

T-86.5161 Special Course in Information System Integration

# Evaluation of Intalio BPM Tool

Pekka Helkiö, 58490K  
Antti Seppälä, 63212W  
Ossi Syd, 63513T

# Table of Contents

1. Abstract.....	1
2. Introduction.....	2
2.1 Background.....	2
2.2 Objective and Research Problem.....	2
2.3 Methodology.....	2
2.4 Scoping and Constraints.....	2
3. Method for Assessing a BPM Modeller.....	4
4. Test methods.....	6
4.1 Modelling a Process.....	6
4.2 Conformance to BPMN.....	6
4.3 Conformance to BPEL.....	6
4.4 Usage Experiences.....	6
4.5 Review of Intalio Licensing and Documentation.....	6
5. Overview of Intalio tools.....	7
5.1 Intalio Business Process Management suite.....	7
5.2 Intalio Designer.....	7
5.3 Intalio Server.....	7
6. Modelling a Process.....	8
6.1 Setting Up a Project.....	8
6.2 Basic Model Elements.....	9
6.3 Message Schemas.....	10
6.4 Generation of BPEL.....	12
7. Conformance to BPMN.....	13
7.1 Importance of BPMN standard.....	13
7.2 Available element categories.....	13
7.3 Available Basic Shapes.....	13
7.4 Available Start Event Triggers.....	14
7.5 Available Intermediate Event Shapes.....	14
7.6 Available End Event Shapes.....	15
7.7 Available Gateway Shapes.....	16
7.8 Available Pools and Lanes.....	16
7.9 Available Connecting Objects.....	17
7.10 Artifacts Defined by the BPMN Standard.....	17
7.11 Sequence and Message Flows Supported in Standard and in Intalio Designer.....	17
7.12 Analysis of Conformance to BPMN.....	19
8. cursory study of BPEL conformance.....	20
8.1 BPEL standard used by Intalio.....	20
8.2 Comparison with standard.....	20
8.3 Conclusions of conformance to BPEL.....	22
9. Usage experiences.....	23
10. Review of Intalio Licensing and Documentation.....	24
10.1 Licensing and Pricing.....	24
10.2 Documentation.....	24

11. Conclusions.....26



## 1. Abstract

Intalio provides an interesting alternative for business process modelling (BPM) tools. The advantages of the provided solution are wide standards support, clear and well established user interface and low price for especially the modeller part of the tools. The modeller is evaluated against a method constructed in this study. The evaluation method consists of comparing the visual elements and capabilities of the modeller against the Business Process Modeling Notation (BPMN) -standard and constructing a process model with the tool. The usability of the tool is evaluated as exâert analysis and the resulting Business Process Executable Language (BPEL) is compared against latest draft of WS-PBEL 2.0. The modeller is usable, and supports BPMN elements well enough to be used for everyday modelling. The tool has some stability issues. No evidence of non-conformance to BPEL standard was found.

## 2. Introduction

### 2.1 Background

Business process management, BPM, is one of the hyped things at the moment. In an article published in June 2006 Wolf's and Harmon's charted the current state of BPM in their report "The State of Business Process Management" [1]. Current situation is that 7 % of companies in study used a BPM Suite and in the other end 23% of companies have only simple graphical tools such as MS Visio and Powerpoint. Graphical tools were also found most important(24%) along with process modelling tools (22%). When asked about the products and services companies will acquire in 2006, BPM Suites was chosen by 11%. Total amount of companies considering to acquire a modelling tool (BPM suites included) was 24%. Interestingly enough, of this total 48% companies considered some kind of training. In our opinion this speaks that many companies are certainly interested in BPM but not quite yet ready for investments.

This study is done as a part of a HUT course T-86.5161 Special Course in Information System Integration. The approach into the world of BPM tools is hands-on. As the percentages in previous paragraph imply, much more understanding about BPM tools in general is needed. This work attempts to do a part of that, on a detailed and rather practical level.

Our study is focused around Intalio BPM Suite. It is advertised as a open source BPM tool which is rather exceptional in the range of current BPM products. While many licensing restrictions are in place as described later in this paper, the community edition is an easy way for a company to start experimenting with BPM.

### 2.2 Objective and Research Problem

The purpose of this study is to get an understanding about Intalio's BPM tool and form grounded opinion about it. To reach this objective we formulated following research problem and sub-problems.

- How does Intalio function as a BPM modeller?
  1. How to asses a BPM modeller?
  2. Asses Intalio with the discovered method.

To answer the research problem we will solve the sub-problems. The focus of this study is in assesing the Intalio modelling tool with the method discovered in sub-problem one. The terminological difference between BPM modeller and BPM tool is explained in Scoping and Constraints chapter.

### 2.3 Methodology

The method for answering sub-problem one is literature study. Sources of information are both academic articles and more news like writings. Method for the testing is more of practical hands-dirty exploration and comparison with reference material. Usability issues are evaluated in a case-study fashion.

### 2.4 Scoping and Constraints

The scope of this study is to mainly assess the modeller of the Intalio BPM Suite. The modeller, also known as designer, contains most of the interesting features and limitations of the set of tools provided by Intalio. The study focuses primarily on the standards compliance of the designer and its ability to function as a process modelling tool. The server part of the tool

set is given an overview of the suitability as a part of BPM tool set. Thorough testing of the server for different functionalities is outside of the scope of this study.

### 3. Method for Assessing a BPM Modeller

By exploring literature, a suitable model to assess quality of a BPM tool was found and thus it was unnecessary to start creating a model from scratch. In their article "Development of Quality Evaluation Metrics for BPM (Business Process Management) System" YeongSeok et al. construct a framework for analysing quality of BPM systems. From this generic model we later derive our method for analysing Intalio's BPM modeller.

The bulk of the generic quality evaluation framework is based on ISO/IEC 9126 software quality characteristics and barometer guideline which determines the quality from the point of view of the customer. In addition to the basic software quality YeongSeok et al. include BPM specific criteria. Their criteria is based on following research papers by Gartner: "Creating a BPM and Workflow Automation Vendor Checklist", "A BPM Taxonomy" and "Magic Quadrant for pure-play BPM".

The BPM system quality metrics proposed by YeongSeok et al. are presented on the following table(case examples omitted). [2]

Quality Characteristics	Subcharacteristics	BPM Component	Descriptions
Functionality	Correctness	BPM Modeller	Ability to model Business Process that conforms to BPEL and BPML specifications
	Accuracy	BPM Server	Ability to execute Business Process Model accurately
	Interoperability	BPM Server	Extended ability for connecting to each BPM server between organizations
	Security	BPM Server	Ability of data security
Reliability	Maturity	BPM Server	Ability to endure various situations
	Fault Tolerance	BPM Server	Ability to sustain whole systems despite of particular problems
	Recoverability	BPM Server	Ability to handle bad situations
Usability	Understandability	Administration tool	Ability to easily manage business flow without additional explanation
	Learnability	Whole system	General perspective
	Operability	Whole system	General perspective
Efficiency	Time behaviour	BPM Server	Speed of BPM server
	Resource utilization	BPM Server	Resource efficiency of BPM server
Maintainability	Analysability	Whole system	Ability to help administrators resolve problems
	Changeability	BPM Server's Agility	Ability to agilely change business process
	Stability	BPM Server's Agility	Ability to handle frequently occurred problems in unexpected situations
	Testability	BPM Server's Agility	Ability to simulate changed process before executing the process

Quality Characteristics	Subcharacteristics	BPM Component	Descriptions
Portability	Adaptability	Whole system	General perspective
	Installability	Whole system	General perspective
	Coexistence	Whole system	General perspective
	Replaceability	BPM Server	Ability to replace BPM server with other servers
Integrability	Data Integration	BPM Server	Ability to integrate software that is considered as Activity in Business Process Model through data exchange
	Process Integration	BPM Server	Ability to integrate software that is considered as Activity in Business Process Model through inter-function call
Domain Specific Needs	BAM(Business Activity Monitoring)	BPM Monitoring	Ability monitor currently operating business process
	Development Environment	BPM IDE	Ability to support the development of software systems based on designed business process
	Industrial Template Supportability	BPM Builder	Ability to provide Best Practise Process templates

The above list of metrics by YeongSeok et al. is on a rather high level in many cases. They themselves conclude that future research is needed to create checklists for each subcharacteristic. It is also clear that a full test of all the characteristics is quite a sizeable undertaking, even after applying this study's scoping to BPM Modelling. Therefore to establish a feasible method for evaluating Intalio we need to decide which characteristics we focus on and how to measure those.

