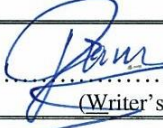




University of  
Stavanger

Faculty of Science and Technology

## MASTER'S THESIS

Study program/ Specialization:  Computer Science	Spring semester, 2012  <u>Open</u> / Restricted access
Writer: Dmitrijus Pantiuchovas	 ..... (Writer's signature)
Faculty supervisor: Chunming Rong External supervisor(s): Jochen Müller	
Titel of thesis:  <b>Extending Enterprise Architecture models for Cloud Computing</b>	
Credits (ECTS): 120	
Key words: BPMN 2.0 standard, Ontology, Protege, jBPM, Cloud Computing service providers, Enterprise architecture models, Jess rule engine	Pages: 73  + enclosure: CD  Stavanger, 15-06-2012 Date/year

# Extending Enterprise Architecture models for Cloud Computing

## Master thesis

Dmitrijus Pantiuchovas

Department of Computer Science, University of Stavanger  
Stavanger, Norway  
d.pantuchovas@stud.uis.no

Faculty Supervisor: Jochen Müller, Bouvet, jochen.mueller@bouvet.com  
External Supervisor: Chunming Rong, UIS, chunming.rong@uis.no

**Abstract:** The new wave of technology changes has introduced cloud computing. For an enterprise this innovation can bring great cost savings as well as risks. Therefore a special analysis process shall be done before the decision is made. This work focuses on Enterprise Architecture business process models as the main source of information about applications and data used by the company. We extend Business Process Modelling and Notation (BPMN) standard to include Cloud computing related information. In addition, we create software, which is able of processing BPMN models and defining whether element needs to be stored inside the company or placed in a private/public cloud. User can also define the preferred level of security. As an output the user is provided with modified BPMN model and a list of recommended service providers.

## Table of contents

Table of contents .....	3
1 Introduction .....	6
1.1 Motivation .....	6
2 Enterprise Architecture .....	6
2.1 EA roots and Zachman framework.....	6
2.1.1 Reasons to build Enterprise Architecture .....	8
2.1.2 Zachman framework evaluation.....	9
2.2 The Open Group Architecture Framework .....	9
2.2.1 TOGAF framework evaluation .....	10
2.3 Gartner Description of EA.....	11
2.4 EA model standard .....	11
2.5 Master thesis EA scope and description .....	12
3 Cloud Computing.....	13
3.1 Brief history of Cloud computing .....	13
3.2 NIST definition of Cloud .....	13
3.3 Private, Public, Hybrid and Community Clouds. ....	13
3.4 Service layers in Cloud computing [3] .....	14
3.5 Service Level agreements.....	14
3.6 Advantages and Disadvantages of Cloud [4],[5].....	15
4 Problem description.....	16
4.1 Challenges implementing the cloud. ....	16
4.2 Relation between Cloud computing and EA. ....	17
5 Analysis of enterprise data. ....	18
5.1 Enterprise data analysis solutions. ....	18
5.2 Ontology and Enterprise Architecture.....	19
5.3 Proposed concept .....	20
6 BPMN standard analysis .....	21
6.1 Process modeling conformance.....	21
6.1.1 BPMN Process Types .....	21
6.1.2 BPMN Process Elements .....	22
6.1.3 Visual Appearance .....	22
6.1.4 Interchange format .....	22
6.2 Process execution conformance.....	23
6.2.1 Operational semantics .....	23
6.2.2 Activity lifecycle .....	23
6.3 Choreography Modeling Conformance .....	23
6.3.1 BPMN Choreography Types.....	23
6.3.2 BPMN Choreography Elements.....	24
6.4 BPMN Process Elements .....	25
6.4.1 Sequence flow connection rules .....	28
6.4.2 BPMN extensibility.....	29

6.5	BPMN Infrastructure package.....	29
6.5.1	Definition attribute table.....	30
6.5.2	Imports.....	30
6.5.3	XML schemas .....	30
6.6	BPMN Foundation package .....	31
6.7	BPMN Common elements package.....	31
6.8	BPMN Services package.....	32
6.9	Collaboration package. ....	32
6.10	Process diagrams .....	33
6.10.1	Activity element .....	34
6.10.2	Task.....	34
6.10.3	Sub-process element .....	35
6.10.4	Transaction .....	36
6.10.5	Ad-hoc sub-process .....	36
6.10.6	Call activity.....	36
6.10.7	Global Task.....	37
6.11	Items and Data .....	37
6.11.1	Data stores .....	38
6.11.2	Xpath expressions for data .....	38
6.12	Event Element.....	38
6.13	Lessons learned from extending BPMN standard.....	39
7	Description of BPMN Extension for Cloud Computing.....	39
7.1	Related work.....	39
7.2	Extension requirements .....	40
7.3	Activity element extension.....	41
7.3.1	Extension Class Diagram .....	41
7.3.2	Extension attributes and model associations .....	44
7.3.3	Extension XML Schema.....	44
7.4	Extension for Items and Data elements .....	44
7.4.1	Extension Class Diagram .....	44
7.4.2	Extension attributes and model associations .....	47
7.4.3	Extension XML Schema.....	48
8	Proof of concept .....	48
8.1	BPMN Extension Implementation .....	48
8.2	Extension import and usage.....	50
8.3	Creating the initial BPMN model .....	51
8.4	Graphical user interface.....	51
8.5	Exporting the extended BPMN model.....	52
8.6	Ontology implementation .....	53
8.7	Analyzing framework .....	55
8.8	Analysis of the ontology.....	55
8.9	Exporting the modified model.....	58
9	Evaluation.....	61
9.1	Testing .....	61
9.2	Evaluation of the Proof of concept .....	63

9.3	Feasibility study.....	64
9.3.1	Technical feasibility .....	64
9.3.2	Economic feasibility .....	65
9.3.3	Legal feasibility.....	66
9.3.4	Operational feasibility .....	66
9.3.5	Schedule feasibility.....	67
10	Conclusions.....	67
11	Future work.....	67
	APPENDIX.....	71

## **1 Introduction**

### **1.1 Motivation**

Constantly changing business environment makes enterprises to become very flexible and agile. The bigger enterprise the more difficult for it is to adapt to new requirements. Now there is a new wave of market changes coming with the introduction of Cloud computing. Computing as a service brings not only unlimited scalability and cost reduction but also challenges related to implementing the Cloud in the enterprise. Privacy and sensitive data safety issues need to be solved in advance. In order to succeed, the enterprise has to efficiently and cost effectively overcome mentioned challenges. In this paper we extend enterprise architecture models in order to visualize whether the element has to be stored internally or can be placed into private/public cloud.

## **2 Enterprise Architecture**

Many enterprise architecture (EA) modeling frameworks exist. They have different scope and characteristics but their mission is the same – to represent business structure in order to maintain enterprise evolution. In this paper we will focus on two frameworks, which represent the essence of the EA.

### **2.1 EA roots and Zachman framework**

It all began with Dewey Walker - IBM's Director of Architecture in the late 1960s. He is called the grandfather of architecture methodologies. He produced architecture planning documents that later became known as Business Systems Planning. Later on during the mid 1980s, the employee of the IBM John Zachman, contributed to the evolution of BSP. He identified the need of using logical construction architectures to define and control integration of the system components.

Zachman framework created in 1987 presents a table with grouped architectural artifacts, such as models, specifications and documents. Each artifact position is defined by a particular issue and user role. User roles are the following:

1. Planner – people who invest into enterprise. Conceptual model is needed. Main area of interest is the scope of the system: overview and estimate of the system, costs, relation to the general environment.

2. Owner – person who runs the company. This role needs conceptual business models, which represent processes and entities with their relationships.
3. Designer – needs logical model of the enterprise, which specifies data elements, logical process flows, functions in the business model.
4. Builder – needs physical model with the main focus on technology, I/O devices, and tools.
5. Subcontractor – needs narrowed out-of-context model in order to create / upgrade and implement a particular module into existing system.
6. Enterprise – the functioning enterprise model.

Issues are:

1. What – data description, which is different for every role;
2. How – function description;
3. Where – network description;
4. Who – description of people involved;
5. When – time description;
6. Why – description of the motivation behind.

Every role (rows) has its own understanding of the appropriate issue (columns). *Figure 1* below provides the representation of the Zachman framework<sup>1</sup>. Each of 36 intersection cells represents artifacts.

---

<sup>1</sup> <http://www.thinkingprocess.co.uk/images/zachmanframework.png>

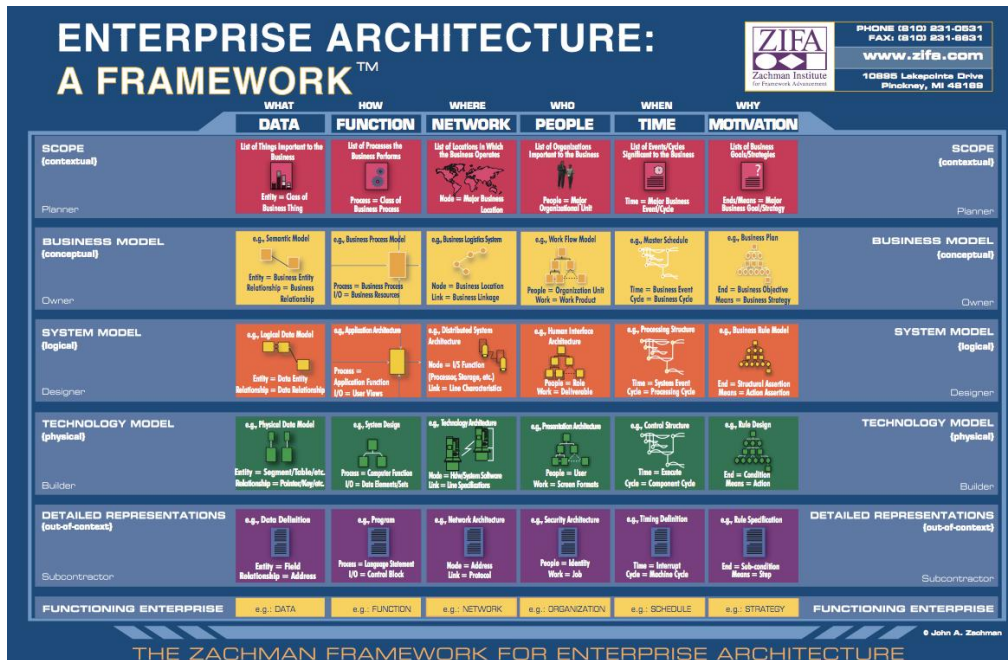


Figure 1 Zachman Framework

### 2.1.1 Reasons to build Enterprise Architecture<sup>2</sup>

According to Zachman framework there are several reasons to build Enterprise Architecture for a company:

Alignment – different roles aim the same goal and implement solutions decided by owner.

Integration – information availability for different roles eliminates inner communication impediments and creates area for further improvement. Integration enables different units of the company create, modify and share standardized data.

Change – changing the company structure in order to gain more flexibility. Agility brought by enterprise architecture guarantees fast, low risk, cheap adoption to a frequently changing market.

Reducing “time to market” – increasing speed of production due to integration of the processes and data standards. With a help of EA the whole process of production takes less time, all information in standardized format is available for different units.

<sup>2</sup> Zachman, J.A., *You can't "Cost-Justify" Architecture* 2001



### **2.1.2 Zachman framework evaluation**

Advantages of the Zachman framework are:

1. Each role is aware of all issues and therefore has understanding of the whole process flow.
2. Each artifact is well defined for the appropriate role.
3. Each issue can be traced down from requirements to the technical implementation. For example: Data requirements from owner can be traced down to the implementation of Designer.

Nevertheless, Zachman framework is limited to representation of the existing architecture and can not provide information about:

1. Need of creating a new architecture;
2. Process for creating a new architecture;
3. Way to evaluate the new architecture.

## **2.2 The Open Group Architecture Framework**

EA Framework evolution led to creation of government structure oriented architecture and standardization techniques: Technical Architecture Framework for Information Management (TAFIM), Federal Enterprise Architecture Framework (FEAF). Further EA frameworks evolution and innovation grow made TAFIM to be adopted by public sector. It was passed to Open Group management and renamed into TOGAF.

The Open Group Architecture Framework (TOGAF) comes not only with a framework itself but also with Architecture Development Method (ADM) and, in a way, complements Zachman provided model. TOGAF divides Enterprise architecture into 4 sections:

- Business architecture – description of the main business processes;
- Application architecture – description of applications design and interaction;
- Data architecture –database organization and access;
- Technical architecture –hardware and software supporting applications description.

TOGAF represented as an Enterprise Continuum – architecture layered approach where each layer closer to the circle centre is more specific then preceding layer. Ability to move between layers is provided by ADM.

ADM phase cycle is provided on *Figure 2* and described below.

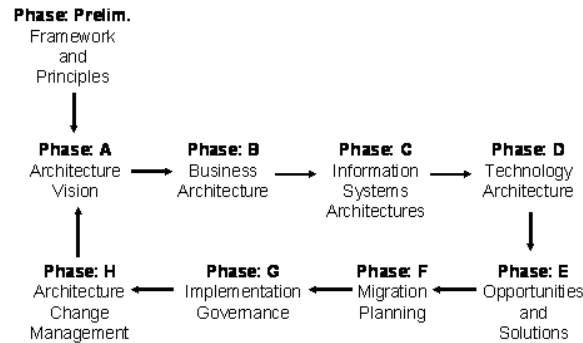


Figure 2. TOGAF process diagram

- **Preliminary Phase:** modifying TOGAF process to meet business specifics;
- **Phase A:** Architecture vision of how to pass through the ADM cycle. Setting scope, requirements, high level definitions for existing and future architectures;
- **Phase B:** Existing business architecture analysis, looking for gaps and drawbacks;
- **Phase C:** Existing information system analysis, creation of application architecture;
- **Phase D:** Deciding on infrastructure needed for the new architecture;
- **Phase E:** Different implementations evaluation, looking for the best operation;
- **Phase F:** Implementations analysis and sorting according to their risk, price and benefits;
- **Phase G:** Architectural specifications for prioritized implementations;
- **Phase H:** Update of the existing management process with the new artifacts.

### 2.2.1 TOGAF framework evaluation

In addition to Zachman artefact concept TOGAF framework provides an accurate method of improving the existing architecture. It is also widely available for different size companies in the private sector.

Nevertheless TOGAF does not guarantee that the new architecture is better than the old one, as long as there is no comparison. The quality depends exclusively on experience of involved parties.

### 2.3 Gartner Description of EA

Gartner Group is one of the leading IT research and advisory companies. High IT field research achievements made them to state their own EA definition:

*“Enterprise architecture is the process of translating business vision and strategy into effective enterprise change by creating, communicating and improving the key requirements, principles and models that describe the enterprise's future state and enable its evolution. The scope of the enterprise architecture includes the people, processes, information and technology of the enterprise, and their relationships to one another and to the external environment. Enterprise architects compose holistic solutions that address the business challenges of the enterprise and support the governance needed to implement them”<sup>3</sup>.*

In other words EA is always ongoing and non stopping process that converts business strategy and its main guiding principles into real, measurable change. This is achieved with the help of enterprise future state description, which clarifies how existing requirements and models need to be changed. EA is much more than IT process. It covers people, processes, data and technology, including their inner relationships and connections with outer environment. Solutions provided by EA affect not only separate units, but the whole company.

### 2.4 EA model standard

All mentioned frameworks store architecture in various artefacts. Depending on data type the appropriate method is used, this could be UML diagram, flow chart diagram, Microsoft Excel table, different specifications, etc. When it comes to process architecture, business process model and notation (BPMN) standard[1] is used. It is widely accepted object management group (OMG) standard for EA visual process models representation. BPMN represented by flow-chart and comes as an output of many EA modeling tools. BPMN example on *Figure 3*, shows a chronological process flow of an

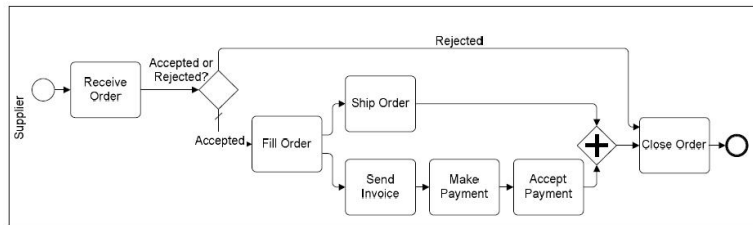


Figure 3. An example of BPMN (adopted from [www.bpmn.org](http://www.bpmn.org))

<sup>3</sup> A. Lapkin, P. Allega, B. Burke, *Gartner Clarifies the Definition of the Term Enterprise Architecture* (Gartner ID #G00156559). 12-08-2008.

order. Due to BPMN open source availability and popularity I will be using it as an Enterprise Architecture model representation.

