POLITECNICO DI MILANO

Scuola di Ingegneria dell'informazione

Corso di Laurea Specialistica in Ingegneria Informatica

PORTABILITY BENCHMARK OF X-PDL FILES AMONG WFMSs FROM DIFFERENT VENDORS

Supervisor: Prof. Giuseppe Pozzi

Master Thesis by:

Vesa Jonuzi

Matr. 780378

Academic Year 2013 / 2014

Table of contents

1. Introduction 3
   1.1 The need for standardization and its benefits 4
   1.2 Standards 6
      1.2.1 Graphical Standards 6
      1.2.2 Execution Standards 7
      1.2.3 Interchange Standards 7
      1.2.4 Diagnosis Standards 7
   1.3 Why portability is critical? 8
2. State of the Art 9
   2.1. Workflow Management Coalition (WfMC) 9
   2.2. Workflow Reference Model 9
      2.2.1 Workflow Enactment service 11
      2.2.2 Interfaces 12
      2.2.3 Workflow Definition Interchange (Interface 1) 13
   2.3. XML Process Definition Language (XPDL) 14
      2.3.1 XPDL history, weaknesses and advantages 14
      2.3.2 XPDL structure 16
   2.4. BPMN 19
   2.5 Mapping from BPMN to XPDL 20
   2.6. BPMN/XPDL portability 21
      2.6.1 Portability Classes 22
      2.6.2 BPMN Model Portability Validation test 22
   2.7. WFMC vs. DBMS similarities 24
3. Goals of the Thesis 26
4. Design choices 28
5. Results 32
   5.1 The reference process model 32
   5.2 Tests 32
6. Conclusion 51
7. Bibliography 53
8. Glossary 56

Abstract

The evolution of workflow as a technology has covered a number of different product areas which in turn has raised the need for standardization. In order to support the use of workflow and develop interoperability and portability standards, the Workflow Management Coalition was established. The WfMC created the reference model to deal with the standardization. This thesis focuses in the usability of the Interface 1 of the reference model. The interface provides an interchange format the XML Process Definition Language that can support the exchange of the process definitions among different workflow vendors.

During the thesis there were made some tests in order to evaluate whether this standard truly ensures business process model portability among different vendors as well as different XPDL dialects. As a first step a business process model named “Purchase Request” was designed, which served as a reference example throughout the whole project. This model was tested on top of a several commercial and open-source workflow management products previously selected.

Assessments that were made, specifically included these aspects: Design portability that is whether the same model can be developed in approximately the same way in all the tools, Execution portability- if the reference example is properly imported and Translatability- in cases when the model was not properly imported what are the most common portability problems and how we can manage them.

As a result, the tests concluded that although only a single standard was used, a process model with the same content can be interpreted in different ways and so these custom behaviors cannot be easily understood by all parties. As a future work we suggested the adoption of a cross compiler which would be able to automatically translate the source XPDL into a target tool-specific format XPDL.

Sommario

L’evoluzione della tecnologia del workflow ha coperto diverse aree di produzione, rendendo necessario un astandardizzazione. La Workflow Management Coalition (WfMC) è una organizzazione non-profit finalizzata a supportare l’uso dei workflow e a sviluppare standard di interoperabilità e portabilità. La WfMC ha così creato il modello di riferimento (reference model) rivolto alla standardizzazione. La presente tesi si concentra sull’usabilità dell’Interfaccia 1 di tale modello di riferimento. L’interfaccia fornisce un formato di scambio, l’XML Process Definition Language, che può supportare lo scambio di definizioni di processi tra workflow management systems (WfMS) di differenti produttori.

Durante la tesi sono stati svolti alcuni test per valutare se lo standard assicuri effettivamente la portabilità di modelli di processi aziendali tra diversiproduttori, così come tra differenti linguaggi e dialetti di XPDL. Come primo passo, è stato progettato un modello di processo aziendale chiamato “Richiesta d’Acquisto”, utilizzato come esempio di riferimento durante l’intero progetto. Il modello è stato testato con differenti WfMS sia commerciali che open-source, precedentemente selezionati.

Sono state effettuate alcune considerazioni, includendo specificatamente i seguenti aspetti: la portabilità del progetto, ovvero la possibilità che lo stesso modello possa essere sviluppato approssimativamente nello stesso modo nei vari ambienti di progettazione; la portabilità di esecuzione – se l’esempio di riferimento viene importato correttamente nei vari WfMS; la traducibilità, ovvero quali siano i problemi di portabilità più comuni nei casi in cui il modello non venga importato correttamente, e come possano essere gestiti tali problemi.

I risultati del test hanno portato alla conclusione che, nonostante fosse stato usato un solo standard raccomandato dalla WfMS, il modello di processo di riferimento (“Richiesta di Acquisto”) viene interpretato in modi differenti dai vari WfMS, e tali comportamenti differenti non possono essere facilmente previsti.

Come sviluppo futuro, si suggerisce l’adozione di un cross compilatore che possa automaticamente tradurre la definizione di un processo data in un formato XPDL nello specifico formato XPDL caratteristico del WfMS destinazione.

1. Introduction

Today we are facing a very tough war between businesses that starve to remain competitive in the so called industrial and commercial environment. Businesses have to operate at all costs tirelessly for the constantly growing demand. The main strategy that they follow is to maintain the efficiency and effectiveness at the proper level, as a result of which they would be able to increase the quality and customer satisfaction. This strategy must be progressively more focused on fundamental changes with respect to markets, products and services (Rouse, 2005). In information technologies perspective this means supporting effective management of critical changes to existing processes and workflows *and* efficient support for the deployment of new processes and workflows (Caverlee, Bae, & Wu, 2007). In the beginning, those processes were performed manually by individuals who particularly manipulated with physical objects. However, as time goes by and with the introduction of the new technologies these processes started to be replaced by automated ones; now performed with computer programs.

Under the umbrella of the term "workflow", people may be referring to a business process, specification of process software that implements and automates a process, or software that simply supports the coordination and collaboration of people that implement a process (Georgakopoulos & Hornick, 1995). Various concepts attributed to the term workflow are illustrated in Figure 1 (Georgakopoulos & Hornick, 1995).

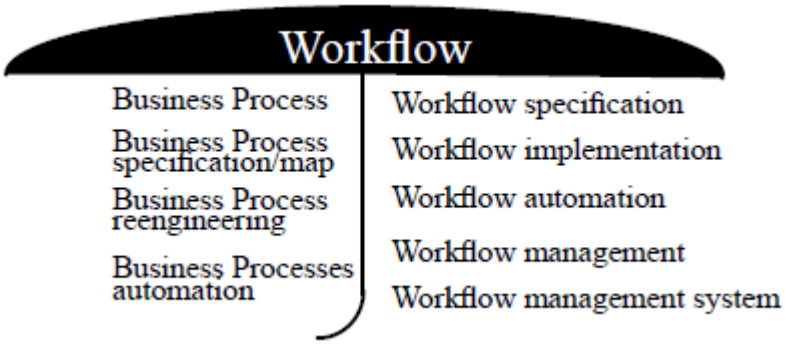


Figure 1 The "Workflow umbrella" (Georgakopoulos & Hornick, 1995)

This concept exists since industrialization, and has evolved from a research that was aimed at increasing efficacy by concentrating on the routine aspects of work activities, they typically separate work activities into well-defined tasks, roles, rules, and procedures which regulate most of the work in manufacturing and the office (Georgakopoulos & Hornick, 1995). Moreover, Workflow management is a technology that has been around since the early nineties (Reijersa & van der Aalstb, 2005) and have become ''one of the most successful genres of systems supporting cooperative working (Dourish, 2001). They help organizations to support their business processes. According to Van der Aalst and Van Hee (van der Aalst & van Hee, 2002) a workflow management system can be compared with an operating system i.e. It controls the workflows between the various resources-people or applications.

Lawrence hypothesis (Lawrence, 1997) would be enough to state why WFMSs should be

adopted. According to him by having a dedicated automated system in place for the logistics management of a business process, such processes could theoretically be executed faster and more efficiently. Based on (Caverlee, Bae, & Wu, 2007) benefits of WfMSs consists of benefits from business and IT perspective. Business benefits are summarized as:

- Process improvement
- Enterprise flexibility - adjust existing processes and to deploy new processes as the business needs change.

Whereas benefits from IT perspective:

- Ease of software development
- Reduction of risk for overall system development

WFM is becoming more than what it is, it is becoming a way to develop new business applications. Advocates point to the fact that 75% of workflow projects succeed while 75% of application development projects fail. It appears that defining a business system in terms of work item flow is easier, and more flexible, than trying to develop the same functionality as bespoke software (Smith & Fingar, 2003).

1.1 The need for standardization and its benefits

Today we see a huge demand in the market for workflow products; in response to this a large number of vendors are providing different workflow solutions. In such a diversified market, organizations wishing to invest in modeling and automating processes want to feel safe about their investment. By that we mean that they want to function efficiently well without being reliant to a single vendor. On the other hand, the lack of predefined principles during the workflow modeling process may lead to misunderstandings. This might be the case when there would be a need for a model exchange between other business partners or consultants. Different people may have different way of expressions, thus they might have difficulties understanding and using models developed by others. Furthermore, these models have to be deployed, which means that they have to be fully understandable also from the deployment environment. As a matter of fact there was a need for standards that actually enables more efficient use of workflows. A definition given by ITI (ITU Telecommunication Standardization Sector) states clearly the role of a standard: *"'Open Standards' are standards made available to the general public and are developed (or approved) and maintained via a collaborative and consensus-driven process. 'Open Standards' facilitate interoperability and data exchange among different products or services and are intended for widespread adoption."*

There are many benefits that result from adoption of standards. Khan categorizes them into the following six categories (M. Khan, 2005):

- **Further understanding of BPM:** promotes the use of standard terminology, reduces confusion and makes easier for customers to compare competitive products.
- **Make it Easier to Develop Business Processes:** by developing a standard language that is widely used, it will be easier for companies to develop, maintain, and upgrade business processes.
- **Facilitate Interaction among BPM Systems from Different Vendors:** A large company may use a different BPM system for different types of processes. If both systems use the same underlying language for defining the processes, it will make it easier to support their interactions.
- **Enable Migration and Transportability of BPM Processes:** they will allow companies to easily migrate processes from one BPM system to another. Customers are no longer tied to one vendor.
- **Reduce Cost for the Customer by Increasing Competition:** By increasing understanding, among BPM systems, and enabling the migration of processes from one system to another, the emergence of BPM standards will reduce cost to the customer.
- ***Develop more Robust Processes based on a Solid Mathematical Foundation:*** languages for expressing processes are based on a solid mathematical foundation of *Pi Calculus*. The claim is that because of the mathematical foundations, business processes that are defined by using these languages are robust and reliable, as compared to processes that are not defined by using Pi Calculus (Smith & Fingar, 2003).

1.2 Standards

There are several organizations that are publishing workflow standards such as: Object Management Group, the Workflow Management Coalition, and the Business Process Management Initiative (in June 2005, BPMI and OMG announced the merger of their BPM activities (OMG), and the Organization for the Advancement of Structured Information Standards.

Ko at al. (K L Ko, S G Lee, & Lee, 2009) classify standards based on process lifecycle and divide them into four groups:

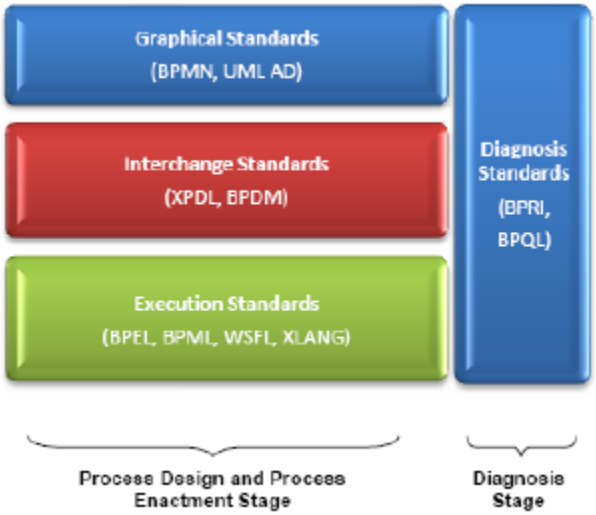


Figure 2 Classification of standards (K L Ko, S G Lee, & Lee, 2009)

1.2.1 Graphical Standards

Above all the standards, graphical standards are the most readable and easy to understand standards that do not require prior technical knowledge. They are used to express the work flow, the decision points and other process meanings graphically.

The most common standards that belong to this group are:

- Unified Modeling Language Activity Diagrams (UML AD) defined by Object Management Group (OMG, UML 2.0 Superstructure Specification, 2004)
- Business Process Modeling Notation (BPMN) defined by Object Management Group (OMG, Business Process Modeling Notation (BPMN), 2004), we will elaborate this standard in more detail in the following chapter.
- Event- driven Process Chains developed by the Institute for Information Systems (Scheer, 1992)

1.2.2 Execution Standards

Ko at al. define execution standards as those that enable business process designs to be deployed and automated. There are currently two important execution standards:

- Business Process Modeling Language (BPML) developed by Business Process Management Initiative (BPMI)
- Business Process Execution Language (BPEL) standardized by Organization for the Advancement of Structured Information Standards (OASIS) in 2004.

1.2.3 Interchange Standards

According to Mendling and Neumann (Mendling & Neumann, 2005), interchange standards are needed to translate graphical standards to execution standards and exchange business process models between different systems. There are currently two prominent interchange standards:

- Business Process Definition Metamodel (BPDM) by OMG 2008 (OMG, Buiness Porcess Definition Metamodel (BPDM), 2008)
- XML Process Definition Language by the Workflow Management Coalition (WfMC) at 1995 (WfMC). We will go further about this standard in the following chapter.

1.2.4 Diagnosis Standards

Diagnosis Standards monitor and optimize business processes running in and across companies' systems. Those are:

- Business Process Runtime Interface (BPRI) developed by OMG at 2002.
- Business Process Query Language (BPQL) developed by BPMI respectively OMG.

1.3 Why portability is critical?

Software portability is very crucial and it has been chosen as one of the most prominent characteristics of software quality (Ortega, P' erez, & Rojas, 2003). A product is considered portable if it is significantly less expensive to adapt the product to run on the new computer than to write a new product from scratch (MOONEY, 1990). In our situation this is the case regarding process models i.e. we want to be able to shift them between different editor vendors as well as engine vendors with no or limited attempt to rewrite them.

In the case of workflow management systems portability is realized through XPDL standard, developed by WfMC. It is able to port all graphical information as well as executable aspects.

1.4Thesis outline

The goal of this master thesis is to test whether this standard truly ensures XPDL portability among different vendors and different XPDL dialects. In the following sections we shall go through WfMC's definition of XPDL, including the workflow reference model with a greater focus on the Interface 1. Next, we shall explain XPDL and BPMN evolutions and their structures. Chapter 3 will go through the goals and the main obstacles that we have faced during the testing phase. Chapter 4 will consider the design choices regarding the workflow editors and the engines. Chapter 5 and 6 will present the process model of reference and the achieved results. Finally, Chapter 7 draws some conclusions, depicts the lessons learnt and highlights some tips on the future research directions, including how to develop an automatic translator for XPDL.

2. State of the Art

2.1. Workflow Management Coalition (WfMC)

The evolution of workflow as a technology has covered a number of different product areas which in turn has raised the need for standardization. Workflow management coalition has been founded in august 1993 as an international non-profit organization. The aim of the WfMC is to support the use of workflow through developing interoperability and connectivity standards and to define a common terminology for workflow vendors. Today WfMC has around 300 members being those developers, consultants, analysts, as well as university and research groups (WfMC).

According to the WfMC, workflow is defined as: “The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.” The definition is very broad; it can include different contexts of workflow implementations. Participants can be humans or computers, tasks can be of different kind and rules are preliminary defined to be tailored to the specific process. Based on this, WfMC defines the workflow management system as a technology that is able to develop and maintain communication and coordination between components of different workflows: “A system that defines, creates, and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with other workflow participants and, where required, invoke the use of IT tools and applications.”

2.2. Workflow Reference Model

The WfMC created the reference model to deal with the standardization. The workflow reference model provides three guidelines (Chang, 2006).The first guideline is a common terminology which will help customers to understand the workflow product. The second guideline is about the functional components that are necessary in a WFMS. This guideline is about the approach how workflow vendors should design their workflow products which in case of successful implementation may increase the interoperability. Finally, the third guideline is the set of interfaces that connect the various functional components.