Focusing the above framework for quality analysis is done by practical limits. As the selected approach is hands-on, functionality is the most logical choice to start from. To test it a purchasing process is modelled and conformance to BPEL and BPMN standards is tested. The latter replaces BPML standard in YeongSeok et al's testing framework. BPEL is more important when considering execution of modelled processes and and BPMN when evaluating the usability and general capability of the modeller.

Other areas in the framework will be merely commented by our experiences. An example of this is reliability, extensive reliability testing is hard to perform but apparent issues in reliability, such as program crashes can be reported. As we focus on the modelling aspect of BPM, server side characteristics can be excluded.

Summary of our tests and the characteristics tested are listed in the following table:

Test	Characteristic tested
Modelling a process	Usability, Functionality
Cursory conformance to BPEL	Functionality:correctness
Conformance to BPMN	Functionality:correctness (replaces BPML)
Usage experiences	Usability, Efficiency, Reliability, Portability(partly)
Review of Intalio documentation and licensing	Domain specific needs, Integrability

## 4. Test methods

### 4.1 Modelling a Process

In this test we evaluated Intalio Designer by modelling RosettaNet Partner Interface Process 3A1: Request Quote [3]. The modelling was done in explorative way, without any reference material or tutorials. The workflow of the modelling process was written down and usability related issues were monitored during the modelling.

### 4.2 Conformance to BPMN

The method constructed for evaluating conformance to BPMN was based on the BPMN standard [4]. BPMN is a standard which describes a high level visual approach to Business process modelling. It contains different elements which describe the business process and the messaging between the process. The study of BPMN conformance was conducted by first examining the standard, and applying the constructs described in the standard to Intalio. This includes setting up different model situations where the statements in the standard could be tested as well as testing the availability and usability of different graphical elements available in Intalio. We used the elements provided by the tool and compared how the tool allows them to be used versus how the standard defines the usage of the elements. Most significant deviations from the standard are described in this study. The findings are further analysed and their impact on the modelling capabilities of the tools are evaluated.

### 4.3 Conformance to BPEL

In this section the BPEL generated by the modeller is analysed for possible non-conformance with the WS-BPEL standard [5]. The evaluation was performed by first studying the standard. Then the output of the modeller is compared against the standard definition and possible non-conforming items are reported.

### 4.4 Usage Experiences

This testing was done two-fold: On the other hand usability issues were monitored during the modelling process and analysed based on general usability guidelines. Another approach was taken by analysing usability from a role of business analyst with minor experience on technical aspect of the business process modelling.

The suitability of Eclipse framework for graphical modelling was evaluated and also possible handicap and blocker issues were elicited.

### 4.5 Review of Intalio Licensing and Documentation

In this part of the study we also examine the licence that comes with Intalio Community edition and compare it against the licence of the commercial Enterprise edition. We also review the documentation that is provided by Intalio.

## 5. Overview of Intalio tools

### 5.1 Intalio Business Process Management suite

Intalio offers a range of solutions for business process management and control. All the tools are available as separate downloads from Intalio web site [6]. Each of the tools is directed to addressing different aspects of business process management. Below is an explanation of the issues two of the most interesting provided tools are designed to solve.

### 5.2 Intalio Designer

Intalio Designer is a tool for modelling a business process with Business Process Modeling Notation (BPMN). This notation can be transformed to BPEL [7] by the Designer. The process which the Designer uses to transform BPMN to BPEL is not documented by Intalio. We believe that the transforming is based on the methodologies described in various papers published by van der Aalst et al. [8]. This generated executable language is then intended to be executed by for example the Intalio Server. An important usage for the Designer is also to communicate functionalities of processes graphically. This paper focuses mainly on the usage and standards conformance of the Designer.

### 5.3 Intalio Server

Intalio server is a separate part of the Intalio tool suite. It can be used to execute the processes modelled by the Designer as well as execute other processes. The Server can integrate seamlessly with the Designer so processes modelled in the Designer can be exported directly to the Server. The Server then executes the designed process. The server can be used also without the Designer to set up a framework for executing business processes modelled with other tools instead of Intalio Designer. This modularity allows for greater customization of the used tools and on the other hand provides more possibilities to integrate or test the server in an already set up environment for example as a replacement or back-up system. In this study the licensing of the server is reviewed.

## 6. Modelling a Process

A RosettaNet standard process called Partner Interface Process 3A1: Request Quote [3] was used to evaluate the modelling capabilities of Intalio Designer. This process has two participants, Buyer and Seller and two messages, Request for Quote from Buyer to Seller and Quote confirmation as a reply from Seller.

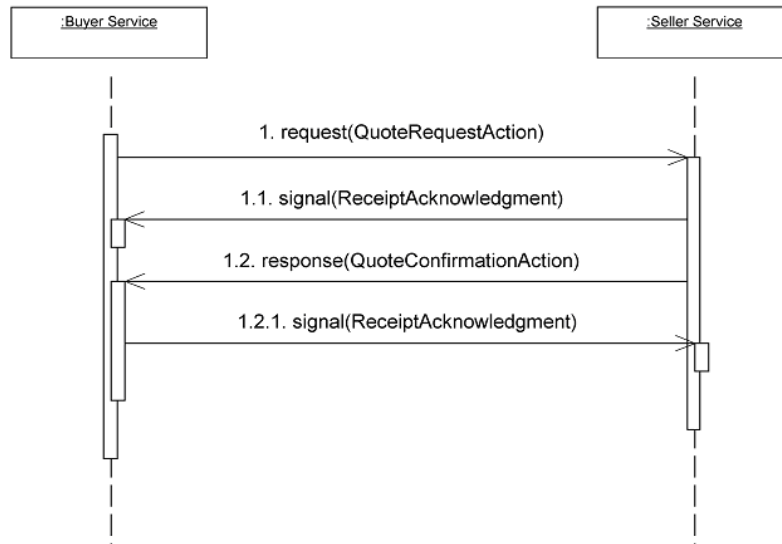


Figure 1: RosettaNet PIP3A1: Request Quote, Copyright 2003 RosettaNet

### 6.1 Setting Up a Project

In order to be able to create a BPMN model with Intalio, a project needs to be set up. In short, project is a folder in a specific location (called workspace) where all files related to the project are located.

There is no default project open when Intalio starts, so it needs to be created via Process Explorer tab context menu or through File -menu. Selecting New / Project brings up a menu where the type of the project can be selected from a tree view. Business Process Project is the appropriate project type for BPMN modelling. The project is also

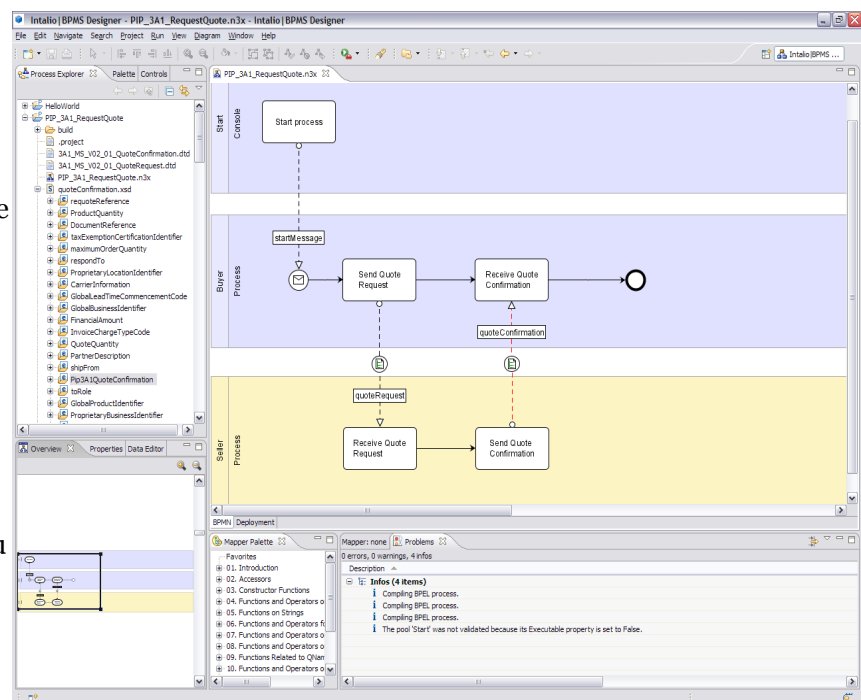


Figure 2: Intalio Designer UI in default configuration. Process Explorer and Overview views on the left. Mapper palette and Problems view at the bottom. BPMN model fills the largest window.

given a name in the same dialog.

After the project is created, a file for the BPMN model needs to be added to the project. This can be accomplished in a similar way to project creation, by selecting New Diagram either from Process Explorer context menu or from File -menu.

