen click add "
                                            + "message receiving activities.");

              JLabel lblNewLabel = new JLabel("");
              panel_3.add(lblNewLabel, BorderLayout.SOUTH);
              lblNewLabel.setIcon(new ImageIcon(
                             "C:\\Users\\nek\\Desktop\\Capture.PNG"));

              JPanel panel_1 = new JPanel();
              panel_1.setBackground(Color.WHITE);
              contentPane.add(panel_1, BorderLayout.CENTER);
              panel_1.setLayout(new BorderLayout(0, 0));

              JPanel panel_4 = new JPanel();
              panel_4.setBackground(Color.WHITE);
              panel_4.setBorder(new LineBorder(new Color(0, 0, 0)));
              panel_1.add(panel_4, BorderLayout.CENTER);
              panel_4.setLayout(new BorderLayout(0, 0));

              JLabel lblNewLabel_1 = new JLabel("possible receivers");
              lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
              panel_4.add(lblNewLabel_1, BorderLayout.NORTH);

              JScrollPane scrollPane = new JScrollPane();
              panel_4.add(scrollPane);

              list = new JList<String>();
              panel_4.add(list, BorderLayout.CENTER);

              JButton btnNewButton = new JButton("add message receiving
activities");
              panel_4.add(btnNewButton, BorderLayout.SOUTH);

              JPanel panel_5 = new JPanel();
              panel_5.setBackground(Color.WHITE);
              panel_1.add(panel_5, BorderLayout.SOUTH);
              panel_5.setLayout(new BorderLayout(0, 0));

              JButton btnNewButton_1 = new JButton("save & exit");
              panel_5.add(btnNewButton_1);

              pkg1 = getthePackage();
              WorkflowProcess wp = (WorkflowProcess)
pkg1.getWorkflowProcesses().get(
                             0);
```

```
             int i = 0;
             int k = 0;
             String roleterm = "";
             for (i = 0; i < wp.getActivities().size() - 1; i++)
                    ;
             {
                    while (i == wp.getActivities().size() - 1) {
                           Activity temp = (Activity) wp.getActivities().get(i -
k);
                           k = k + 1;
                           if (temp.getName().startsWith("Send")) {
                                  roleterm = temp.getId().replaceAll("[^a-zA-
Z]+", "");
                                  senderList.add(roleterm + ":" +
temp.getName());
                                  senderid.add(temp.getId());
                           } else if (temp.getName().startsWith("Receive")) {
                                  roleterm = temp.getId().replaceAll("[^a-zA-
Z]+", "");
                                  receiverList.add(roleterm + ":" +
temp.getName());
                                  receiverid.add(temp.getId());
                           }
                           if (k == wp.getActivities().size() - 1) {
                                  break;
                           }
                    }

             }

             System.out.println("Print sender list" + senderList);
             System.out.println("Print sender id" + senderid);
             System.out.println("Print receiver list" + receiverList);
             System.out.println("Print receiver id" + receiverid);

             if (!(senderList.isEmpty())) {

                    int n = 0;
                    for (n = 0; n < senderList.size(); n++) {
                           comboBox.addItem(senderList.get(n));
                    }

                    ActionListener actionListener = new ActionListener() {

                           public void actionPerformed(ActionEvent actionEvent)
{
                                  model = new DefaultListModel<String>();
                                  list.setModel(model);
                                  ItemSelectable is = (ItemSelectable)
actionEvent
                                                .getSource();
                                  String name = selectedString(is);
                                  setsendername(name);
                                  JOptionPane.showMessageDialog(null, "You have
Selected: "
                                                + name);
                                  StringTokenizer t1 = new StringTokenizer(name,
":");
                                  t1.nextToken();
                                  String currentTerm1 = t1.nextToken();
                    System.out.println("What is the sendername?" +
name);
                    int m = 0;
                    for (m = 0; m < receiverList.size(); m++) {
                        String tempname;
                        try {
                            tempname = receiverList.get(m);
                            StringTokenizer t = new
StringTokenizer(tempname,
                                    ":");
                            String currentTerm = new
String();
                            t.nextToken();
                            currentTerm = t.nextToken();
                            if
(currentTerm1.substring(4).equals(

    currentTerm.substring(7))) {

    model.addElement(receiverList.get(m));
                                list.setModel(model);
                            }
                        } catch (Exception e) {
                            // TODO Auto-generated catch
block
                            e.printStackTrace();
                        }
                    }
                }

                //
                private String selectedString(ItemSelectable is) {
                    Object selected[] = is.getSelectedObjects();
                    return ((selected.length == 0) ? "null"
                            : (String) selected[0]);
                }
            };
            comboBox.addActionListener(actionListener);

            btnNewButton_1.addMouseListener(new MouseListener() {
                // @Override
                public void mouseClicked(MouseEvent e) {
                    String newfile = "";
                    try {
                        JFileChooser fc = new JFileChooser();
                        int returnval =
fc.showSaveDialog(frame2);
                        if (returnval ==
JFileChooser.APPROVE_OPTION) {
                            File file = fc.getSelectedFile();
                            newfile = file.toString();
                        }
                        Liveness2XPDL.writeToFile(newfile,
pkg1);
                    } catch (Exception e1) {
                        // TODO Auto-generated catch block
                    }
                    System.out.println("\nWritting XPDL model into
file \""
```

```
                            + newfile + "\".");

                    frame2.dispose();
                }

                //
                // @Override
                public void mouseEntered(MouseEvent e) {
                    // // TODO Auto-generated method stub
                    //
                }

                //
                // @Override
                public void mouseExited(MouseEvent e) {
                    // // TODO Auto-generated method stub
                    //
                }

                //
                // @Override
                public void mousePressed(MouseEvent e) {
                    // // TODO Auto-generated method stub
                    //
                }

                //
                // @Override
                public void mouseReleased(MouseEvent e) {
                    // // TODO Auto-generated method stub
                    //
                }
                //
            });

            btnNewButton.addMouseListener(new MouseListener() {
                // @Override
                public void mouseClicked(MouseEvent e) {
                    // // TODO Auto-generated method stub
                    JOptionPane
                            .showMessageDialog(null,
                                    "You will create a
message please select a sender and one or more receivers.");
                    String sendername = getsendername();
                    System.out.println("The sender is:" +
sendername);
                    String tempsender = "";
                    String tempsenderid = "";
                    int i = 0;
                    for (i = 0; i < senderList.size(); i++) {
                        tempsender = senderList.get(i);
                        if (tempsender == sendername) {
                            tempsenderid = senderid.get(i);
                        }
                    }
                    System.out.println("What is the tempsenderid"
                            + tempsenderid);