Regardless of their variability workflow management systems demonstrate some common characteristics based on which WfMC determines three common functional areas (WfMC, 1995):

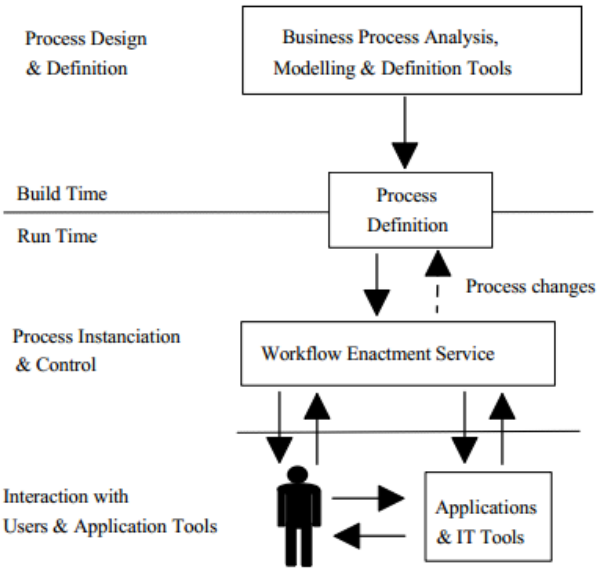


Figure 3 Workflow System Characteristics (WfMC, 1995)

- the Build-time functions, concerned with defining, and possibly modeling, the workflow process and its constituent activities
- the Run-time control functions concerned with managing the workflow processes in an operational environment and sequencing the various activities to be handled as part of each process
- the Run-time interactions with human users and IT application tools for processing the various activity steps

Taking into account these common characteristics and using the major components of generic workflow management system architecture (figure 4); WfMC has created a reference model (figure 5) which also includes interfaces that allow a standardized way of interaction between components.

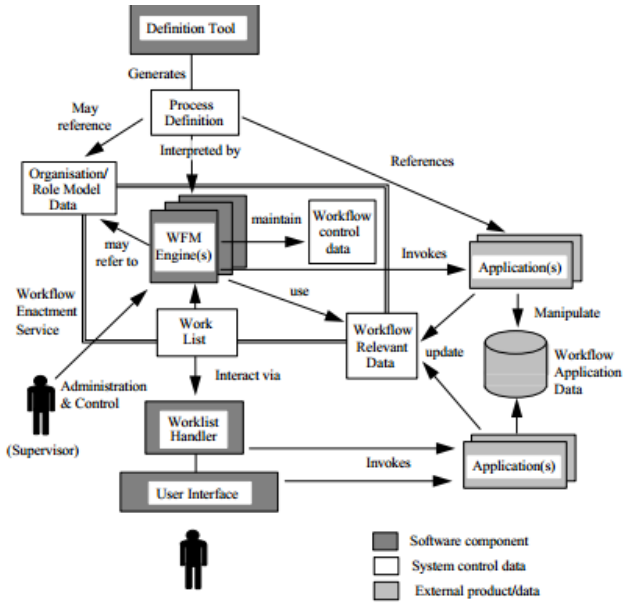


Figure 4 Generic Workflow Product (WfMC, 1995)

2.2.1 Workflow Enactment service

Workflow Enactment is the core of the workflow reference model which actually controls and coordinates the whole model. The workflow management coalition defines the workflow enactment service as a software service that may consist of one or more workflow engines in order to create manage and execute a workflow process (WfMC, 1995). A workflow engine is responsible for part (or all) of the runtime control environment within an enactment service. WfMC defines Workflow Engine as A software service or "engine" that provides the run time execution environment for a workflow instance.

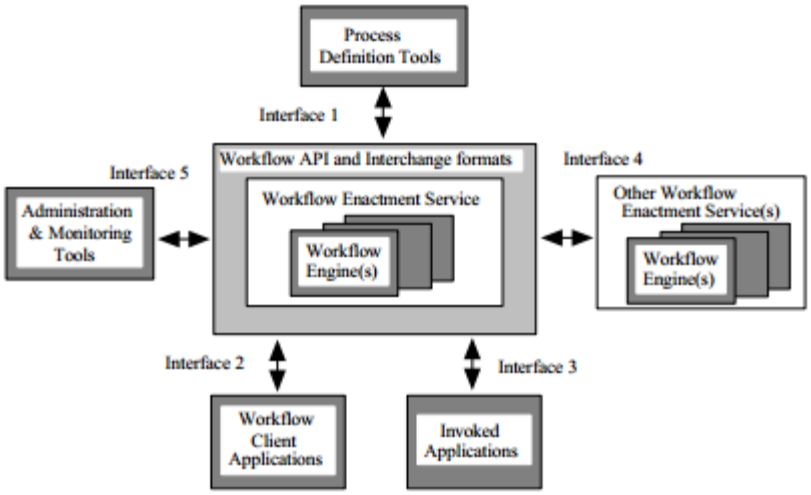


Figure 5 Workflow Reference Model – Components and Interfaces (WfMC, 1995)

2.2.2 Interfaces

2.2.2.1 Process Definition Tools

There exists a variety of tools that are able to design and document business processes. Those tools might be provided within the workflow solution product or separately. In the former case, the process definition is implemented within the workflow product domain whereas in the latter one the process definition might need to be transferred between different products or stored in a place where it can be proficiently accessed. The workflow model is not aware about the nature of these tools, nor the approach how they interact with the enactment service. The workflow definition interchange job is to assure that their process definitions are efficiently deployed within the workflow engine. The interface between the modeling and definition tools and the runtime workflow management software is termed the process definition import/export interface (WfMC, 1995).

2.2.2.2 Workflow Client Applications

Workflow Client Application provides an interface to the users who participate in workflow execution (van der Aalst & van Hee, 2002). It includes the work list manager whose primary job is to provide to the workflow participant list of tasks or the so called work items that may be executed in the process. Chang (Chang, 2006) depicts it with a good example. In his scenario a workplace portal acts as a workflow client application which extracts the work items from the workflow engine and presents them to each user for action. Users can select tasks and invoke the workflow engine to start execution of a task. This is realized through interface 2 for the interactions between workflow engine and the workflow client application (WfMC, 1995).

2.2.2.3 Invoked Applications

Execution of a process might require an invocation of other external applications. In a workflow process, the invoked application is a system participant (Chang, 2006). Invoked applications include interactive applications (need human intervention) as well as fully automated applications (van der Aalst & van Hee, 2002). The engine invokes the applications through interface 3 of the Workflow Reference Model (WfMC, 1995).

2.2.2.4 Other workflow enactment services

Workflow management systems might consist of several workflow engines. In these cases a key objective of the coalition is to define standards that will allow workflow systems produced by different vendors to pass work items seamlessly between one another through interface 4 (WfMC, 1995).

2.2.2.5 Administration and monitoring tools

These are typically tools that allow system administrators to manage WFMS. They include the addition or removal of the resources as well as statistical data about the processes. WfMC proposed a common interface standard for administration and monitoring functions which will allow one vendor's management application to work with another's engines (WfMC, 1995).

2.2.3 Workflow Definition Interchange (Interface 1)

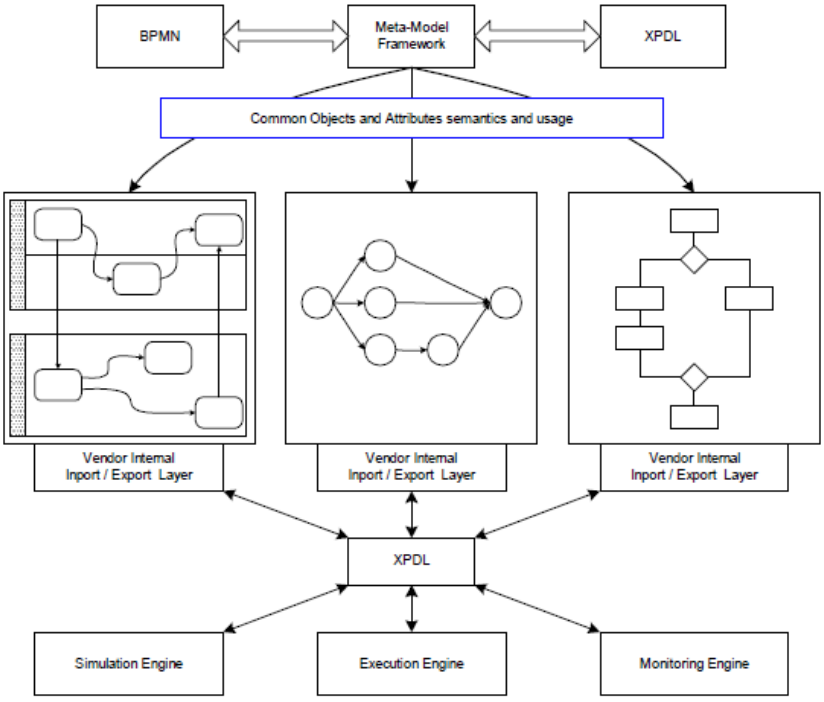


Figure 6 The concept of the Process Definition Interchange (WfMC, 2005)

The interface that links the process definition tool with the workflow engine is termed as the process definition import/export interface (WfMC, 1995). The interface provides an interchange format that can support the exchange of the process definition among different workflow vendors. Process definition is defined from WfMC as follows “The representation of a business process in a form which supports automated manipulation, such as modeling, or enactment by a workflow management system. There are several reasons why a process definition would require to be exchanged. The first and most common case is when a process definition has to be deployed to a workflow engine, thus there must exist a connection between the modeling tool and the execution environment. A further case might be when there is a necessity to change the design tool and finally there might be a need to exchange a process definition from one engine to another. In all of the cases there is a need for a common interchange format. This format must be derived from the common set off objects, relationships and attributes expressing its underlying concepts (WfMC, 2005).

In 1999 WfMC represented a standard language called Workflow Process Definition Language (WfMC, 1999). The Workflow Process Definition Language (WPDL) provides a formal language for the definition and exchange of a process definition using the objects and attributes defined within the meta-model (WfMC, 1998).

With the introduction of XML in 2002, WPDL was substituted with XML process definition language (XPDL) (WfMC, 2005)

2.3. XML Process Definition Language (XPDL)

2.3.1 XPDL history, weaknesses and advantages

XDPL was presented by WfMC as a successor of WPDL from after the manifestation of the XML as an interchange format. Its goal is to ensure process definitions model portability among different workflow products that use different modeling techniques or serve for different purposes (see Figure 7) (Dumar & van der Aalst, 2005)

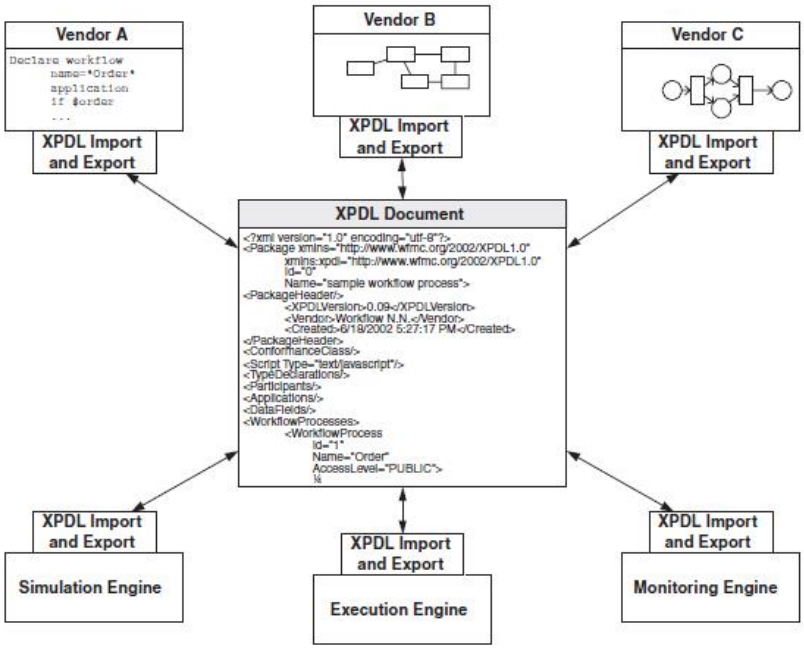


Figure 7 Interchanging XPDL process definition (Dumar & van der Aalst, 2005)

In order to support portability of process definition workflow vendors must provide these two functionalities:

- Export a process definition to a file that is conventional to an interchange format the target system can support
- Import efficiently an interchange format exported by another system.

XPDL assists this process by providing an XML-based lingua franca through which two

different vendors would be able to communicate. In addition, XML allows extensibility. This characteristic allows vendors to add vendor specific functionalities or the so called extended attributes to their process definition. However, these functionalities are useful only if the engine understands them. In contrary the process definitions can be used only if they support the minimum set of attributes that XPDL requires (Chang, 2006)

The Workflow Management Coalition has announced the first release of XPDL in 2002.During these last years there have been several improvements and we have witnessed new releases. The milestones of XPDL are summarized in the following figure:

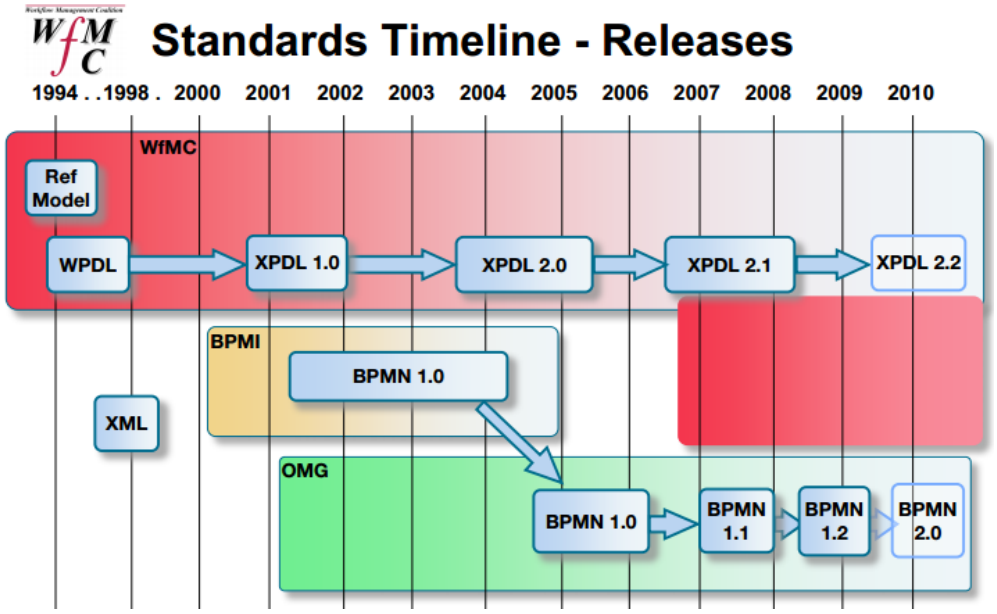


Figure 8 Milestones of XPDL (WfMC)

XPDL 1.0 was officially released in 2002. It retained the semantics of WPDL but defined a new syntax using an XML schema. As we can see from the figure, neither WPDL nor XPDL 1.0 proposed a specific graphical representation, although the underlying Meta model for a process was based on a directed graph structure consisting of activities as nodes and transitions as the edges or pathways between them (Shapiro & WfMC, 2008).

XPDL 2.0 was introduced in 2005 as a result of enhancements based on feedback on XPDL 1.0. From here on XPDL was designed to support graphical interpretation (can store process definitions drawn in BPMN) (Swenson & Brunt, 2006).

In 2007 the OMG published an updated version of BPMN: version 1.1. There were numerous changes to the graphics and semantics. WfMC has incorporated these changes, along with other improvements to XPDL 2.0 to create XPDL 2.1. (Shapiro & WfMC, 2008).

In 2012, the WfMC completed XPDL 2.2 as the fifth revision of this specification. XPDL 2.2 introduces support for the process modeling extensions added to BPMN 2.0 (Shapiro R. M., 2010).

Currently XPDL 3 is initiated, which will be focused on covering the complete specifications of BPMN 2.0 (Gagne & Trisotech, 2011).

One of the strong advantages of XPDL is that any process diagram can be expressed; it allows to extend any process by using the </extend> element. In contrary to other standards it has the ability interpret both graphical representation and semantic data.

Above all, XPDL is becoming a universal standard by having the widest adoption among the workflow system vendors (Ko, 2009)

Translating data from graphical representation to execution is easier rather than that from execution to graphical. This limitation raises doubts whether XPDL fully accomplishes “the bridge between the business analyst and IT specialist” (K L Ko, S G Lee, & Lee, 2009).

XPDL has to maintain backward compatibility such as: old names, old structure, and old relationships. However, this can cause more complexity which might result some misunderstandings as well as fragmentation of information (Chinosi).

2.3.2 XPDL structure

The following figure shows the structure of a basic XPDL file.