## 6.2 Basic Model Elements

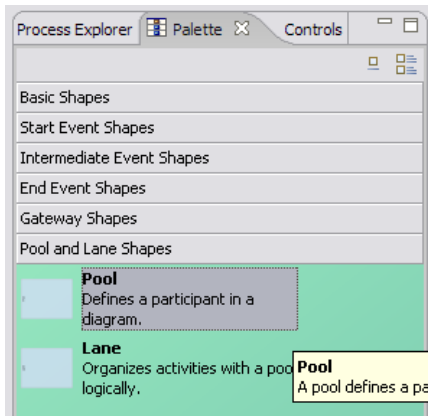


Figure 3: Palette tab

The process modelling starts by placing two Pool -elements, one for Buyer and one for Seller. No tool for adding a Pool is visible in the initial Designer interface, but is found behind a tab named Palette. Palette contains BPMN elements grouped into categories. Pool element is easily found in category named Pool and Lane Shapes. The element must be dragged-and-dropped in order to be added to the model. There is no explicit UI element which hints for the need to drag-and-drop, except the overall similarity to other graphical UIs which use drag-and-drop -feature.

After the Pools are added to the model, there are two Pools with one Lane and one Task in the BPMN model view. The Pools need to be renamed in order to the specify the name of the participant. Renaming function is intuitively placed in the context menu which is reached by right mouse-clicking above the desired object. The tool is called Edit text. However, it was found that Edit text tool was vaguely dependent on the placement of BPMN model window – occasionally choosing the Edit text tool lead to no result. This tiny disturbing factor led to many unnecessary clicks during the modelling. The Lanes are named in the same manner.

After the Pools are in place, Tasks of Request for Quote RosettaNet process are added. The addition of Tasks follows the same pattern as addition of Pools. The BPMN-element for Task is found from Palette tab under category Basic Shapes. Four Task elements are dragged-and-dropped into the model, two into each Pool. These Task all have a default name "Task", so they need to be renamed with Edit Text -tool to depict the name of the activity. The Tasks of the Buyer are named Send Quote Request and Receive Quote Confirmations. Seller Tasks are Receive Quote Request and Send Quote Confirmation.

Now that the Tasks are in place, sequence flow needs to be modelled. In other words this means that arrow shapes which present the order of activities and messages passed between participants are added into the model. Unlike user might expect from the experience cumulated this far, these shapes are not available from Palette -tab, but from the top menu under a tool icon which also contains selection, pan and zoom tools. Sequence Flow Tool is selected from a drop down menu and Tasks are connected by dragging a line from a Task to another. Intalio Designer automatically determines whether the connections are sequence flow connections or message passings to other participants.

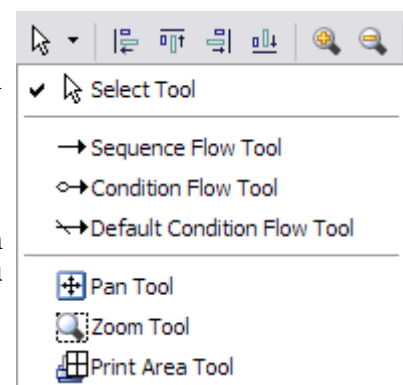


Figure 4: Sequence Flow Tool in drop-down menu.

Send Quote Request -Task is connected to Receive Quote Information -Task and the same is done for the Tasks in Seller -Pool. Message connections are added from Send Quote Request to Receive Quote Request and from Send Quote Confirmation to Receive Quote Confirmation.

These messages are named with already familiar Edit text -tool. Names given are "quoteRequest" and "quoteConfirmation".

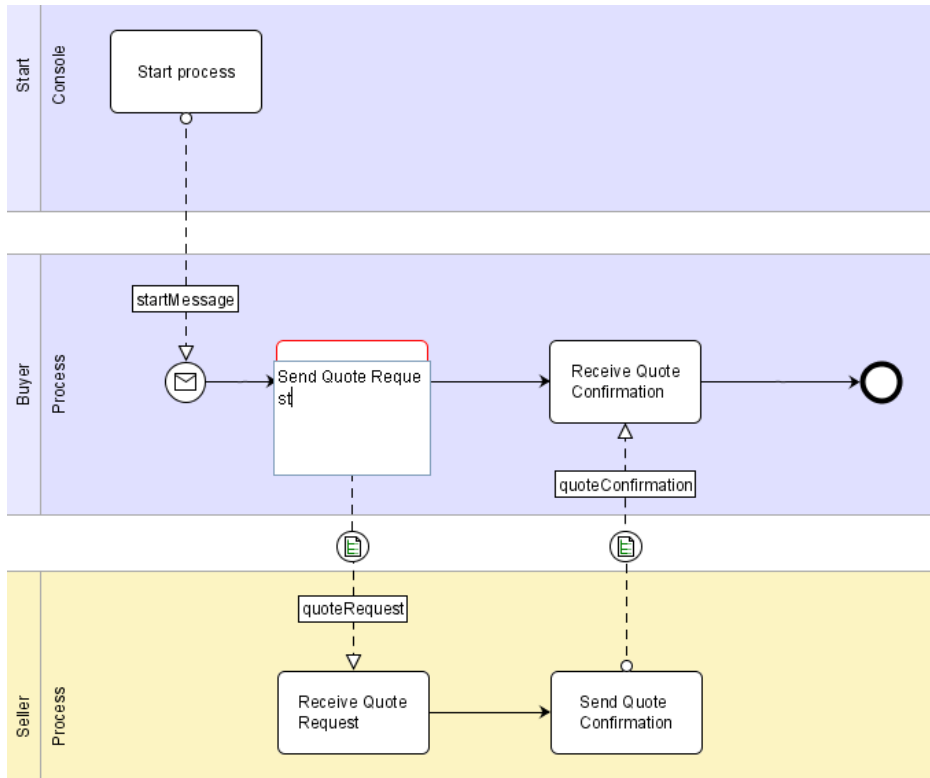


Figure 5: BPMN model with three lanes and Edit text tool in use.

As a next step, start and end points of the process need to be modelled. The BPMN elements to depict these are available in Palette -tab, under categories Start Event Shapes and End Event Shapes. Intuitively, an Empty Start Event Shape is added in front of Send Quote Request task and Empty End Event Shape is added after Receive Quote Confirmation. These shapes are connected to the aforementioned tasks with Sequence Flow Tool.

However, at this point Intalio Designer reports an error in the Problems tab. Error message states the following: "An executable Pool 'Buyer' must start with an Event Gateway or element that receives a 'One-way' or 'Request' message". Without deeper investigation, the problem is fixed as the error message suggests. Another Pool is added to the process model and it's Executable Process -property is set to "false". A single task named "Start process" is placed into this pool. Empty Start Event Shape in Buyer -Pool is replaced with Message Start Event Shape and Start Process is connected to it with the help of Connect Tool. The message is named "startMessage". At this point, no error is reported by the Designer.

### 6.3 Message Schemas

Messages passed between two process participants need to have an message schema associated to them. Schema establishes the rules to which the structure of the document must conform in order to be considered valid. RosettaNet 3A1 specification package includes ready-made XML schemas in DTD (Document Type Definition) for both Quote Request and Quote Confirmation messages.

These DTD -files were imported to Intalio Designer for testing. It was found out that Intalio Designer doesn't support DTD -format. Designer only views DTD-files as text and is unable to parse their structure. For this reason, it is also not possible to associate DTD schemas to messages.

While DTD is considered somewhat obsolete and limited in expressive power, it is nevertheless widely used as it is a part of XML 1.0 specification [9]. The lack of DTD support and lack of integrated conversion tools from DTD to supported schema language can thus be considered as a handicap for Intalio Designer.

The natively supported schema language of Intalio Designer is XSD (XML Schema Definition) [10]. XSD files are XML schema files written in XML Schema language and like DTD, XSD -files define the structure, relation and data types of elements in a document. XSD file can be used to validate whether an XML file conforms to certain format.

Conversion from DTD to XSD is possible, but as mentioned, Intalio Designer doesn't offer tools for it by default. Therefore a commercial third party product called Stylus Studio Enterprise [11] was used to convert DTD files to XSD format.

After XSD conversion is performed, Intalio Designer is able to recognize and parse the schema files. Designer lists the elements of the XSD-schema in Process Explorer -tab, from which they can be dragged-and-dropped to the BPMN model. However, tool based conversion from DTD to XSD was not perfect and Designer reported several errors during the modelling and during the attempt to convert the model into BPEL.