                    String tempreceiver = "";
                    String tempreceiverid = "";
```

```
                                  int z = 0;
                                  int k = 0;
                                  int associationid = 1;
                                  for (z = 0; z < receiverList.size(); z++) {
                                        tempreceiver = receiverList.get(z);
                                        for (k = 0; k < tempsenderlist.size();
k++)
                                               if
(tempsenderlist.get(k).equals(tempreceiver)) {
                                                      tempreceiverid =
receiverid.get(z);
                                                      associationid =
associationid + 1;
                                                      System.out

       .println("What is the tempreceiver id"

       + tempreceiverid);
                                                      Association asoc =
(Association) pkg1

       .getAssociations().generateNewElement();
                                                      asoc.setId("" +
associationid);

       asoc.setSource(tempsenderid);

       asoc.setTarget(tempreceiverid);

       asoc.setAssociationDirectionFROM();

       pkg1.getAssociations().add(asoc);
                                               }
                                  }
                                  JOptionPane.showMessageDialog(null,
                                               "Messages created and
submitted");
                                  tempsenderlist = new LinkedList<String>();
                           }

                           //
                           // @Override
                           public void mouseEntered(MouseEvent e) {
                                  // // TODO Auto-generated method stub
                                  //
                           }

                           //
                           // @Override
                           public void mouseExited(MouseEvent e) {
                                  // // TODO Auto-generated method stub
                                  //
                           }

                           //
                           // @Override
                           public void mousePressed(MouseEvent e) {
                                  // // TODO Auto-generated method stub
                                  //
                           }
```

```
                //
                // @Override
                public void mouseReleased(MouseEvent e) {
                    // // TODO Auto-generated method stub
                    //
                }
                //
            });

            list.addListSelectionListener(new ListSelectionListener() {
                public void valueChanged(ListSelectionEvent evt) {
                    if (evt.getValueIsAdjusting())
                        return;
                    tempsenderlist = list.getSelectedValuesList();
                    System.out.println("The minirolelist is:" +
tempsenderlist);
                }

            });

        }
    }

    protected static void setsendername(String name) {
        sendername = name;
    }

    protected static String getsendername() {
        // TODO Auto-generated method stub
        return sendername;
    }

    public static Package getthePackage() {
        // TODO Auto-generated method stub
        return pkg1;
    }

    public static void setthePackage(Package pkg) {
        pkg1 = pkg;
    }

    public static void setFrame(JFrame frame1) {

        frame2 = frame1;
    }

    public static JFrame getFrame() {
        return frame2;
    }

}
```

Appendix B XPDL files

The personal assistant

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```xml
<xpdl:Package xmlns:xpdl="http://www.wfmc.org/2008/XPDL2.1"
xmlns="http://www.wfmc.org/2008/XPDL2.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" Id="Myfirst" Name="Myfirst"
xsi:schemaLocation="http://www.wfmc.org/2008/XPDL2.1
http://www.wfmc.org/standards/docs/bpmnxpdl_31.xsd">
    <xpdl:PackageHeader>
        <xpdl:XPDLVersion>2.1</xpdl:XPDLVersion>
        <xpdl:Vendor>TUC</xpdl:Vendor>
        <xpdl:Created>2014-09-25 13:35:01</xpdl:Created>
    </xpdl:PackageHeader>
    <xpdl:Script Type="http://www.w3.org/1999/XPath"/>
    <xpdl:Participants>
        <xpdl:Participant Id="PersonalAssistant" Name="PersonalAssistant">
            <xpdl:ParticipantType Type="ROLE"/>
        </xpdl:Participant>
    </xpdl:Participants>
    <xpdl:Pools>
        <xpdl:Pool BoundaryVisible="true" Id="PersonalAssistant" MainPool="false"
Name="PersonalAssistant" Orientation="HORIZONTAL">
            <xpdl:NodeGraphicsInfos>
                <xpdl:NodeGraphicsInfo Height="250" IsVisible="true" Width="350">
                    <xpdl:Coordinates XCoordinate="0" YCoordinate="0"/>
                </xpdl:NodeGraphicsInfo>
            </xpdl:NodeGraphicsInfos>
        </xpdl:Pool>
        <xpdl:Pool BoundaryVisible="true" Id="Myfirst_pool1" MainPool="true"
Name="PersonalAssistant" Orientation="HORIZONTAL" Process="PersonalAssistant">
            <xpdl:Lanes>
                <xpdl:Lane Id="Myfirst_pool1_lan1" Name="PersonalAssistant">
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="220,220,220" IsVisible="true" ToolId="JaWE"/>
                    </xpdl:NodeGraphicsInfos>
                    <xpdl:Performers>
                        <xpdl:Performer>PersonalAssistant</xpdl:Performer>
                    </xpdl:Performers>
                </xpdl:Lane>
            </xpdl:Lanes>
            <xpdl:NodeGraphicsInfos>
                <xpdl:NodeGraphicsInfo BorderColor="0,0,0" FillColor="255,255,215"
IsVisible="true" ToolId="JaWE"/>
            </xpdl:NodeGraphicsInfos>
        </xpdl:Pool>
    </xpdl:Pools>
    <xpdl:WorkflowProcesses>
        <xpdl:WorkflowProcess Id="PersonalAssistant" Name="PersonalAssistant">
            <xpdl:ProcessHeader/>
            <xpdl:Activities>
                <xpdl:Activity Id="1411641301247PersonalAssistant1" Name="start">
                    <xpdl:Event>
                        <xpdl:StartEvent Trigger="None"/>
                    </xpdl:Event>
                    <xpdl:Performers>
                        <xpdl:Performer>PersonalAssistant</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="PersonalAssistant" Width="31">
                            <xpdl:Coordinates XCoordinate="50" YCoordinate="60"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="31">
                            <xpdl:Coordinates XCoordinate="220" YCoordinate="29"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411641301274PersonalAssistant5"
Name="SendRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>PersonalAssistant</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="PersonalAssistant" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200" YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="305" YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411641301279PersonalAssistant8"
Name="ReceiveResponseMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>PersonalAssistant</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="PersonalAssistant" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200" YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="440" YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411641301247PersonalAssistant10" Name="end">
                    <xpdl:Event>
                        <xpdl:EndEvent Result="None"/>
                    </xpdl:Event>
                    <xpdl:Performers>
                        <xpdl:Performer>PersonalAssistant</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="PersonalAssistant" Shape="annotation" Width="31">
                            <xpdl:Coordinates XCoordinate="50" YCoordinate="250"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="31">
                            <xpdl:Coordinates XCoordinate="569" YCoordinate="31"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
            </xpdl:Activities>
            <xpdl:Transitions>
                <xpdl:Transition From="1411641301247PersonalAssistant1"
Id="1411641301274PersonalAssistant6" To="1411641301274PersonalAssistant5">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0" IsVisible="true"
ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411641301274PersonalAssistant5"
Id="1411641301279PersonalAssistant9" To="1411641301279PersonalAssistant8">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0" IsVisible="true"
ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411641301279PersonalAssistant8" Id="11"
To="1411641301247PersonalAssistant10">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0" IsVisible="true"
ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
            </xpdl:Transitions>
        </xpdl:WorkflowProcess>
    </xpdl:WorkflowProcesses>
    <xpdl:ExtendedAttributes>
        <xpdl:ExtendedAttribute Name="EDITING_TOOL" Value="Together Workflow Editor"/>
        <xpdl:ExtendedAttribute Name="EDITING_TOOL_VERSION" Value="4.5-1-20120518-
1200-TAB-2.0-1"/>
        <xpdl:ExtendedAttribute Name="JaWE_CONFIGURATION" Value="default"/>
    </xpdl:ExtendedAttributes>
</xpdl:Package>
```