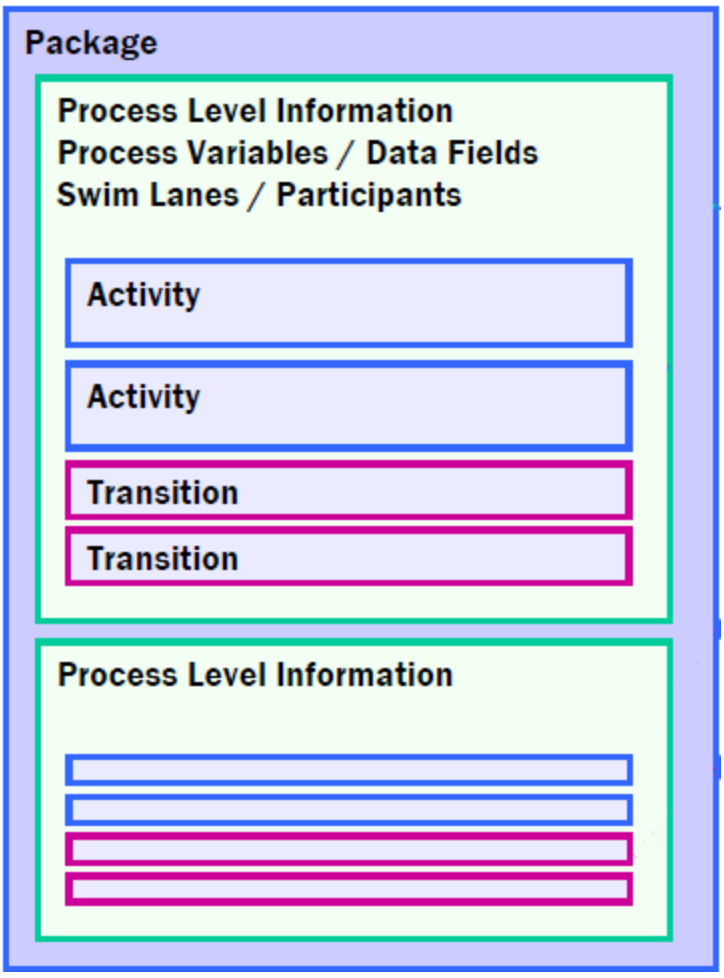


Figure 9 Basic Structure of XPDL (Swenson & Brunt, 2006)

From the figure we can see that the package is the root node of the XPDL file. It acts as a container for the grouping of common data entities from a number of different process

definitions, to avoid redefinition within each individual process definition. (WfMC, 2012).It may consist of one or more separate processes. These processes can inherit anytime several common attributes from the package (shared area). At the level below the package there is a process level that encapsulates information about a single process. It is made up of activities and transition.

XPDL uses an XML-based syntax, specified by an XML schema (XSD). It is based on an underlying meta-model defined by a WfMC. WfMC identifies two Meta models: Package and Process (WfMC, 2012).

2.3.2.1 Package meta model

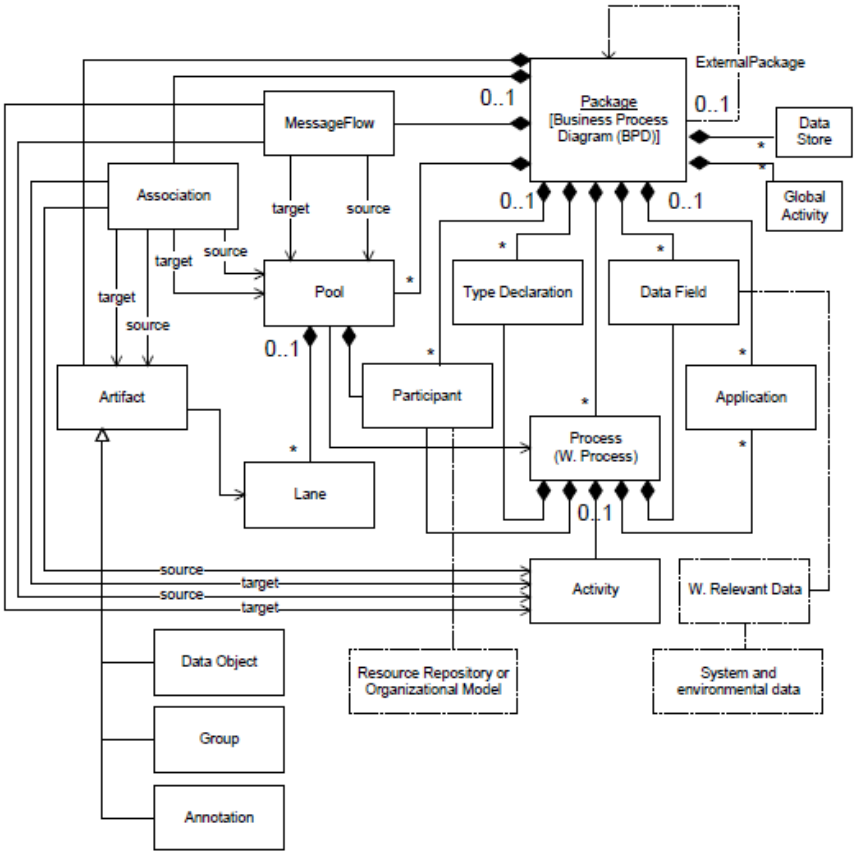


Figure 10 Package Meta model (WfMC, 2012)

The meta-model for the Package identifies the entities and attributes for the exchange, or storage, of process models. It defines various rules of inheritance to associate an individual process definition with entity definitions for participant specification, application declaration and relevant data field, which may be defined at the package level rather than at the level of individual process definitions (WfMC, 2012).

2.3.2.2 Process Meta model

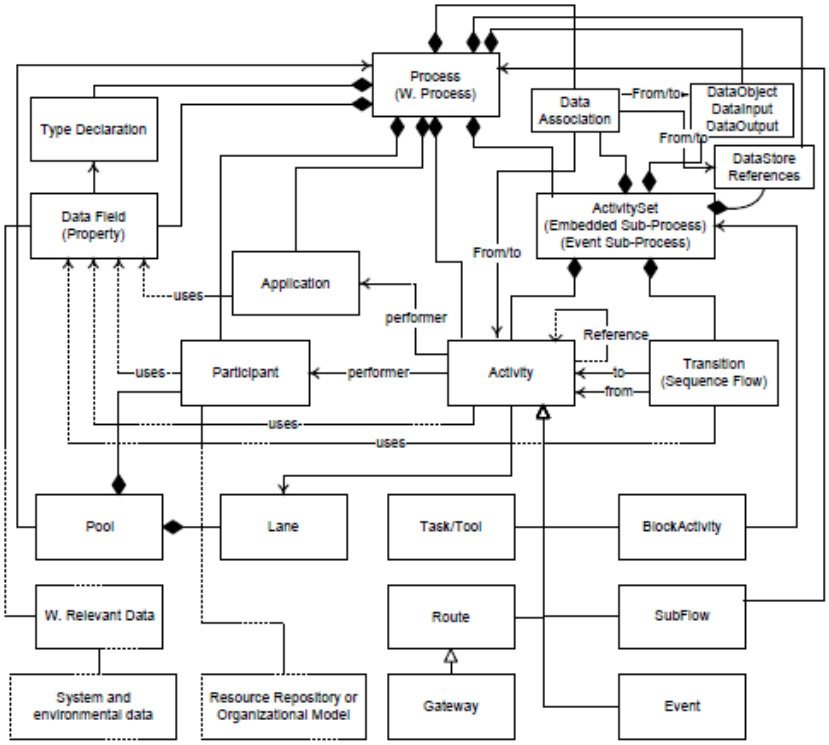


Figure 11 Process Meta model (WfMC, 2012)

WfMC defines the entities that comprise the Process Meta model as follows (WfMC, 2012):

Application: Includes IT applications or interfaces that can be invoked by the service to support or automate an activity. It is identified within the activity by the application assignment attribute.

Process Activity: A single process consists of one or more activities. The Meta model defines several types of activities. One kind is a routing activity or Gateways which is used during the implementation of the decisions which in fact changes the Sequence Flow path throughout the process. An activity contains its performer which might be human or computer program as well as other optional information regarding the starting mode etc.

Transition: Activities are related to one another via sequence flows (transitions). Each individual transition has three elementary properties, the from-activity, the to-activity and the condition under which the transition is made. Transition from one activity to another may be conditional or unconditional.

Participant: Represents the resources that can be assigned to the various activities of the process definition. Those resources are assigned as a specified attribute of an activity. The participant assertion might identify human, set of people (e.g. department) or machine.

Swim lanes: Swim lanes are used to facilitate the graphical layout of a collection of processes and the activities they contain. The swim lane consists by a collection of non-overlapping rectangles called Pools. Each Pool may be further subdivided into a number of Lanes.

Workflow Relevant Data: The data is made available to activities or applications executed during the process and may be used to pass persistent information or intermediate results between activities and/or for evaluation in conditional expressions such as in transitions or performers.

2.4. BPMN

After 2001 (see figure 8), when executable languages already existed, vendors realized that there is a need for a graphical representation language that will have in target business users. This language, rather than being a pure execution language will be structured in such a way that it will be understandable by every generic business user. In response to this need BPMI started an initiative to develop the business process management notation (Miers & White, 2008).

The notation working group was formed in August 2001 and it was formed from 35 modeling companies, organizations and individuals who developed BPMN 1.0 (Miers & White, 2008). Their aim was to create a standardized language which can be used by all business process modeling vendors and users. Another goal was to ensure that XML processes that have been designed for process execution can be graphically presented.

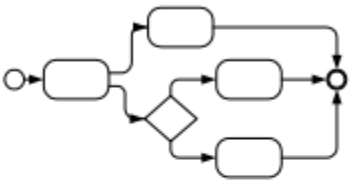
In 2004 BPMN 1.0 was released to the public and in 2006 BPMN 1.0 specification was adopted as an OMG standard (after BPMI was merged into OMG) (Miers & White, 2008).

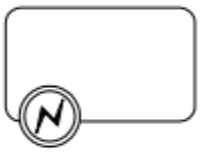
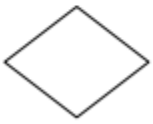
In 2008, OMG released version 1.1, and after one year BPMN 1.2 was in the market which did not have included any major changes, but only editorial changes.

The enormous improvements are done in version 2.0 of BPMN which was released in 2011.the graphical representation of the process definition is linked to a Meta model and schema for BPMN 2.0 Diagram Interchange (BPMN DI) (OMG, 2010). The BPMN DI is meant to facilitate interchange of BPMN diagrams between tools. This meta model is MOF – based which is OMG's industry-standard environment where models can be exported from one application, imported into another, transported across a network, stored in a repository and then retrieved, rendered into different formats (OMG). BPMN DI Meta model enables interchange, interoperability and execution of models.

2.5 Mapping from BPMN to XPDL

XPDL and BPMN are structurally very similar to each other since both of them have a flow-chart structure (White, 2003).The following figure shows the mapping from BPMN to XPDL:

BPMN Graphical Object	***Mapping to XPDL***
 The details of a Pool or an Expanded Sub-Process	<WorkflowProcess/>
 Start Event	<Activity> <Route/> </Activity>
	<Transition/>

Sequence Flow	
 Task	<Activity> <Implementation> <Tool/> <Performer/> </Implementation> </Activities>
 Sub-Process	<Activity> <Implementation> <SubFlow/> </Implementation> </Activities>
 Intermediate Event attached to activity boundary	<Activity> <Implementation/> <TransitionRestriction> <Split Type="XOR"/> </TransitionRestriction> </Activities> Combined with a: <Transition> <Condition Type="EXCEPTION"/> <Transition>
 Decision	<Activity> <Route/> <TransitionRestriction> <Split Type="XOR"/> </TransitionRestriction> </Activities> Combined with a: <Transition> <Condition/> <Transition>

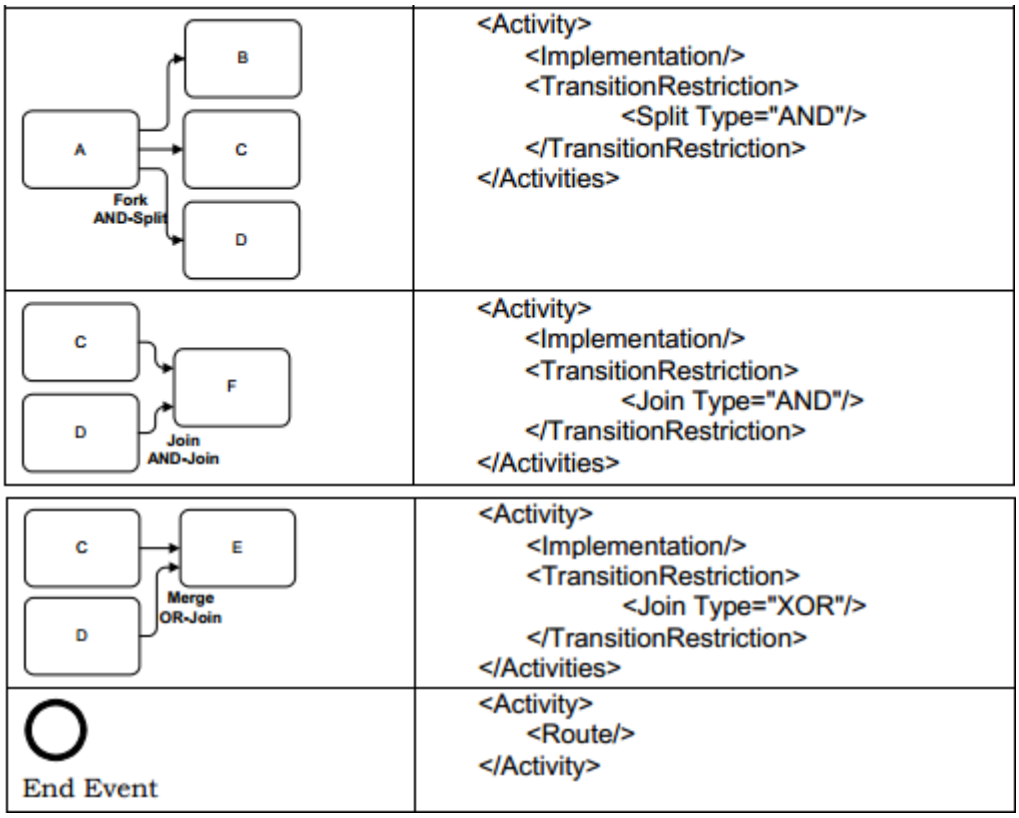


Figure 12 BPMN objects and their mappings to XPDL (White, 2003)

In the penultimate row we can observe the <implementation/> XPDL element. This element represents the applications or the implementation alternative that is assigned to the Tasks. We can note that it is not visible in the BPMN diagram. Implementation and such activities (e.g. performer assignment, data etc) are not explicitly depicted in the BPMN diagram; they are assigned in the tool as task properties which are attainable from XPDL.

2.6. BPMN/XPDL portability

As results of the new innovations in BPMN, XPDL 2.0 has integrated these features and provided an extended meta model which actually unified XPDL and BPMN. This was officially approved in 2005 (Shapiro & WfMC, 2008). This partnership continues to this day where the two organizations continue to improve their standards in synch with each other.
BPMN is used to model both abstract processes and those with executive nature. However, it is usually used to model abstract processes while executive characteristics of the model are defined through vendor specific functionalities (Shapiro, 2010). One goal of the XPDL is to separate the abstract process flow of the BPMN from the execution design of the model and it realizes this through the BPMN Model Portability conformance classes (WfMC, 2012).

2.6.1 Portability Classes

BPMN is composed of large number of objects from which some might be more usable than other. Based on this XPDL has defined three conformance classes: SIMPLE, STANDARD and COMPLETE (WfMC, 2008):

SIMPLE class includes the following BPMN objects: task, collapsed sub-process, gateway (exclusive data-based, inclusive, parallel), None start and None end events, pool, lane, data object, text annotation, sequence flow (uncontrolled, conditional, default), and association.

STANDARD class includes the following BPMN objects: task (task type User, Service, Send, Receive); collapsed and expanded sub-process, looping or multi-instance activity, gateway (inclusive, exclusive data-based, exclusive event-based, parallel), start events (None, message, timer), catching intermediate events in sequence flow (timer, message), throwing intermediate events in sequence flow (message), attached intermediate events (timer, message, error), end events (None, error, message, terminate), pool, lane, data object, text annotation, sequence flow (uncontrolled, conditional, de-fault), and association.

COMPLETE class includes all task types, all event types, and all gate-way types described by BPMN 1.1, message flow, transactional sub-process, and ad hoc sub-process.

To comply with BPMN 2.0, XPDL 2.2 has set three new conformance subclasses (WfMC, 2012):

DESCRIPTIVE class concerned with visible elements and attributes used in high-level modeling.

ANALYTIC contains all of DESCRIPTIVE and in total about half of the constructs in the Full Conformance Class

COMMONEXECUTABLE focuses on what is needed for executable process models.

2.6.2 BPMN Model Portability Validation test

In 2009, Workflow Management Coalition announced the results of the BPMN Model Portability Validation test and declared that at that time over 80 identified products use the WfMC's standard XPDL (WfMC)

This is a test that certified tools for XPDL/BPMN portability conformance, which means that a BPMN diagrams designed from the certified tools can be efficiently exchanged. WfMC has prepared test diagrams for different conformance classes for vendors that have been

willing to take the test. WfMC classifies the certificates furthermore into input, output and input/output certification. For an output certification the vendor should be able to build the test model. The designed model first should be check for XLST validation which is a XLS Transform developed by WfMC that performs structural and conformance checks. If the tool passes the XLST, it should attempt to export the test model as XPDL whose schema is correct and be able to load the same model into one of the already certified tools. For an input certification the vendor should be able to import the test XPDL into the tool, display it graphically correct and be able to modify the model. Finally, the input/output certification includes all the above and in addition import the XPDL model together with additional extensions and export the model with some other elements attached.