## 2.5 Master thesis EA scope and description.

In order to be able to better analyze and provide valuable solutions the scope of interest and description need to be narrowed.

EA covers whole company structure, its domains described below and on *Figure 4*<sup>4</sup>:

1. Business process architecture: tasks composing critical business processes
2. Information architecture: shared data standards
3. Application architecture: applications in use and their interfaces
4. Technology architecture: infrastructure services and standards they are build on

### Enterprise Architecture Domains

**Sub-Domains:** EA Domains are further divided into sub-domains depending on the elements within a domain.

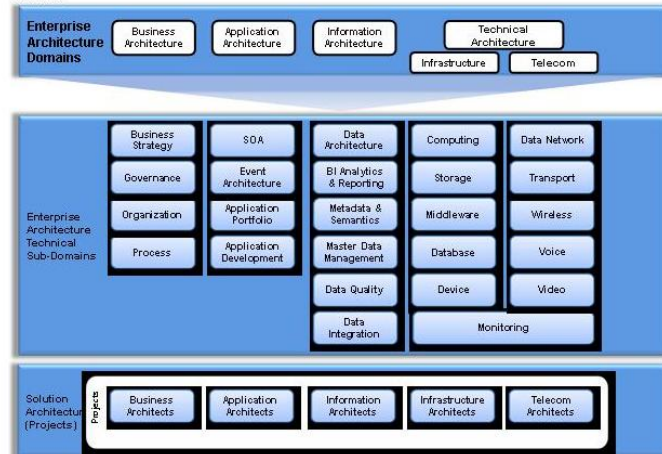


Figure 4. Enterprise Architecture domains

There are too many domains to be included in this paper therefore we are narrowing the scope of our interest. This paper will take Process sub domain from Business architecture and will analyze business process models in order to improve them.

<sup>4</sup> Adopted from: [http://en.wikipedia.org/wiki/Enterprise\\_architecture](http://en.wikipedia.org/wiki/Enterprise_architecture), 2011-10-04

## 3 Cloud Computing

### 3.1 Brief history of Cloud computing

Origin of cloud computing term came from Salesforce.com, which in 1999 provided a way of delivering enterprise applications via a simple website. Amazon was the next one, who expanded cloud capabilities by launching Amazon Web Service in 2002. Later in 2006 Amazon's Elastic Compute cloud (EC2) was introduced. This web service allowed small companies and private sector to rent computers with high computational power to run their own computer applications. The first open source AWS API compatible platform Eucalyptus was introduced in 2008. It allowed deployment of private clouds and was followed by OpenNebula, the first open source software for deploying private and hybrid clouds.

### 3.2 NIST definition of Cloud

According to National Institute of standards and technology (NIST) Cloud computing description [2] provided on September 2011 there are several main characteristics:

**On-demand self service** – a client can have as much resources (storage, servers) as needed automatically upon request without human interaction.

**Broad network access** – all services provided via network and accessible through various type of devices (laptops, mobile phones, servers, etc.)

**Resource pooling** – providers store their resource in a broadly located pool. Client has no particular influence or information about the location except choosing the location for the higher level of abstraction (country, datacenter).

**Rapid elasticity** – unlimited provisioning capabilities provided at any time and quantity upon the demand of the client.

**Measured service** – special pay-per-use measurement system is introduced, preventing waste of money for resources that are not required (provided by internal IT datacenter).

### 3.3 Private, Public, Hybrid and Community Clouds.

There are 4 types of Cloud computing:

1. Public cloud is the one of the most commonly used cloud computing types. It assumes a client to store the application/data on the 3<sup>rd</sup> party servers using their resources. Using public cloud allows to reduce the size of internal infrastructure and therefore save money.

2. Private cloud assumes proprietary computing architecture built inside the corporation. Services within the company can be provided. This is useful when several company offices in different geographical locations need to access the same data.
3. Community cloud unites several organizations which have common interests (security, policy, mission) to have a common Cloud computing system. E.g. Oil and Gas community cloud can be created by several petroleum companies in order to maintain a database for not confidential information.
4. Hybrid cloud assumes some features of an application or entire application to be run in public cloud, while other, more sensitive applications will be run only in private or community cloud.

In this paper we focus on 3 possibilities of storing an application of data. Public/Private clouds or store it internally. Other alternatives can be implemented if needed by simply adding their definitions into the concept.

### **3.4 Service layers in Cloud computing [3]**

There are three typical service layers offered to Enterprises:

1. Infrastructure as a Service (IaaS) – Includes virtualized and shared network, servers, storage with ability of dynamically provision
2. Platform as a Service (PaaS) – Databases, runtimes, middleware, development tooling – basis for application deployment
3. Software as a Service (SaaS) – Industry apps, business processes, ERP, HR, collaboration services and applications stored and accessed in cloud by multiple users.

In this paper we focus on PaaS as a way of storing enterprise applications and data.

### **3.5 Service Level agreements**

Service-level agreements define policies between third party and enterprise dynamically. Each service provider guarantees a level of service. Usually it is expressed in percents of active time over downtime. The higher service level provided the more expensive service can be. So there is a tradeoff in choosing a cloud computing service provider between the price and availability of service, which needs to be assumed when introducing the cloud.

### 3.6 Advantages and Disadvantages of Cloud [4],[5]

At the time of writing this paper there were following advantages and disadvantages of cloud computing.[6]

Advantages:

1. Elastic Scalability. Cloud environments allow businesses to serve larger audiences; gain incremental resources on demand, minimize risk of project failure when starting from small and expanding after.
2. Agility. Efficient shared resource. As long as many applications can be stored allowing good operational level with all users and automated management. Faster reaction to market is given.
3. Reliability and Fault-Tolerance. Built-in redundancy of the high numbers of servers, enables high levels of availability and reliability for applications. In business context this brings more reliable access in the face of possible infrastructure failures.
4. Shared Multi-tenancy. Common infrastructure is shared by multiple users despite their location. Cost reduction, improved efficiency, security of user's data.
5. Cost efficient. Utility-based. Cost efficiency is brought by payment for cloud resources or subscription. SLA-driven.
6. APIs. Application programming interface provided to meet unforeseen business needs.

Nevertheless by the time of writing this paper the following Cloud disadvantages existed.

1. Security. One of the biggest issues as storing business-critical data on 3<sup>rd</sup> party servers raises enormous risk of information to be compromised.
2. Data location and Privacy. As long as data stored on globally located servers, this brings many issues with different countries privacy and data management laws. Such as data openness for government in US, or duration and content restriction in EU. In addition due to disagreement with the provider he is in possession of the information and appropriate safeguards need to be established on the contract level.
3. Internet dependency, performance and latency. Despite the fact that web acceleration technologies evolve, bandwidth is getting always higher and software is scalable some latency still occurs. Business sensitive to latency (Trading companies) can not accept the innovation, though it is now been improved by IBM and Juniper. What is more the problem with internet connection will lead to the loss of whole IT infrastructure.
4. Availability and service levels. Downtime issue is very important for online business services providing companies, as every minute costs dysfunction of particular service. Affecting reputation of the company.

5. Current Enterprise Applications are difficult to migrate. Complex multitier applications enabling different databases, access rights are very hard to switch-over to the cloud. Conversion of applications sometimes is more costly than new application creation on the cloud. Though it is more and more vendors that create application with the possibility to be implemented on cloud.

## 4 Problem description

Numerous papers were written addressing known issues and solutions, such as Cloud-Net platform architecture utilizing virtual private networks for secure networking of enterprise sites and cloud [7], homomorphic token with distributed verification of erasure-coded data for secure storage [8], improving data integrity by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication and public verifiability along with dynamic data operations [9], Aneka solution for load balancing of cloud and executing applications in Cloud, requires standard interface and brings ability to utilize different leverages[10]. In this paper we are aiming to take another approach on providing data safety and privacy based on EA business process models.

### 4.1 Challenges implementing the cloud.

Based on Jerico Forum paper [11] in order to adopt cloud a series of questions should be answered:

1. What is sensitivity level of data.
2. Regulatory restrictions according to the type of information and the place (country) of storage.
3. What layer of Cloud is needed (Infrastructure, Platform, Software, Process) depending on your business specific?

Answering these questions might make creating standards for data, trust levels, metadata attached to each data packet.

Another source [12] claims to be able to migrate to cloud company aside from analyzing privacy and security aspects should go through 3 phases:

1. Make a roadmap of replicating available applications and infrastructure into cloud. Progressive transition to cloud.
2. Design explicitly policy architectures helping maintain agility and innovation.
3. Implement new architectures.

Designing new architecture is supposed to be a very important as long as a company has a chance to re-architect itself into next generation enterprise.



Intel [13] planning implement private cloud in 3 phases estimated to take 3-plus years. Each year they gradually upgrade the following areas: Business transformation, Computing, Storage, Network, Security, Management, Data Centers and Clients. A tool able of improving the speed and quality of migration process is highly needed and valued.

#### 4.2 Relation between Cloud computing and EA.

Enterprise Architecture and Cloud computing need to be compared in order to find their relation and a solution of how EA models can assist implementing the Cloud Computing. With a help of *Table 1* it is possible to see how EA relates to Cloud computing.

**Table 1 Cloud computing and EA comparison**

NIST	Appisty, Kinetix, Communications of the ACM	Ongoing process	People/proc./data/technology oriented	Whole system solutions	Future state description	Gartner
		Change	Alignment	Reduce production cycle time Integration of common data	Alignment	Zachman
		Business architecture	Data architecture	Applications/technical architecture	Business architecture	TOGAF
On-demand self service/ Rapid elasticity / Resource pooling	Scalability/Agility	X	X	X	X	
Broad Network access	Share multi-tenancy		X			
	Reliability		X	X		
Measured service	Cost efficiency	X		X		

	Security / privacy risk		<b>X</b>	<b>X</b>		
--	----------------------------	--	----------	----------	--	--

EA describes an **ongoing process** and computational requirements change as well. Cloud computing provides on demand resource scalability. So more resources are used when needed and abandoned when there is no need in them. This can be efficiently cost evaluated avoiding unnecessary expenses.

EA **orientation on process, people, data and technology** is related to scalable on demand resources depending on actual needs, shared multi-tenancy providing different unit people access the system through different devices and reliability of replicated data preventing it from being lost. On the other security and privacy issues are crucial for enterprise data management and though special attention needs to be paid.

**Whole system solution** characteristic is related to Cloud computing provided scalable service, reliability of replicated data and cost efficiency in maintaining. Security risks also need to be considered because it affects the whole system.

**Future state description characteristic** is related to cloud computing service scalability, which guarantees the required amount of resources needed.

As we can see EA and Cloud computing are closely related. What is more EA models can be enriched with related cloud computing information, such as pricing, security level, availability of service, etc. This additional expressiveness will help in making decisions and avoiding risks.

## 5 Analysis of enterprise data.

### 5.1 Enterprise data analysis solutions.

According to challenges stated in chapter 4 there is a serious barrier for enterprise when migrating into the cloud. EA needs to be changed accordingly assuming all the threats and risks brought by cloud computing. In other words, data sensitivity, privacy and security have to be analyzed before deciding the type of storage (internal storage or private/public cloud) that will be used for a particular service.

Further we discuss several of the existing techniques used to find a data in the enterprise infrastructure.

Specific knowledge ontology based Framework for knowledge discovery [14] Data discovery algorithm use EA database according to Ontology specified data required. Then the rule engine is used for defining similarity of the data stored in databases.

Another approach [15] utilizes references architectures of the enterprise, integration and modeling tools to collect the data and resent knowledge views. Further it references to other tools which deal with knowledge views. Proposed framework uses Object Management Groups (OMG) Model-Driven Architecture (MDA), Common Warehouse Meta-Model (CWM) and was partially implemented in Java, Cobra.

Ontology-based knowledge management system [16] uses mapping of multiple ontologies along with their enhancement process, such as automatic update using metadata from Office applications. Karlsruhe Ontology and Semantic web framework (KAON<sup>5</sup>), which allows querying and maintaining data was used for ontology management.

Kensington data mining framework [17] was built in order to cope with distributed data storage. Utilizing Enterprise JavaBeans (EJB) for secure connection and authentication as well as mining itself. Java and Cobra programmed the framework allows to retrieve data and apply classification, clustering and association rule discovery analytical methods.

As seen from these examples it is difficult to locate information needed. Enterprise architecture models provide a systematic way of displaying elements of the company and therefore can assist with data search.

## 5.2 Ontology and Enterprise Architecture

As seen in previous chapter, most of the projects use ontology as a way of systematic storage and data querying. Visually presented business process models are highly valued by individuals, because of easy readability, however when it comes to interpretation or analysis of large amount of models, additional assistance is needed. Data ontology is one of approaches to systematically group data and query it for required relations. An ontology based enterprise architecture project [18] uses 3 ontologies: business term, EA component, relationships among EAs. This way allows common EA understanding as every user can represent it from his point of view. In addition, inner and outer integration between several business entities is easy. Another approach is Shell company provided 4dimensional ontology for EA concept [19]. Assuming 3D ontology as a physical object description in present, they add time dimension and shift the view from present to future and past. So in a way they replace a “snapshot” of a model with a continuous representation.

In this project we aim to use EA models as a source of information and utilize ontology for additional systematized storage of model elements and data relation querying.

---

<sup>5</sup> <http://kaon.semanticweb.org>

### 5.3 Proposed concept

By the moment of writing this paper there was no accessible application able of modeling the cloud computing in the context of Enterprise Architecture. By combining EA models and BPMN extension we are able to show which elements of the model are located in cloud and which are stored locally. This gives the end user much more expressivity and assistance in decision making while migrating the enterprise infrastructure into cloud.

Graphical representation of the concept is on the *Figure 5*.

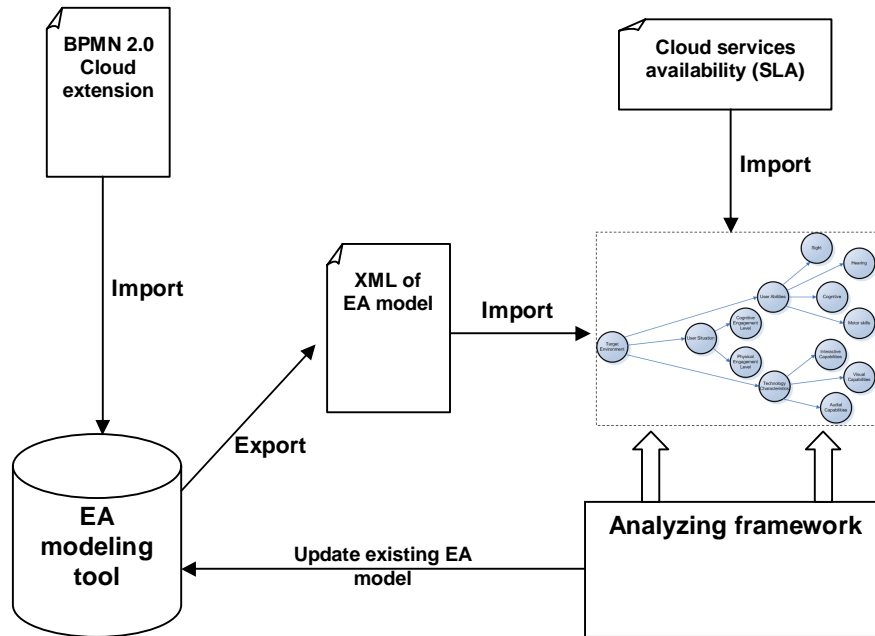


Figure 5. Proposed concept model.

The implementation of the concept consists of several steps:

1. Create BPMN Cloud extension, which will extend current EA model and add more attributes to Data and Activity elements.
2. Create the initial BPMN model with the use of modelling tool.
3. Export enriched with cloud attributes BPMN model to separate file.
4. Import the model into ontology.
5. Analyse and redefine the model taking into consideration:

1. user selected security level defined as a percent;
2. model elements relations to each other;
3. model data elements sensitivity;
6. The BPMN model shall be then updated and imported back into modelling tool. It now describes if the elements needs to be stored in private cloud, public cloud or stored locally.