The broker

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xpdl:Package xmlns:xpdl="http://www.wfmc.org/2008/XPDL2.1"
xmlns="http://www.wfmc.org/2008/XPDL2.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Id="Myfirst"
Name="Myfirst" xsi:schemaLocation="http://www.wfmc.org/2008/XPDL2.1
http://www.wfmc.org/standards/docs/bpmnxpdl_31.xsd">
    <xpdl:PackageHeader>
        <xpdl:XPDLVersion>2.1</xpdl:XPDLVersion>
        <xpdl:Vendor>TUC</xpdl:Vendor>
        <xpdl:Created>2014-09-25 14:19:51</xpdl:Created>
    </xpdl:PackageHeader>
    <xpdl:Script Type="http://www.w3.org/1999/XPath"/>
    <xpdl:Participants>
        <xpdl:Participant Id="Broker" Name="Broker">
            <xpdl:ParticipantType Type="ROLE"/>
        </xpdl:Participant>
    </xpdl:Participants>
    <xpdl:Pools>
        <xpdl:Pool BoundaryVisible="true" Id="Broker" MainPool="false"
Name="Broker" Orientation="HORIZONTAL">
            <xpdl:NodeGraphicsInfos>
```

```
                <xpdl:NodeGraphicsInfo Height="250" IsVisible="true"
Width="350">
                    <xpdl:Coordinates XCoordinate="0" YCoordinate="0"/>
                </xpdl:NodeGraphicsInfo>
            </xpdl:NodeGraphicsInfos>
        </xpdl:Pool>
        <xpdl:Pool BoundaryVisible="true" Id="Myfirst_pool1" MainPool="true"
Name="Broker" Orientation="HORIZONTAL" Process="Broker">
            <xpdl:Lanes>
                <xpdl:Lane Id="Myfirst_pool1_lan1" Name="Broker">
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="220,220,220" IsVisible="true" ToolId="JaWE"/>
                    </xpdl:NodeGraphicsInfos>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                </xpdl:Lane>
                <xpdl:Lane Id="Myfirst_pool1_lan2" Name="Expression lane">
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="255,255,215" IsVisible="true" ToolId="JaWE"/>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Lane>
            </xpdl:Lanes>
            <xpdl:NodeGraphicsInfos>
                <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="255,255,215" IsVisible="true" ToolId="JaWE"/>
            </xpdl:NodeGraphicsInfos>
        </xpdl:Pool>
    </xpdl:Pools>
    <xpdl:WorkflowProcesses>
        <xpdl:WorkflowProcess Id="Broker" Name="Broker">
            <xpdl:ProcessHeader/>
            <xpdl:Activities>
                <xpdl:Activity Id="1411643991086Broker1" Name="start">
                    <xpdl:Event>
                        <xpdl:StartEvent Trigger="None"/>
                    </xpdl:Event>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="Broker" Width="31">
                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="60"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="31">
                            <xpdl:Coordinates XCoordinate="174"
YCoordinate="28"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991113Broker5"
Name="ReceiveRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Join Type="Exclusive"/>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true" LaneId="Broker"
Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="305"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991121Broker10" Name="ServiceMatch">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true" LaneId="Broker"
Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="511"
YCoordinate="22"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991123Broker14"
Name="Decision:[(InvokeDataManagement|RequestForServicesSR)]">
                    <xpdl:Route GatewayType="Exclusive"/>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Split Type="Exclusive">
                                <xpdl:TransitionRefs>
                                    <xpdl:TransitionRef
Id="1411643991123Broker18"/>
                                    <xpdl:TransitionRef
Id="1411643991129Broker25"/>
                                </xpdl:TransitionRefs>
                            </xpdl:Split>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Broker" Width="40">
```

```
                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="40">
                            <xpdl:Coordinates XCoordinate="712"
YCoordinate="10"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991123Broker16"
Name="Decision2:[(InvokeDataManagement|RequestForServicesSR)]">
                    <xpdl:Route GatewayType="Exclusive"/>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Join Type="Exclusive"/>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Broker" Width="40">
                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="40">
                            <xpdl:Coordinates XCoordinate="663"
YCoordinate="115"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991129Broker22" Name="Decision">
                    <xpdl:Route GatewayType="Exclusive"/>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Split Type="Exclusive">
                                <xpdl:TransitionRefs>
                                    <xpdl:TransitionRef
Id="1411643991148Broker28"/>
                                    <xpdl:TransitionRef
Id="1411643991154Broker34"/>
                                </xpdl:TransitionRefs>
                            </xpdl:Split>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Broker" Width="40">
                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="40">
                            <xpdl:Coordinates XCoordinate="851"
YCoordinate="4"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991129Broker24" Name="Decision2">
                    <xpdl:Route GatewayType="Exclusive"/>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Join Type="Exclusive"/>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Broker" Width="40">
                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="40">
                            <xpdl:Coordinates XCoordinate="856"
YCoordinate="116"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991148Broker27"
Name="InvokeDataManagement">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true" LaneId="Broker"
Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="986"
YCoordinate="30"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991154Broker33"
Name="SendRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true" LaneId="Broker"
Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="752"
YCoordinate="69"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991155Broker36"
Name="ReceiveResponseMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true" LaneId="Broker"
Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="987"
YCoordinate="104"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991157Broker41"
Name="SendResponseMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Split Type="Parallel">
                                <xpdl:TransitionRefs>
                                    <xpdl:TransitionRef
Id="1411643991102Broker43"/>
                                    <xpdl:TransitionRef Id="45"/>
                                </xpdl:TransitionRefs>
                            </xpdl:Split>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true" LaneId="Broker"
Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="447"
YCoordinate="109"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411643991086Broker44" Name="end">
                    <xpdl:Event>
                        <xpdl:EndEvent Result="None"/>
                    </xpdl:Event>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="Broker" Shape="annotation" Width="31">
                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="250"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="31">
                            <xpdl:Coordinates XCoordinate="265"
YCoordinate="126"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
            </xpdl:Activities>
            <xpdl:Transitions>
                <xpdl:Transition From="1411643991086Broker1"
Id="1411643991113Broker6" To="1411643991113Broker5">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991113Broker5"
Id="1411643991121Broker11" To="1411643991121Broker10">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991121Broker10"
Id="1411643991123Broker17" To="1411643991123Broker14">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991123Broker14"
Id="1411643991123Broker18" To="1411643991123Broker16">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991123Broker14"
Id="1411643991129Broker25" To="1411643991129Broker22">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991129Broker22"
Id="1411643991148Broker28" To="1411643991148Broker27">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991148Broker27"
Id="1411643991129Broker29" To="1411643991129Broker24">
                    <xpdl:ConnectorGraphicsInfos>
```