Above all, WfMC has partnered with Business Process Incubator in order to provide web based XPDL Conformance Verification Self Services. The web widget is called Cloud Apps. This widget appears in the left side of the WfMC website. Once the app is opened the user needs simply need to select the format of the source file as a MS Visio, XPDL, BPMN or MS Project. Beside the verifications other actions also can be taken such as: conversion, transformation, validation (depending on the source file).

Figure 13 Snapshot from the Cloud Apps accessed from xpdl.org

2.7. WFMC vs. DBMS similarities

Just as data management applications are developed and used with the assistance of a database management system (DBMS), so business process execution is performed on top of workflow *management systems (*WFMS). Workflow management systems enable the "extraction" of process management from the application software, just as database management systems extract data management from the application software (van der Aalst & van Hee, 2002). Despite this, WFMSs and DBMSs possess many similarities also in their design.

From an architectural point of view (see figure 14) (University of Kaiserslautern, 2003) workflow Meta model can be compared with relational algebra Meta model where the tool used for workflow design that supports BPMN can be seen– just as like a data modeling tool supporting an entity-relationship model.

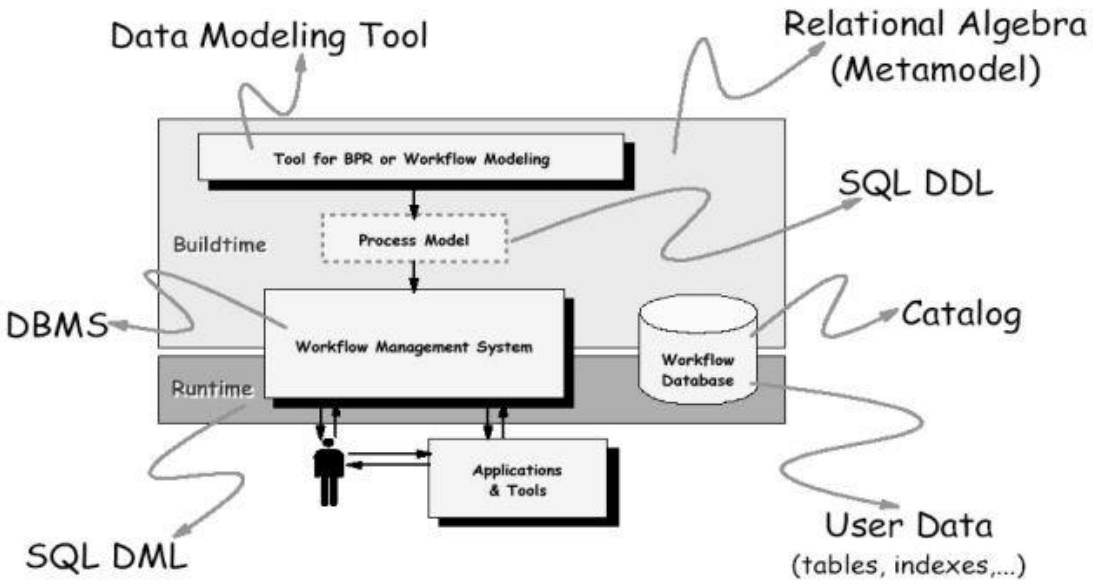


Figure 14 WFMC building blocks and their correspondence to DBMS (University of Kaiserslautern, 2003)

The process model definition (workflow schema) exported from the modeling tool to be deployed in the workflow management system can be mapped to the input of a database scheme into a database management system.

The first step in database schema design is a conceptual design which produces conceptual schema that is entity relationship diagrams. At a later stage ER diagrams are mapped to the logical level where relational tables are constructed onto which DBMS can perform operations. Comparable procedure is followed in workflow design. In workflow schema design one first develops a conceptual design which generates not executable abstract entities that is BPMN graphs. Next we have mapping of the BPMN diagrams to a lower (logical) level that is to executable XPDL files that can be read, interpreted and executed by WFMSs.

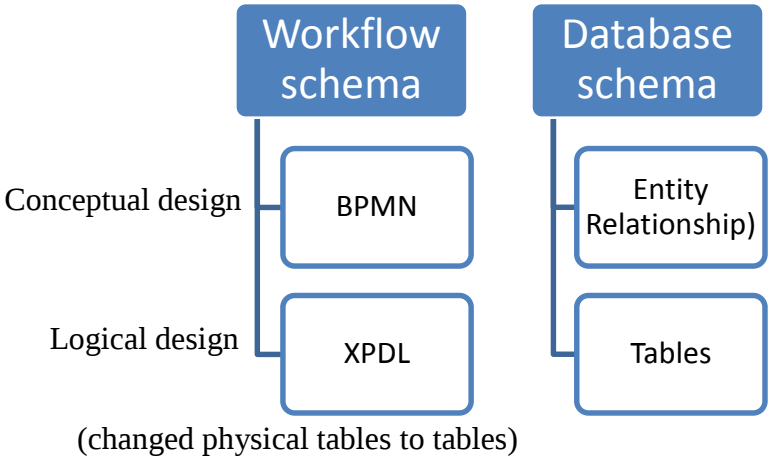


Figure 15 Conceptual and Logical Levels in WFMS and DBMS

Considering the refined WFMS Meta model (Figure 15) (Leymann & al., 2010), today process modeling is seen as a multistep effort where domain experts first create conceptual process models i.e. BPMN process definitions which are then transformed into a logical process model or into an executable XPDL format. However, in some cases logical process layer might be present in the execution engine, which is the case in BPM suites when execution aspects are added to the process definition through vendor specific modules.

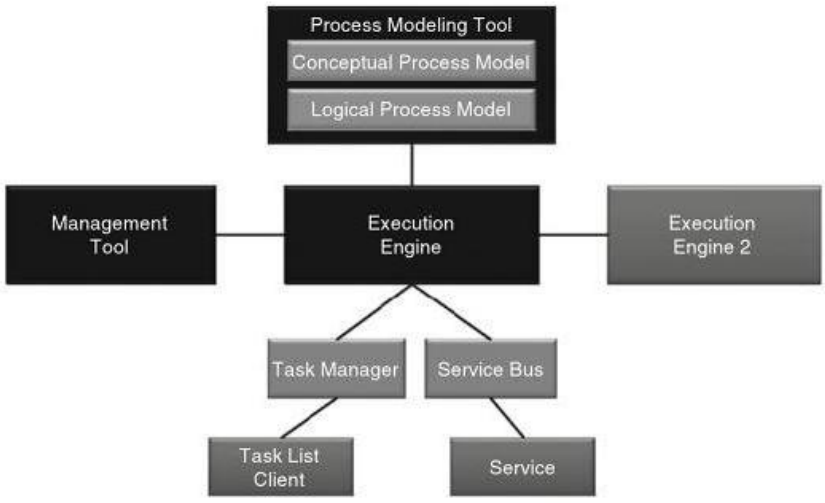


Figure 16 WFMS refined Meta model (Leymann & al., 2010)

3. Goals of the Thesis

The WfMC (Workflow Management Coalition) is the non-profit organization of reference, aiming at defining a standard glossary as well as a standard definition language for business processes. Such a language is named XPDL (XML Process Definition Language).

Over the last decade, XPDL has been becoming the standard language for process definition, accepted by many vendors as the primary object for the implementation of portability. Nowadays, we have a large number of vendors that have adopted the specifications proposed by WfMC as a design requirement for their system and declare their products to be XPDL compliant. However, the diversification of products and their ability to understand different and specific dialects of XPDL, raises the question whether XPDL can really, totally achieve its original purpose.

The goal of the current thesis is to test on the field the portability of process definitions. In order to do it, as a first step we design a process definition model that would be considered as a reference. Specifically, we want to test the following aspects:

1. Design portability: we aim at checking if this model can be built in approximately the same way in all the modeling tools that we have chosen, to evaluate if all the tools use a standardized way of design.

2. Execution portability: we aim at checking if the engines/editors which present themselves as XPDL compliant, are able to properly import the reference process model.

3. Translatability: when the engines / editors mentioned above are not able to successfully complete the process of importing, we aim at evaluating what are the most common obstacles or porting problems and how they can be managed.

During the development of the thesis we faced some obstacles. Just as we mentioned above, there exist several workflow products in the market; however, gaining access to them was not as easy as we expected.

There are several open source products as well as commercial ones. However, the problem lays in the fact that most of the open source WFMSs' are supporting elder versions of XPDL, while commercial ones support the newer versions. Meanwhile process editors (to which we had no access problem) are supporting newer versions of XPDL therefore; we experienced several difficulties about the engines. In some cases we have had access to evaluation versions of the products, but in these cases not all the functionalities were provided.

All the vendors aim at distinguishing themselves from their competitors in order to gain competitive advantages. Thus, they are trying to provide the most convenient and appropriate ways of designing and interpreting business process models. Consequently, they are generating different or vendor specific XPDL dialects.

Considering these obstacles or facts, there was a need to investigate the schemas/structures of XPDL files version: 1.0, 2.1, 2.2, and to study a tool-specific way of interpretation and translation.

4. Design choices

As we mentioned in the earlier chapter, we have faced difficulties in accessing commercial workflow products. Consequently, our choices were limited, so for this reason our first attempt was to investigate all the open source workflow products in the market that pretend to be XPDL complaint. On the other hand some of the commercial workflow products offer complete workflow solutions that is they combine their workflow modules/products in a single suite. In some cases only one module (which usually was the business process editor) was provided as standalone and a freeware product. While other modules were provided free for only 30 days trial. Nevertheless, the only criteria that determined our decision was that these products should be compliant to WfMCs' XPDL specifications.

As workflow products we considered modelers or process editors and workflow engines. We categorize editors in proprietary and public domain editors which are further classified based on the version of XPDL that they support. Whereas the workflow engines we classify as standalone engine and those embedded in a BPMN suite.

Public domain editors:

XPDL version 2.1:

- **Together Workflow editor (TWE)**

Together XPDL and BPMN Workflow Editor also known as Enhydra JaWE is a Java workflow editor which is fully implementing the WfMCs XPDL specifications and uses BPMN as a graphical notation. It is an Open Source project which is publicly available on SourceForge under the GPL V3 license as "Enhydra JaWE". Specifications of this tool claim that it is able to read (understand) and edit every file that is WfMC compliant. LDAP connections to MS Active Directory or OpenLDAP are supported for importing process participant data. Apart from simple XML syntax checks, it performs the extensive logical validation of XPDL automatically (Together Teamsolutions Co., Ltd.). Current version of TWE is 5.0.1, this version supports XPDL 2.1.

- **TIBCO Business Studio**

It is Eclipse-based and serves as the common design environment for TIBCO's business process management (BPM) and service-oriented architecture (SOA) platforms. TIBCO Business Studio Community Edition is standards based and freely available software that allows users to build end-to-end business models that are compliant with the BPMN, XPDL, and UML standards. It also validates process models to ensure users that their model is compliant to standards. Except modeling, it also allows the simulation of the business processes based on real data or sample data. There are two views of TIBCO Business Studio, TIBCO Business Studio for Analysts and TIBCO Business Studio for Designers (TIBCO Software Inc.). TIBCO Business Studio for Analysts is freely available to public and it is a

business friendly non-technical version of the design environment (TIBCO Software Inc.). The latest and the version that we have used throughout the thesis is 3.5. This version supports XPDL 2.1.

- **YAOQIANG XPDL Editor**

YAOQIANG XPDL Editor is a freeware graphical editor for creating, editing, managing and reviewing workflow process definitions compliant with WfMC specifications (XPDL 2.1 and 1.0). It represents all XPDL elements graphically through property panels and a graph component to give the user a better understanding and an overview of the process definitions. The final output of the editor is an XML file (using the standardized WfMC XPDL schema) which can then be interpreted and executed by all WfMC XPDL compliant workflow engines. YAOQIANG XPDL Editor accomplishes three main goals: reading of WfMC XPDL files from the file system, graphical representation and guided editing/modeling of process definitions, writing of WfMC XPDL process definition XML files to the file system (Yaoqiang). The version of the editor used in the thesis is 2.1.21 and support XPDL 2.1.

XPDL version 2.1:

- **BizAgi Process Modeler**

BizAgi Ltd. is an active member of the group in charge of defining the standard in the OMG. It provides a workflow solution product called BizAgi Suite composed of the following modules: BizAgi Process Modeler: Process Modeling and Documentation environment, BizAgi BPM Studio: Automation Module for creating business process models and create process applications and BizAgi BPM Server: Execution engine that generates portal views for human interactions and integrates to back-end applications according to process model created with BizAgi Studio. BizAgi Process Modeler is the only one of the above mentioned products that is standalone and available as freeware. It supports BPMN 2.0 and XPDL 2.2 (Bizagi Ltd.) .The version of the modeler used in thesis is 2.7.0.2 and the last available version of the suite is 10.4.

Standalone Workflow Engines:

- **Together Workflow Serve**

Together Workflow Server (TWS) – also known as Enhydra Shark is an open-source project which is publicly available. It is a flexible and extendable WfMC XPDL compliant Java workflow engine which can be used in both ways: embedded or standalone. Process definitions are based on WfMC XML Process Definition Language. Together Workflow Server Together Workflow Server includes an advanced Swing administration client, a command line client and a web-based work list and administration client for managing installed workflow servers. In addition to the execution of business processes, TWS offers additional features like tracking of all business processes within the system (Together Teamsolutions Co., Ltd.). The latest release is version 6.0.1, which will also be used in this thesis.

- **WfMOpen**

WfMOpen is a J2EE based implementation of a workflow engine. The workflow component is based on a set of Java interfaces that defines API for workflow management facility. Workflows are specified using WfMC's XML Process Definition Language (XPDL) with some extensions. However, it is only partially conformal to the WfMC model, since it does not implement the Interface 4 of the WfMC that is it does not interact with other workflow enactment services (Sourceforge). Version 2.2.1 of the WfmOpen has been evaluated during thesis. This version supports XPDL 1.0.

Workflow engines embedded in a BPM suite:

- **Bonita**

Bonita BPM is complete BPM solution software which is composed of several components: Bonita Studio, BPM Engine and Bonita Portal. Its graphical environment for creating processes Bonita Studio contains two major design tools: the whiteboard, for drawing a process flow diagram and defining the detail of steps and the form builder, which is used to create forms used in process web applications. Bonita Studio allows the user to get started with processes designed with other standards and technologies such as XPDL. Bonita BPM Engine is the process execution engine of Bonita BPM. Bonita BPM Portal is the part of Bonita BPM that is visible to process users, who use it to view tasks and take actions. Bonita BPM is provided in four different editions: Community, Teamwork, Efficiency, and Performance. The latest release of Bonita is version 6.3 (Bonitasoft Inc,).

Besides the above mentioned software's, we tried to put in use also other software's:

WorkflowGen – we were able to download the manual installation of this software from the Workflowgen community website. After a long installation procedure the software was unable to work, due to a lack of some extensions which we assume were absent because it was not a commercial version.

IGrafX – is offering number of products which together offer a workflow solution. One of the products; IgrafX flowcharter was offered as trial version. However, in order to be able to import/export XPDL a file, XPDL/XML Interface plug-in was needed which was not available for evaluation purposes.

Above all, we have sent several requests to some vendors (Chalexcorp, Comactivity, Workflogen, ActiveVos, Agile Point Server) with the expectation that they might offer an evaluation copy of their products, but this attempt failed in all cases.

5. Results

5.1 The reference process model

During the thesis we have used a reference process model which was based on a very common process used within many companies, that is Purchase Request process.

The structure of the organization that would take part in this process is comprised of: Requesting Department, Financial Department (which is composed of: Supervisor, Department Manager and Vice President), and finally Purchasing Department. Initialization of the process is performed by the requesting department which will create and submit the purchase request (PR). Depending on the total amount of the PR, the request will be routed for approval within the organization. If the amount will be greater than 500 and smaller than 1500 the request will be send to the Supervisor, if it will be greater than 1500 and smaller than 5000 will be sent to the Department Manager, greater than 5000 to the vice president and otherwise directly to the Purchasing Department. If the request is sent to the Financial Department, then taking into consideration the final outcome of the decision, the PR is either send to the Purchasing department for further processing or back to the Requesting department notifying the rejection of the request and the process ends. On the other case that is, if the PR is directly send to the Purchasing Department without a need of confirmation from the Financial Department the order is processed without any interruption. Once the application has reached the Purchasing Department (irrespective of the fact whether it has arrived there directly or through the Financial Department) quotations are performed and as soon while the processing of the order begins a notification about the acceptance is sent to the Requesting Department and the process ends.