## 6 BPMN standard analysis

In this project we use BPMN standard for business process models and here we will introduce main modeling elements and extension requirements.

Object Management Group (OMG) created Business Process Model and Notation (BPMN) standard in order to assist an organization in managing their processes. With a help of a single BPMN model business analysts are able to modify process scheme, developers to implement changes and users to utilize functionality. In addition, BPMN is XML based, this helps for business process execution languages to interpret and visualize models.

Software or BPMN extension, which complies with all requirements is said to conform to BPMN standard. If the compliance is only partial the software is said to be based on BPMN standard. There are 4 types of conformance:

1. Process modeling conformance (described in chapter 6.1)
2. Process execution conformance (described in chapter 6.2)
3. BPEL process execution conformance (BPMN mapping to WS-BPEL supporting tool which also comply with Process Execution Conformance)
4. Choreography modeling conformance (described in chapter 6.3)

If the software complies with one of the 4 types it is assumed to conform to BPMN.

### 6.1 Process modeling conformance

Process modeling conformance claiming implementation MUST comply with the following requirements.

#### 6.1.1 BPMN Process Types

Implementation must comply with BPMN core elements defined in the Infrastructure, Foundation, Common, and Service packages. All of them are described in chapters 6.5 – 6.8.

Process diagrams must be able to include the Process, Activities, Data, Human Interaction elements. Described in chapter 6.10

Collaboration diagrams including Pools and Message Flow. Described in chapter 6.9.

Conversation diagrams, including Pools, Conversations and Conversation Links Described in chapter 6.9.

### 6.1.2 BPMN Process Elements

The Process Modeling Conformance type set consists of:

- Collaboration and Process diagram elements
  - Task types, embedded Sub-Processes;
  - CallActivity;
  - Gateway types;
  - Event types (Start, Intermediate, End);
  - Lanes;
  - Participants;
  - Data Object (including DataInput and DataOutput);
  - Message;
  - Group;
  - Text Annotation;
  - Sequence Flow (including conditional and default flows);
  - Message Flow;
  - Conversations (limited to grouping Message Flow, and associating correlations);
  - Correlation;
  - Association (including Compensation Association).
- Markers
  - Loop, Multi-Instance, Transaction, Compensation for Tasks and their sub-processes

### 6.1.3 Visual Appearance

In order for BPMN to be widely used the standard graphical elements, shapes, markers and their connections are defined for model visualizing. Diagram elements have strictly defined semantics which must be followed. Semantic element representation can include additional attributes, which are set and modified with a help of implementation methods. Additionally, BPMN Diagram can be extended with user created graphical elements or attributes, with condition that they do not conflict with the existing standard.

### 6.1.4 Interchange format

With a help of the standard a universal model definition format is created so that implementation software should be able to import, modify and export BPMN model to other tools.

## **6.2 Process execution conformance**

The tool claiming for conformance must implement operational semantics and activity life cycle

### **6.2.1 Operational semantics**

1. Semantics of loop activity
2. Semantics of multiple instances (MI) defining if instances shall be created sequentially or in parallel
3. Semantics defining parallel, exclusive, inclusive and complex gateway operation
4. Semantics of event sub-process, which define states and operation of the element.
5. Semantics of compensation, which define states and operation of the element.

### **6.2.2 Activity lifecycle**

1. READY state. When there is enough tokens to start the activity
2. ACTIVE state. When there is enough tokens and data InpuSet is available
3. WITHDRAWN state. When activity is not activated due to race condition (exclusive gateway defines only one activity to be activated)
4. FAILED state. When error happens during the execution
5. TERMINATED state. When interrupting event occurs during the execution
6. COMPLETING state. Additional intermediate state where there is a possibility to insert non-interrupting event handlers before the activity is set to COMPLETED.
7. FAILING or TERMINATING state. When error occurs an intermediate state allows terminating nested activities and releasing data.
8. COMPLETED state. Output data is created, predefined number of tokens is released into further flow.
9. COMPENSATING state. After completed state there can be compensation if needed. Tools claiming for conformance must also support BPMN diagram import.

## **6.3 Choreography Modeling Conformance**

### **6.3.1 BPMN Choreography Types**

Following packages must be supported:

1. The BPMN core elements which required by Process Modeling Conformance
2. Choreography diagrams, define in Choreography packages

3. Collaboration diagrams, which required by Process Modeling Conformance

### 6.3.2 BPMN Choreography Elements

The Choreography Conformance set includes

1. Message;
2. Choreography
  1. Choreography Task;
  2. Global Choreography Task;
  3. Sub-Choreography;
3. Events
  1. Start Events (e.g., None, Timer, Conditional, Signal, and Multiple);
  2. Intermediate Events (None, Timer, Cancel, Conditional, Signal, Multiple, Link,...)
  3. End Events (None, Terminate),
4. Gateways;
5. Pools and Message Flow if Choreography used within Collaboration.

Like in Process Modeling Conformance visual appearance requirements, an implementation shall follow graphical and semantic requirements defined by the standard. Choreography diagram shall be imported, modified and exported by the tool provided methods.

Comparison of all 4 conformance types provided in the *Table 2*. Based on it we can decide which conformance type the extension shall comply with.

**Table 2 Comparing of 4 BPMN conformance types**

Category	Process Modeling Conformance	Process Execution Conformance	BPEL Process Execution Conformance	Choreography Conformance
Visual representation of BPMN Diagram Types	BPMN Process Types	N/A	N/A	BPMN Choreography Types
BPMN Diagram Elements that need to be supported.	BPMN process elements	N/A	N/A	BPMN Choreography elements
Import/Export of diagram types	Yes	Yes	Yes	Yes
Support for Graphical syntax and semantics	Process and Collaboration diagrams	N/A	N/A	Choreography and Collaboration diagrams
Support for Execution Semantics	N/A	Yes for Process diagrams	Yes for Process diagrams	Choreography execution semantics

















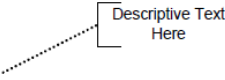
As long as our project aims to utilize BPMN process models it is essential to describe Process Modeling Conformance (omitting execution and choreography conformance) requirements in more details. In addition we will need to create a BPMN Cloud extension which will comply with requirements and provide problem oriented BPMN solution, which can be reused afterwards and extended if needed.

#### 6.4 BPMN Process Elements

In order for implementation for conform it should satisfy modeling elements requirements. Five main elements types are presented in the *Table 3*.




**Table 3 Main BPMN elements' description and visualization**

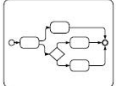


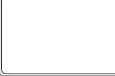


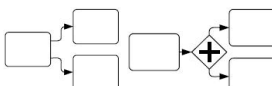
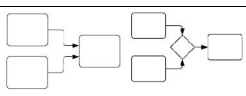
<b>Flow Objects</b>	Events stand for some action which affects the model. They have a trigger or result	
	Activities a BPMN model element, which represents the work process. They can be atomic or non-atomic	
	Gateways used for branching the process flow	
<b>Data</b>	Data Objects provides information of input and/or output of each activity	
	Data Input shows what data is "fed" into the process	
	Data Output shows what data is received after process is executed.	
	Data Store stands for external data storage, where activities can save information and use it in the future.	
<b>Connecting Objects</b>	Sequence flows represents order of process execution in the model	
	Message Flows represents message flow between communicating parties	
	Associations used like comments to attach additional information	

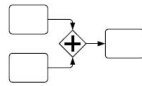
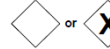








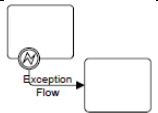
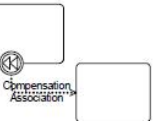
	to elements	
	Data associations used when associated data flow direction needs to be shown	
<b>Swim lanes</b>	Pools help to visually separate activities belonging to different parties. It also used to distinguish participant in Collaboration	
	Lanes used for additional grouping of processes inside one participant area	
<b>Artifacts</b>	Groups are used to display categories, help to visually separate a part of activities not affecting the common model flow.	
	Text acts as comments helping to better understand the elements of the model.	

Additional extension of several main elements, required to be conformed to, is provided in the *Table 4*.

**Table 4 Main elements extension and visualization**

<b>Events</b>	Start event initiates the sequence flow of the model. Started when the trigger condition is met.	
	Intermediate event executes during the flow when the trigger condition is met.	
	End event executed at the end of the process flow with the confirmation of the required conditions.	

	<p>Events can be triggered by different elements: message, time, occurred error, requested escalation to other process, cancel of the process, compensation of the process, fulfilling some conditions, linking to other model diagram part, signal received, termination action request, multiple sequence or parallel events which can start the process. Start and Intermediate events are the one which are catching. End and intermediate events are those which throwing. All events assumed to be interrupting process execution, unless otherwise is stated. Then the event is marked as non-interrupting.</p>	<table border="0"> <thead> <tr> <th></th> <th>"Catching"</th> <th>"Throwing"</th> <th>Non-Interrupting</th> </tr> </thead> <tbody> <tr> <td>Message</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Timer</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Error</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Escalation</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Cancel</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Compensation</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Conditional</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Link</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Signal</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Terminate</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Multiple</td> <td></td> <td></td> <td></td> </tr> <tr> <td>Parallel Multiple</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		"Catching"	"Throwing"	Non-Interrupting	Message				Timer				Error				Escalation				Cancel				Compensation				Conditional				Link				Signal				Terminate				Multiple				Parallel Multiple			
	"Catching"	"Throwing"	Non-Interrupting																																																			
Message																																																						
Timer																																																						
Error																																																						
Escalation																																																						
Cancel																																																						
Compensation																																																						
Conditional																																																						
Link																																																						
Signal																																																						
Terminate																																																						
Multiple																																																						
Parallel Multiple																																																						
<b>Activities</b>	<p>Sub-process activity hides the complexity of the process and can be expanded on request</p>	 <div style="border: 1px solid black; padding: 5px; display: inline-block;">       Sub-Process Name   </div>																																																				
	<p>Looping activity which defines how many time the process or sub-process shall be executed</p>																																																					
	<p>Transaction activity defines external process while committing or canceling the transaction</p>																																																					
	<p>Task atomic activity defining the actual work</p>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">       Task Name     </div>																																																				
	<p>Multiple instances creating a desired number of activity instances and running them either in sequential or parallel.</p>																																																					
<b>Gateways</b>	<p>Branching used to expand the flow from single process to multiple. Different gateway types will define the processes that follow the execution.</p>																																																					
	<p>Forking used for parallel process execution. Example of usage is synchronization of processes.</p>																																																					
	<p>Merging used when several processes result with the same process execution.</p>																																																					

	Joining used when the process execution condition can be met by several other processes combined result.	
	<p>Gateways can be of different types:</p> <ul style="list-style-type: none"> <li>• Exclusive based is the regular one, depending on condition a single particular path is chosen;</li> <li>• Event-based and parallel event based rely on incoming events and provide output tokens either in parallel or sequentially respectively;</li> <li>• Inclusive gateways provide tokens for all events that comply with the conditions, multiple paths are possible;</li> <li>• Complex allows enhanced decision making (<i>m</i> of <i>n</i> tokens needed to launch, time based launching, etc.)</li> <li>• Parallel used for synchronization, if one event is triggered and process is started it is still waiting for other events to trigger before completing the process.</li> </ul>	<p>Exclusive </p> <p>Event-Based </p> <p>Parallel Event-Based </p> <p>Inclusive </p> <p>Complex </p> <p>Parallel </p>
<b>Sequence flows</b>	Uncontrolled	
	Conditional allows condition to be attached to the sequence flow	
	Default executed then no other conditional sequence flow run	
	Exception flow executed when error event occurs, forwards to exception handling process	
<b>Associa-tions</b>	Compensation association occurs when compensation event is met, links for compensation process execution.	

#### 6.4.1 Sequence flow connection rules

Connection rules of the Flow elements allow connecting sub-processes, gateways, intermediate events among each other and with the sequence flow element (black arrow). Other elements have limitations, like end event can have only incoming sequence flow, start event can have only outgoing.

Message flow connection is allowed for pools, activities, sub-processes among each other and for catching intermediate message. Throwing intermediate message and Throwing end message can be connected with all above mentioned elements. Throwing end message can not have incoming message flows and start catching message can have neither of the message flows.

## 6.4.2 BPMN extensibility

Extensibility is allowed for elements' attributes, introducing new elements or artifacts if they do not contradict to BPMN element semantics. In addition, the core elements like events, activities and gateways shall be changed minimally. So that intuitive understanding of the model would remain as with the regular BPMN standard visualization.

## 6.5 BPMN Infrastructure package

Each BPMN element is described with Definition class diagram, attribute table, Import attribute table and XML schema.

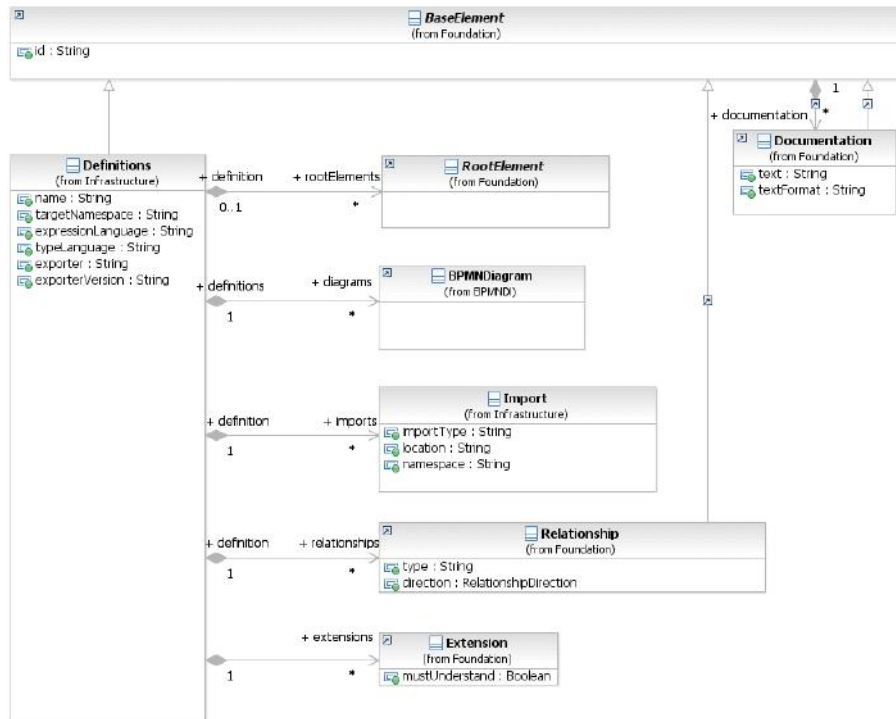


Figure 6. Definition class Diagram

Definition Class Diagram *Figure 6* displays the class relation for the exact element. Using it we can visually define class dependencies and easily extend the existing model.

### 6.5.1 Definition attribute table

Definition attribute table (example *Figure 7*) contains internal element attributes and model association attributes. They define internal values for the element along with defining connection with other entities (root elements, associated diagrams, imports, extensions, relationships, model exporting tool).

Table 10.1 – Process Attributes & Model Associations

Attribute Name	Description/Usage
processType: ProcessType = none { None   Private   Public }	<p>The processType attribute Provides additional information about the level of abstraction modeled by this Process.</p> <p>A public Process shows only those flow elements that are relevant to external consumers. Internal details are not modeled. These Processes are publicly visible and can be used within a Collaboration. Note that the public processType was named abstract in BPMN 1.2.</p> <p>A private Process is one that is internal to a specific organization. By default, the processType is "none," meaning undefined.</p>

Figure 7 Example of attribute table

### 6.5.2 Imports

Import classes MUST be defined externally. They serve as a link with NON-BPMN elements or other BPMN models. They include attributes: importType (Number, String, etc), location (string), namespace (string)

### 6.5.3 XML schemas

XML schemas (example on *Figure 8*) actually present a whole structure of the element, its attributes and imports.