In order to be able to use ready-made RosettaNet messages schemas in the process, XSD schema was manually altered to get rid of error messages reported by Designer. The most significant change was the addition of namespaces to the XSD file. The problems in DTD to XSD conversion cannot be considered as flaws of Intalio, but in practice they may result in a significant extra work and possible appearance of points of incompatibility in a implementation project where messages schemas are not directly available in XSD format.

```
<!--
RosettaNet XML Message Schema
3A1_MS_V02_01_QuoteRequest.dtd (22-May-2003
1:17:46 PM)
This document has been prepared by RosettaNet
(http://www.rosettanet.org) from requirements
gathered during the cluster/segment workshops and
in conformance with the RosettaNet methodology.
-->

<!ENTITY % common-attributes "id CDATA #IMPLIED">

<!ELEMENT Pip3A1QuoteRequest
(fromRole,
GlobalDocumentFunctionCode,
Quote,
thisDocumentGenerationDateTime,
thisDocumentIdentifier,
toRole)>

<!ELEMENT fromRole
(PartnerRoleDescription)>

<!ELEMENT PartnerRoleDescription
(ContactInformation?,
GlobalPartnerRoleClassificationCode,
PartnerDescription)>

<!ELEMENT ContactInformation
(contactName,
EmailAddress,
facsimileNumber?,
telephoneNumber?)>
```