5.2 Tests

5.2.1 Tests with the process designed in TWE

The design of the above mentioned process in Together Workflow Editor looks as in figure 17. As shall be seen, the PR process is a large process and consequently, for visual reasons it can be better illustrated if it is vertically oriented. This tool supports such functionality. In BPMN lanes represent Roles and organizational structures. Since, inside the Finance Department, there are three entities respectively roles who take part in the process: supervisor, department manager and vice president we decided to create them inside the lane Finance Department, as nested lanes and other departments as a whole organization structure inside a single lane. Tasks, exclusive and parallel gateways are created as follows (see figure 17). Next, the tool allowed us to define workflow variables that would be necessary later for defining expressions in the conditions. We created two workflow variables: Total price as Float type and Approved as Boolean. In order to be able to assess the direction to which the PR will be sent for authorization, we had to define the conditions in the outgoing transitions of the exclusive gateway. Scripting language used in expressions in TWE can be of these types: text/JavaScript, text/VBScript, text/TCL, text/ECMAScript, and text/xml. It can be set from the package properties. Our choice was text/JavaScript type. Three of the

transitions have the condition type “Condition” that is we have to define expressions on them and the fourth one (the yellowish one, see figure 17) has condition type “Otherwise”. The conditions were set and the process model was ready to be exported. TWE automatically saved the file in XPDL so there was no need for any specific exportation procedure.

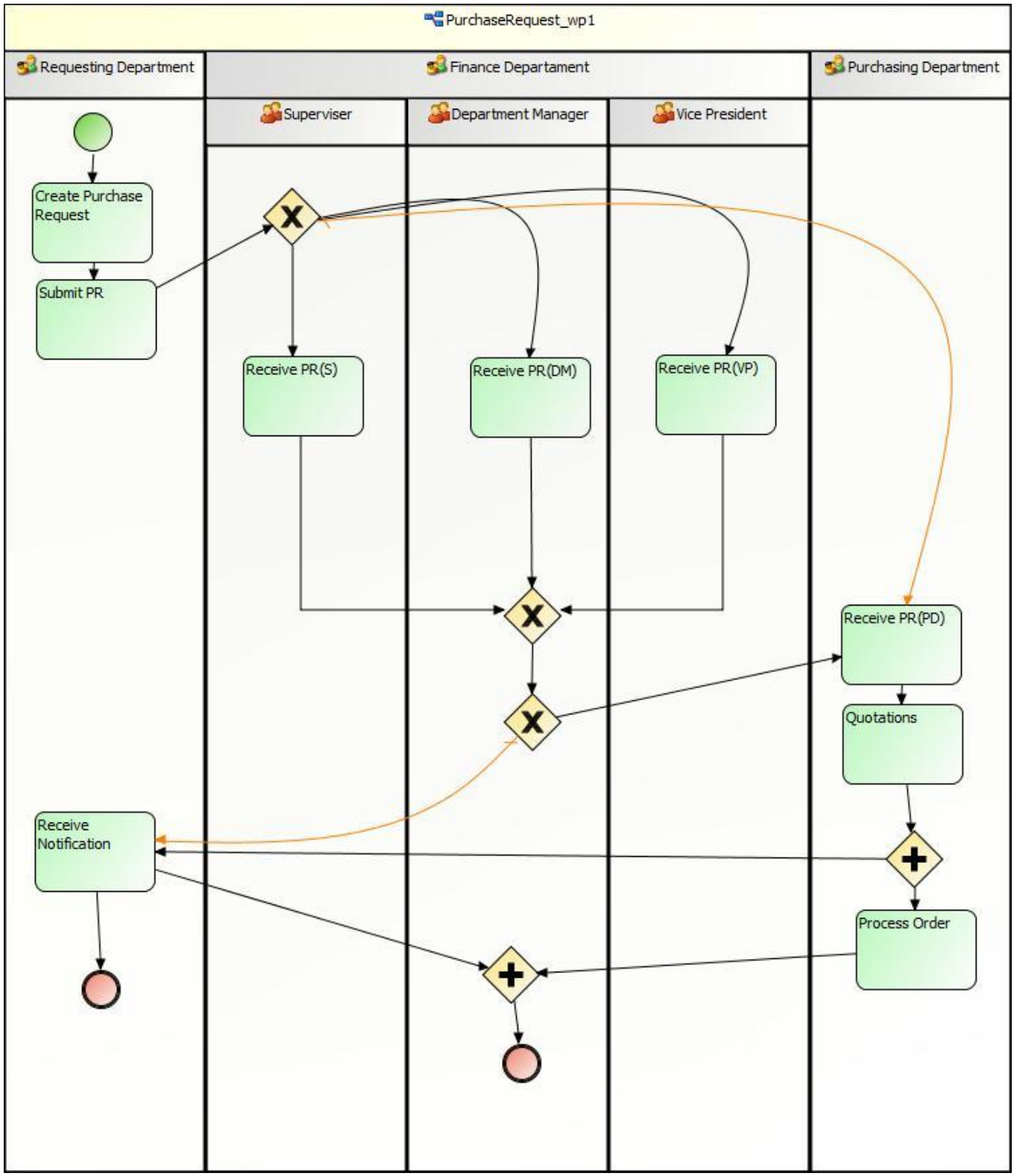


Figure 17 Purchase Request Process designed in TWE

TWEs' XPDL in TWS

The first and successful attempt of importing the XPDL file generated by TWE was at Together Workflow Server (TWS). The file was imported without any problem and it was immediately ready for deployment.
The import was performed through TWSs' Swing Admin application which is meant to be used both by administrators to manage TWS engine, and ordinary users to execute work lists. We used the default configuration that is; the TWS was running in the same VM as the admin application.
the repository management is displayed IN TWS Admin. It shows all available files in chosen XPDL repository where the upload possibility is offered. Our upload try was successful and Purchase Request XPDL file was shown in the repository (Figure 18).

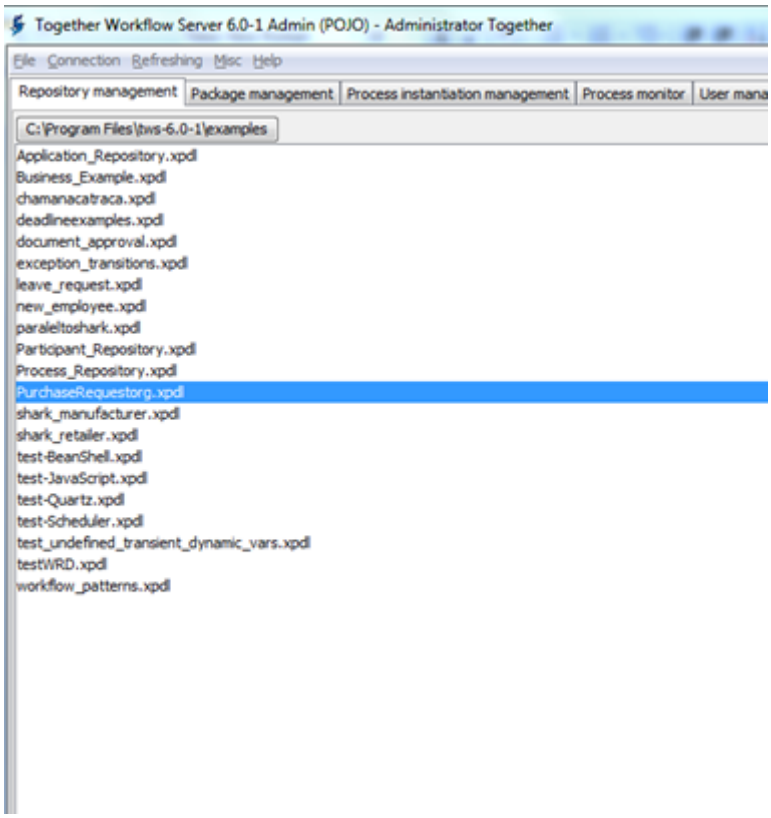


Figure 18 The Repository Management in TWS

After the file was uploaded into XPDL repository, we were able to load it into the engine. The package management was displaying all the files that were loaded into engine. Next, the process was instantiated through the process instantiation management. The "Process monitor" division allowed us to graphically monitor the implementation of the process. Above all, here we had the opportunity to play with the values of workflow variables in order to make sure that the expressions in the conditions are correctly understood and properly executed by the engine. Participants' data were also accurately imported. They were visible in the User management area from where we were able to map XPDL Participants to the real TWS users or groups (Figure 19).

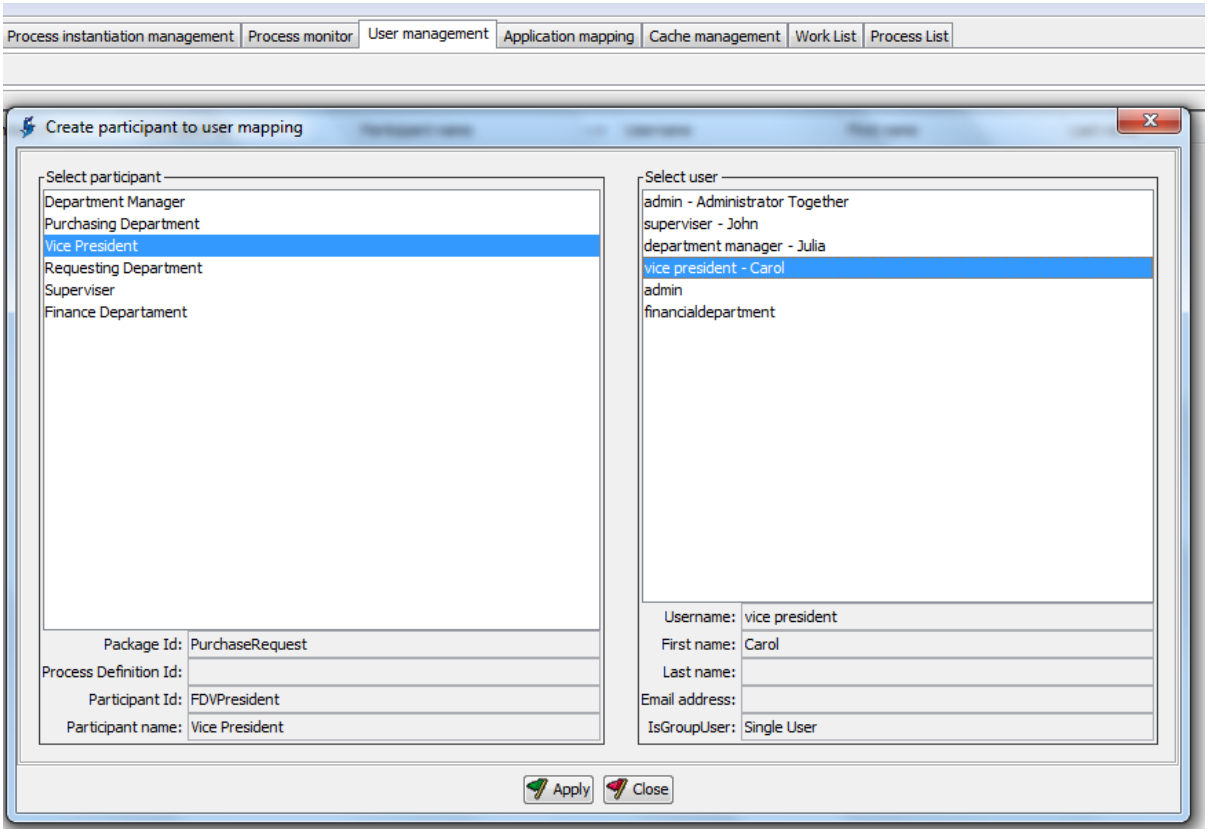


Figure 19 User Management in TWS

Since they are both produced by the same vendor and somehow have common origins, at the first glance it seems that TWS is totally able to understand and correctly interpret the XPDL syntax produced by TWE. However, this was not completely true; we came across a problem while executing the process. The problem was with the synchronized merge; that is with the implementation of the parallel activities (in our case Receive Notification and Process Order). This pattern had a non deterministic behavior in TWS as we can see from the figures 20 and 21 (snapshot: from the Process monitor, the highlighted activities are enabled), Receive Notification activity is enabled twice.

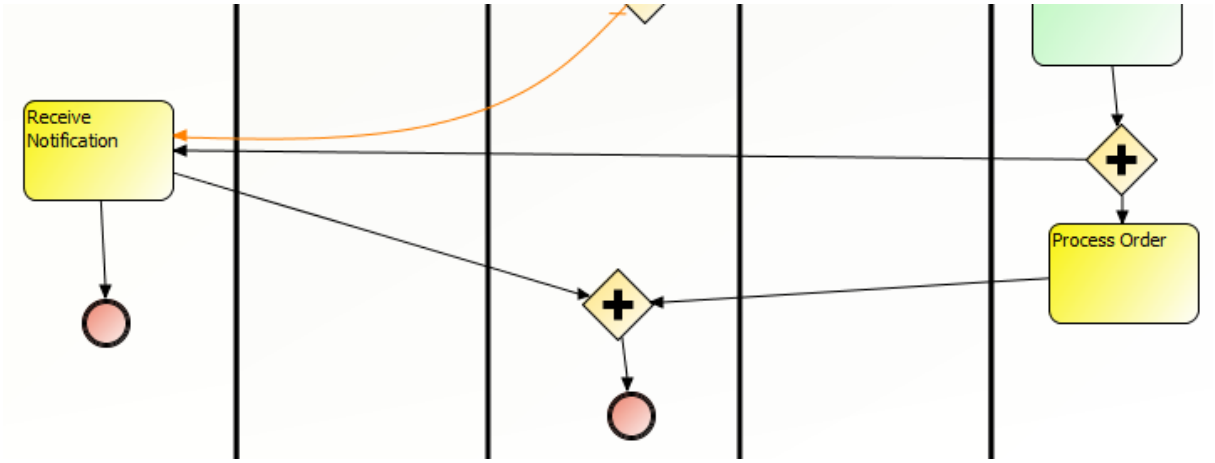


Figure 20 The first step in the process after the split

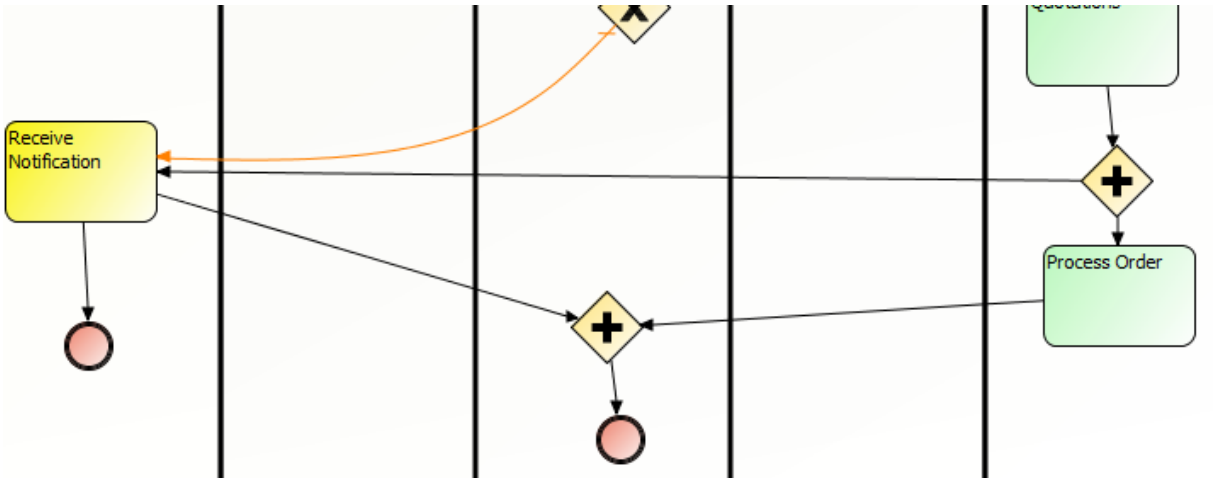


Figure 21 The second step in the process after the split

TWEs' XPDL in BizAgi

BizAgi process modeler acts as a mediator through which the XPDL packages from other tools are imported in BizAgi suite for execution. Our first import attempt in the modeler was not successful. The error encountered was:

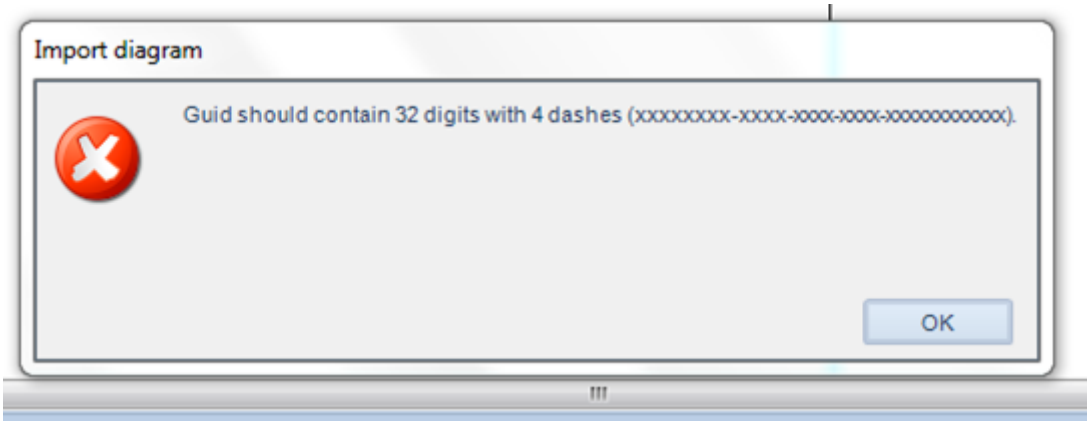


Figure 22 First error in BizAgi

The error was ambiguous. Since we already knew that both tools were using different versions of XPDL, we started the adaptation of the schema with changing the version of the XPDL file. TWE exports in version 2.2 while BizAgi supports XPDL 2.2. So, the following modifications were made:

Original	Modified
<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="newpkg1" Name="newpkg1" 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>	<?xml version="1.0" encoding="utf-8"?> <Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" Id="6854e0e3-1444-4853-87ae-48bc6b3c69f1" Name="Diagram 2" xmlns="http://www.wfmc.org/2009/XPDL2.2"> <PackageHeader> <XPDLVersion>2.2</XPDLVersion>

Table 1 Mapping TWE to BizAgi – Adapting versions of XPDL

Where also the prefix ":xpdl" was removed from all the tags.