```
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991129Broker22"
Id="1411643991154Broker34" To="1411643991154Broker33">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991154Broker33"
Id="1411643991155Broker37" To="1411643991155Broker36">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991155Broker36"
Id="1411643991129Broker38" To="1411643991129Broker24">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991129Broker24"
Id="1411643991123Broker39" To="1411643991123Broker16">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991123Broker16"
Id="1411643991157Broker42" To="1411643991157Broker41">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991157Broker41"
Id="1411643991102Broker43" To="1411643991113Broker5">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411643991157Broker41" Id="45"
To="1411643991086Broker44">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
            </xpdl:Transitions>
        </xpdl:WorkflowProcess>
    </xpdl:WorkflowProcesses>
    <xpdl:ExtendedAttributes>
        <xpdl:ExtendedAttribute Name="EDITING_TOOL" Value="Together Workflow
Editor"/>
        <xpdl:ExtendedAttribute Name="EDITING_TOOL_VERSION" Value="4.5-1-
20120518-1200-TAB-2.0-1"/>
```

```
        <xpdl:ExtendedAttribute Name="JaWE_CONFIGURATION" Value="default"/>
    </xpdl:ExtendedAttributes>
</xpdl:Package>
```

The complex provider

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xpdl:Package xmlns:xpdl="http://www.wfmc.org/2008/XPDL2.1"
xmlns="http://www.wfmc.org/2008/XPDL2.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Id="Myfirst"
Name="Myfirst" xsi:schemaLocation="http://www.wfmc.org/2008/XPDL2.1
http://www.wfmc.org/standards/docs/bpmnxpdl_31.xsd">
    <xpdl:PackageHeader>
        <xpdl:XPDLVersion>2.1</xpdl:XPDLVersion>
        <xpdl:Vendor>TUC</xpdl:Vendor>
        <xpdl:Created>2014-09-25 14:47:47</xpdl:Created>
    </xpdl:PackageHeader>
    <xpdl:Script Type="http://www.w3.org/1999/XPath"/>
    <xpdl:Participants>
        <xpdl:Participant Id="ComplexProvider" Name="ComplexProvider">
            <xpdl:ParticipantType Type="ROLE"/>
        </xpdl:Participant>
    </xpdl:Participants>
    <xpdl:Pools>
        <xpdl:Pool BoundaryVisible="true" Id="ComplexProvider" MainPool="false"
Name="ComplexProvider" Orientation="HORIZONTAL">
            <xpdl:NodeGraphicsInfos>
                <xpdl:NodeGraphicsInfo Height="250" IsVisible="true"
Width="350">
                    <xpdl:Coordinates XCoordinate="0" YCoordinate="0"/>
                </xpdl:NodeGraphicsInfo>
            </xpdl:NodeGraphicsInfos>
        </xpdl:Pool>
        <xpdl:Pool BoundaryVisible="true" Id="Myfirst_pool1" MainPool="true"
Name="ComplexProvider" Orientation="HORIZONTAL" Process="ComplexProvider">
            <xpdl:Lanes>
                <xpdl:Lane Id="Myfirst_pool1_lan1" Name="ComplexProvider">
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="220,220,220" IsVisible="true" ToolId="JaWE"/>
                    </xpdl:NodeGraphicsInfos>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                </xpdl:Lane>
                <xpdl:Lane Id="Myfirst_pool1_lan2" Name="Expression lane">
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="255,255,215" IsVisible="true" ToolId="JaWE"/>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Lane>
            </xpdl:Lanes>
            <xpdl:NodeGraphicsInfos>
                <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="255,255,215" IsVisible="true" ToolId="JaWE"/>
            </xpdl:NodeGraphicsInfos>
        </xpdl:Pool>
    </xpdl:Pools>
    <xpdl:WorkflowProcesses>
```

```
        <xpdl:WorkflowProcess Id="ComplexProvider" Name="ComplexProvider">
            <xpdl:ProcessHeader/>
            <xpdl:Activities>
                <xpdl:Activity Id="1411645667374ComplexProvider1" Name="start">
                    <xpdl:Event>
                        <xpdl:StartEvent Trigger="None"/>
                    </xpdl:Event>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="ComplexProvider" Width="31">
                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="60"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="31">
                            <xpdl:Coordinates XCoordinate="378"
YCoordinate="34"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667402ComplexProvider5"
Name="ReceiveRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Join Type="Exclusive"/>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="432"
YCoordinate="6"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667409ComplexProvider10"
Name="Decision">
                    <xpdl:Route GatewayType="Exclusive"/>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Split Type="Exclusive">
                                <xpdl:TransitionRefs>
                                    <xpdl:TransitionRef
Id="1411645667454ComplexProvider35"/>
                                    <xpdl:TransitionRef
Id="1411645667425ComplexProvider18"/>
```

```
                                </xpdl:TransitionRefs>
                            </xpdl:Split>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="ComplexProvider" Width="40">
                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="40">
                            <xpdl:Coordinates XCoordinate="508"
YCoordinate="76"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667409ComplexProvider12"
Name="Decision2">
                    <xpdl:Route GatewayType="Exclusive"/>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Join Type="Exclusive"/>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="ComplexProvider" Width="40">
                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="40" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="40">
                            <xpdl:Coordinates XCoordinate="1813"
YCoordinate="61"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667425ComplexProvider17"
Name="DecideRouteType">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="575"
YCoordinate="105"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667446ComplexProvider22"
Name="SendRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="804"
YCoordinate="101"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667448ComplexProvider25"
Name="ReceiveResponseMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="1019"
YCoordinate="91"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667449ComplexProvider28"
Name="SortRoutes">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="1415"
YCoordinate="104"/>
```

```
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667454ComplexProvider34"
Name="DecidePOITypes">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="575"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667458ComplexProvider39"
Name="SendRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="710"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667459ComplexProvider42"
Name="ReceiveResponseMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="980"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667461ComplexProvider45"
Name="DecidePOIs">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="1250"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667464ComplexProvider50"
Name="SendRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="1437"
YCoordinate="10"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667465ComplexProvider53"
Name="ReceiveResponseMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
```

```
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="1606"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667466ComplexProvider57"
Name="SendResponseMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Split Type="Parallel">
                                <xpdl:TransitionRefs>
                                    <xpdl:TransitionRef
Id="1411645667389ComplexProvider59"/>
                                    <xpdl:TransitionRef Id="61"/>
                                </xpdl:TransitionRefs>
                            </xpdl:Split>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="ComplexProvider" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="1899"
YCoordinate="105"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1411645667374ComplexProvider60" Name="end">
                    <xpdl:Event>
                        <xpdl:EndEvent Result="None"/>
                    </xpdl:Event>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="ComplexProvider" Shape="annotation" Width="31">
                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="250"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="31" IsVisible="true"
LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="31">
                            <xpdl:Coordinates XCoordinate="2055"
YCoordinate="123"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
            </xpdl:Activities>
            <xpdl:Transitions>
                <xpdl:Transition From="1411645667374ComplexProvider1"
Id="1411645667402ComplexProvider6" To="1411645667402ComplexProvider5">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667402ComplexProvider5"
Id="1411645667409ComplexProvider13" To="1411645667409ComplexProvider10">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667409ComplexProvider10"
Id="1411645667425ComplexProvider18" To="1411645667425ComplexProvider17">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667425ComplexProvider17"
Id="1411645667446ComplexProvider23" To="1411645667446ComplexProvider22">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667446ComplexProvider22"
Id="1411645667448ComplexProvider26" To="1411645667448ComplexProvider25">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667448ComplexProvider25"
Id="1411645667449ComplexProvider29" To="1411645667449ComplexProvider28">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667449ComplexProvider28"
Id="1411645667409ComplexProvider30" To="1411645667409ComplexProvider12">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667409ComplexProvider10"
Id="1411645667454ComplexProvider35" To="1411645667454ComplexProvider34">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667454ComplexProvider34"
Id="1411645667458ComplexProvider40" To="1411645667458ComplexProvider39">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667458ComplexProvider39"
Id="1411645667459ComplexProvider43" To="1411645667459ComplexProvider42">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667459ComplexProvider42"
Id="1411645667461ComplexProvider46" To="1411645667461ComplexProvider45">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667461ComplexProvider45"
Id="1411645667464ComplexProvider51" To="1411645667464ComplexProvider50">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667464ComplexProvider50"
Id="1411645667465ComplexProvider54" To="1411645667465ComplexProvider53">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667465ComplexProvider53"
Id="1411645667409ComplexProvider55" To="1411645667409ComplexProvider12">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667409ComplexProvider12"
Id="1411645667466ComplexProvider58" To="1411645667466ComplexProvider57">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667466ComplexProvider57"
Id="1411645667389ComplexProvider59" To="1411645667402ComplexProvider5">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1411645667466ComplexProvider57" Id="61"
To="1411645667374ComplexProvider60">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
```

```
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
            </xpdl:Transitions>
        </xpdl:WorkflowProcess>
    </xpdl:WorkflowProcesses>
    <xpdl:ExtendedAttributes>
        <xpdl:ExtendedAttribute Name="EDITING_TOOL" Value="Together Workflow
Editor"/>
        <xpdl:ExtendedAttribute Name="EDITING_TOOL_VERSION" Value="4.5-1-
20120518-1200-TAB-2.0-1"/>
        <xpdl:ExtendedAttribute Name="JaWE_CONFIGURATION" Value="default"/>
    </xpdl:ExtendedAttributes>
</xpdl:Package>
```