Example 1: A part of RosettaNet DTD file.

```
<?xml version="1.0" ?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.intalio.com/"
xmlns:tns="http://www.intalio.com/">

<xsd:element name='Pip3A1QuoteRequest'>
<xsd:complexType>
<xsd:sequence>
<xsd:element ref='tns:fromRole' />
<xsd:element
ref='tns:GlobalDocumentFunctionCode' />
<xsd:element ref='tns:Quote' />
<xsd:element
ref='tns:thisDocumentGenerationDateTime' />
<xsd:element
ref='tns:thisDocumentIdentifier' />
<xsd:element ref='tns:toRole' />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name='fromRole'>
<xsd:complexType>
<xsd:sequence>
<xsd:element
ref='tns:PartnerRoleDescription' />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name='PartnerRoleDescription'>
<xsd:complexType>
<xsd:sequence>
<xsd:element ref='tns:ContactInformation'
minOccurs='0' maxOccurs='1' />
<xsd:element
ref='tns:GlobalPartnerRoleClassificationCode' />
<xsd:element ref='tns:PartnerDescription' />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Example 2: A part RosettaNet DTD converted to XSD.

## 6.4 Generation of BPEL

The validity of designed BPMN model can be checked during any phase of the modelling process by choosing Validate the Diagram -tool from the top menu. This tool validates the diagram in the background and reports possible errors in Problems -view.

Generation of BPEL is as straightforward as validation of the diagram. By choosing Generate BPEL, WSDL and Compile for PXE -tool Intalio Designer generates the BPEL files for the process. Also WSDL (Web Services Description Language) [12] and PXE files are generated.

WSDL files are XML-format descriptor files which describe how the process should be used as a web service; it establishes the messages and protocol formats which should be used to access the web service and service set the web service offers.

PXE [13] is a "process execution engine" developed by Intalio and Apache Software Foundation [14]. It is used to execute processes modelled in BPEL by Intalio Server, among others. Generated files are in binary format.

After the BPMN model has passed validation and has been successfully compiled, it can be deployed to the Intalio Server with single click on Deploy the Diagram to Server -tool.



Figure 6: Shortcuts to model validation, BPEL generation (including WSDL and PXE compilation) and server deployment available in top menu.

### Generated Files

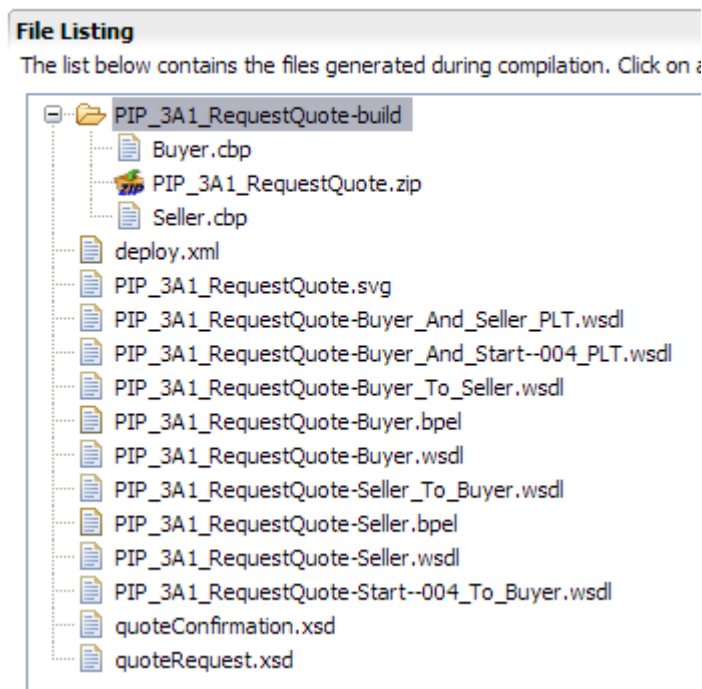


Figure 7: Files generated by compilation process.

## 7. Conformance to BPMN

### 7.1 Importance of BPMN standard

Intalio Designer is a tool for modelling business processes with Business Process Modelling Notation (BPMN). It is a standard for modelling business processes graphically. It is defined and currently maintained by Object Management Group (OMG). The elements available in Intalio Designer and their usage descriptions are represented in tables below. Also the standardized elements which are missing from the tool are represented as a separate section.

While the importance of conformance to prominent standards such as BPMN can be disputed it should be noted that without any standards conformance different modelling tools would be incompatible with each other. Even though BPMN can be regarded as a recommendation of how business process elements are visualized, we feel that evaluating the conformance of BPMN standard in a stricter sense helps in understanding whether the tool is suitable for modelling certain types of processes. Following the standard closely allows tools to produce portable models where the same model can be viewed and edited in different environments. Good standards conformance is also an advantage in the usability of the tool as the modelling process and notations does not have to be learned from beginning if changing tools.

### 7.2 Available element categories

The Designer -tool contains most of its BPMN -elements in a Graphical User Interface element called palette. The contents of the palette are organized into different categories for easier accessibility. The organization is done by categorizing the functionalities element shapes into logical groups. The groups are *basic shapes*, *start event shapes*, *intermediate event shapes*, *end event shapes*, *gateway shapes* and *pool and lane shapes*. These groups appear to correspond loosely on the elements defined in the standard. The apparent corresponding standard element definitions are *Activities*, *Events*, *Gateways*, and *Swimlanes*. Each of the different group is described below in more detail.

### 7.3 Available Basic Shapes

The basic shapes contain the fundamental elements for representing activities in BPMN -diagram. Generally they are tasks or sets of tasks combined into elements called processes. Basic shapes are based on the BPMN -standard section 9.4 Activities [15] p. 63

Shape name	Description	Notes
Task	A basic activity	
Sub-Process	An activity that contains other activities	
Looping task	A repeatable activity	Looping tasks are not separate entities according to BPMN standard.
Looping sub-process	A repeatable activity which contains other activities	Looping sub-processes are not separate entities according to BPMN standard.
Annotation	Annotation provides information about the diagram	Annotations are generally not compiled into BPEL

Table 1: Available basic shapes in Intalio Designer

When examining the BPMN -standard we found that annotations are not categorized under the same category as the tasks and processes and their loop -variants. Annotations are usually defined as Artifacts [15] p. 91, elements which give the modeller a possibility to embed information into diagrams. Combining Artifact -elements with Activity elements is a minor inconsis-

ency in the categorization as Artifacts are generally not compiled into executable language. The style of separating the entities for looping tasks and sub-processes is not clearly encouraged in the standard. Instead the graphical element should change its entity type automatically if the task is executed in loops. Intalio behaves differently, the task type does not change automatically, but it can however be later changed when editing the model.

According to BPMN standard a task type named Multi-instance should be available. This type of task is a form of looping task which specifies how many loops or instances of the task will be executed. According to our studies support for multi-instance type tasks is missing from Intalio.

## 7.4 Available Start Event Triggers

Start event triggers are based on the section 9.3.2 [15] p. 35 of the standard. These elements indicate the starting of process flow usually when a predefined trigger condition occurs.

Shape name	Description	Notes
Empty	Indicator for execution beginning	Name used in Intalio differs from the standard. Element is called None in standard BPMN.
Message	Indicator for execution beginning after receiving a message	
Rule	Triggering execution when certain conditions are met	Intalio description does not match the BPMN standard. According to Intalio "Rule Starts a process after receiving a message"

Table 2: Start event triggers in Intalio

Start event triggers supported in Intalio Designer are a subset of the elements defined in the standard. According to standard there are additionally elements of named *Timer*, *Link*, and *Multiple*. The *Timer* triggers are used to initiate starting of the process execution based on a pre-defined time or interval. *Link* -triggers allow initiating the process based on the outcome of a previous process and *Multiple* -triggers allow modelling processes which can start with multiple possible events or messages.

It is apparent that without these triggers starting of different processes cannot be modelled at the same preciseness as described in the BPMN -standard. Especially modelling multiple start events or a process that runs periodically becomes difficult. Some of these restrictions could be worked around for example by creating special processes where execution is delayed and then triggered to move forward when conditions matching e.g. the multiple -trigger are all matched. This however can bring an unnecessary level of complexity to the models which could be avoided if all the elements described in the standard were supported.

## 7.5 Available Intermediate Event Shapes

Intermediate events are generally events that occur between start and end events. Intermediate events are described in the standard section 9.3.4 [15] p. 43. Intermediate events are used to model delays in process execution, message handling or handle other exceptions in the process flow.

Shape name	Description	Notes
Empty	An event occurring in the middle of the process	Name used in Intalio differs from the standard. Element is called None in standard BPMN.
Message	A message can be used to trigger the execution of a process forward	
Timer	An event which can be set to trigger at a specific time or specific intervals	
Error	An element capable of creating error messages as well as catching them	
Compensation	Indicator how a compensation is applied to a sub-processes	
Rule	An event which triggers when a certain rule comes true	Intalio description for Rule does not match the standard. According to Intalio the element "Waits for a message from a participant"

Table 3: Intermediate event shapes in Intalio

Intermediate event triggers are also a subset of BPMN -standard. In addition to the triggers in Intalio Designer also the following triggers are recognized by the standard: *Cancel*, *Link* and *Multiple*. The *Link* and *Multiple* -event triggers behave in much the same way as described in the start event section. The *Cancel* -event is an event trigger which should cancel a sub-process or task immediately when triggered. The difference between cancel and *Terminate* -end events is that cancelling should also negate the effects the process has performed so far.

Diverting from the standard by missing the *Cancel*, *Link* and *Multiple* -triggers can cause either incompatibilities with other tools and difficulties with modelling certain processes relying on these triggers. A such a process is for example a situation where there are competing processes and completing one of them should cancel the execution of others. [16] p. 21

## 7.6 Available End Event Shapes

End event shapes are triggers that terminate the execution of a process. They may occur in the middle of a process denoting ending of process when a certain trigger event arrives or in the end of the process when the process is completed. End events are described in BPMN standard section 9.3.3 [15] p. 40.

Shape name	Description	Notes
Empty	An indicator for ceasing the execution of a process	Name used in Intalio differs from the standard. Element is called None in standard BPMN.
Message	Delivers a message when a process is ended	
Error	Ends the execution with an error message. The message will be caught by another intermediate message event	
Compensation	Identifies that a compensation is necessary when the business process ends	
Terminate	Forces an immediate termination of the process	

Table 4: End event shapes in Intalio

The standard defines additional elements. The end event triggers defined in standard are

*Cancel*, *Link* and *Multiple*. The definitions of each are similar as described earlier in this paper. In process end situation the triggering of the events cause the execution either to jump to other process (*Link*), Cancel a sub-process transaction (*Cancel*) or cause multiple other events to be forwarded to other triggers (*Multiple*). Missing these adversely affects the ability of the Intalio to accurately model process ending events.

## 7.7 Available Gateway Shapes

Gateway shapes are elements in BPMN which allow for execution flow in a process to interact with the process content. The elements can be used to e.g. model situations where flow is either divided or joined to/from separate flows, or where there are multiple alternatives for the flow to proceed. Gateways are described in section 9.5 of the BPMN -standard [15] p. 68.

Shape name	Description	Notes
Data-based exclusive	(XOR), Only one of the alternative paths is possible	Decision is made based on the input data
Event-based exclusive	(XOR), Only one of the alternative paths is possible	Decision is made based on an event happening in the process
Data-based inclusive	(OR) One or more of the alternative paths are possible at the same time	
Parallel	A gateway which synchronizes parallel paths either by joining or dividing them at the same time	

Table 5: Gateways available in Intalio Designer

The separation between inclusive and exclusive gateways is not defined as a separation of the graphical elements in the standard. In standard the type of the element is defined as selectable attribute. The functionality for selecting the type of the element is currently missing from Intalio Designer and the type of the element cannot be set via a parameter, but it is included as a built-in assumption in the graphical element instead. This causes a restriction which disables run-time changes of element types which should be possible if following the standard.

An element not present in the selection of gateway elements provided by Intalio is *Complex* gateways. This is a form of gateway which allows creating more complex gateways, which would be difficult to create with the standard simple gateways. Complex gateway is defined in the standard as “a set of linked simple gateways into a single, more compact situation” [15] p. 82. We found in our studies that complex gateways are not necessary for representation of BPMN -standard capabilities as they can be replaced with sets of other gateways. However the possibility of merging gateways into a complex gateway would improve the visual look of the model, as there would be less elements in a graph.

## 7.8 Available Pools and Lanes

Pools and swimlanes are used to partition and sort the activities in a diagram. They are defined in the BPMN standard section 9.6 [15] p. 86.

Shape name	Description	Notes
Pool	Represents a participant in a process	
Lane	Organizes and categorizes activities in a pool	

Table 6: Swimlane types available in Intalio Designer

Intalio Designer appears to implement all the functionalities necessary for representing swimlanes defined in the standard.

## 7.9 Available Connecting Objects

Connecting shapes are used to represent the flow of process and describing the messages that can be sent between processes. Messages are an integral part of BPMN standard and are described in chapter 10 [15] p. 99

Shape name	Description	Notes
Sequence flow	Shows the order of the tasks and activities in a diagram	
Condition flow	Flow element indicating relation to a gateway	
Default condition flow	Flow element which is selected as a default from a gateway	

Table 7: Connecting objects available in Intalio Designer

Connector objects implemented in Intalio Designer conform to the standard. The usage of the messages is examined more closely in section Sequence and message flows supported in standard and in Intalio Designer.

## 7.10 Artifacts Defined by the BPMN Standard

The BPMN standard defines an additional set of elements named Artifacts. These are elements which generally do not affect the execution of the created process but instead provide additional information of the process to the viewer. This category of elements is not available in Intalio at all. The only element belonging to the artifacts -category was located in Basic shapes -category under the name Annotation. Other elements belonging to Artifact -category are *data objects* and *groups*. *Data objects* are used to represent actual data handled in different phases of a process, such as documents, reports and general data provided and needed by the process.

*Groups* are visual elements which are used to represent elements which form a logical entities. According to our studies the grouping function in Intalio Designer acts differently than what could be expected. Currently the grouping only creates a sub-process of the selected tasks. The semantic meaning of sub-process is different from the meaning of group -function in the BPMN -standard. It is also not possible to group tasks in different pools together as is illustrated in BPMN standard Fig. 9.39 [15] p. 96 and in examples given by S. White [17] p. 17.

## 7.11 Sequence and Message Flows Supported in Standard and in Intalio Designer

The BPMN standard defines how process- and message flows should function in a BPMN model. This definition is represented in tables 8.4 and 8.5 [15] 27-28. The table contents are represented below together with our findings of how Intalio Designer follows the standard definitions. For this study we created a test diagram which contained two identical lanes with all possible components. We then tried to connect the components in a single swimlane via

sequence flows, or connect elements in different swimlanes with messages. The results are represented in a table below.

From/TO	Start event	Sub-process	Task	Gateway	Pool	Inter-mediate event	End event
Start event		S s	S s	S s		S s	S s
Sub-process	M m*, s	M, S m*, s	M, S m*, s	S s	M	M, S m*, s	S s
Task	M m	M, S m, s	M, S m, s	S s	M	M, S m, s	S s
Gateway		S s	S s	S s		S s	S s
Pool	M	M	M		M	M	
Intermediate event		S s	S s	S s		S s	S s
End event	M m	M m	M m		M	M m	

Table 8: Connections possible in BPMN standard and in Intalio Designer

(M), Message flow possible according to standard

(S), Sequence flow possible according to standard

(m), Message flow supported in Intalio Designer

(s), Sequence flow supported in Intalio Designer

(\*) Intalio allows message delivery to a task inside a sub-process, but does not support delivering messages to the sub-process itself

Our study reveals that Intalio's support for BPMN standard message- and sequence flow is not a complete set of all the flow possibilities. Intalio Designer follows the standard closely in a sense that it does not allow messages or sequence flows which are not defined in the standard. However the support for all the possible flows is also incomplete. Most complete support is in sequence flow, where Intalio performs well. Sequence flows were found out to be available between all important elements.

Message flow support lacks the support of messaging between pools and swimlanes. It was found out that the pools in Intalio Designer do not support sending or receiving messages, and the message must be originating from a graphical entity residing inside a pool. Apparently pools are used only as contexts in Intalio and sending messages always requires an activity.

Sending messages between sub-processes in different pools was found out to perform differently when compared to standard. In Intalio it is only possible to send messages to tasks inside sub-processes but not directly to the sub-process itself. The standard states that sending messages to sub-processes should be possible. Interestingly the sequence flow could be linked to directly to sub-processes, but not into the tasks inside a sub-process.

## 7.12 Analysis of Conformance to BPMN

In the section above we have studied the support of different BPMN -elements in Intalio Designer. We have concluded that support for different graphical elements in BPMN varies from well supported to partially supported to not supported at all depending on the element in question.

Our findings indicate that there could be problems modelling certain business processes which rely on the features that are unsupported by Intalio. On a general level the missing support for the *Multiple*, *Link*, and *Cancel* -elements causes Intalio Designer to not be able to model situations related to these messages or sequences. We noted however that some of these elements can be modelled equivalently with other notations which are supported in Intalio Designer. The missing support of sending and receiving messages to lanes and pools may be more severe as these kinds of situations cannot be modelled with an easy work around. This may however be negligible, as such processes would in our opinion be difficult to create in real world and thus not necessary to model.

Another issue of not supporting all the Artifacts does not limit Intalio's ability to actually model a physical process as these elements do not affect the BPEL even though applied to the notation. The problem of not supporting Artifacts is that the semantic meaning of the business process may be more difficult to communicate to the viewer, as there are not enough capabilities to embed information about task groups and necessary physical documents related to the process.

We also find that the provided object groups of Intalio could be enhanced to conform more to the standard. Currently Intalio uses a categorization scheme which deviates from the standard by dividing events separately to start- intermediate- and end events. This is probably done in order to improve the usability of the tool. Also other minor deviations from the standard exist such as representing gateway types as graphical entity properties instead of gateway attributes.

The basic ability of Intalio Designer to model business processes is in theory on a reasonable level. There are enough graphical elements and despite incompatibilities with the standard notation, the tool is theoretically able to represent most of the business processes without problems. When performing more difficult modelling the tool may not have sufficient elements to model everything directly. We believe however that most of the incompatibilities faced when modelling a process can be at least worked around by using other notations for the model. According to Srikarsemsira & Roongruangsuwan [18] incompatibilities with the standard exist also in tools that have been on the market longer than Intalio.

## 8. cursory study of BPEL conformance

Purpose of this chapter is to search BPEL code generated by Intalio Designer from a BPMN model for evidence of non-conformance to BPEL standard. Full code is presented in Appendix I.

### 8.1 BPEL standard used by Intalio

To examine conformance to BPEL we need to determine which standard Intalio attempts to implement. As Intalio doesn't provide this kind of information, first possible source is the generated BPEL code.

Following contains two parts of the process definition from the code generated.