The second import attempt was not successful. The next error message shown was:

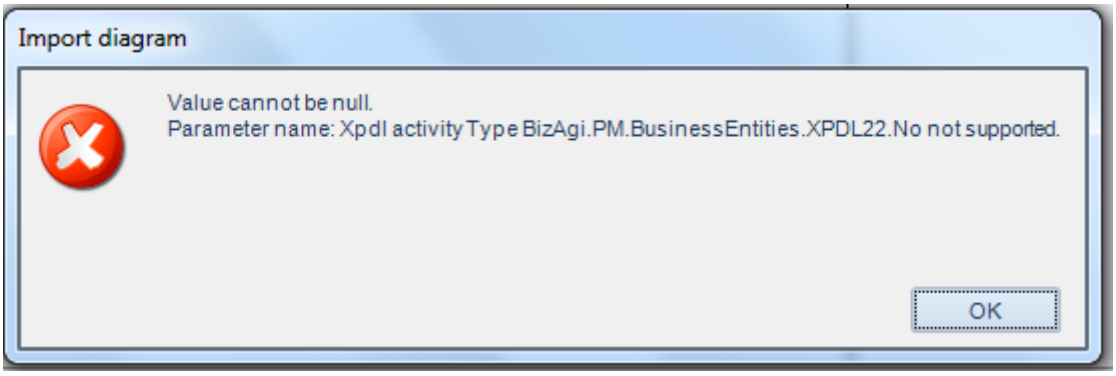


Figure 23 Second error in BizAgi

Activity types in TWE can be of type “No Implementation” and “Task Application” and they are distinguished within the XPDL file with the tags: <Implementation> <No /> or <Implementation> <Task>. However, this is not understandable by BizAgi. Despite TWE, BizAgi has a richer palette (also supports newer version of BPMN) and offers more choices for activity types such as: None Task, User Task, Services, Receive, Send Tasks, etc. Consequently, we improved the fragment in question in order to be understandable for BizAgi as follows:

Original	Modified
< Implementation> <No/> </Implementation>	<Implementation> <Task/> </Implementation>

Table 2 Mapping TWE to BizAgi: Adapting activity definitions

Even this modification did not enable the import. The next error message was more ambiguous:

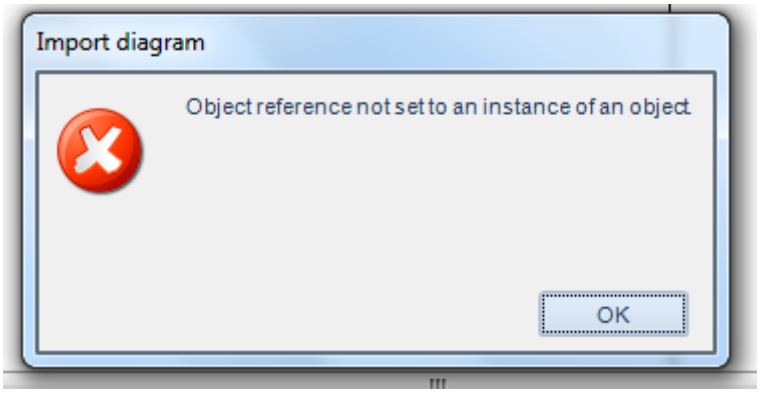


Figure 24Third error in BizAgi

Since our model was a large one and the error message was not so specific we decided to test the import with one other simpler process model. We designed a very simple model (figure 25) with one participant, one activity and the associated Start and End Events. After many attempts we still did not have any result yet and the error message was showing continuously (Figure 24).

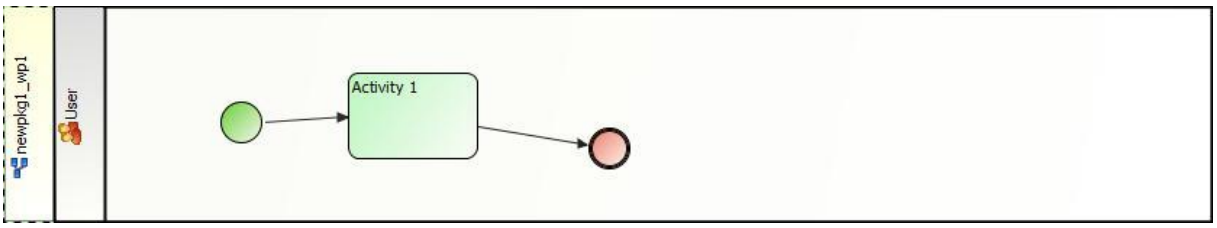


Figure 25 Simple Process Model with TWE

Finally, we decided to design a similar model in BizAgi, export it in XPDL and compare the two files. And finally, the only modification in the XPDL file which made it able to be imported in BizAgi was the addition of the <description /> tag in the following fragment:

Original	Modified
<xpdl:Participant Id="newpkg1_par1" Name="user"> <xpdl:ParticipantType Type="ROLE"/> <xpdl:ExtendedAttributes> <xpdl:ExtendedAttribute Name="JaWE_TYPE" Value="LANE_DEFAULT"/> </xpdl:ExtendedAttributes> </xpdl:Participant>	<Participants> <Participant Id="newpkg1_par1" Name="user"> <ParticipantType Type="ROLE"/> <Description /> <ExtendedAttributes> <ExtendedAttribute Name="JaWE_TYPE" Value="LANE_DEFAULT"/> </ExtendedAttributes> </Participant>

Table 3 Mapping TWE to BizAgi- Adapting XPDL fragments

However, the imported model in the BizAgi process modeler was not so encouraging (see figure 26 a).As seems TWE does not define any graphical information regarding Pool and Lane dimensions. It gives no information about the coordinates of transitions but only defines properties such as "From" and "To". As a result, we were supposed to add properties about the dimensions and coordinates manually on the XPDL file.

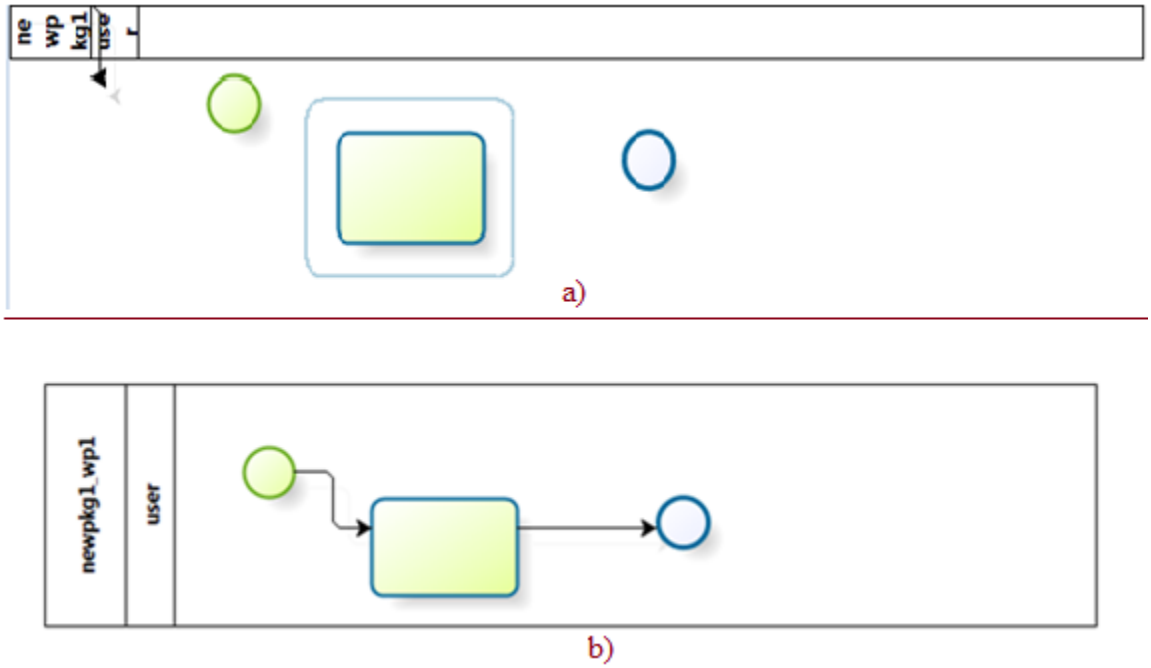


Figure 26 a) The process without modifications, b) The process with modifications

The original model also should undergo the same modifications that we performed in the simple model. However, the original model is very large and the performance of these same modifications can be more time consuming. Something additional that was not present in the simple process model were the Nested Lanes and the Expressions in conditions. BizAgi Process Modeler is not able to read Nested Lanes and Expressions; we will go through these properties of the tool later in the chapter when we will be investigating the import of files generated from BizAgi process modeler. After the modifications, runtime properties can be added to the model through the BizAgi studio and then run through the BizAgi server. These steps are not explained in this thesis since they do not use XPDL specifications thus are out of the scope.

Software's not able to understand TWE

We test the import of the XPDL generated from TWE also at WFMOPEN and BONITA. After all the efforts we were not able to import it to the tools mentioned. This may be explained well relying on the fact that both WFMOPEN and BONITA support older versions of XPDL. XPDL supports BPMN notation since XPDL 2.1, so that is a reason why XPDL 1.0 and 2.0 cannot be imported to vendors supporting latter versions or vice versa. XPDL's 2.x versions together with BPMN have brought some new concepts, though it is difficult and time consuming to adjust the file to the latter version; it is as writing it from the scratch. Since we could not find any editor which is able to export XPDL in older versions, the investigation of these tools (WFMOPEN and BONITA) will not be further considered.

5.2.2 Tests with the process designed in BizAgi

A similar process to the PurchaseRequest.xpdl designed with TWE was modeled also within BizAgi (see Figure 27). However, the resulting model had some changes.

1. The model in TWE was aligned vertically (it was our choice due to visual reasons); however, this was not possible with BizAgi Process Modeler.

2. Hierarchically structured Lanes (nested Lanes) are not supported in BizAgi Process Modeler. In fact a lane does not represent by all means a participant in BizAgi, it is only a graphical sub-partition of a Pool, the participants are defined in the Pool properties and explicitly assigned to Participants, and they have nothing to do with lanes.

3. Expressions in conditional transitions in BizAgi are defined as a free text. Since the modeler is a “part” of a suite it only performs the part of the design. Consecutively, even workflow variables cannot be set. The rest of the activities such as: data definitions, rules construction etc. are performed during the automation phase. As a result, BizAgi Process Modeler uses a different approach from TWE, it creates a non-executable model. BizAgi Studio is an environment where the necessary information for process execution are set. The resulting model is stored in the server repository and is interpreted and executed by the BizAgi BPM Server. However, this model is not based on the XPDL specifications thus cannot be exported as such format.

Even though BizAgi Process Modeler does not generate an executable model, we decided to test at least whether that abstract model can be imported to the other engines/editors.

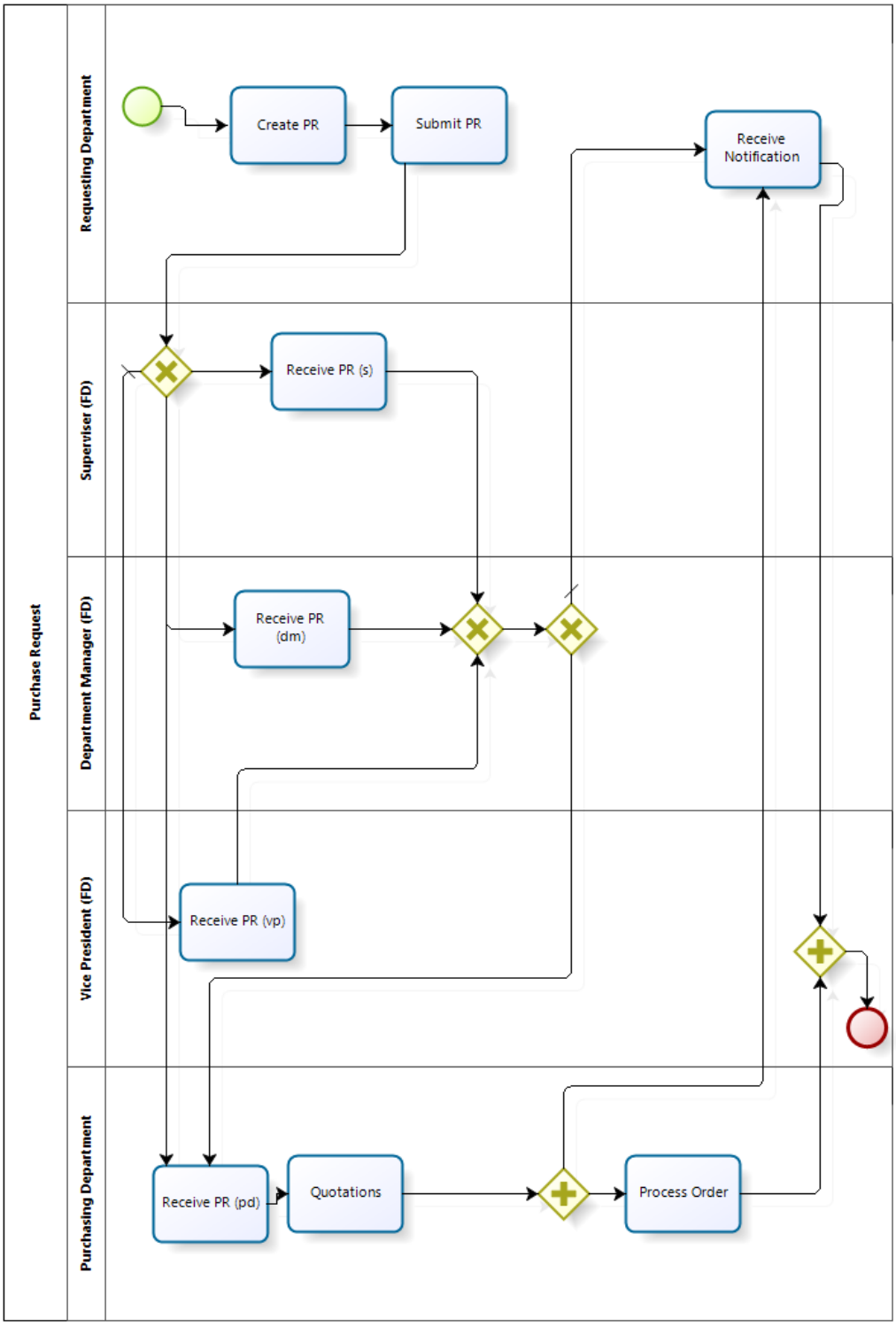


Figure 27 Purchase Request process designed with BizAgi Process Modeler

BizAgi's XPDL in TWS

In the beginning, the XPDL file was uploaded successfully, however, whenever we tried to load the model an error encountered (Figure 28):