Table 10.32 – Activity XML schema

```

<xsd:element name="activity" type="tActivity"/>
<xsd:complexType name="tActivity" abstract="true">
  <xsd:complexContent>
    <xsd:extension base="tFlowNode">
      <xsd:sequence>
        <xsd:element ref="ioSpecification" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="property" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="dataInputAssociation" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="dataOutputAssociation" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="resourceRole" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="loopCharacteristics" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="isForCompensation" type="xsd:boolean" default="false"/>
      <xsd:attribute name="startQuantity" type="xsd:integer" default="1"/>
      <xsd:attribute name="completionQuantity" type="xsd:integer" default="1"/>
      <xsd:attribute name="default" type="xsd:IDREF" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

Figure 8 Example of XML schema

## 6.6 BPMN Foundation package

Foundation package presents core classes related to Definition class and the core BPMN class model.

1. **Base Element** class is a superclass for the most of the elements. They inherit attributes like: id, documentation, extensionDefinition and extensionValues.
2. **Documentation** class inherits attributes and model associations from Base Element class and adds its own attributes. In this way it extends the functionality of description or documentation elements with text and textFormat attributes, which describes the elements.
3. **Extension** class provides elements that allow adding attributes and new elements to the existing base elements. During the model interchange standard elements will always remain the same however extended attributes might be lost. Extension consists of 4 elements:
  1. **Extension** element is connected to element Definition (by extension attribute inside Definition class) and with the help of its attributes: definition and mustUnderstand flag. mustUnderstand Boolean flag defines if the semantics of extension definition MUST be understood by BPMN adopter for a correct model processing.
  2. **ExtensionDefinition** has its name attribute and is connected with Extension element. In addition it further connects to ExtensionAttributeDefinitions.
  3. **ExtensionAttributeDefinitions** connected to Extension definition define the name, type and isReference flag (if the value is contained inside or it is only reference).
  4. **ExtensionAttributeValue** element connects for ExtensionAttributeDefinition and provides one of the 2 values. Either it is value or valueReference, depending on isReference flag set in associated attribute definition.
4. **External Relationships** class adds possibility to map the existing process artifacts to external resource documentation. As an example the UML diagram artifacts can be non intrusively related to BPMN Artifacts. It has a source, target, type and direction (direction of the relationship None/Forward/Backward/Both) attributes.
5. **Root element** inherits attributes from Base Element. It provides a way to avoid unneeded maintenance by allowing reuse of the element itself with its references and attributes.

## 6.7 BPMN Common elements package

Common elements package include following elements:

1. Artifacts – as described earlier, this is a type of comment used in BPMN
2. Group – used for logical grouping but not affecting the model flow.
3. Correlation – association of two process instances with a help of a particular message element.
4. Error – event is raised when there is a fault while running the activity. It is described with the name and errorCode.

5. Escalation – due to some business situation changes the particular process might need to change itself accordingly. This is analog to error event and also described with the name and escalationCode.
6. Event – this is something that happens during the course of the process and affecting its flow.
7. Expressions – are natural language text formed statements, which are used for evaluation and decision making.
8. Flow element – is the super class of all possible model flow elements
9. Flow elements container – is the super class of BPMN model diagrams. It contains Flow Elements.
10. Gateways – provides a decision point mechanism. Based on received information it can change the flow of the process execution according to defined conditions.
11. Item Definition – assumes data object or messages. They are defined with itemKind (physical or information data type), structureRef (reference to data structure to be used), import (location of data structure and format), isCollection (Boolean flag) attributes.
12. Message – element representing communication content between two parties.
13. Resources – class that specify resources used by activity. The reference attribute defines a link to resource location.
14. Sequence flow – element showing the order of the flow.

## 6.8 BPMN Services package

Provides ability to build services, interfaces or operations.

Interface class is defined with the name, list of operations provided, callable elements list and implementation reference which defines the artifact with concrete implementation.

Operation defines input and output messages which might occur during its' call.

## 6.9 Collaboration package.

Collaboration mechanism assists when there is a need to model communication of several participants. With the help of pool and message flow elements it is possible to encapsulate each participant process inside its own pool. When there is a need to communicate, a message flow will connect both parties as on *Figure 9*.

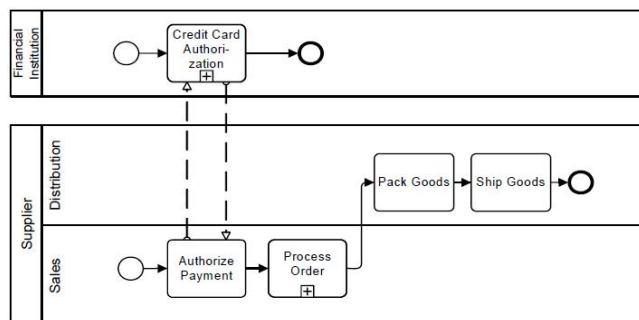


Figure 9 Example of Collaboration diagram.



Conversation element can replace a set of message flows between participants. Then it simplifies the visual understanding of the model by hiding details. When user wants to be able to see the content of the conversation a sub-conversation element can be created. It allows expanding the view into a set of message flows. Conversation consists of conversation node and conversation link displayed on *Figure 11*.

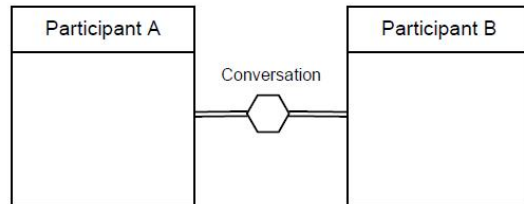


Figure 11. Conversation diagram

## 6.10 Process diagrams

Process is a set of graphical elements used to describe something that is happening in business environment. It visualized as a combination of Activities, Events and Gateways interconnected by sequence flows.

Process entity has three attributes.

- **ProcessType** describes if the process is internal or public. Public processes are used in collaboration when communicating with other participants. They present communication flow, while hiding internal (private) processes. Private processes describe activities inside the company.
- **IsExecutable** attribute describes if the process executes some action or it documents behavior. Executable processes need some input and when the condition is met they are launched. Non executable process does not need any input. Public processes are non executable.
- **IsClosed** defines if non modeled activities, such as sending/receiving messages or events, can appear during the process execution.

Additional model associations can provide the process with auditing links, monitoring links, list of supporting artifacts, connection with other processes, modeler defined additional properties, process responsible resources, correlation subscriptions (matching incoming messages with internal process data correlation subscriptions) and definitional collaboration references (describes which participant element and how the process communicates with).

### 6.10.1 Activity element


Activity element is one of the main BPMN model elements. It describes the work that is performed. Activity class is an abstract super class for all Activity types. This means that in case of creating a new element is should inherit Activity class attributes and functionality. Activity has the following attributes and model associations:







- **IsForCompensation** flag defines if activity is run as one of the regular flow elements or it is initiated only when compensation action is required.
- **LoopCharacteristics** define if the activity should be repeated.
- **Resources** attribute defines entities responsible for the activity. It can be for example a user role or organization. Further more resource class is defined its own attributes like associated parameter bindings, expressions and parameters references
- **Default** attribute defines a sequence flow element that will be launched when no other flows satisfy their initiation conditions.
- **InputOutputSpecification** defines the InputSet and OutputSet for a particular activity
- **Properties** association is a modeler defined additional functionality of the activity.
- **boundaryEventRefs** displays an array of events which are attached to the boundary of the activity.
- **DataInputAssociations** define how the InputSet data shall be extracted and DataOutputAssociations defines the same for OutputSet.
- **startQuantity** defines the sufficient number of tokens passed by sequence flow element in order for activity to start.
- **CompletionQuantity** defines the number of output tokens generated after the activity is finished.

### 6.10.2 Task

Task is atomic activity which is executed by a role or application. It can not be divided into sub-processes. Task types and their characteristics provided in the *Table 5* below. Task type is illustrated by putting a special symbol in the upper left corner of the task element.

**Table 5 Task Types with characteristics**

Task Type	Type visual representation	Attributes	Description
Service task		Inherited from Activity class <b>implementation</b> – defines the technology used (web services or automated application) <b>operationRef</b> – invoked operation definition	Task that is completed with a help of some technology. With a help of message flow a Service task can be associated with partici-

			part, whose service is used.
<b>Send task</b>		Same as service task plus <b>messageRef</b>	Task that sends a message to external participant
<b>Receive task</b>		Same as send task plus <b>instantiate</b> – when this flag set to true the second symbol is used. When the message received the process is instantiated.	Task that waits for a message from external participant. Task connected to a participant via message flow identifies the sender
<b>User task</b>		<b>implementation</b> – defines the technology used (web services or automated application) <b>renderings</b> – define task rendering attributes by connecting to BPMN extension adopters <b>actualOwner</b> – defines the user id who performs the task <b>taskPriority</b> – defines the priority of the task	Assumes that task is performed by a human with the assistance of application
<b>Manual task</b>		Attributes inherited from Activity class only	Task that is done by a user without any aid.
<b>Business rule</b>			Task that provides an input or gets an output from the business rule engine
<b>Script task</b>		<b>script</b> – defines if the task run the script or it acts as an abstract task <b>scriptFormat</b> – defines the format of the script in mime-format	Task that initializes business process engine with a user made script file.

### 6.10.3 Sub-process element

Sub-processes define expanded view of the process. It is a marker with a plus sign place in square and located at the bottom of the task element. Sub-process can be expanded showing the whole process. In addition to sub-process marker there can be loop, compensation (displayed with two arrows, right directed), ad-hoc (displayed with tilde symbol) and multi instance markers (displayed with triple vertical lines). Sub-process has two attributes:

- **triggeredByEvent** – set to true when it is an event sub-process
  - Event sub-process is triggered whenever the start event is triggered. It is displayed as task however uses dashed line instead if solid line.
- **artifacts** – list of artifacts contained in the sub-process.

#### 6.10.4 Transaction

Transaction is a type of sub-process which is triggered by transaction protocol. It is visualized by the same symbol as activity but uses double line. Transaction has three output states:

- **Success** – produces any special output. The normal sequence flow is proceeds.
- **Failed (cancel)** – when transaction is cancelled by event or incoming message appropriate compensation and roll back actions shall be executed.
- **Hazard** – when error intermediate events occur the transaction flow is interrupted and proceeds with error handling.
- Besides inheriting attributes from Activity class Transaction has additional attribute. **method** – defines the method used when committing or canceling transaction. It is set by URI link.

#### 6.10.5 Ad-hoc sub-process

Ad-hoc sub process is another special type. It has only predefined activities, but no predefined process sequence flow can be assigned. Activities are not connected among themselves and it is performer, who decides when and what activity shall be executed. Nevertheless it is possible to assign data inputs and sequence flow for elements, which appear in a mandatory sequence. While inheriting attributes from Activity, ad-hoc sub-process has its own attributes:

- **completionCondition** – checking the expression condition and when it is met the sub-process is finished;
- **ordering** – defines if activities performed in parallel or sequentially
- **cancelRemainingInstances** – used with parallel ordering and defines if remaining activities shall be cancelled when the completion condition is met.

#### 6.10.6 Call activity

Call activity is used during the process execution when a global task or process needs to be run. It uses the same shape as a task element however when it is a global task it is displayed with thick solid line and task type marker in the upper left corner. When it is a process it is displayed as a sub-process element with thick solid line. Besides inheriting attributes from Activity class Call activity has its own model associations. **calledElement** – defines a reference to either a global task or a process element.

### 6.10.7 Global Task

Global Task is an activity that can be called from any other process in the model. Attribute **resources** define the performer of the task if it is not overridden by call activity. Global task types are the same as of the regular task (user, manual, service, business rule, etc.).

### 6.11 Items and Data

BPMN itself does not provide the structure of the data however it comes with a good mechanism for attaching of external expression languages and data structures. Nevertheless its default data structure association is XML Schema and Xpath. BPMN allows multiple expression languages or data structures within the same model.

ItemAwareElement is one of the core classes used for data structure referencing. It provides two model associations: **itemSubjectRef** for instance specification and **dataState** for possible instance state description. DataState class can also be referenced, which will further describe states of the element.

DataObject elements are the ones that are displayed on the BPMN diagram and included within the process. DataObject class inherits attributes from ItemAwareElement and FlowElement adding own state flag isCollection, which defines if the data object is a single value or collection. If it is a collection elements shaped is modified with three vertical lines (as sub-process multi-instance element with sequential execution). Lifecycle accessibility constraints are used:

```
Process A
  Data object 1
  Task A
  Sub-process A
    Data Object 2
    Task B
  Sub-process B
```

According to this example the Data Object 1 can be accessed by all lower elements: Task A, Sub-process A, Task B, Sub-process B. While Data Object 2 can be accessed only by Task B and Sub-process A. This means that the data object can be accessed either by its immediate parent or siblings.

DataObjectReference class further extends ItemAwareElement and FlowElement classes and adds one more attribute for data object. dataObjectRef defines the DataObject which is referenced by DataObjectReference. This class is mostly used when the same data element used within the model and different object states are needed.

### 6.11.1 Data stores

Data stores like Data objects can be placed within the same process and referenced by DataStoreReference class. The difference is that data store information remains after the process is over and is stored externally. Besides inheriting attributes it has its own: **name**, **isUnlimited** flag and **capacity** – when the storage is limited the capacity value must be set. dataStoreRef model association provides with ability to reuse the same data store. It is also possible to assign modeler define properties. Accessibility constraints of data store properties are nearly opposite to data object. The process property can be accessed only by itself or children processes (including elements).

Some of the task might require or produce the data. DataInputs and DataOutputs are used then. Inheriting the InputOutputSpecification class they are able of defining if it is a collection or set of data that needs to be output or input. In addition the define data set references.

### 6.11.2 Xpath expressions for data

BPMN implements several xpath expressions for accessing the data inputs and outputs. Each Item definition is defined as XSD complex type or element. Data objects can be accessed through Xpath function **Element getDataObject('processName','DataObjectName')** – data object name is optional and return of this function is data object value. Data inputs and outputs are accessed by functions **Element getDataInput('dataInputName')** and **Element getDataOutput('dataOutputName')** both returns the value of the element requested. Properties can be accessed by **getProcessProperty('processName','propertyName')**, **getActivityProperty('activityName','propertyName')** and **getEventProperty('eventName','propertyName')**. Instance attributes are accessed through the functions **getProcessInstanceAttribute('processName','attributeName')**, **get ChoreographyInstanceAttribute('attributeName')** and **getActivityInstanceAttribute('activityName','attributeName')**.

## 6.12 Event Element

BPMN event element stands for something that occurs during the process execution. It can be catching or throwing events. Events are triggered by messages. There are following strategies of handling the catching events:

1. Publication – the trigger is caught by all events in the pool. So when the message arrives all corresponding events are triggered.
2. Propagation – the trigger is caught by the innermost enclosed scope instance with catching event. This can be an error handling or escalation. If the error is triggered the execution of the process is stopped. During the escalation triggering the process executes.

3. Termination – if the trigger is thrown the process terminates without compensation or further event handling.
4. Compensation – if the process completes successfully it can be compensated. It is either implicit by executing all the steps in backwards order or user defined.
5. Cancellation – if triggered it will cancel all running activities and compensate committed. In case of transaction it will also roll back the whole transaction.

### **6.13 Lessons learned from extending BPMN standard**

To conclude this chapter we highlight the main lessons learned:

1. BPMN standard is a basic set of process diagram elements and logic for connecting them. It is used in business to systematically visualize any kind of ongoing processes.
2. In order to extend the standard, a proposed solution has to comply with at least one of the conformance types: Process modeling, Process execution, Choreography modeling conformance. In our case, the extension should comply with Process modeling conformance.
3. BPMN process elements are divided into packages. Several packages are included into Process modeling conformance requirements: Infrastructure, Foundation, Common, and Service packages. All of the elements defined in them, has to be supported by the extension.
4. Process Diagrams describe the visual appearance of the elements, their attributes, properties and underlying XML code. We went through only elements of our interest: activities, data items, events.

After getting familiar with the standard and its requirements we are ready to create an BPMN Cloud extension for additional highlighting of the elements, which are stored in public or private clouds.

## **7 Description of BPMN Extension for Cloud Computing.**

### **7.1 Related work**

Extending the BPMN has been in practice for the past few years. One of the extensions [20] aims to extend BPMN with execution of business processes which is originally provided by WS-BPEL and lacking in BPMN. Created Resource and User Task class extensions increased user levels of interoperability and portability.

Another extension [21] proposes security requirements integration into a business process elements. Pool, activity, data object, lane and group elements are extended with security requirement property which defines type of attacks (non repudiation, attack/harm detection, integrity, privacy, access control) the element needs to cope with. Object Constraint Language (OCL) was used in order to create this extension.

As a continuation of the security extension the [22] paper proposes usage of Query View Transformation (QVT) to generate UML use cases from the security aware business process specification.

Nevertheless none of the papers provides a detailed description of the implementation and the way of using the created extension.

## **7.2 Extension requirements**

Now when we have described BPMN process modeling conformance and checked the related work we can create an extension which must comply with all requirements. This extension can be reused and expanded by anyone else interested in integrating Cloud computing into Enterprise architecture. Following the master thesis context we want our Enterprise Architecture model to be more expressive for Cloud Computing. Requirements for BPMN extension are the following:

1. Easy for human visual distinction of processes and data elements, which are stored in cloud;
2. Additional activity element semantic extension for finding out if the element is in cloud or not;
3. Additional data element semantic extension for:
  1. finding out if the element is in cloud or not;
  2. the sensitivity level of data;
4. Conformance with existing BPMN standard requirements;
5. Corresponding Class diagram, attribute table, model associations table and XML Schema;
6. Extension implementation;
7. Extension import and usage.



### **7.3 Activity element extension**

#### **7.3.1 Extension Class Diagram**

Proposed extension is aimed to slightly enrich the existing model and therefore does not introduce any additional elements, but rather adds attributes and intermediate classes.

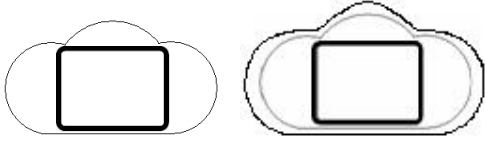
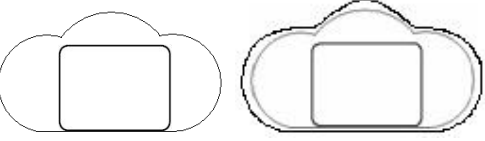
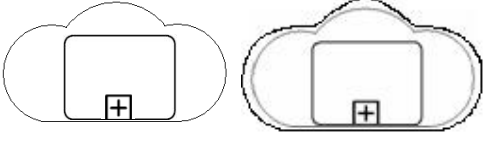
Elements which we modify are Data Element and Activity Element. The UML class diagram of the original model and extension are displayed on the *Figure 11*.



Model class relations remain the same except for intermediate **tCloudAwareActivity** class with the attribute **isInPublicCloud**. This attribute will be inherited by classes: **Task**, **Call Activity**, **Sub Process**, which means that they will be able to define whether the application is in public cloud or not.

When **isInPublicCloud** flag is present the visual appearance of the activity element changes. Depending on the type of the cloud one or two additional border lines are added. In the *Table 6* new representations are provided.

**Table 6 Visual activity element extension**

Description of additional functionality	Visual representation
<p><b>Call Activity</b> is added with additional cloud form solid line border which allows easily distinguish the activity among others. The border is displayed when the <b>isInPublicCloud</b> flag is present. If the flags value set to true single line cloud shape is shown, if false double line shape is shown.</p>	
<p><b>Task</b> is added with additional cloud form solid line border which allows easily distinguish the activity among others. The border is displayed when the <b>isInPublicCloud</b> flag is present. If the flags value set to true single line cloud shape is shown, if false double line shape is shown.</p>	
<p><b>Sub Process</b> is added with additional cloud form solid line border which allows easily distinguish the activity among others. The border is displayed when the <b>isInPublicCloud</b> flag is present. If the flags value set to true single line cloud shape is shown, if false double line shape is shown.</p>	

The marker for activity element extension is cloud form solid single or double line border which is drawn around the main element. Element inside should be placed in the center horizontally and bottom by vertical alignment. Sequence flow lines shall be connected to cloud border. All types of activity elements are possible, including all types of tasks, sub processes, call activities, transactions.

### 7.3.2 Extension attributes and model associations

Table 7 presents attribute which extends the activity attribute set. Model associations are left unchanged.

**Table 7 Activity element extension attributes**

Attribute Name	Description/Usage
<b>isInPublicCloud</b> : boolean = false	<p>The flag that identifies whether activity happens in public cloud or not. Default value is <i>false</i>.</p> <p>If <i>false</i>, then this activity and all corresponding software applications are assumed to be stored locally, in private or hybrid (application storage is local server) cloud. Additional cloud form border is <i>invisible</i>.</p> <p>If <i>true</i>, then it is assumed that all of the software applications that assist activity fulfillment are stored in public cloud. Additional cloud form border is <i>visible</i>.</p>

The attribute is meant to graphically represent elements which are stored in cloud as long as these have one of the biggest security and privacy requirements. Therefore when the user looks at the model he is able to distinct elements without getting deep into their characteristics. It is also possible to group cloud stored activities into Pool element adding more expressivity to the model.

### 7.3.3 Extension XML Schema

XML Schema and XSD document are provided in APPENDIX.

## 7.4 Extension for Items and Data elements

Data elements represent the information that is used or produced by activities. It is data sensitivity that takes most of the attention when deciding whether a process shall be moved into public cloud or not. In this extension no new data elements are introduced, however new attributes, intermediate class and visual elements enrichment are proposed.

### 7.4.1 Extension Class Diagram

In order to comply with requirements we added intermediate class **tSensitivityAwareElement**, which defines two attributes: **sensitivity** and **isInPublicCloud**, which are inherited by **DataInput**, **DataOutput**, **DataStore** and **DataObject**. Therefore we can check the data sensitivity and decide if it should be placed in public cloud or not. As

long as sensitivity can have three possible values, we created an enumeration class **SensitivityType**. We provide standard and extended UML class diagrams on *Figure 12*.

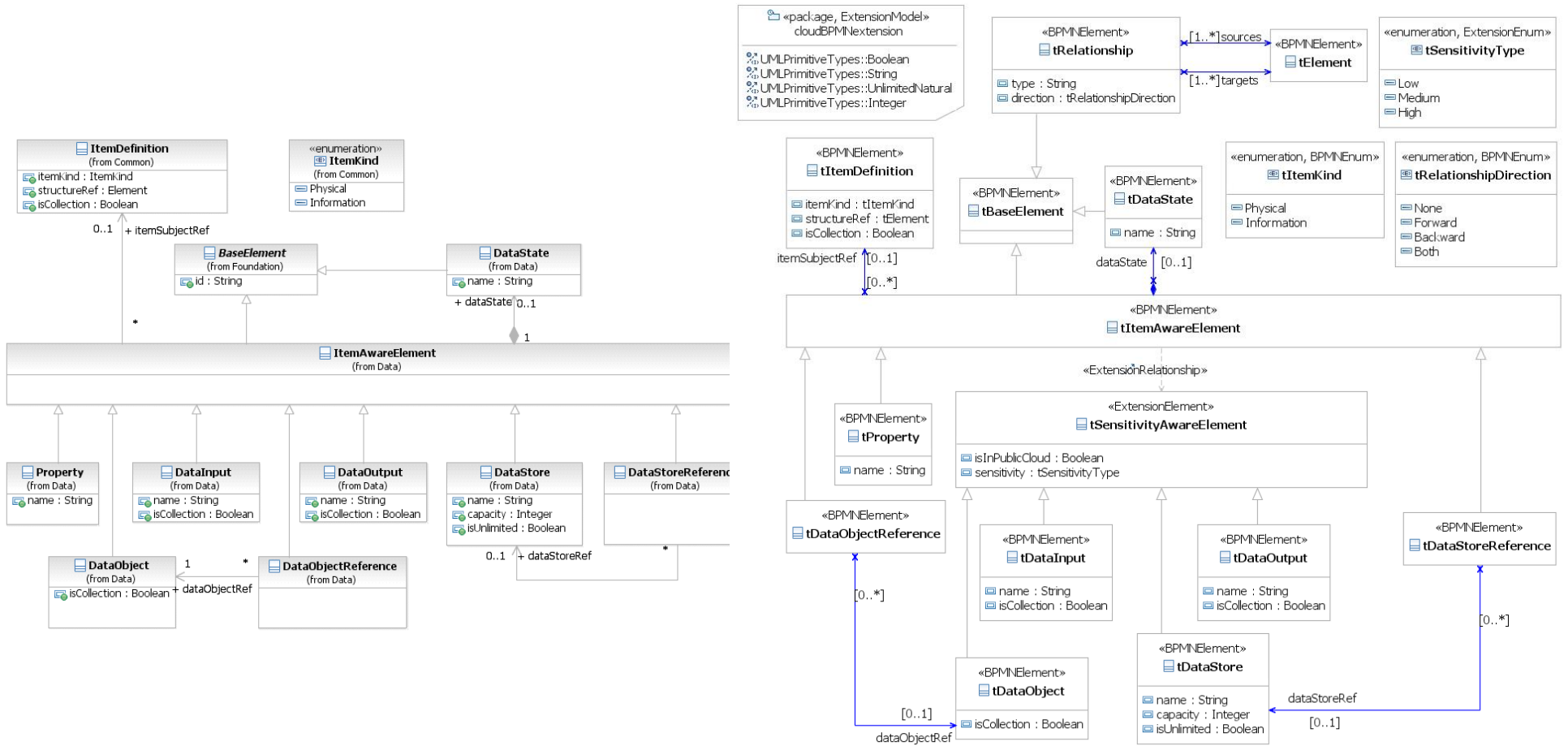


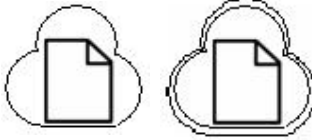



Figure 12 Data Item standard and extended UML class diagrams.

When **isInPublicCloud** flag is present in the element description the visual appearance of the data element changes. Depending on the type of the cloud one or two additional border lines are added. In the *Table 8* new representation is provided.

**Table 8 Visual data element extension**

Description of additional functionality	Visual representation
<p><b>Data Input</b> is added with additional cloud form solid line border, which is visible when the attribute <b>isInPublicCloud</b> flag is present. If the flags value set to true single line cloud shape is shown, if false double line shape is shown.</p>	
<p><b>Data Output</b> is added with additional cloud form solid line border, which is visible when <b>isInPublicCloud</b> flag is present. If the flags value set to true single line cloud shape is shown, if false double line shape is shown.</p>	
<p><b>Data Object</b> is added with additional cloud form solid line border, which is visible when <b>isInPublicCloud</b> flag is present. If the flags value set to true single line cloud shape is shown, if false double line shape is shown.</p>	
<p><b>Data Store</b> is added with additional cloud form solid line border, which is visible when <b>isInPublicCloud</b> flag is present. If the flags value set to true single line cloud shape is shown, if false double line shape is shown.</p>	

The marker for activity element extension is cloud form solid single or double line border which is drawn around the main element. Element inside should be placed in the center horizontally and bottom in vertical alignment. Sequence flow lines shall be connected to cloud border.

#### 7.4.2 Extension attributes and model associations

*Table 9* presents attribute which extends Data Input, Data Output, Data Store and Data Object elements attribute set. Model associations are left unchanged.

**Table 9 Common data element extension attributes**

Attribute Name	Description/Usage
----------------	-------------------

<b>isInPublicCloud</b> : boolean = false	<p>The flag that identifies whether data element stored in public cloud or not. Default value is <i>false</i>.</p> <p>If <i>false</i>, then this data entity is assumed to be stored locally, in private or hybrid (data storage is local server) cloud. Additional cloud form border is <i>invisible</i>.</p> <p>If <i>true</i>, then it is assumed that data entity is stored in public cloud. Additional cloud form border is <i>visible</i>.</p>
<b>sensitivity</b> : SensitivityType = Medium {Low   Medium   High}	<p>Attribute identifies the business sensitivity of data. Default value is Medium.</p> <p><i>Low</i> sensitivity of data can be combined with <b>isInPublicCloud</b> attribute <i>true</i> value.</p> <p><i>High</i> sensitivity of data can NOT be used when <b>isInPublicCloud</b> attribute has <i>true</i> value.</p>

As long as visual representation is directly dependent on attribute value the user does not have to go deep into characteristics when he sees the cloud border around the element. Additional model expressivity brings value for both business and technical people.

### 7.4.3 Extension XML Schema

Data Item extension XML Schema along with Activity element is provided in APPENDIX.

## 8 Proof of concept

The project is open source and can be accessed online:  
<http://sourceforge.net/p/bpmn4cloud/wiki/Home/>

### 8.1 BPMN Extension Implementation

The created extension is open source and can be used for other research or commercial needs.

Java based Eclipse Modeling Tool and BPMN plug-in were used to implement the extension. As a documentation source we use presentation paper [23] and UML 2 user manual [24]. There were 4 steps in creating the BPMN extension:

1. Creating a Conceptual Domain Model of an Extension (CDME) UML diagram which includes the original classes and extending functionality. The result looks as a



regular UML diagram where classes and their relationships are displayed. UML2 class diagram is used.

- Using predefined BPMN Extension UML profile (UML diagram displayed on *Figure 13*) we distinguish the extension elements from the standard ones.

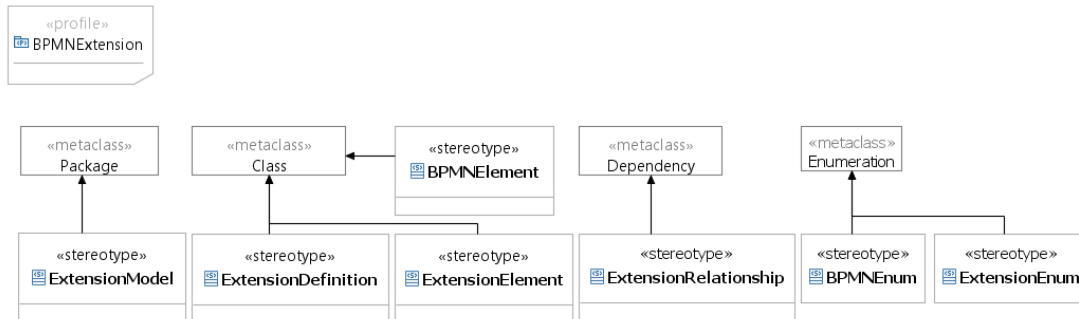


Figure 13 BPMN Extension UML profile diagram.

- With assistance of OMG operational Query View Transformation (QVT) plug in we convert the UML diagram into XML Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema:Schema          xmi:version="2.0"          xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:schema="http://xmlschema/1.0"
xsi:schemaLocation="http://xmlschema/1.0
../org.cidisi.bpmn.extension.transformation/model/XMLSchema.ecore"          targetNames-
pace="http://www.org.eclipse.emf.ecore.impl.DynamicEObjectImpl@179e500          (eClass:
org.eclipse.emf.ecore.impl.EClassImpl@126ad8d (name: OclInvalid_Class) (instanceClassName:
null)          (abstract:          false,          interface:
false)).null/org.eclipse.emf.ecore.impl.DynamicEObjectImpl@179e500          (eClass:
org.eclipse.emf.ecore.impl.EClassImpl@126ad8d (name: OclInvalid_Class) (instanceClassName:
null) (abstract: false, interface: false))">
  <definitions xsi:type="schema:SimpleTypeDefinition" name="xsd:string"/>
  <definitions xsi:type="schema:SimpleTypeDefinition" name="xsd:QName"/>
  <definitions xsi:type="schema:SimpleTypeDefinition" name="xsd:boolean"/>
  <definitions xsi:type="schema:SimpleTypeDefinition" name="xsd:integer"/>
  <definitions xsi:type="schema:EnumerationTypeDefinition" name="tSensitivityType">
    <values value="Low"/>
    <values value="Medium"/>
    <values value="High"/>
  </definitions>
  <definitions xsi:type="schema:ComplexTypeDefinition" name="tCloudAwareActivity">
    <attributeUses required="true">
      <attributeDeclaration name="isInPublicCloud" typeDefinition="//@definitions.2"/>
    </attributeUses>
  </definitions>
</schema:Schema>
```

```

</definitions>
<definitions xsi:type="schema:ComplexTypeDefinition" name="tSensitivityAwareElement">
  <attributeUses required="true">
    <attributeDeclaration name="isInPublicCloud" typeDefinition="//@definitions.2"/>
  </attributeUses>
  <attributeUses required="true">
    <attributeDeclaration name="sensitivity" typeDefinition="//@definitions.4"/>
  </attributeUses>
</definitions>
</schema:Schema>

```

4. The last step is to generate XSD documents. Using JSP based code generator framework JET we transform the XML Schema into package of XSD files. As a result we have 3 files defining the extension and 5 XSD files defining BPMN interchange format, so that importing into another tool would be easy as possible.