```
<bpel:process
  xmlns:bpel="http://schemas.xmlsoap.org/ws/2004/03/business-process/"
```

```
  queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0"
  expressionLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0"
  suppressJoinFailure="yes" abstractProcess="no" exitOnStandardFault="no">
```

The second line would suggest the standard being used, but the link leads to a document that does not exist. The second part of code suggest WS-PBEL 2.0. To confirm this we need to look for other sources. From Intalio's community forums we found a post with same question. A forum administrator answered with this reply: " Intalio|Designer supports translation of BPMN to BPEL 2.0 while Intalio|Server executes BPEL 2.0 and BPEL 1.1" [19]. Based on experiences from gathering information WS-BPEL 2.0 seems to be what is meant by BPEL 2.0. In our study applicability of this standard is increased by the fact that one of the standards editors, Assaf Arkin, is representing Intalio [20].

### 8.2 Comparison with standard

The business process we compared with standard is RosettaNet PIP\_3A1\_RequestQuote/Buyer. [3]

#### Top-level attributes

Top-level attributes are defined in opening process -element. Attributes defined by standard are: queryLanguage, expressionLanguage, suppressJoinFailure, abstractProcess, exitOnStandardFault [20] (p.23). All these are present in the code generated and they are given feasible values.

#### Document Linking

To get information on things such as partnerlink types, variable types and other constructs the process definition can be extended by other documents. This is done with import -element. It has mandatory attribute, importType, and two optional attributes: location and namespace. In our example the element is following: [20](p.32)

```
<bpel:import
  namespace="http://www.example.com/PIP_3A1_RequestQuote/Buyer"
  location="PIP_3A1_RequestQuote-Buyer.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
```

Though [http://www.example.com/PIP\\_3A1\\_RequestQuote/Buyer](http://www.example.com/PIP_3A1_RequestQuote/Buyer) does not exist, the syntax is correct.

## Partner Links

Interaction of business process and other services are modelled by partner links. Partnerlink-Type specify the relationships and roles played between two services. Required attributes are myRole and/or partnerRole. If initializePartnerRole is used, then partnerRole must exist. [20](p.37-38)

```
<bpel:partnerLinks>
  <bpel:partnerLink name="Buyer_And_Seller_PLink"
    partnerLinkType="wsdl:ns4:Buyer_And_Seller_PLT"
    myRole="Buyer_Role" partnerRole="Seller_Role" initializePartnerRole="yes"/>
  <bpel:partnerLink name="Buyer_And_Start--004_PLink"
    partnerLinkType="wsdl:ns3:Buyer_And_Start--004_PLT"
    myRole="Buyer_Role" initializePartnerRole="no"/>
</bpel:partnerLinks>
```

The restrictions mentioned are met in this example.

## Variables

Variables are needed for holding messages and data when executing business process. Following restrictions apply: First variable name must not contain '.'-mark. It must also have exactly one of these attributes: messageType, type or element. [20](p.46-57)

```
<bpel:variables>
  <bpel:variable name="quoteRequest" messageType="Seller:quoteRequest"/>
  <bpel:variable name="quoteConfirmation" messageType="Seller1:quoteConfirmation"/>
  <bpel:variable name="startMessage" messageType="Start:startMessage"/>
</bpel:variables>
```

The restrictions mentioned are met in this example.

## Main activity

Each business process has one main activity [20](p.24). In our code this is a sequence starting on line 41 with opening element: `<bpel:sequence name="Buyer-Sequence">`. This sequence is composed of one receive -activity which initiates the process, one invoke -activity which sends the request and another receive -activity for confirmation. The activities in sequence are performed in order.

First receive -activity of the sequence:

```
<bpel:receive bpmn:id="ID-2a04fdb4-c127-1004-8ed3-6d9599fe9bb8"
  name="Name_Not_Specified-Receive"
  partnerLink="Buyer_And_Start--004_PLink"
  portType="Start:Start--004_To_Buyer_PortType"
  operation="Name_Not_Specified" variable="startMessage" createInstance="yes"/>
```

According to standard portType is optional, but if it exists it must match the portType specified in partnerLink [20](p.24). This portType should be specified in "[http://www.example.com/PIP\\_3A1\\_RequestQuote/Buyer To Seller](http://www.example.com/PIP_3A1_RequestQuote/Buyer_To_Seller)" if the mentioned page existed.

Attribute createInstance=Yes means that this activity starts the process. Each process must contain at least one of these. [20](p.36)

Then the invoke -activity:

```
<bpel:invoke bpmn:id="ID-216441ca-c127-1004-8c91-58328da8f429"  
  name="Send_Quote_Request-Invoke"  
  partnerLink="Buyer_And_Seller_PLink"  
  portType="Seller:Buyer_To_Seller_PortType"  
  operation="Receive_Quote_Request" inputValue="quoteRequest">  
  <rdfs:label xml:space="preserve">Send Quote Request</rdfs:label>  
</bpel:invoke>
```

Similarly to receive, same constraints for portType apply [20](p.25). The sixth line is Resource Description Framework(RDF) vocabulary definition, not BPEL [21].

The final receive activity contains no new points of interest. The sequence itself and the activities it is composed of meet restrictions presented.

### 8.3 Conclusions of conformance to BPEL

The BPEL code generated was rather simple and no evidence of non-conformance to WS-PBEL 2.0 standard was found. The output of the modeller leads us to believe that the generated BPEL is syntactically conformant to the standard. However this study does not focus on the semantic conformance and resemblance of BPMN used when to generated BPEL. In this brief analysis main focus was on syntax, as references in the code did not exist. It is also worth noting that WS-PBEL 2.0 is still in draft stage.

## 9. Usage experiences

Intalio Designer uses components from the projects of Eclipse community [22][23]. Eclipse community is especially known for its popular integrated development environment for Java programming language. If the user of Intalio Designer has previous experience on Eclipse, the UI components and their capabilities, menus and general usability logic are immediately familiar to the user. If user has no previous experience on Eclipse, it is likely that Intalio UI is somewhat complex for the new user: There is a lot of options available in menus and a lot of views open in the default perspective.

It should be noted that Eclipse framework has been originally designed as an application development environment; to handle multiple text files open simultaneously and to assist developer in code-writing, formatting and building. Graphical BPMN modelling is a very different task from code writing and this could be used as argument whether Eclipse-like UI is suitable for graphical modelling, although BPMN modelling also includes elements similar to code development.

Intalio Designer installation files are delivered as a single Java Archive (JAR) file. It may not be apparent for the user that this file needs to be executed, as it can appear as an archive file depending on the specific system configuration. If the system environment is appropriately configured with Java, the installation file can be executed simply by double-clicking the file. This opens a straightforward graphical installation dialog in which the licence, installation path and components to be installed are presented. There is a possibility to edit installation path, but it is not required. The installation dialog is highly analogous to installation dialogs in other Windows applications.

When Designer is started, there is no default model template open so a first time user has to first to discover how to create a project and how to create a diagram file inside the project. On the other hand, the project and diagram creation is very analogous to file management in any modern desktop environment.

Dragging-and-dropping model elements from Palette-tab is very intuitive and allows fast and easy placement of model elements. Nevertheless, naming of the model elements is not as usable as it could be. This simple tool fails to function occasionally and relatively long time can be wasted on retries. Connecting the elements with sequence flow arrows is also slower than it could be, as selection of sequence flow tool from the top or context menu takes some precise mouse pointing. This can be easily circumvented if user discovers that sequence flow tool can be accessed through space -key.

The functionality of Undo -tool in Designer is unpredictable, as some operations cannot be undone without any apparent reason. This is likely to result in problems with more complex models, if erroneous operations cannot be properly undone by the user.

Designer evaluates the validity of XSD -files on parallel with the editing – similarly to Eclipse source code validation. This feature functions generally slow and sometimes causes the whole application not to respond. This often happens when file is under editing and is temporarily in non-validating state. This kind of stability issue is a severe blocker for Intalio Designer production usage.

Despite the noted problems, it can be estimated that Intalio Designer still has better than average usability and learnability, as the user interface inherits the good sides of very mature and many times trialled Eclipse interface and graphical model editing tools are analogous to other such tools on modern graphical desktop environments. Through good learnability user can also accomplish high work efficiency. As for reliability aspect, Intalio Designer has still to mature. In general, from the perspective of usability aspect Intalio Designer would be suitable for small enterprise use.

## 10. Review of Intalio Licensing and Documentation

### 10.1 Licensing and Pricing

Intalio web site states that the tools are Open source[24]. When examining the licence it was found out that this is not entirely true in a meaning of traditional open source. In traditional open source the actual source codes for creating the binary files is distributed with the product or upon separate request [25]. Intalio is basing their products on open source tools and projects, but the source code of the actual modifications to these projects done by Intalio is not publicly available.

Intalio licences the BPM tool suite in two different licences, the Community Edition and the Enterprise Edition [26]. The community edition for each of the tools is available for free. Interestingly there are no differences in the functionalities of the Community vs. Enterprise editions of the tool suite, the only differences are the limitations described in the licence agreements of the tools.

The community edition licence agreement [27] states that users of Intalio products have *”limited license to (a) download the Intalio Software in object code form only”*. The tools however are available for download for free.

Further examination of the community edition licence [27] reveals that some restrictions are placed on which products can be used together with Intalio server: [27] *”Developer shall not (and shall not allow any third party to):” “(ii) use the Intalio Software in live production or with live data except on allowed databases or application server”*. The licence actually restricts the allowed databases which can be used in conjunction with Intalio server. The only allowed database is the MySQL database bundled with the server distribution. This licensing constraint can adversely affect the portability and functionality of the server product provided by Intalio. According to Intalio the Community Edition is also bundled with Apache Geronimo and Apache Derby databases. Enterprise edition of the server does not have the bundled database and server constraints [26].

It was also found out that Intalio tools have used some third party components which stop functioning over time. The effect of this is that Intalio is rendered unusable unless new version of the tool is available for download in which the licence to third party components is renewed. These licensing issues are generally interesting in the context of portability and maintainability of BPM -tools. With less licensing constraints the Intalio products could be more maintainable.

Licence	Intalio Designer	Intalio Server
Community edition (free)	No constraints to the usage of the Designer enforced by the licence.	Can only be used together with Apache Geronimo server, Apache Derby database and MySQL database.
Enterprise edition	No constraints to the usage of the Designer enforced by the licence.  May include source codes to the tool. Functionally the same as Community edition	No constraints to the usage of server- or database software. Virtually any deployment environment supported  Price: \$25,000.00 per CPU per year, including support & maintenance.

Table 9: Comparison of the licensing effects to Intalio Designer and Server

### 10.2 Documentation

Documentation of Intalio is scarce. There are tutorials with screenshots and reference manuals to basic operations (also with screenshots) available on Intalio website for registered users [28]. As tutorials and reference manuals are presented with numbered steps and screenshots, these sources of help are easy to follow and replicate and they offer a good starting point

for an Intalio Designer user. Despite this, the tutorial and reference manual are difficult to search through and they only cover the basic usage of Designer, not technical details on how some specific task should be done in actual production environment.

There are also forums available for registered users where it is possible to get help from other Intalio users and from Intalio employees. The forums suffer from myriad messages which are so context-specific that they have virtually no value for other users than the poster. Also the search features for the forums are limited.

The scarcity of Intalio documentation can be explained through the business model of Intalio. The company offers a free product but gets the revenue and profit by offering training and support services for the product. The “1-2-3” concept presented at the Intalio web site supports this conclusion [29].

## 11. Conclusions

In this study we have established a method for evaluation of generic BPM tool based on earlier research done by YeongSeok et al [2]. Some of the methods presented by YeongSeok were discarded as unsuitable for this project and the rest were altered to better suit the hands-on and quite practical approach taken in this study.

It was established that the assessment of BPM modeller should include testing of the modeller for conformance to BPMN and BPEL standards. Also usability and stability should be evaluated and documentation should be reviewed. In the specific case of open source involved also the licensing terms should be verified.

As the scope of the study was on the level of a general analysis, the evaluation of the Intalio BPM tool was done with explorative methods. The different aspects of the software were reviewed and some specific tasks were done as practical walkthrough through the features of the software. The experiences were written down and analysed in parallel with continued practical testing.

We were not able to find any critical deficiencies in the modelling capabilities of the Designer. However there were numerous smaller shortcomings for example in respect to the conformance to messaging defined in BPMN standard. The practical modelling capabilities were also hindered by the non-responsiveness and occasional lock-ups of the modelling tool. Usability of the modeller has also some minor issues. However, in overall the usability of the modeller is on a good level.

The cursory study of generated BPEL revealed no evidence of non-conformance to standards, but certain issues with references can be found in our example model. It is worth noting that WS-BPEL 2.0 is still in draft stage so full conformance analysis would not be feasible. One can also assume that no major long term conformance issues will arise being that Intalio is represented in the editor staff of the standard.

In overall, when taking into account the examined capabilities of the Intalio toolset and the expected low starting costs, it is estimated that the gained value from the product is on a good level. When all the findings of this study are summed, it can be said that Intalio could be a suitable starting point for a small scale enterprise looking for BPM modeller.

The scope of this study was limited only to certain most essential aspects available in Intalio tools. Further research especially in the field of execution of real world business processes is imperative to gain more conclusive answers to complete this cursory analysis.

This study could be expanded by analysing various different BPM tools and subject them to a generic modelling problem rather than analysing what features specific tool offers in isolation. The performance and features of each tool could then be evaluated more objectively and more comparable analysis could be presented.

## References

- [1] Wolf C., Harmon P. The State of Business Process Management. BPTrends, June 2006.
- [2] YengSeok L., JuhHyun B., Seokkoo S. Development of Quality Evaluation Metrics for BPM System. Fourth Annual ACIS International Conference on Computer and Information Science, 2005.
- [3] RosettaNet Standards, PIP 3A1: Request Quote, Online. <http://rosettanel.org/pip3a1>, Referenced 19.10.2006
- [4] Object Management Group (OMG), Business Process Modeling Notation Specification 1.0, Final adopted specification, 01.02.2006, Online. <http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf>, Referenced 19.10.2006
- [5] OASIS web services, Business Process Execution Language Version 2.0 Public Review Draft, 23rd August, 2006, Online. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v20-rddl.html>, Referenced 10.01.2007
- [6] Intalio, Product List, Online. <http://www.intalio.com/products>, Referenced 11.11.2006
- [7] Ouyang C., Dumas M., van der Aalst W.M.P, ter Hofstede A.H.M. From Business Process Models to Process-oriented Software Systems: The BPMN to BPEL Way. BPM Center Report BPM-06-27, 2006.
- [8] Ouyang C., van der Aalst W.M.P., Dumas M., ter Hofstede A.H.M. Translating BPMN to BPEL. BPM Center Report BPM-06-02, 2006.
- [9] W3C Recommendation, Extensible Markup Language (XML) 1.0 (Fourth Edition), August 16 2006 Online <http://www.w3.org/TR/xml>, Referenced 19.10.2006
- [10] W3C, XML Schema Part 2: Datatypes Second Edition, October 28 2004 Online <http://www.w3.org/TR/xmlschema-2>, Referenced 01.11.2006
- [11] Stylus Studio, DataDirect, Online. <http://www.stylusstudio.com>, Referenced 01.11.2006
- [12] W3C Note, Web services Description Language (WSDL) 1.1, Online. <http://www.w3.org/TR/wsdl>, Referenced 19.10.2006
- [13] Intalio, PXE Confluence Wiki, Online. <http://pxe.intalio.org>, Referenced 10.11.2006
- [14] Ghalimi, I., Donation to Apache, Online. <http://itredux.com/blog/2006/02/15/donation-to-apache>, Referenced 10.11.2006
- [15] Object Management Group (OMG), Business Process Modeling Notation Specification 1.0, Final adopted specification, 01.02.2006, Online. <http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf>,
- [16] White, S., Process Modeling Notations and Workflow Patterns. BPTrends, March 2004.
- [17] White, S., Using BPMN to Model a BPEL Process. BPTrends, March 2005.

[18] Srikarsemsira W., Roongruangsuwan S., Comparative analysis of Business Process Diagram Conventional Forms and Vendor specific standards, International Journal of the Computer, the Internet and Management. Vol. 13, No SP3. November 2005

[19] Intalio, Community forums thread “BPEL Standards and Intalia” 7.12.2006, Online. [http://bpms.intalio.com/component/option,com\\_joomlboard/Itemid,47/func,view/id,3788/catid,8/](http://bpms.intalio.com/component/option,com_joomlboard/Itemid,47/func,view/id,3788/catid,8/), Referenced 10.01.2007

[20] OASIS Web services, Business Process Execution Language Version 2.0 Public Review Draft, 23rd August, 2006, Online. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v20-rddl.html>, Referenced 10.01.2007

[21] Brickley D., Guha R.V., RDF Vocabulary Description Language 1.0: RDF Schema, Online. <http://www.w3.org/TR/rdf-schema/>, Referenced 10.01.2007

[22] Eclipse Foundation, Eclipse Website, Online. <http://www.eclipse.org>, Referenced: 10.11.2006

[23] Ghalimi, I, Why Eclipse?, Online. <http://www.intalio.com/news/bpm-20-blog-post/?post=2006/03/27,->, Referenced 19.10.2006

[24] Intalio, Reasons to try, Online. <http://www.intalio.com>, Referenced 11.11.2006

[25] Cooper D., DiBona C., Stone, M, Open sources 2.0. O'Reilly, October 2005. 488p. ISBN: 0-596-00802-3

[26] Intalio, Pricing section, Online. <http://bpms.intalio.com/content/view/138/98>, Referenced 30.11.2006

[27] Intalio. BPMS Community edition Licence agreement, 2006, Version 4.3.1 Referenced 11.11.2006

[28] Intalio, Help section, Online. <http://bpms.intalio.com/content/section/4/87>, Referenced: 06.11.2006

[29] Intalio, Leader in Open Source BPMS, Online. <http://www.intalio.com/>, Referenced 17.11.2006

## Appendix I

The BPEL message generated with Intalio Designer from a BPMN model.