The Multi-Agent Personal Assistant, Broker, Complex Provider with associations

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<xpdl:Package xmlns:xpdl="http://www.wfmc.org/2008/XPDL2.1"
xmlns="http://www.wfmc.org/2008/XPDL2.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Id="Myfirst"
Name="Myfirst" xsi:schemaLocation="http://www.wfmc.org/2008/XPDL2.1
http://www.wfmc.org/standards/docs/bpmnxpdl_31.xsd">

    <xpdl:PackageHeader>

        <xpdl:XPDLVersion>2.1</xpdl:XPDLVersion>

        <xpdl:Vendor>TUC</xpdl:Vendor>

        <xpdl:Created>2014-09-30 16:26:58</xpdl:Created>

    </xpdl:PackageHeader>

    <xpdl:Script Type="http://www.w3.org/1999/XPath"/>

    <xpdl:Participants>

        <xpdl:Participant Id="Broker" Name="Broker">

            <xpdl:ParticipantType Type="ROLE"/>

        </xpdl:Participant>

        <xpdl:Participant Id="ComplexProvider" Name="ComplexProvider">

            <xpdl:ParticipantType Type="ROLE"/>

        </xpdl:Participant>

        <xpdl:Participant Id="PersonalAssistant"
Name="PersonalAssistant">

            <xpdl:ParticipantType Type="ROLE"/>

        </xpdl:Participant>

    </xpdl:Participants>

    <xpdl:Pools>
```

```
        <xpdl:Pool BoundaryVisible="true" Id="Broker" MainPool="false"
Name="Broker" Orientation="HORIZONTAL">

            <xpdl:NodeGraphicsInfos>

                <xpdl:NodeGraphicsInfo Height="250" IsVisible="true"
Width="350">

                    <xpdl:Coordinates XCoordinate="0" YCoordinate="0"/>

                </xpdl:NodeGraphicsInfo>

            </xpdl:NodeGraphicsInfos>

        </xpdl:Pool>

        <xpdl:Pool BoundaryVisible="true" Id="Myfirst_pool1"
MainPool="true" Name="Broker" Orientation="HORIZONTAL" Process="Broker">

            <xpdl:Lanes>

                <xpdl:Lane Id="Myfirst_pool1_lan1" Name="Broker">

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="220,220,220" IsVisible="true" ToolId="JaWE"/>

                    </xpdl:NodeGraphicsInfos>

                    <xpdl:Performers>

                        <xpdl:Performer>Broker</xpdl:Performer>

                    </xpdl:Performers>

                </xpdl:Lane>

                <xpdl:Lane Id="Myfirst_pool1_lan2"
Name="ComplexProvider">

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="220,220,220" IsVisible="true" ToolId="JaWE"/>

                    </xpdl:NodeGraphicsInfos>

                    <xpdl:Performers>

                        <xpdl:Performer>ComplexProvider</xpdl:Performer>

                    </xpdl:Performers>

                </xpdl:Lane>

                <xpdl:Lane Id="Myfirst_pool1_lan3"
Name="PersonalAssistant">

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="220,220,220" IsVisible="true" ToolId="JaWE"/>
```