The created implementation does not assume changes of the BPMN model shapes, therefore additional modifications shall be applied when importing the model. Extension implementation Java project is attached to this paper.

## 8.2 Extension import and usage.

After creating the extension it needs to be imported into modeling tool and used for making business process models. There are numerous modeling tools in the market. Signavio<sup>6</sup>, Modelio<sup>7</sup>, Eclipse OCL<sup>8</sup> (Object constraint language) all provide expressive modeling however limited to standard BPMN elements. Importing extensions is either impossible or extremely difficult and requires rewriting the source code. Red Hat jBPM is a java based Eclipse extension which provides rich extensibility for basic BPMN elements. User defined java code defines variables which extend modeling elements property list. Even though importing of the existing BPMN extensions is limited and all of the extension functionality needs to be programmed, jBPM is still one of the best available solutions which can provide the required operation. In addition jBPM is an open source tool with a good support, which makes it very attractive for enterprise use. Companies like British Telecom, Tieto, Alfresco, Hudson, SeeWhy, Nuxeo, SNS bank, Shatel, etc. use this tool due to its proven reliability and performance. Therefore there is a good reason for creating the extension with jBPM help.

---

<sup>6</sup> <http://www.signavio.com/en.html>

<sup>7</sup> <http://www.modeliosoft.com/>

<sup>8</sup> <http://www.eclipse.org/modeling/mdt/?project=ocl>

### 8.3 Creating the initial BPMN model

Testing BPMN models were created using jBPM Eclipse plug-in, based on real life enterprise process models. They are advanced enough to serve as a test case of the concept evaluation.

### 8.4 Graphical user interface

The concept implementation assumes user participation during the process. As far as users might not be related to IT, additional graphical user interface is needed.



Figure 14 Application screen

Execution of the program assumes 3 main steps in the navigation screen. *Figure 14* presents the program screen.

When the program is launched only left side of the window is active, the rest remains disabled until the BPMN model is loaded. After the user opens a model, all of the data elements are listed in the text field below the Open File button and Convert to BPMN with radio buttons are activated.

When user has defined the sensitivity for each data element and pressed Convert to BPMN button the lower part of the screen is activated, while the rest is disabled. User can set the required sensitivity level and then press Modify model to apply the changes and upgrade the initial model to processed one.

A list with data description before modifying, after modifications and the list of service providers are displayed for estimation purposes. *Figure 15* presents output of the application in the GUI.

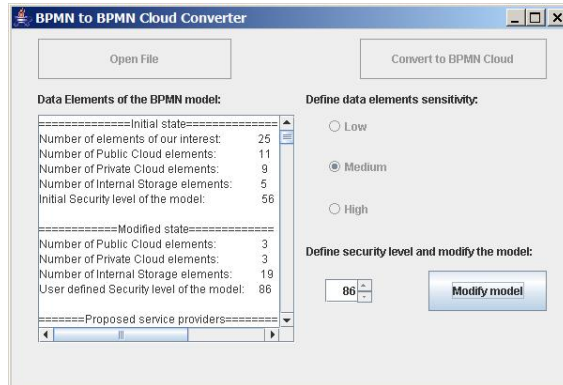


Figure 16. Application output. Graphical user interface

After this the user should open jBPM process modeling tool and verify the updated model.

### 8.5 Exporting the extended BPMN model

Exporting application was developed using Java programming language. It provides a graphical user interface and some user input to export the BPMN model. Model processing steps are the following:

1. The BPMN model is read into program and converted into XML document, which allows accessing and navigating through models nodes.
2. User is provided with the list of data elements and needs to define the sensitivity level of each item:
  1. **High** sensitivity corresponds to business valuable data, such as client information, price calculations, etc.
  2. **Medium** level means that data is not that sensitive and can be place in organization private cloud.
  3. **Low** level means that information needs to be accessed from multiple places and does not posses any commercial secret.
3. BPMN model is updated with defined data sensitivity attributes and also with Boolean flag attribute, which defines if the element is located in local storage, private or public clouds. By default each element is assigned to have public cloud storage location.
4. The initial BPMN model is updated and saved for further usage.

## 8.6 Ontology implementation

Ontology is created using Protégé modeling tool. Graphical representation can be seen on the *Figure 16*.

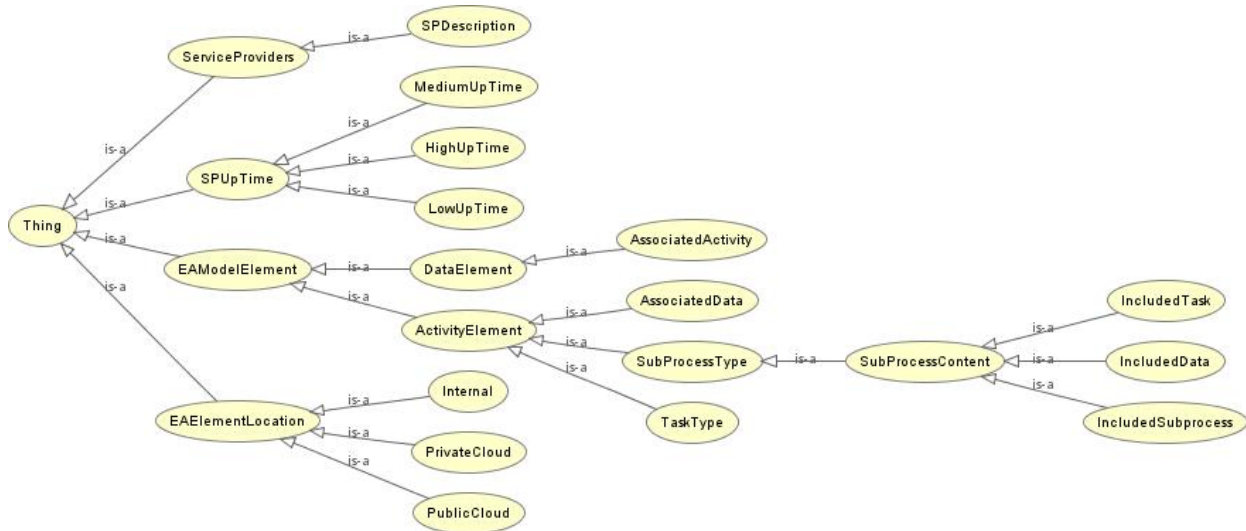


Figure 16 Ontology of EA model and cloud computing service providers

Ontology does not include all of the BPMN defined elements but only those which can be stored in cloud:

1. Data represents various information sources used in EA model.
2. Activity elements represent software and applications which can use or produce data during their execution.

EA ontology model tree consists of several branches:

1. **EAElementLocation** – defines three alternatives of element storage:
  - Internal – stands for existing company infrastructure which is the most secure way of storing information it is also reliable and the has the highest maintenance costs.
  - Private – stands for private cloud network, which is less expensive and secure comparing to internal storage.
  - Public – stands for public cloud network, which is the cheapest and the least secure.

Decision on element placement in one of the mentioned locations depends on its sensitivity level.

2. **ServiceProviders** and **SPUPTIME** help to sort different cloud service providers according to their UP running time.
3. **EAModelElement** – defines Data and Activity elements. Each of them can have associated activities or data. In addition activity elements are divided into regular task elements and sub process elements. Sub process can include more data, activities or sub processes inside, therefore it has to be processed in advanced way.

*Table 10* below presents object properties of the ontology. Each property its domain (elements which hold that property) and range (range of elements the property is applied for)

**Table 10 Object Properties of the Ontology**

Property name	Domain	Range	Description
hasAssociatedActivity	DataElement	AssociatedActivity	Each data element should have associated process
hasAssociatedData	TaskType, Sub-ProcessType	AssociatedData	Some of processes can have associated data element
hasLocation	DataElement, TaskType, Sub-ProcessType	EAElementLocation	Each element must have a defined storage location: internal, public cloud or private cloud
hasDataContent	SubProcessType	IncludedData	Sub process can have data elements inside.
hasTaskContent	SubProcessType	IncludedSubProcess	Sub process can have another sub process element inside.
hasSubProcessContent	SubProcessType	IncludedTask	Sub process can have task elements inside.
hasUPTime	ServiceProviders	SPUPTIME	Providers are sorted according to their service level.
hasDescription	ServiceProviders	SPDescription	Each service provider has a short description

In addition some of the elements have data properties defined. Data properties describe the element with a predefined type property. *Table 11* presents all data properties.

**Table 11 Data Properties of the Ontology**

Data Property Name	Domain	Data property type

Name	DataElement, TaskType, Subprocess-Type, Associated Data, AssociatedActivity, IncludedData, IncludedTask, IncludedSubprocess,	String
Sensitivity	DataElement	String, has 3 values: High, Medium, Low

### 8.7 Analyzing framework

Before starting analysis we have to make two assumptions:

- Internal Storage is the most expensive way of storing applications and data. Private cloud reduces some cost and therefore is cheaper to maintain than internal storage. Public cloud is the cheapest solution.
- Service providers can be divided into 3 groups. Most reliable are those who provide the highest SLA level for the last 365days. These are assumed also as the most expensive. Medium are less expensive and provide less service guarantee. The least expensive service providers have lowest SLA level.

Based on these two assumptions we can conclude that when user is choosing lower level of security he understands the advantages of cost reduction and disadvantages or extra risk taken.

Analyzing framework involves several steps:

1. Define security requirements level. User has to set a percentage 0%-100% of the security required.
2. Import the BPMN model into ontology. User able to import a converted BPMN model.
3. Query the ontology. Querying is based on logical rules which are defined in the following chapter.
4. Apply the security level. There is a specific way of applying security level to the existing model which will be described in the following chapter.
5. Export the model. The original xml file of the model has to be changed in order to apply the results of analysis step. Elements are moved to the location pools, connections are left untouched.

### 8.8 Analysis of the ontology

Analysis of the ontology depends on preconditions and logical rules. Preconditions are:

1. Model data elements can have three types of sensitivity: low, medium and high.
2. There are three alternatives of storing the software and data represented by the model. Public cloud, private organizational cloud or leave them untouched in the internal storage, which is the most secure alternative.
3. Security level. User can reduce or add the overall model security level.
4. The more secure storage required the more it costs. That means that public cloud storage is the cheapest solution while internal storage will bring the highest expenses.

Logical rules were implemented using SQWRL query language and Jess<sup>9</sup> engine for Java platform. The output of running the rules is inferred ontology model, where classes Internal, PublicCloud and PrivateCloud are filled with copies of TaskType, SubProcessType and DataElement instances. Therefore we are able to process the data further by applying user defined security level. Some of the most important logic included in rules:

1. All processes by default are assigned with public cloud location. If data is attached the location of the process is defined by data elements' sensitivity.
2. Storage location of the data element depends on its sensitivity. Three sensitivity levels correspond to three storage alternatives.
3. If the same process has several data elements attached, it will be assigned the storage location of the most sensitive data element. That means that if a process has two data elements attached. One of them has medium sensitivity and the other has high. While placing one of the data elements in a private cloud and another store internally the process element will be stored internally. So that the highest possible security level is kept.
4. Sub process storage location is defined in two steps. First it is assigned default public cloud location unless a data element is attached. Then the content of the sub process is analyzed, if any of the elements is assigned a more secure location the sub process inherits it.
5. If some of the content elements have less secure location they are removed from a sub process and put into appropriate location pool.

We do not include SLA ontology analysis, as far as the approach is straightforward and with the current implementation does not need advanced ontology processing.

Security Level is applied according to the following *Table 12*.

**Table 12 Security level definition**

Security level	Elements in public	Elements in private	Elements in internal
----------------	--------------------	---------------------	----------------------

---

<sup>9</sup> [www.jessrules.com](http://www.jessrules.com)



<b>0%</b>	100%	0%	0%
<b>10%</b>	90%	10%	0%
<b>20%</b>	80%	20%	0%
<b>30%</b>	70%	30%	0%
<b>40%</b>	60%	40%	0%
<b>50%</b>	50%	50%	0%
<b>60%</b>	40%	40%	20%
<b>70%</b>	30%	30%	40%
<b>80%</b>	20%	20%	60%
<b>90%</b>	10%	10%	80%
<b>100%</b>	0%	0%	100%

This simple gradation table allows easily define number of elements in each location based on user set security level. As it can be seen from the table after the level of 50% there are no elements in the internal location. But there can be always elements stored in public cloud as long as when elements have low sensitivity level we do not want them to keep in a private or internal storage.

In addition, client can modify the code and redefine the security level definition, this brings flexibility and scalability to the model.

Operation of applying the user defined security level includes the following steps:

1. Calculate the current model security level.
2. Calculate number of elements in each location according to required security level.
3. Move elements while keeping the highest possible security level.

To calculate the number of elements in each location two main formulas are used.

For the security level range 0%-50%, we use:

$$\text{publicElements\%} = 100\% - \text{definedSecurityLevel\%}.$$

$$\text{privateElements\%} = \text{definedSecurityLevel\%}$$

When the security level range is 51%-100%, we use:

$$\text{publicElements\%} = 100\% - \text{definedSecurityLevel\%}$$

$$\text{privateElements\%} = \text{publicElements\%}$$

$$\text{internalElements\%} = 100\% - ( 2 * \text{publicElements\%} )$$

When number of elements defined for the required security level, we begin with filling the elements into appropriate locations. The pseudo algorithm is described as follows:

```

i = 0;
while i < elementsToThisLocation;
  while publicStack.notEmpty() && i < elementsToThisLocation
    resultArray.add(publicStack.pop())
    i++;
  while privateStack.notEmpty() && i < elementsToThisLocation
    resultArray.add(privateStack.pop())
    i++;
  while internalStack.notEmpty() && i < elementsToThisLocation
    resultArray.add(internalStack.pop())
    i++;

```

When placing elements into result locations we start with filling public location first. According to algorithm we pop elements out of the public stack until it is empty or we have enough of the elements. In case the public stack is empty and we still need elements, we take them from private stack and so on. When it comes to fill the private location we will begin from the place where we finished filling the public location. If there are left elements in public stack we take them first and then come to private ones. In case we are not enough with public and private elements we take them from internal stack. At the end we fill the internal location with the left elements from public/private/internal stacks. So by making N (number of all elements) operations we are able to sort all of the elements into according locations. This way of sorting elements allows us to keep the highest security level as possible, as long as we gradually come from lowest security elements to the highest, while not touching the highest unless we really need to do so.

## 8.9 Exporting the modified model.

In order to be able to export the model, first we need to consider jBPM model xml file structure. The xml file consists of 2 main parts: element description and design description.

Element description provides information about the element, its connections to other elements, content.

Design description describes the common model visual layout. It defines elements measurements and location coordinates.

Initially it was assumed to import the BPMN 2.0 Cloud extension which has redefined some elements attributes and added new visual elements. Assumed output model is provided on *Figure 17*. However in order to implement such functionality a major source code changes of jBPM modeling tool are needed and due to time limit it was decided to take another approach of visualizing whether the element is stored internally, in public cloud or private cloud.