```
<?xml version="1.0" encoding="UTF-8"?>
<bpel:process
  xmlns:bpel="http://schemas.xmlsoap.org/ws/2004/03/business-process/"
  xmlns:wsdlns4="http://www.example.com/PIP_3A1_RequestQuote/Buyer_And_Seller_PLT"
  xmlns:tns="http://www.intalio.com/"
  xmlns:ode="http://www.apache.org/ode/type/extension"
  xmlns:Seller1="http://www.example.com/PIP_3A1_RequestQuote/Seller_To_Buyer"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:wsdlns0="http://www.example.com/PIP_3A1_RequestQuote/Seller_To_Buyer"
  xmlns:Seller="http://www.example.com/PIP_3A1_RequestQuote/Buyer_To_Seller"
  xmlns:wsdlns3="http://www.example.com/PIP_3A1_RequestQuote/Buyer_And_Start--004_PLT"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:wsdlns1="http://www.example.com/PIP_3A1_RequestQuote/Buyer_To_Seller"
  xmlns:bpmn="http://bpms.intalio.com/designer/bpmn"
  xmlns:Start="http://www.example.com/PIP_3A1_RequestQuote/Start--004_To_Buyer"
  xmlns:wsdlns2="http://www.example.com/PIP_3A1_RequestQuote/Start--004_To_Buyer"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:this="http://www.example.com/PIP_3A1_RequestQuote/Buyer"
  bpmn:id="ID-216441c4-c127-1004-8c91-58328da8f429" name="Buyer"
  targetNamespace="http://www.example.com/PIP_3A1_RequestQuote/Buyer"
  queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0"
  expressionLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0"
  suppressJoinFailure="yes" abstractProcess="no" exitOnStandardFault="no">
  <rdfs:label xml:space="preserve">Buyer</rdfs:label>
  <bpel:import
    namespace="http://www.example.com/PIP_3A1_RequestQuote/Buyer"
    location="PIP_3A1_RequestQuote-Buyer.wsdl" importType="http://schemas.xmlsoap.org/wsdl/">
  <bpel:partnerLinks>
    <bpel:partnerLink name="Buyer_And_Seller_PLink"
      partnerLinkType="wsdlns4:Buyer_And_Seller_PLT"
      myRole="Buyer_Role" partnerRole="Seller_Role" initializePartnerRole="yes"/>
    <bpel:partnerLink name="Buyer_And_Start--004_PLink"
      partnerLinkType="wsdlns3:Buyer_And_Start--004_PLT"
      myRole="Buyer_Role" initializePartnerRole="no"/>
  </bpel:partnerLinks>
  <bpel:variables>
    <bpel:variable name="quoteRequest" messageType="Seller:quoteRequest"/>
    <bpel:variable name="quoteConfirmation" messageType="Seller1:quoteConfirmation"/>
    <bpel:variable name="startMessage" messageType="Start:startMessage"/>
  </bpel:variables>
```

```
<bpel:sequence name="Buyer-Sequence">
  <bpel:receive bpmn:id="ID-2a04fdb4-c127-1004-8ed3-6d9599fe9bb8"
    name="Name_Not_Specified-Receive"
    partnerLink="Buyer_And_Start--004_PLink"
    portType="Start:Start--004_To_Buyer_PortType"
    operation="Name_Not_Specified" variable="startMessage" createInstance="yes"/>
  <bpel:invoke bpmn:id="ID-216441ca-c127-1004-8c91-58328da8f429"
    name="Send_Quote_Request-Invoke"
    partnerLink="Buyer_And_Seller_PLink"
    portType="Seller:Buyer_To_Seller_PortType"
    operation="Receive_Quote_Request" inputVariable="quoteRequest">
    <rdfs:label xml:space="preserve">Send Quote Request</rdfs:label>
  </bpel:invoke>
  <bpel:receive bpmn:id="ID-25eeab80-c127-1004-8ed3-6d9599fe9bb8"
    name="Receive_Quote_Confirmation-Receive"
    partnerLink="Buyer_And_Seller_PLink"
    portType="Seller1:Seller_To_Buyer_PortType"
    operation="Receive_Quote_Confirmation"
    variable="quoteConfirmation" createInstance="no">
    <rdfs:label xml:space="preserve">Receive Quote Confirmation</rdfs:label>
  </bpel:receive>
  <bpel:empty bpmn:id="ID-2be53090-c127-1004-8ed3-6d9599fe9bb8" name="Name_Not_Specified--005"/>
</bpel:sequence>
</bpel:process>
```