```
                    </xpdl:NodeGraphicsInfos>

                    <xpdl:Performers>


<xpdl:Performer>PersonalAssistant</xpdl:Performer>

                    </xpdl:Performers>

                </xpdl:Lane>

                <xpdl:Lane Id="Myfirst_pool1_lan4" Name="Expression
lane">

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="255,255,215" IsVisible="true" ToolId="JaWE"/>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Lane>

            </xpdl:Lanes>

            <xpdl:NodeGraphicsInfos>

                <xpdl:NodeGraphicsInfo BorderColor="0,0,0"
FillColor="255,255,215" IsVisible="true" ToolId="JaWE"/>

            </xpdl:NodeGraphicsInfos>

        </xpdl:Pool>

    </xpdl:Pools>

    <xpdl:Associations>

        <xpdl:Association AssociationDirection="From" Id="118"
Name="1412083618667PersonalAssistant111ReceiveRequestMessage"
Source="1412083618667PersonalAssistant111" Target="1412083618576Broker5">

            <xpdl:Object Id="0" Name=""/>

        </xpdl:Association>

        <xpdl:Association AssociationDirection="From" Id="119"
Name="1412083618667PersonalAssistant111ReceiveRequestMessage"
Source="1412083618667PersonalAssistant111"
Target="1412083618625ComplexProvider50">

            <xpdl:Object Id="0" Name=""/>

        </xpdl:Association>

        <xpdl:Association AssociationDirection="From" Id="120"
Name="1412083618667PersonalAssistant111ReceiveRequestMessage"
Source="1412083618667PersonalAssistant111" Target="1412083618576Broker5">

            <xpdl:Object Id="0" Name=""/>

        </xpdl:Association>
        <xpdl:Association AssociationDirection="From" Id="121"
Name="1412083618667PersonalAssistant111ReceiveRequestMessage"
Source="1412083618667PersonalAssistant111" Target="1412083618576Broker5">

            <xpdl:Object Id="0" Name=""/>

        </xpdl:Association>

        <xpdl:Association AssociationDirection="From" Id="122"
Name="1412083618667PersonalAssistant111ReceiveRequestMessage"
Source="1412083618667PersonalAssistant111"
Target="1412083618625ComplexProvider50">

            <xpdl:Object Id="0" Name=""/>

        </xpdl:Association>

        <xpdl:Association AssociationDirection="From" Id="123"
Name="1412083618667PersonalAssistant111ReceiveRequestMessage"
Source="1412083618667PersonalAssistant111"
Target="1412083618625ComplexProvider50">

            <xpdl:Object Id="0" Name=""/>

        </xpdl:Association>

    </xpdl:Associations>

    <xpdl:WorkflowProcesses>

        <xpdl:WorkflowProcess Id="Broker" Name="Broker">

            <xpdl:ProcessHeader/>

            <xpdl:Activities>

                <xpdl:Activity Id="1412083618466Broker1" Name="start">

                    <xpdl:Event>

                        <xpdl:StartEvent Trigger="None"/>

                    </xpdl:Event>

                    <xpdl:Performers>

                        <xpdl:Performer>Broker</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Broker" Width="31">

                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="60"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="31">
```

```
                            <xpdl:Coordinates XCoordinate="170" 
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618576Broker5" 
Name="ReceiveRequestMessage">

                    <xpdl:Implementation>

                        <xpdl:No/>

                    </xpdl:Implementation>

                    <xpdl:Performers>

                        <xpdl:Performer>Broker</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:TransitionRestrictions>

                        <xpdl:TransitionRestriction>

                            <xpdl:Join Type="Exclusive"/>

                        </xpdl:TransitionRestriction>

                    </xpdl:TransitionRestrictions>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo IsVisible="true" 
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200" 
YCoordinate="040"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60" 
IsVisible="true" LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="305" 
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618583Broker10" 
Name="ServiceMatch">

                    <xpdl:Implementation>

                        <xpdl:No/>
                    </xpdl:Implementation>

                    <xpdl:Performers>

                        <xpdl:Performer>Broker</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="710"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618585Broker14"
Name="Decision:[(InvokeDataManagement|RequestForServicesSR)]">

                    <xpdl:Route GatewayType="Exclusive"/>

                    <xpdl:TransitionRestrictions>

                        <xpdl:TransitionRestriction>

                            <xpdl:Split Type="Exclusive">

                                <xpdl:TransitionRefs>

                                    <xpdl:TransitionRef
Id="1412083618585Broker18"/>

                                    <xpdl:TransitionRef
Id="1412083618605Broker25"/>

                                </xpdl:TransitionRefs>

                            </xpdl:Split>

                        </xpdl:TransitionRestriction>

                    </xpdl:TransitionRestrictions>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Broker" Width="40">
```

```
                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Myfirst_pool1_lan4" ToolId="JaWE" Width="40">

                            <xpdl:Coordinates XCoordinate="1115"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618585Broker16"
Name="Decision2:[(InvokeDataManagement|RequestForServicesSR)]">

                    <xpdl:Route GatewayType="Exclusive"/>

                    <xpdl:TransitionRestrictions>

                        <xpdl:TransitionRestriction>

                            <xpdl:Join Type="Exclusive"/>

                        </xpdl:TransitionRestriction>

                    </xpdl:TransitionRestrictions>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Broker" Width="40">

                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Myfirst_pool1_lan4" ToolId="JaWE" Width="40">

                            <xpdl:Coordinates XCoordinate="1655"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618605Broker22"
Name="Decision">

                    <xpdl:Route GatewayType="Exclusive"/>

                    <xpdl:TransitionRestrictions>

                        <xpdl:TransitionRestriction>
```

```
                            <xpdl:Split Type="Exclusive">

                                <xpdl:TransitionRefs>

                                    <xpdl:TransitionRef
Id="1412083618609Broker28"/>

                                    <xpdl:TransitionRef
Id="1412083618613Broker34"/>

                                </xpdl:TransitionRefs>

                            </xpdl:Split>

                        </xpdl:TransitionRestriction>

                    </xpdl:TransitionRestrictions>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Broker" Width="40">

                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Myfirst_pool1_lan4" ToolId="JaWE" Width="40">

                            <xpdl:Coordinates XCoordinate="1655"
YCoordinate="105"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618605Broker24"
Name="Decision2">

                    <xpdl:Route GatewayType="Exclusive"/>

                    <xpdl:TransitionRestrictions>

                        <xpdl:TransitionRestriction>

                            <xpdl:Join Type="Exclusive"/>

                        </xpdl:TransitionRestriction>

                    </xpdl:TransitionRestrictions>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Broker" Width="40">