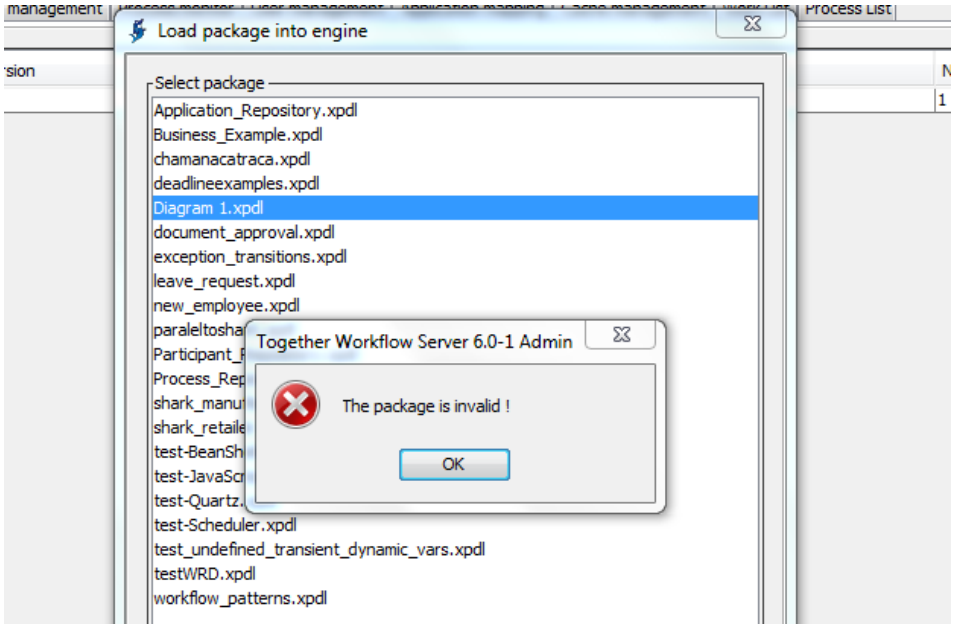


Figure 28 First Error in TWS

And the details where shown as following:

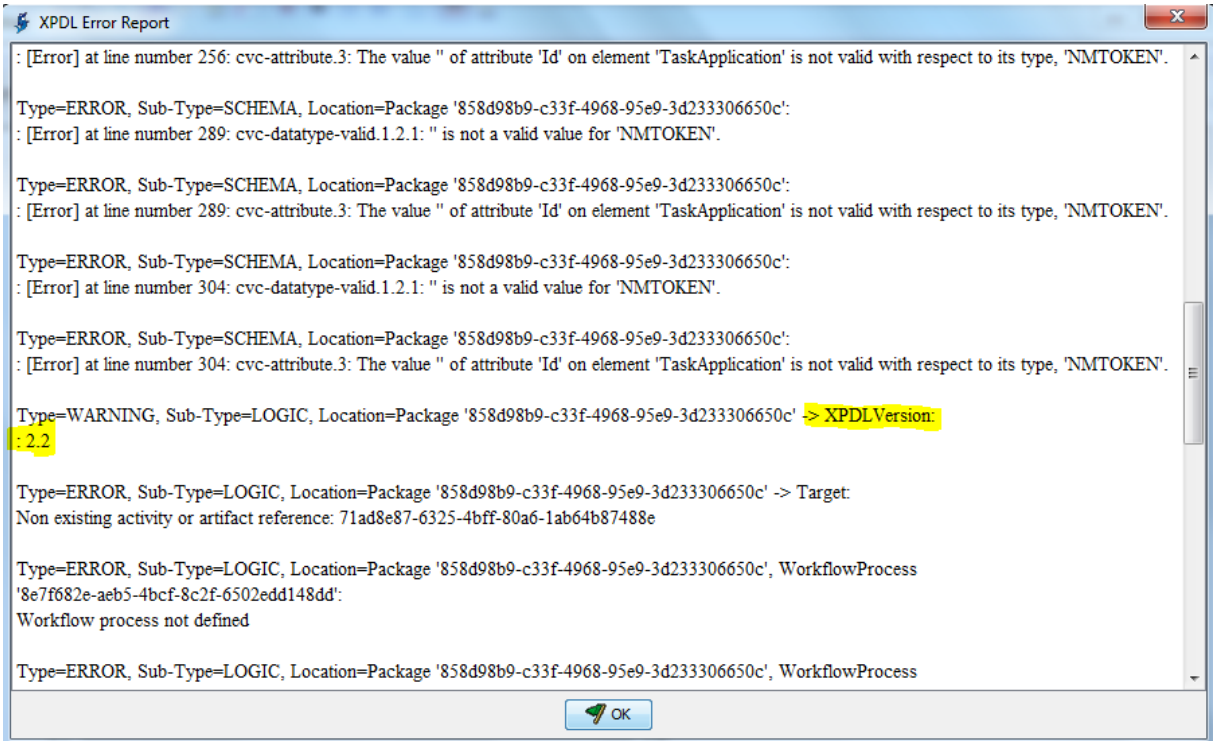


Figure 29 Details of the error encountered while loading the BizAgi model into Shark

The models exported from BizAgi are in the XPDL 2.2 version and the ones that Shark (TWS) supports are of version 2.1, so, we first tried to modify the version respectively the header of the package. The following fragments were replaced:

Original	Modified
<Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" Id="858d98b9-c33f-4968-95e9-3d233306650c" Name="Diagram 1" xmlns="http://www.wfmc.org/2009/XPDL2.2">	<Package xmlns="http://www.wfmc.org/2008/XPDL2.1" xmlns:xpdl="http://www.wfmc.org/2008/XPDL2.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.wfmc.org/2008/XPDL2.1 http://www.wfmc.org/standards/docs/bpmnxpdl_31.xsd" Id="858d98b9-c33f-4968-95e9-3d233306650c" Name="Diagram 1">
<XPDLVersion>2.2</XPDLVersion>	<XPDLVersion>2.1</XPDLVersion>

Table 4 Mapping BizAgi to TWS- Adapting XPDL versions

The next attempt was following with the errors:

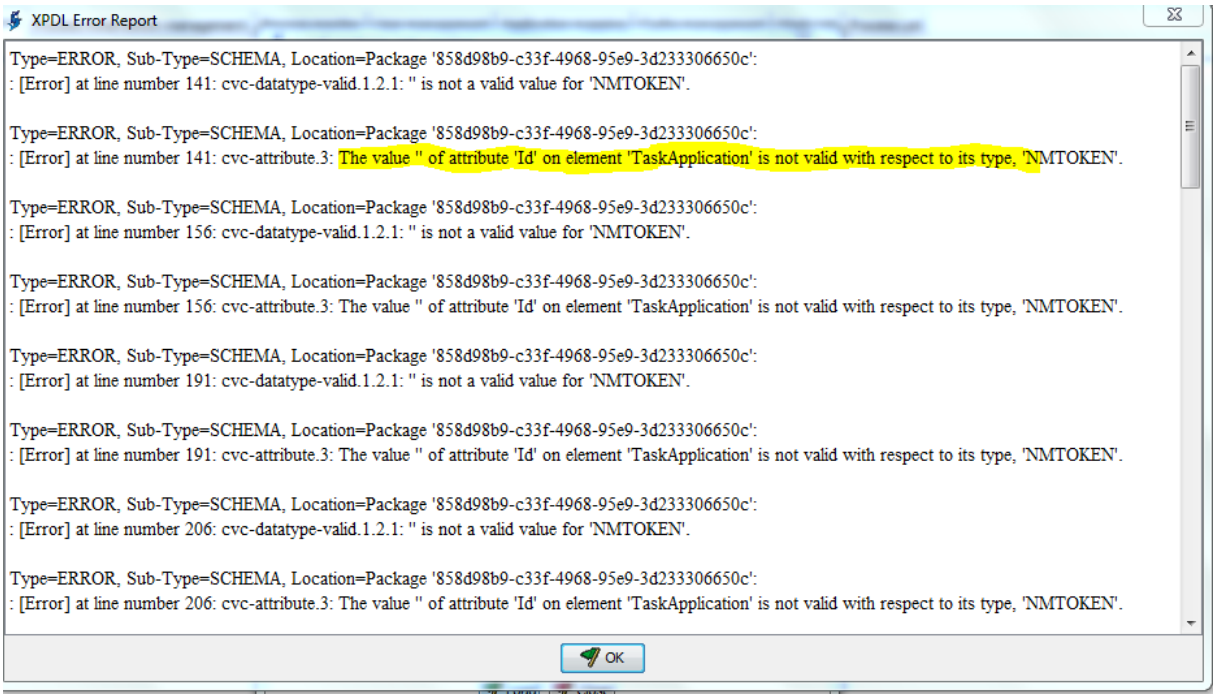


Figure 30 Second Error message in TWS

We referred previously about the different definitions of Activities among TWE and BizAgi when we were evaluating TWE import in BizAgi. Here we applied the opposite modifications (assuming that TWE interprets in a way that is understandable by TWS):

Original	Modified
< Implementation> <Task/> </Implementation>	<Implementation> <No/> </Implementation>

Table 5 Mapping BizAgi to TWS- Adapting Activity definitions

Finally, the remaining error was:

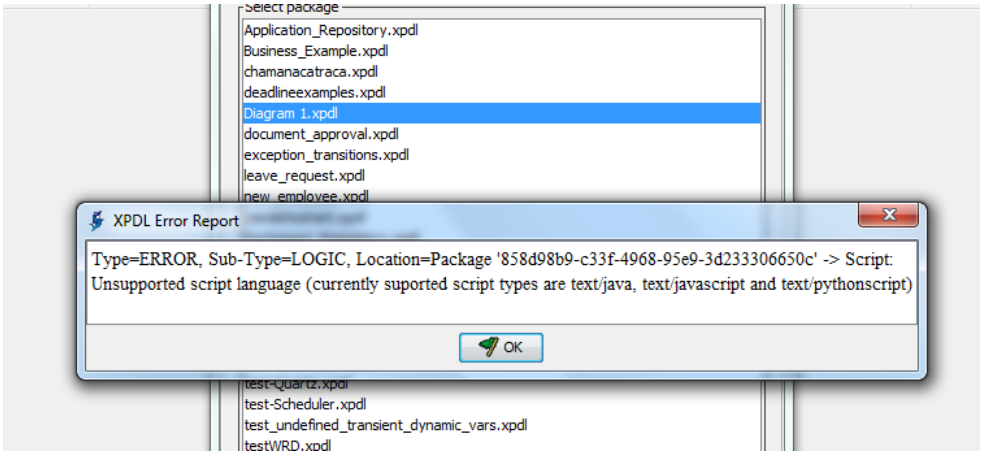


Figure 31 Final error in TWS

This error is obvious. We stated before that BizAgi defines the conditions through plain text. It actually does not label the scripting language in the XPDL file at all. To test whether the file will be imported without this problem, we defined scripting language manually (added "<Script Type="text/javascript"/> in the XPDL file). In this case the file was successfully imported.

First of all, as it can be seen from the Figure 32 activities were not well positioned. Second, lanes are in the form of expression lanes which in the "together workflow family" are known as free text expression lanes. Those lanes are not representation of the participants from the model but in fact are special Graph swim-lanes for holding activities which performers are defined as expressions. As to the execution, the implementation of the process was obviously following a single path from the exclusive gateway due to the lack of rule definitions.

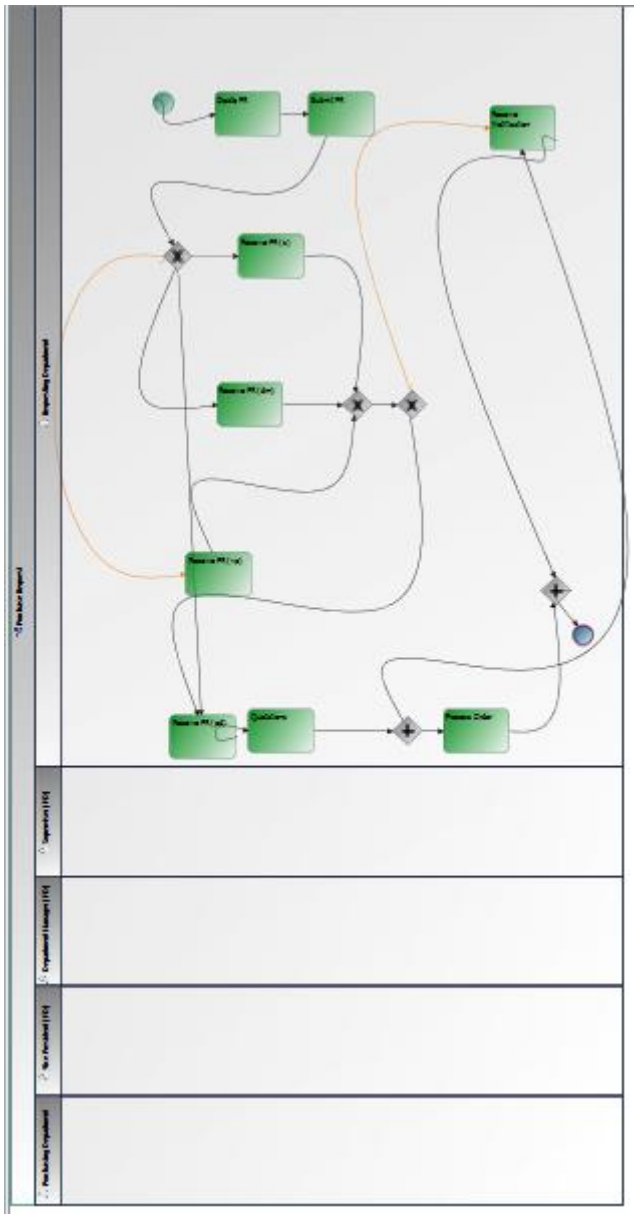


Figure 32 Purchase Request Process from BizAgi imported in TWS

5.2.3 Tests with the process designed in TIBCO

Similarly as we did with other tools, we tried to design the same model also with TIBCO Business Studio. Here we faced with the same obstacles that we did with BizAgi. So, the model (figure 33) was not identical to the original Purchase Request process, but it suffered some changes due to the lack of some properties that TIBCO didn't have. The process looked more similar to the one generated from BizAgi and the same explanations apply here as well.

TIBCO's XPDL in TWS

In contrary from BizAgi, we did not have many difficulties in adapting the file from TIBCO to accurate TWS importation. This happened thanks to the fact that both vendors support the same standard version, which is: 2.1. The only error that prevented the importation was the" Unsupported script language". Just as BizAgi, also TIBCO did not provide the possibility to define the expressions in the conditions with other format then "free text". Once we overcome this error (in the same way as we have done before), the file was successfully imported. Since BizAgi and TIBCO in some way use the same approach in the process design we came across the same failures such as: the graphical representation was not imported accurately, lanes were not defining participants and the XPDL was not executable.

TIBCO's XPDL in BizAgi

In this case we were confronted with similar problems that encountered during the importation of TWEs' XPDL into BizAgi. These were basically the incompatibilities due to different XPDL versions. We managed these problems in the same way as we did with the file from TWE, that is; modification on schema definition, addition of the "description" tag within the participant definition tag and modifications of the Activity definitions. Although after these modifications, the file was imported successfully, its appearance was not correct at all.
From figure 33 can be seen that BizAgi yet again, just as was the case with TWE, is unable to understand information about x y coordinates of activities. But differing from TWE, they (BizAgi and TIBCO) have the same way of expressing lanes, for this reason they are interpreted properly.

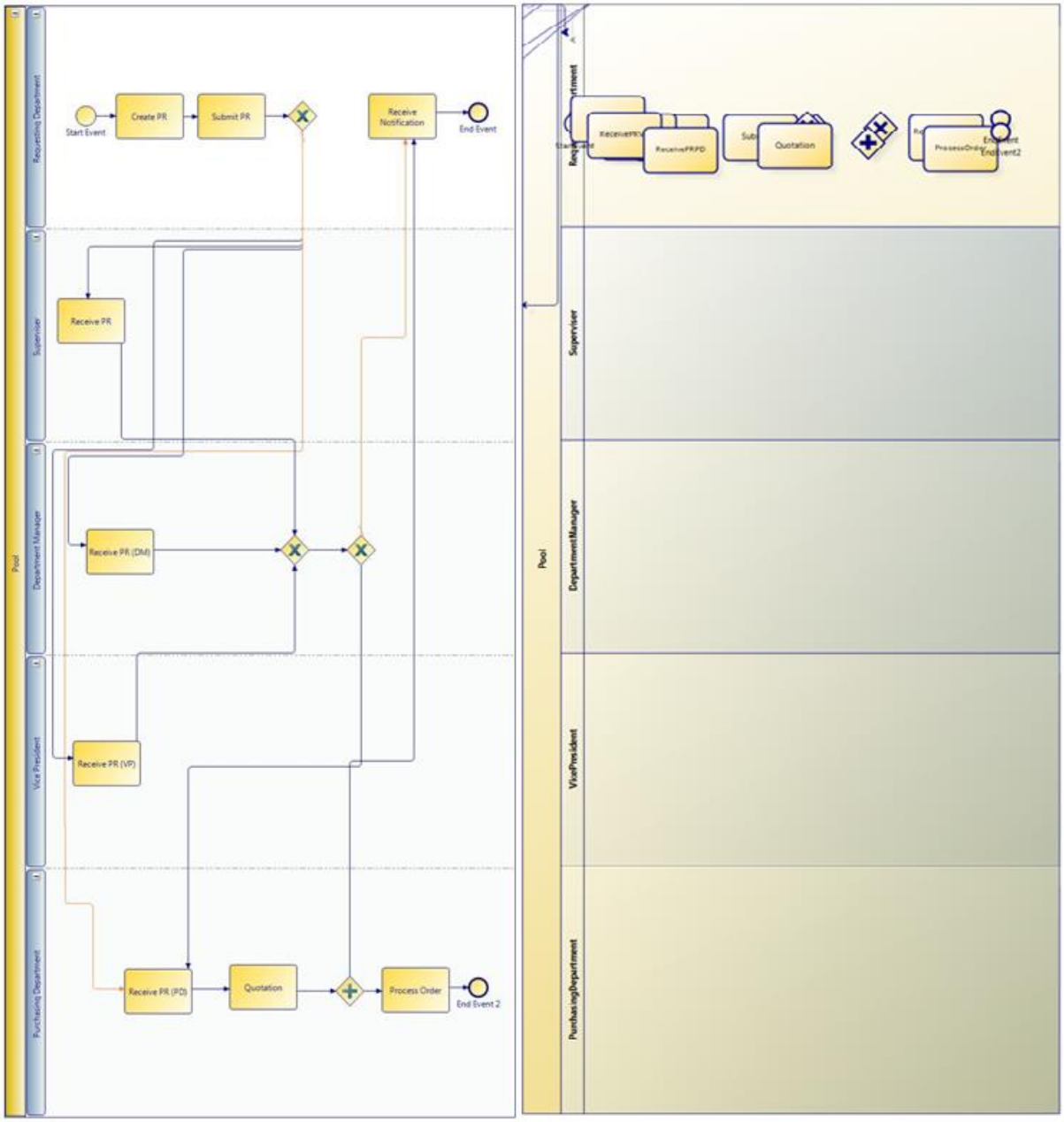


Figure 33 Purchase Request in TIBCO vs. TIBCO imported in BizAgi

5.2.4 Tests with the process designed in YAOQIANG XPDL editor

The way of designing processes in this tool was almost identical to that of TWE. The appearance of the generated Purchase Request process model was also similar to that of TWE, apart from the inability of orienting the model vertically (as was the case with many other tools).