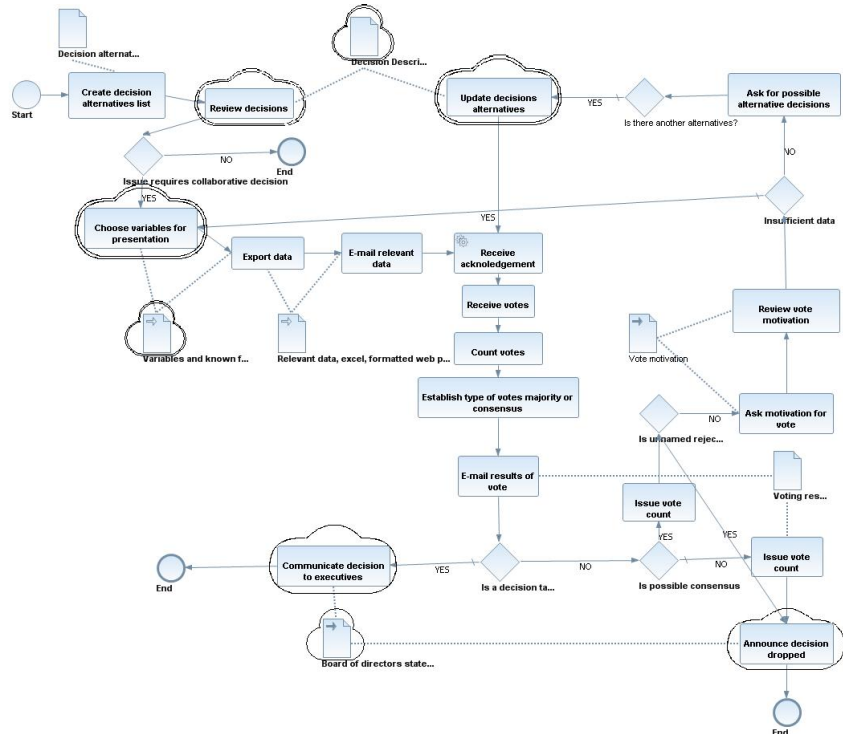


Figure 17 Initially planned output model

In order to implement model changes we first create two pools for public and private cloud storage. All the elements placed in one of them will be then easily distinguished from other location elements. After having created pools, we move element description into appropriate location depending on our made analysis and then change it coordinates in the design part of the xml document. The example of application output is shown on the *Figure 18*.

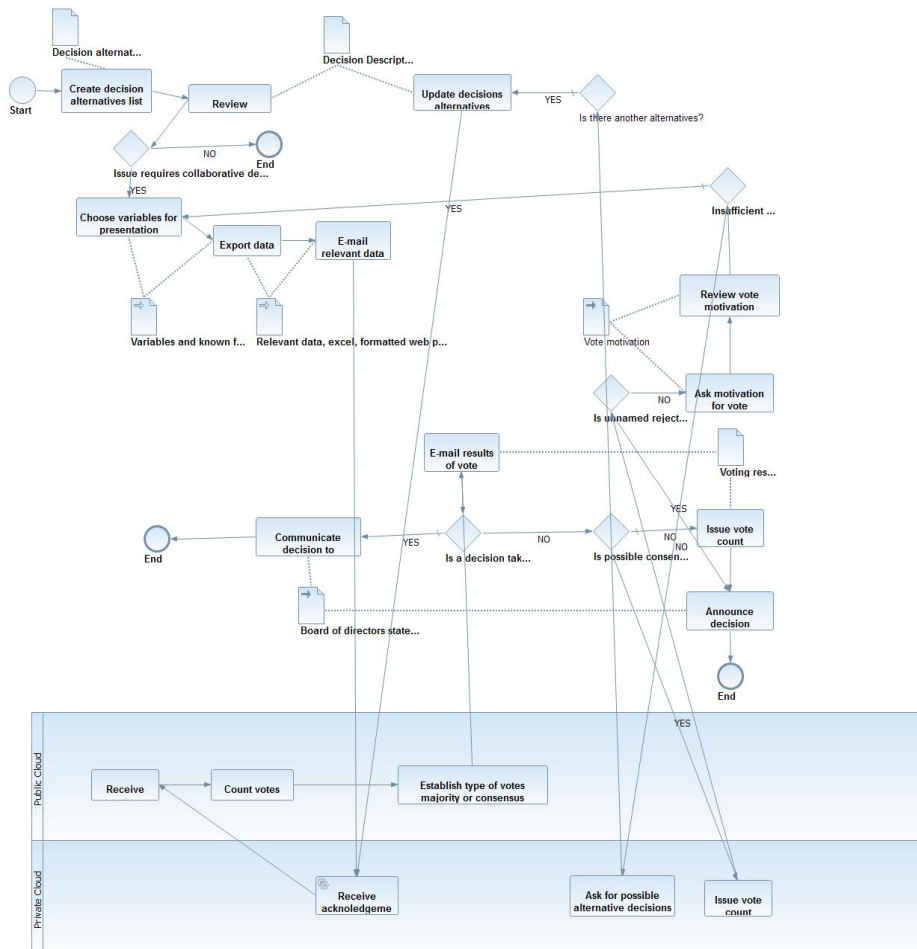


Figure 18 Application output. Modified model

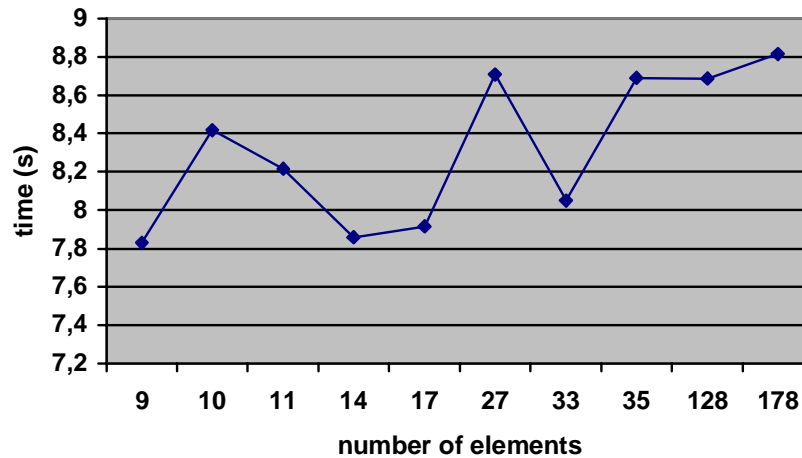
Changing visual representation this way may sometimes cause some inconsistencies in viewing the model, as long as we do it automatically. Therefore minor manual model modifications might be required.

## 9 Evaluation

Regular comparison evaluation techniques (processing time, storage, scalability, etc.) are not applicable in our case as long as there is no other solution that our concept can be compared with. Due to little research in cloud computing and enterprise architecture fields we have to take another approach.

### 9.1 Testing

The only quantitative evaluation available in our case is time testing. 10 different size BPMN models were created from real life examples. All of the models are presented in APPENDIX 1 Three size groups were defined: 9-11, 17-35, 128-178. Each model was processed 12 times and the average execution time taken. *Figure 19* presents the result



of the testing.

As it can be seen from the result diagram the time of running 10 element model is 0,4 seconds less than for 178 element model, while the second model is 19 times bigger than the first.

Another approach separated the common execution time into 2 parts:

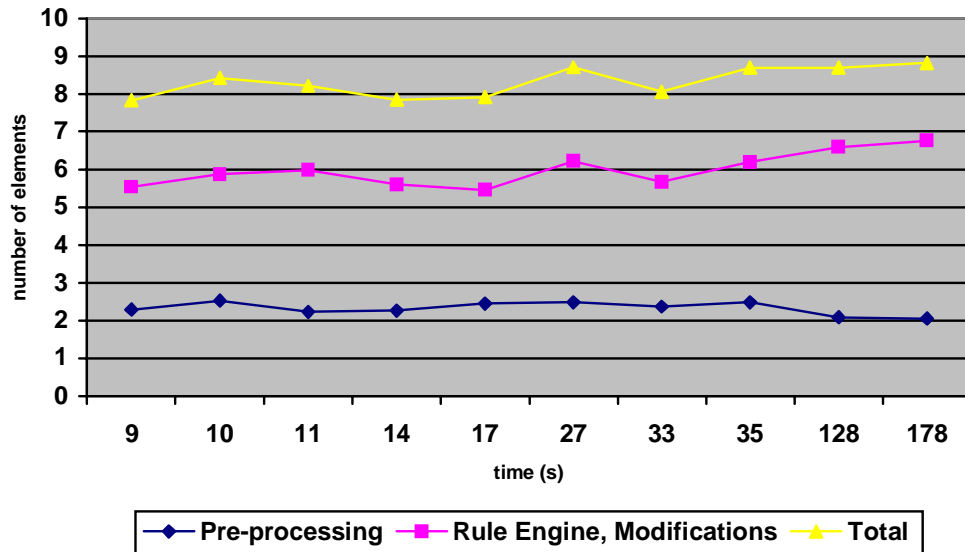
1. pre-processing – the time taken to read the model from file and create temporary data structure. It is dependent on the speed of the computer and the number of elements.

2. Rule Engine, Modification – the time taken to feed data into ontology, apply logical rules and make the reasoning, apply user defined security level and modify the original model.

As it can be seen from results, while pre-processing time remains quite constant, the further processing takes approximately 0,8 seconds more for the model with 178 comparing to one with 10 elements.

Based on results we can claim that using ontology with the rule engine for logical reasoning This proves the performance of the concept and using ontology with rule engine.

Positive aspects:



1. The application was tested with different size BPMN models the smallest has 10 elements, the biggest more than 70 elements. Independently of the number of elements the program takes the same amount of time to execute.
2. User was able to set sensitivity level for each of the existing data elements
3. Additional security level setting modifies the model according to defined logic.
4. Modified models are valid bpmn files which can be opened in jBPM modeling tool.

Negative aspects:

1. Due to current way of implementation during the execution of the big model a lot of elements are moved into location pools, the visual representation suffers and the

model needs additional manual modifications. The fix of this drawback is included in future work description.

2. Current implementation operates on the same file which is opened in the beginning resulting with overwriting it. Another copy of original file has to be made before modifying it.

## 9.2 Evaluation of the Proof of concept

The implementation proves that the proposed concept can be achieved. Further investigation of the implementation quality can then be evaluated by feasibility study. In order to prove the concept we have to define the requirements which were raised and check if they were implemented. For better evaluation we abstract graphical representation layer from the logical analysis, therefore we create two sets of requirements: Graphical representation and Analysis.

Graphical representation requirements:

1. extend BPMN in order to visualize Cloud Computing,
2. use modeling tool to create a BPMN process model and then modify it into extended,
3. present the result in the same modeling tool.

Analysis requirements:

1. the solution has to be scalable
2. logical reasoning shall be fast and easy to implement. User should be able to extend the set of logical rules.
3. allow user to define model security level.

After requirements are defined we check the implementation and confirm if it has considered all of them.

Graphical requirements implementation:

1. BPMN standard was extended and implementation along with results was presented in chapter 8.1.
2. The initial model data elements were added sensitivity and location attributes and initiated. Based on initial data the model was fed into ontology. The logical reasoning rule engine reanalyzed elements' locations and provided an output model which complies with the BPMN extension and jBPM modeling tool requirements. As a result the user got a modified model where elements are clearly separated into storage locations (public cloud pool, private cloud pool or no changes applied).

3. jBPM process modeling tool plug-in for Eclipse was used to create BPMN 2.0 models. The created initial model file was modified and updated to represent different location pools and elements stored inside them.
4. Additionally, a graphical user interface was created to simplify data inputs and outputs. User is able to set initial data sensitivity level and after the current models security level is calculated he can redefine it.

Analysis requirements implementation:

1. Ontology allows us to store as much elements as we need. In addition, the quantity of elements does not affect the overall performance and consequently the user can scale the ontology size easily.
2. Reasoning is done by Jess rule engine which queries the ontology model with logical rule statements. Query statements' syntax is a mix of SQL and discrete logic. All of the logical rules defined directly in Java code and executed fast. Due to application open source availability, rules can be additionally changed by users according to their needs.
3. A special security level setting algorithm was implemented and described in chapter 8.7. It allows the user to redefine models security level and based on obtained result to compare costs associated. The higher risk the cheaper maintenance of the system. On the other hand user is also able to make the security level higher than the original one.

All of the defined requirements were implemented, which allows us to state that the provided implementation is a proof of the proposed concept. Next we make feasibility study of the proposed concept and its implementation.

### **9.3 Feasibility study**

The feasibility study will be done concerning 5 major aspects abbreviated as TELOS [25]. This term stands for: technical, economic, legal, operational, schedule. It is assumed to have a study on each of the aspects and if the implementation satisfies all of them the software project can be positively evaluated. Descriptions of the aspects are described in the following sections.

#### **9.3.1 Technical feasibility**

Technical feasibility describes if the existing company technology is capable of processing the software. Changing the existing technology can sometimes cause unpredicted expenses and therefore the solution shall be possible to implement and run using the existing technical infrastructure.



The proposed concept is capable of running on any Java supporting platform. Therefore any existing infrastructure can process and extend the solution. Nevertheless there is a need for several software tools in order to extend the solution:

- Open source Eclipse platform is needed along with jBPM plug in. They will provide the required environment for developing the application and creating BPMN 2.0 models.
- If there is a need to modify or extend the ontology, the open source Protégé application is needed.
- Rule processing engine Jess is needed in order to run the application. Its' trial version is available for 30 days either a commercial license can be obtained for approx 5000\$. Open source alternatives exist and could also replace proposed rule engine.

As a result, in order to run the software the company will need to have the valid Jess rule engine library and Eclipse platform with jBPM plug-in installed. More technology upgrade is needed in case of further extending. In this case ontology building application might be required. This shows that the proposed concept can be run with minimal existing technology changes.

### **9.3.2 Economic feasibility**

Economic feasibility assumes costs needed to implement the solution. Additional “cost – benefit” analysis can be included to prove the profitability of the software project, which means that the costs of implementing the solution should be lower, than the benefit obtained after the work is done.

In order to estimate the economic feasibility we have to consider several most probable expenses:

1. Startup cost. In order to start using the application it has to be installed locally. As far as it is a standalone application written in java. There will be spent max 30min on installing the software and max 30min for user to get familiar with the product.
2. Running cost. To run the software no server space is needed. And it takes very little of the local storage. Therefore the cost of the computing resources is assumed to be 0\$.
3. Maintaining cost. The proposed implementation does not impose any maintaining, except for updating the jess library if you are using the trial academic version and do not redistribute the software. Updating the library takes 10min of developers' time once in a month and does not influence the total cost.
4. Cost for adding additional functionality. In order to extend the existing solution some more work should be done. Costs depend on requirements and their scope.

5. Software cost. Most of the proposed components are open source. It is only rule engine which can be used inside the company for free. This is described by academic license and limits the usage of the software to a single seat. If there is a need to obtain a commercial license, it will cost 15000\$<sup>10</sup> to have unlimited usage for one application.

To sum up, in order to use the solution only 1hr of user time has to be spent in order to start using the software. Maintaining and running costs are very low and not accounted.

There are several most probable benefits from using the software:

1. The time of analyst will be reduced. With the help of this solution there will be no need in manual investigation of the model, but rather the software will display all of the data elements involved and the user is to define their sensitivity.
2. Human factor error avoidance. As long as the work is done by machine the human factor error is minimized.
3. No need to outsource. Solution allows the company to analyze their models on their own without outsourcing it.

The solution helps to save on the most expensive type of resources – time of personnel. After cost-benefit comparison we can conclude, that the proposed implementation can save a great amount of personnel time and costs associated with it.

### **9.3.3 Legal feasibility**

Legal feasibility describes if the proposed solution complies with legal requirements and regulations of the company. Privacy and confidentiality are of a great concern.

As long as the solution meant to be run locally it does not impose any privacy risks.

### **9.3.4 Operational feasibility**

Operational feasibility describes if the existing personnel and procedures are enough to cope with the new solution. If the innovation requires personnel retraining or procedure updates it must be clearly defined whether existing resources can meet the requirements.

In order to use the solution no additional training is required. In case of extending the solution software manuals and online sources are sufficient.

---

<sup>10</sup> Price information received by email from sales representative on 2012-06-07.

### **9.3.5 Schedule feasibility**

Schedule feasibility describes if the time needed to implement the solution is acceptable or implementation has to be outsourced. Lack of the time can also lead to modification of the scope of the software project or rescheduling of the development plan.

As described in technology feasibility section it will take approximately 1hr of personnel time to set up the application and start using. Additional extending of the solution will require time for manual reading and planning. Time spent on learning will depend on personnel experience and abilities.

Feasibility study proved that the model can be easily integrated with the existing technology and minimum additional personnel knowledge requirements. It also saves costs on human resources. To evaluate the concept even more we arranged short interviews with several EA and BPMN experts, who were presented the implementation and left their feedback.

## **10 Conclusions**

In this work we make the link between Enterprise architecture (EA) and Cloud computing in order to see how they relate to each other. We describe both of them, their specifics, analyze approaches to integrate Cloud computing into enterprise.