                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>
```

```
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Myfirst_pool1_lan4" ToolId="JaWE" Width="40">
                            <xpdl:Coordinates XCoordinate="2870"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618609Broker27"
Name="InvokeDataManagement">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="2195"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618613Broker33"
Name="SendRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>Broker</xpdl:Performer>
                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="2195"
YCoordinate="105"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618614Broker36"
Name="ReceiveResponseMessage">

                    <xpdl:Implementation>

                        <xpdl:No/>

                    </xpdl:Implementation>

                    <xpdl:Performers>

                        <xpdl:Performer>Broker</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="3140"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618616Broker41"
Name="SendResponseMessage">
```

```
                    <xpdl:Implementation>

                        <xpdl:No/>

                    </xpdl:Implementation>

                    <xpdl:Performers>

                        <xpdl:Performer>Broker</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:TransitionRestrictions>

                        <xpdl:TransitionRestriction>

                            <xpdl:Split Type="Parallel">

                                <xpdl:TransitionRefs>

                                    <xpdl:TransitionRef Id="45"/>

                                    <xpdl:TransitionRef
Id="1412083618563Broker43"/>

                                </xpdl:TransitionRefs>

                            </xpdl:Split>

                        </xpdl:TransitionRestriction>

                    </xpdl:TransitionRestrictions>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="2060"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618466Broker44" Name="end">

                    <xpdl:Event>

                        <xpdl:EndEvent Result="None"/>

                    </xpdl:Event>
                    <xpdl:Performers>

                        <xpdl:Performer>Broker</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Broker" Shape="annotation" Width="31">

                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="250"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Myfirst_pool1_lan1" ToolId="JaWE" Width="31">

                            <xpdl:Coordinates XCoordinate="2600"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618622ComplexProvider46"
Name="start">

                    <xpdl:Event>

                        <xpdl:StartEvent Trigger="None"/>

                    </xpdl:Event>

                    <xpdl:Performers>

                        <xpdl:Performer>ComplexProvider</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Broker" Width="31">

                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="60"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="31">

                            <xpdl:Coordinates XCoordinate="176"
YCoordinate="108"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618625ComplexProvider50"
Name="ReceiveRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Join Type="Exclusive"/>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="575"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618627ComplexProvider55"
Name="Decision">
                    <xpdl:Route GatewayType="Exclusive"/>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Split Type="Exclusive">
                                <xpdl:TransitionRefs>
```

```
                                    <xpdl:TransitionRef
Id="1412083618641ComplexProvider80"/>

                                    <xpdl:TransitionRef
Id="1412083618632ComplexProvider63"/>

                                </xpdl:TransitionRefs>

                            </xpdl:Split>

                        </xpdl:TransitionRestriction>

                    </xpdl:TransitionRestrictions>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Broker" Width="40">

                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Myfirst_pool1_lan4" ToolId="JaWE" Width="40">

                            <xpdl:Coordinates XCoordinate="980"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618627ComplexProvider57"
Name="Decision2">

                    <xpdl:Route GatewayType="Exclusive"/>

                    <xpdl:TransitionRestrictions>

                        <xpdl:TransitionRestriction>

                            <xpdl:Join Type="Exclusive"/>

                        </xpdl:TransitionRestriction>

                    </xpdl:TransitionRestrictions>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Broker" Width="40">

                            <xpdl:Coordinates XCoordinate="200"
YCoordinate="60"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="40"
IsVisible="true" LaneId="Myfirst_pool1_lan4" ToolId="JaWE" Width="40">
```

```
                            <xpdl:Coordinates XCoordinate="3815"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618632ComplexProvider62"
Name="DecideRouteType">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="1520"
YCoordinate="105"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618635ComplexProvider67"
Name="SendRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
```

```
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="1925"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618636ComplexProvider70"
Name="ReceiveResponseMessage">

                    <xpdl:Implementation>

                        <xpdl:No/>

                    </xpdl:Implementation>

                    <xpdl:Performers>

                        <xpdl:Performer>ComplexProvider</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="2465"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618637ComplexProvider73"
Name="SortRoutes">

                    <xpdl:Implementation>

                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="3545"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618641ComplexProvider79"
Name="DecidePOITypes">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="1520"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618643ComplexProvider84"
Name="SendRequestMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="1790"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618644ComplexProvider87"
Name="ReceiveResponseMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
```

```
                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="2330"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618646ComplexProvider90"
Name="DecidePOIs">

                    <xpdl:Implementation>

                        <xpdl:No/>

                    </xpdl:Implementation>

                    <xpdl:Performers>

                        <xpdl:Performer>ComplexProvider</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="3410"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618662ComplexProvider95"
Name="SendRequestMessage">

                    <xpdl:Implementation>

                        <xpdl:No/>

                    </xpdl:Implementation>

                    <xpdl:Performers>

                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="3680"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618662ComplexProvider98"
Name="ReceiveResponseMessage">
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="3950"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618663ComplexProvider102"
Name="SendResponseMessage">
```

```
                    <xpdl:Implementation>
                        <xpdl:No/>
                    </xpdl:Implementation>
                    <xpdl:Performers>
                        <xpdl:Performer>ComplexProvider</xpdl:Performer>
                    </xpdl:Performers>
                    <xpdl:TransitionRestrictions>
                        <xpdl:TransitionRestriction>
                            <xpdl:Split Type="Parallel">
                                <xpdl:TransitionRefs>
                                    <xpdl:TransitionRef
Id="1412083618623ComplexProvider104"/>
                                    <xpdl:TransitionRef Id="106"/>
                                </xpdl:TransitionRefs>
                            </xpdl:Split>
                        </xpdl:TransitionRestriction>
                    </xpdl:TransitionRestrictions>
                    <xpdl:NodeGraphicsInfos>
                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">
                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
                        </xpdl:NodeGraphicsInfo>
                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="90">
                            <xpdl:Coordinates XCoordinate="4220"
YCoordinate="15"/>
                        </xpdl:NodeGraphicsInfo>
                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>
                <xpdl:Activity Id="1412083618622ComplexProvider105"
Name="end">
                    <xpdl:Event>
                        <xpdl:EndEvent Result="None"/>
                    </xpdl:Event>
                    <xpdl:Performers>

                        <xpdl:Performer>ComplexProvider</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Broker" Shape="annotation" Width="31">

                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="250"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Myfirst_pool1_lan2" ToolId="JaWE" Width="31">

                            <xpdl:Coordinates XCoordinate="4355"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618665PersonalAssistant107"
Name="start">

                    <xpdl:Event>

                        <xpdl:StartEvent Trigger="None"/>

                    </xpdl:Event>

                    <xpdl:Performers>


<xpdl:Performer>PersonalAssistant</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Broker" Width="31">

                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="60"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Myfirst_pool1_lan3" ToolId="JaWE" Width="31">

                            <xpdl:Coordinates XCoordinate="170"
YCoordinate="105"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>
                </xpdl:Activity>