YAOQIANG's XPDL in TWS

Not surprisingly, the model was imported in TWS without any obstacle. Besides that, what happened with most of the tool did not happen this time; seemingly the process had an appropriate graphical view. However, the problem appeared at the time when we tried to test the implementation of the process. In the division "Process Monitor" of the TWS, through the graphical view we noticed the following discrepancy:

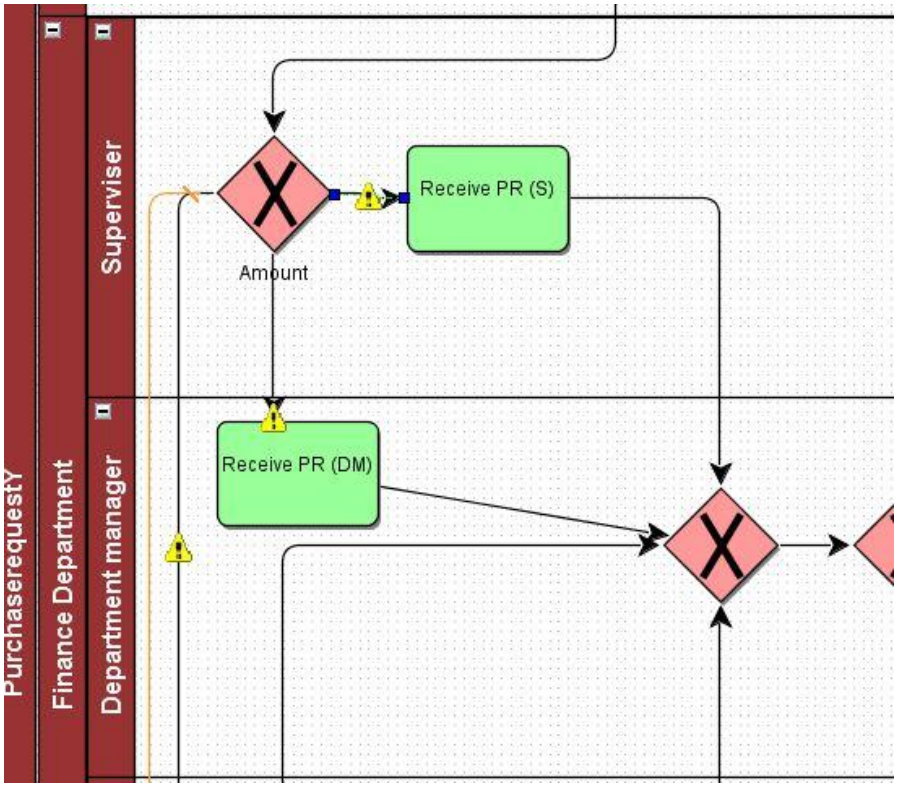


Figure 34 Purchase Request Process from YAOQIANG imported in TWS

Expressions in the conditions defined with YAOQIANG Editor are not recognized from TWS. To understand the mismatch, we decided to compare the XPDL created by YAOQIANG and TWS.

The problem was standing at the discrepancy among the fragments that were defining the expressions. The TWS has not been able to understand the <xpdl:Expression> tag:

YAOQIANG XPDL Editor	TWE
<xpdl:Condition Type="CONDITION"> <xpdl:Expression> newpkg1_wp1_fp1>500&&newpkg1_wp1_fp1<=1500 </xpdl:Expression> </xpdl:Condition>	<xpdl:Condition Type="CONDITION">PR_Total_Price>500&&PR_Total_Price<=1500</xpdl:Condition>

Table 6 Mapping YAOQIANG to TWS – Adapting Condition definitions

Correction of this tag ensured that the process was executed without any problem, except the execution of the merge which we already elaborated before.

The interpretation of the YAOQIANG XPDL editor was already identical to the one of TWE, which was further confirmed during testing the XPDL file generated from it in other tools. Although they possess a difference (YAOQIANG&TWE), which is the interpretation of the rules/expressions in conditions, again this difference did not have any effect on the results of the tests since other tools do not support defining expressions at all. So, during testing of the XPDL file generated from YAOQIANG XPDL editor, we dealt with the same problems and managed them in a same way as we did with TWE.

6. Conclusions

The thesis aimed at testing the portability of the process definitions among different workflow vendors. This was achieved by evaluating different aspects encountered during the import of XPDL files generated from diverse tools.

Those aspects in particular included: 1. if the referent model can be approximately designed in a same way from all the tools selected, respectively the design portability aspect, 2. if these models can be properly imported in the tools presented as XPDL compliant that is the Execution portability aspect and finally 3. The translatability aspect, which actually explores the most common porting problems that occur during import, and specifically evaluates how these problems, can be accurately managed.

The main difficulties we encountered refer to the availability of fully-fledged WfMSs (Workflow Management Systems), as most of the vendors did not enable us to obtain the full versions of their systems.

Moreover, most of the WfMSs that declare themselves to be X-PDL compliant actually implement their own version of an X-PDL dialect.

Table 7 summarizes the main results we achieved during our experimentation. From the results we can conclude that despite the fact that there is a single standard used, there may exist different ways of modeling and interpretation of the same content (process model) among different workflow tools. Some of the tools did not pass the test of the portability at all, while some of them managed to pass being subject to numerous modifications on their XPDL files.

Workflow Products	TWS	WFMOPEN	BONITA	BizAgi
TWE	Success/ problems in *merge* during execution	Import failed	Import failed	Version def., Activity def., Participant def., position of icons
TIBCO	Unsupported Script Lang. No execution	Import failed	Import failed	Version def., Activity def., Participant def., position of icons
YAOQIANG	Success/ problems in	Import failed	Import failed	Version def.,

	expression definitions			Activity def., Participant def., position of icons
BizAgi	Version def., Activity def., Unsupported script Lang. Part. Lanes- Expression lanes, position of icons No execution	Import failed	Import failed	

Table 7 Test results

Workflow vendors support different design techniques, some of those may be only partially shared while some might be totally different and provide a different graphical representations. Although they are all based on a same original core, portability problem lies in the fact that they are implemented "inconsistent" with existing specifications. This is perhaps even natural, since tool vendors are trying to differentiate themselves by offering something more and different. Today the trend is to offer all –in – one suites, so most of them present BPM suites. But, just as we observed from testing's, in all of the BPMN suites run time data such as : the logic in the conditions or the workflow data associated to them is always defined in vendor-specific properties rather than using attributes defined by the BPMN. This is due to the fact that most of the BPM suites do not use a single standard for execution. Thus, we cannot expect total portability of execution models.

However, we need to further consider at least the portability of abstract models. When considering these cases we saw that tools where producing XPDL files that either were of different versions or contained tool-specific interpretations. Consequently, they were unable to understand each other. Since getting vendors use a same way of interpretation is not possible, as a future work we suggest the implementation of a specific compiler that will be able to automatically translate the source XPDL into a target tool-specific format XPDL.

As a first step, the analyses of the meta models of XPDLs supported by source and target tool will be taken. The compiler will take as inputs a set of model fragments which are compliant to the source meta model and apply translation rules to them. These rules will be based on specific implementations of the XPDL standard of target tool, which will directly map the source fragment to the target fragment. And finally, the compiler will produce a model with fragments that are compliant to the target XPDL meta model.

7. Bibliography

Artech Consultores. (n.d.). *GeneXus Business Process Management Suite: Genexus*. Retrieved from Genexus Web Site: http://www.genexus.com/products/business-process-management-suite?en

Bizagi Ltd. (n.d.). *Products: Bizagi*. Retrieved from Bizagi website: http://www.bizagi.com/en/products/bizagi-bpm-suite/overview-bpm-suite

Bonitasoft Inc,. (n.d.). *Documnetation: Bonitsoft* . Retrieved from Bonitasoft Inc, Web site: http://documentation.bonitasoft.com/

Caverlee, J., Bae, J., & Wu, Q. a. (2007). Workflow management for enterprise transformation. *Information Knowledge Systems Management* , 61-80.

Chang, J. F. (2006). *Business Process Management Systems. Strategy and Implementation.* New York: Auerbach Publications. Taylor & Fransic Group.

Chinosi, M. (n.d.). BPeX: A New Approach to BPMN Model Portability. University of Insubria- Varese (Italy).

Chinosi, M., & Trombetta, A. (2012). BPMN: An introduction to the standard. *Computer Standards & Interfaces* , 124-134.

Dourish, P. (2001). Process descriptions as organizational accounting devices: The dual use of workflow technologies. *ACM 2001 international conference on supporting group work* (pp. 52-60). New York: ACM Press.

Dumar, M., & van der Aalst, W. M. (2005). *Process-Aware Information Systems.* New Jersey: A JOHN WILEY & SONS, INC., PUBLICATION.

Gagne, D., & Trisotech. (2011). BPMN Standards What is new in BPMN 2.0 and XPDL 2.2.

Georgakopoulos, D., & Hornick, M. (1995). An Overview of Workflow Management:From Process Modeling to Workflow Automation. *Distributed and Parallel Databases* , 119-153.

iGrafx, LLC. (n.d.). *Process modelin &Analysis Producst*. Retrieved from iGrafx Web site: http://www.igrafx.com/products/process-modeling-analysis

ITU Telecommunication Standardization Sector. (n.d.). Retrieved from International Telecommunication Union: http://www.itu.int

K L Ko, R., S G Lee, S., & Lee, W. (2009). Business Process Management (BPM) Standards: A survey. *Business Process Management Journal , 15* (5).

Ko, R. K. (2009). *A computer scientist's introductory guide to business process management.*

Lawrence, P. (1997). *Wokrflow Handbook.* New York.

Leymann, F., & al., a. (2010). Business Process Management Standards. In J. v. Brocke, *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems (International Handbooks on Information Systems)* (pp. 513-542). Berlin.

M. Khan, R. (2005). What Standards really matter for BPM. *BP Trends* .

Mendling, J., & Neumann, G. (2005). A Comparison of XML Interchange Formats for Business Process Modeling. In *Workflow Hanbook 2005.*

Miers, D., & White, S. A. (2008). *BPMN Modeling and Reference Guide: Understanding and Using BPMN.* Future Strategies Inc.

MOONEY, J. D. (1990). Strategies for Supporting Application Portability. *IEEE Computer , 23* (11), 59-70.
OMG. (n.d.). Retrieved from http://www.omg.org
OMG. (2008). Buiness Porcess Definition Metamodel (BPDM). Retrieved from http://www.omg.org/spec/BPDM/1.0/
OMG. (2010). *Business Process Model and Notation (BPMN 2.0).*
OMG. (2004). Business Process Modeling Notation (BPMN). Retrieved from http://www.bpmn.org/
OMG. (n.d.). *MOF*. Retrieved from OMG: http://www.omg.org
OMG. (2004). UML 2.0 Superstructure Specification. Retrieved from http://www.omg.org/cgi-bin/doc?formal/05-07-04
Ortega, M., P' erez, M., & Rojas, T. (2003). Construction of a Systemic Quality Model for evaluating a Software Product. *Software Quality Journal , 11* (3), 219–242.
Reijersa, H. A., & van der Aalstb, W. M. (2005). The Effectiveness of Workflow Management Systems: Predictions and Lessons Learned. *International Journal of Information Management* , 458–472.
Riccardi, G. (2002). *Database Management with Website Development Applications.*
Rouse, W. B. (2005). A Theory of Enterprise Transformation. In *Systems Engineering* (pp. 279-295).
Scheer, I. D. (1992). *ARIS (Architecture of integrated Informatin Systems).* Springer.
Shapiro, R. M. (2010). XPDL 2.2: Incorporating BPMN 2.0 Process Modeling Extensions. In L. Fischer, *2010 BPMN and Workflow Hanbook*. Florida, USA: Future Strategies Inc.
Shapiro, R. M., & WfMC. (2008). XPDL 2.1 Integrating Process Interchange & BPMN.
Smith, H., & Fingar, P. (2003). Workflow is just a Pi process.
Sourceforge. (n.d.). *WfMOpen*. Retrieved from Sourceforge projects: http://wfmopen.sourceforge.net/
Swenson, K., & Brunt, J. (2006). XPDL 2.0 and BPMN 1.0 Tutorial.
TIBCO Software Inc. (n.d.). *Tibco AccessPoint: Tibco Community*. Retrieved from Tibco Community: http://tap.tibco.com/storefront/trialware/tibco-business-studio-community-edition/prod15312.html
TIBCO Software Inc. (n.d.). *Tibco Business Studio: Tibco Software*. Retrieved from TIBCO Software Inc. Web site: http://www.tibco.com/products/automation/business-process-management/activematrix-bpm/business-studio/default.jsp
Together Teamsolutions Co., Ltd. (n.d.). *Together Workflow Editor : Together Teamsolutions*. Retrieved from Together Teamsolutions Web site.
Together Teamsolutions Co., Ltd. (n.d.). *Together Workflow Server*. Retrieved from Together Teamsolutions Web site.
University of Kaiserslautern. (2003). Lecture notes in Workflow Management Systems. Workflow and web services.
van der Aalst, W. v., & van Hee, K. v. (2002). *Workflow Management: Models, Methods, and Systems* . Massachusetts London, England: The MIT Press Cambridge.
WfMC. (n.d.). *Papers.* Retrieved from bpmnhanbook: http://www.bpmnhanbook.com
WfMC. (n.d.). *Press: Workflow Management Coalition*. Retrieved from Workflow Management Coalition: http://www.wfmc.org

WfMC. (1998). *Workflow Management Coalition Interface 1: Process Definition Interchange Process Model (WfMC TC-1016-P).*
WfMC. (1999). *Workflow Management Coalition Workflow Standard: Interface 1 – Process Definition Interchange Process Model (WFMC-TC-1016).* Florida, USA: Lighthouse Point.
WfMC. (2012). *Workflow Management Coalition Workflow Standard: Process Definition Interface -- XML Process Definition Language (XPDL 2.2) (WFMC-TC-1025).*
WfMC. (2005). *Workflow Management Coalition Workflow Standard: Workflow Process Definition Interface – XML Process Definition Language (XPDL) (WFMCTC- 1025).* Florida, USA: Lighthouse Point.
WfMC. (1995). *Workflow Management Coalition. The Workflow Reference Model. Document Number TC00-1003* (1.1 ed.).
WfMC. (2008). *XML process definition language (XPDL 2.1), WFMC-TC-1025.*
WfMC, W. M. (n.d.). Retrieved from http://www.wfmc.org
White, S. A. (2003). XPDL and BPMN . In L. Fischer, *Workflow Handbook 2004* (pp. 221-238). Florida, USA: Future Strategies.
Yaoqiang. (n.d.). Yaoqiang XPDL Editor - User Guide.

8. Glossary

BPMN: Business Process Modeling Notation

XPDL: XML Process Definition Language

WfMC: Workflow Management Coalition

OMG: Object Management Group

BPMI: Business Process Management Initiative

WFMS: Workflow Management Systems

TWS: Together Workflow System

TWE: Together Workflow Editor