                <xpdl:Activity Id="1412083618667PersonalAssistant111"
Name="SendRequestMessage">

                    <xpdl:Implementation>

                        <xpdl:No/>

                    </xpdl:Implementation>

                    <xpdl:Performers>


<xpdl:Performer>PersonalAssistant</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan3" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="440"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618668PersonalAssistant114"
Name="ReceiveResponseMessage">

                    <xpdl:Implementation>

                        <xpdl:No/>

                    </xpdl:Implementation>

                    <xpdl:Performers>


<xpdl:Performer>PersonalAssistant</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo IsVisible="true"
LaneId="Broker" Shape="annotation">

                            <xpdl:Coordinates XCoordinate="0200"
YCoordinate="040"/>
```

```
                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="60"
IsVisible="true" LaneId="Myfirst_pool1_lan3" ToolId="JaWE" Width="90">

                            <xpdl:Coordinates XCoordinate="845"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

                <xpdl:Activity Id="1412083618665PersonalAssistant116"
Name="end">

                    <xpdl:Event>

                        <xpdl:EndEvent Result="None"/>

                    </xpdl:Event>

                    <xpdl:Performers>


<xpdl:Performer>PersonalAssistant</xpdl:Performer>

                    </xpdl:Performers>

                    <xpdl:NodeGraphicsInfos>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Broker" Shape="annotation" Width="31">

                            <xpdl:Coordinates XCoordinate="50"
YCoordinate="250"/>

                        </xpdl:NodeGraphicsInfo>

                        <xpdl:NodeGraphicsInfo Height="31"
IsVisible="true" LaneId="Myfirst_pool1_lan3" ToolId="JaWE" Width="31">

                            <xpdl:Coordinates XCoordinate="1250"
YCoordinate="15"/>

                        </xpdl:NodeGraphicsInfo>

                    </xpdl:NodeGraphicsInfos>

                </xpdl:Activity>

            </xpdl:Activities>

            <xpdl:Transitions>

                <xpdl:Transition From="1412083618466Broker1"
Id="1412083618576Broker6" To="1412083618576Broker5">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
```

```
                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618576Broker5"
Id="1412083618583Broker11" To="1412083618583Broker10">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618583Broker10"
Id="1412083618585Broker17" To="1412083618585Broker14">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618585Broker14"
Id="1412083618585Broker18" To="1412083618585Broker16">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618585Broker14"
Id="1412083618605Broker25" To="1412083618605Broker22">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618605Broker22"
Id="1412083618609Broker28" To="1412083618609Broker27">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>
                <xpdl:Transition From="1412083618609Broker27"
Id="1412083618605Broker29" To="1412083618605Broker24">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618605Broker22"
Id="1412083618613Broker34" To="1412083618613Broker33">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618613Broker33"
Id="1412083618614Broker37" To="1412083618614Broker36">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618614Broker36"
Id="1412083618605Broker38" To="1412083618605Broker24">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618605Broker24"
Id="1412083618585Broker39" To="1412083618585Broker16">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618585Broker16"
Id="1412083618616Broker42" To="1412083618616Broker41">
```

```
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618616Broker41"
Id="1412083618563Broker43" To="1412083618576Broker5">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618616Broker41" Id="45"
To="1412083618466Broker44">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618622ComplexProvider46"
Id="1412083618625ComplexProvider51" To="1412083618625ComplexProvider50">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618625ComplexProvider50"
Id="1412083618627ComplexProvider58" To="1412083618627ComplexProvider55">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618627ComplexProvider55"
Id="1412083618632ComplexProvider63" To="1412083618632ComplexProvider62">
                    <xpdl:ConnectorGraphicsInfos>
```

```
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618632ComplexProvider62"
Id="1412083618635ComplexProvider68" To="1412083618635ComplexProvider67">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618635ComplexProvider67"
Id="1412083618636ComplexProvider71" To="1412083618636ComplexProvider70">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618636ComplexProvider70"
Id="1412083618637ComplexProvider74" To="1412083618637ComplexProvider73">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618637ComplexProvider73"
Id="1412083618627ComplexProvider75" To="1412083618627ComplexProvider57">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618627ComplexProvider55"
Id="1412083618641ComplexProvider80" To="1412083618641ComplexProvider79">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
```

```
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618641ComplexProvider79"
Id="1412083618643ComplexProvider85" To="1412083618643ComplexProvider84">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618643ComplexProvider84"
Id="1412083618644ComplexProvider88" To="1412083618644ComplexProvider87">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618644ComplexProvider87"
Id="1412083618646ComplexProvider91" To="1412083618646ComplexProvider90">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618646ComplexProvider90"
Id="1412083618662ComplexProvider96" To="1412083618662ComplexProvider95">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618662ComplexProvider95"
Id="1412083618662ComplexProvider99" To="1412083618662ComplexProvider98">
                    <xpdl:ConnectorGraphicsInfos>
                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>
                    </xpdl:ConnectorGraphicsInfos>
                </xpdl:Transition>
                <xpdl:Transition From="1412083618662ComplexProvider98"
Id="1412083618627ComplexProvider100" To="1412083618627ComplexProvider57">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618627ComplexProvider57"
Id="1412083618663ComplexProvider103"
To="1412083618663ComplexProvider102">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618663ComplexProvider102"
Id="1412083618623ComplexProvider104" To="1412083618625ComplexProvider50">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618663ComplexProvider102"
Id="106" To="1412083618622ComplexProvider105">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618665PersonalAssistant107"
Id="1412083618667PersonalAssistant112"
To="1412083618667PersonalAssistant111">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>
                <xpdl:Transition From="1412083618667PersonalAssistant111"
Id="1412083618668PersonalAssistant115"
To="1412083618668PersonalAssistant114">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

                <xpdl:Transition From="1412083618668PersonalAssistant114"
Id="117" To="1412083618665PersonalAssistant116">

                    <xpdl:ConnectorGraphicsInfos>

                        <xpdl:ConnectorGraphicsInfo FillColor="0,0,0"
IsVisible="true" ToolId="JaWE"/>

                    </xpdl:ConnectorGraphicsInfos>

                </xpdl:Transition>

            </xpdl:Transitions>

        </xpdl:WorkflowProcess>

    </xpdl:WorkflowProcesses>

    <xpdl:ExtendedAttributes>

        <xpdl:ExtendedAttribute Name="EDITING_TOOL" Value="Together
Workflow Editor"/>

        <xpdl:ExtendedAttribute Name="EDITING_TOOL_VERSION" Value="4.5-1-
20120518-1200-TAB-2.0-1"/>

        <xpdl:ExtendedAttribute Name="JaWE_CONFIGURATION"
Value="default"/>

    </xpdl:ExtendedAttributes>

</xpdl:Package>
```