In order to improve and speed up adopting of Cloud computing we propose a concept consisting of several implementation steps. We extend EA business process models with additional attributes in order to display elements which are stored in Cloud. After that we use rule engine with ontology to apply logical reasoning and upgrade the existing model. We also implement a mechanism for a user to set up his preferred security level. The model is then reconfigured accordingly. The project is open source and source code was published online.

Evaluating the solution proved the feasibility of the proposed concept.

## **11 Future work**

Future work would include following aspects:

1. Extending jBPM modeling tool with required functionality and elements, this was not done due to time limit.
2. Improvement of the graphical user interface to provide a better output.
3. Multiple models handling in order to automate the process even more.

4. Extend BPMN standard for additional IT business oriented elements such as: SAP, Time Management system, ERP, etc. This enhances readability of business process models.

## References

- [1] OMG, "Business Process Model and Notation (BPMN) v 2.0 standard." OMG, 03-Jan-2011.
- [2] National Institute of Standards and Technology NIST, "The NIST Definition of the Cloud Computing Special Publication 800-145." U.S. Department of ommerce, Sep-2011.
- [3] Mario Godinez, Eberhard Hechler, Klaus Koenig, Steve Lockwood, Martin Oberhofer, Michael Schroeck., *Art of Enterprise Information Architecture, The: A Systems-Based Approach for Unlocking Business Insight*, 1st ed. IBM press, 2010.
- [4] Appistry, "White paper - Unlocking the Promise of Cloud Computing for the Enterprise." Appistry, 2009.
- [5] Kynetix, "White Paper - Cloud Computing A Strategy Guide for Board Level Executives." Kynetix Technology Group, 2009.
- [6] Brian Hayres, "Cloud Computing," *Communications of the ACM*, vol. 51, no. 7, Jul. 2008.
- [7] Timothy Wood, "The case for enterprise-ready virtual private clouds," *ACM DL digital library*. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1855537>. [Accessed: 16-Oct-2011].
- [8] Cong Wang; Qian Wang; Kui Ren; Wenjing Lou, "IEEE Xplore - Ensuring data storage security in Cloud Computing." [Online]. Available: [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=5201385](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5201385). [Accessed: 16-Oct-2011].
- [9] Qian Wang, Cong Wang, Jin Li, Kui Ren and Wenjing Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," *SpringerLink*. [Online]. Available: <http://www.springerlink.com/content/3q57638j61n2313k/>. [Accessed: 16-Oct-2011].
- [10] Rajkumar Buyya, Suraj Pandey, Chirstian Vecchiola, "Cloudbus toolkit for Market-Oriented cloud computing." [Online]. Available: [http://books.google.com/books?id=VKHwHChUFtUC&pg=PA24&dq=Cloudbus+Toolkit+for+Market-Oriented+Cloud+Computing&hl=en&ei=pZOaTqaIMYnU4QTXp8nxAw&sa=X&oi=book\\_result&ct=result&resnum=1&ved=0CEsQ6AEwAA#v=onepage&q=Cloudbus%20Toolkit%20for%20Market-Oriented%20Cloud%20Computing&f=false](http://books.google.com/books?id=VKHwHChUFtUC&pg=PA24&dq=Cloudbus+Toolkit+for+Market-Oriented+Cloud+Computing&hl=en&ei=pZOaTqaIMYnU4QTXp8nxAw&sa=X&oi=book_result&ct=result&resnum=1&ved=0CEsQ6AEwAA#v=onepage&q=Cloudbus%20Toolkit%20for%20Market-Oriented%20Cloud%20Computing&f=false). [Accessed: 17-Oct-2011].
- [11] Jericho forum, "Position Paper - Cloud Cube Model: Selecting Cloud formations for Secure Collaboration." Jericho forum, Apr-2009.
- [12] T. B. Winans, J.S. Brown, "Cloud Computing: A collection of working papers." Deloitte center for the Edge, May-2009.
- [13] Sudip Chahal, J. Hahn-Steicher, D. Kamhout, R. Kraemer, H. Li, C. Peters, "White paper - An Enterprise Private Cloud Architecture and Implementation Roadmap." Intel Information Technology, Jun-2010.
- [14] M. El Kourdi, H. Shah, and A. Atkins, "A PROPOSED FRAMEWORK FOR KNOWLEDGE DISCOVERY IN ENTERPRISE ARCHITECTURE." *Via Nova Architectura*, 06-Jun-2007.
- [15] Elena I. Neaga & Jennifer A. Harding, "Taylor & Francis Online :: An enterprise modeling and integration framework based on knowledge discovery and data mining - International Journal of Production Research - Volume 43, Issue 6." [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/00207540412331322939>. [Accessed: 28-Nov-2011].

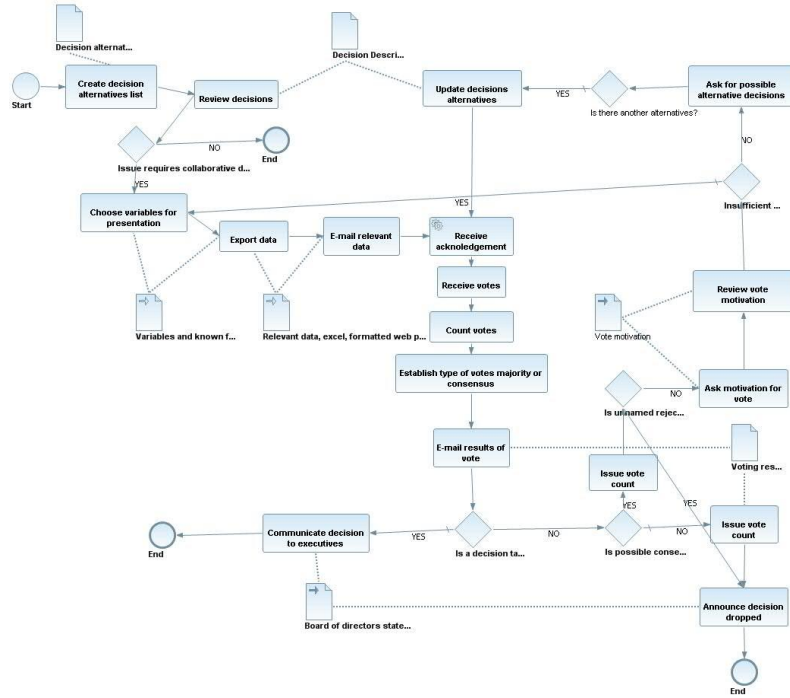
- [16] Maedche, A.; Motik, B.; Stojanovic, L.; Studer, R.; Volz, R.; “IEEE Xplore - Ontologies for enterprise knowledge management,” *IEEEExplore*, Apr-2003. [Online]. Available: [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1193654](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1193654). [Accessed: 28-Nov-2011].
- [17] J. Chattratchat, J. Darlington, Y. Guo, S. Hedvall, M. Köhler and J. Syed, “An architecture for distributed enterprise data mining,” *SpringerLink*. [Online]. Available: <http://www.springerlink.com/content/7k118704w5417764/>. [Accessed: 28-Nov-2011].
- [18] Dongwoo Kanga, Jeongsoo Lee, Sungchul Choib, Kwangsoo Kim, “An ontology-based Enterprise Architecture,” *ScienceDirect.com Elsevier*, 05-Jun-2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417409006368>. [Accessed: 04-Mar-2012].
- [19] “4 dimensional ontology for EA.” [Online]. Available: <http://www.comp.leeds.ac.uk/researchintoindus-try/resources/MatthewWest.ppt>. [Accessed: 04-Mar-2012].
- [20] L. J. R. Stroppi, O. Chiotti, P. D. Villarreal, “A BPMN 2.0 Extension to Define the Resource Perspective of Business Process Models,” <http://www.citeulike.org>, 2011. [Online]. Available: <http://www.citeulike.org/user/mete/article/10181120>. [Accessed: 10-Apr-2012].
- [21] Alfonso Rodriguez, Eduardo Fernandez-Medina, Mario Piattini, “A BPMN Extension for the Modeling of Security Requirements in Business Processes,” *IEICE - Transactions on Information and Systems*, vol. Volume E90-D Issue 4, Mar. 2007.
- [22] Alfonso Rodriguez, Eduardo Fernandez-Medina, and Mario Piattini, “Using QVT to obtain Use Cases from Secure Business Processes modeled with BPMN,” 2008. [Online]. Available: [lams.epfl.ch/conference/bpmds07/program/Rodriguez\\_6.pdf](http://lams.epfl.ch/conference/bpmds07/program/Rodriguez_6.pdf).
- [23] Stroppi, L.J.R., Chiotti, O., Villarreal, P.D, “Extending BPMN 2.0: Method and Tool Support. In: BPMN 2011: Proceedings of the 3rd International Workshop and Practitioner Day on BPMN,” Nov-2011. [Online]. Available: <http://www.slideshare.net/lstroppi/extending-bpmn2-share>. [Accessed: 18-Mar-2012].
- [24] Tatiana Fesenko, “MDT-UML2Tools How To Use UML Profiles - Eclipsepedia.” 12-Nov-2008.
- [25] James A. Hall, *Information Technology Auditing and Assurance*, 3rd ed. USA: South-Western Cengage Learning, 2011.

## APPENDIX

### BPMN Cloud extension XSD file content

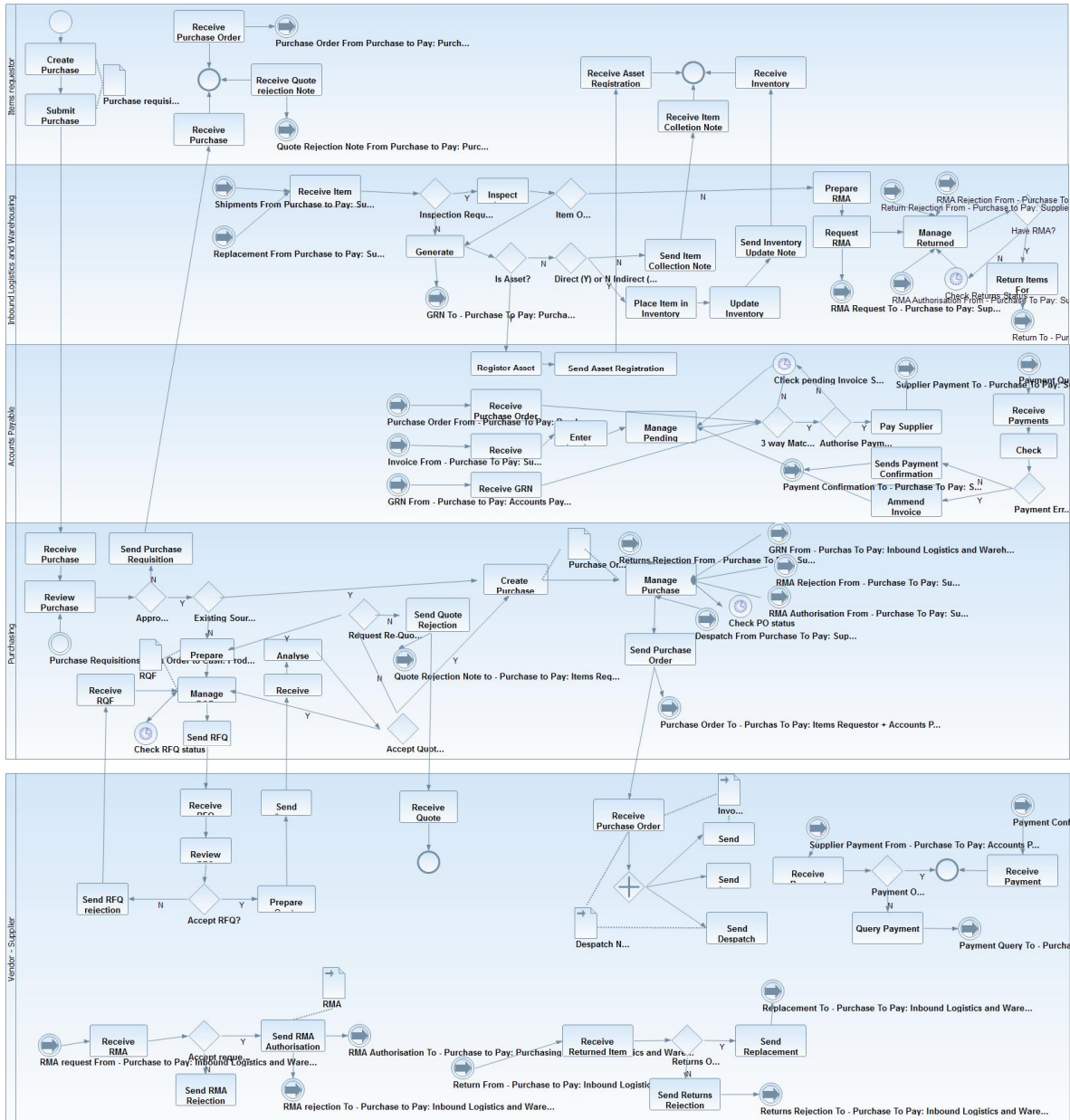
```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
  xmlns="http://www.org.eclipse.emf.ecore.impl.DynamicEObjectImpl@179e500 (eClass:
  org.eclipse.emf.ecore.impl.EClassImpl@126ad8d (name: OclInvalid_Class) (instanceClass-
  Name: null) (abstract: false, interface:
  false)).null/org.eclipse.emf.ecore.impl.DynamicEObjectImpl@179e500 (eClass:
  org.eclipse.emf.ecore.impl.EClassImpl@126ad8d (name: OclInvalid_Class) (instanceClass-
  Name: null) (abstract: false, interface: false))"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL"
  targetNamespace="http://www.org.eclipse.emf.ecore.impl.DynamicEObjectImpl@179e500
  (eClass: org.eclipse.emf.ecore.impl.EClassImpl@126ad8d (name: OclInvalid_Class)
  (instanceClassName: null) (abstract: false, interface:
  false)).null/org.eclipse.emf.ecore.impl.DynamicEObjectImpl@179e500 (eClass:
  org.eclipse.emf.ecore.impl.EClassImpl@126ad8d (name: OclInvalid_Class) (instanceClass-
  Name: null) (abstract: false, interface: false))"
>
<xsd:import namespace="http://www.omg.org/spec/BPMN/20100524/MODEL" schemaLoca-
tion="BPMN20.xsd"/>
  <xsd:simpleType name="tSensitivityType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Low" />
      <xsd:enumeration value="Medium" />
      <xsd:enumeration value="High" />
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="tCloudAwareActivity" abstract="false">
    <xsd:attribute name="isInPublicCloud" type="xsd:boolean" />
  </xsd:complexType>
  <xsd:complexType name="tSensitivityAwareElement" abstract="false">
    <xsd:attribute name="isInPublicCloud" type="xsd:boolean" />
    <xsd:attribute name="sensitivity" type="tSensitivityType" />
  </xsd:complexType>
</xsd:schema>
```

# BPMN model Test1

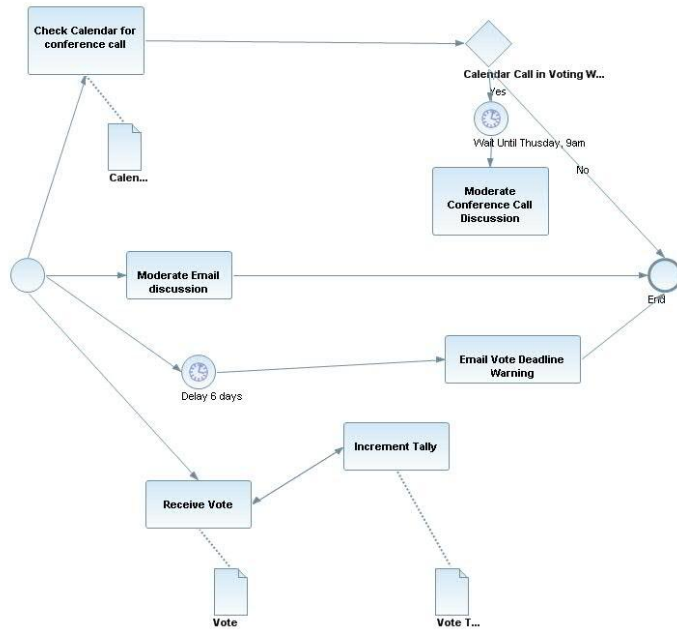




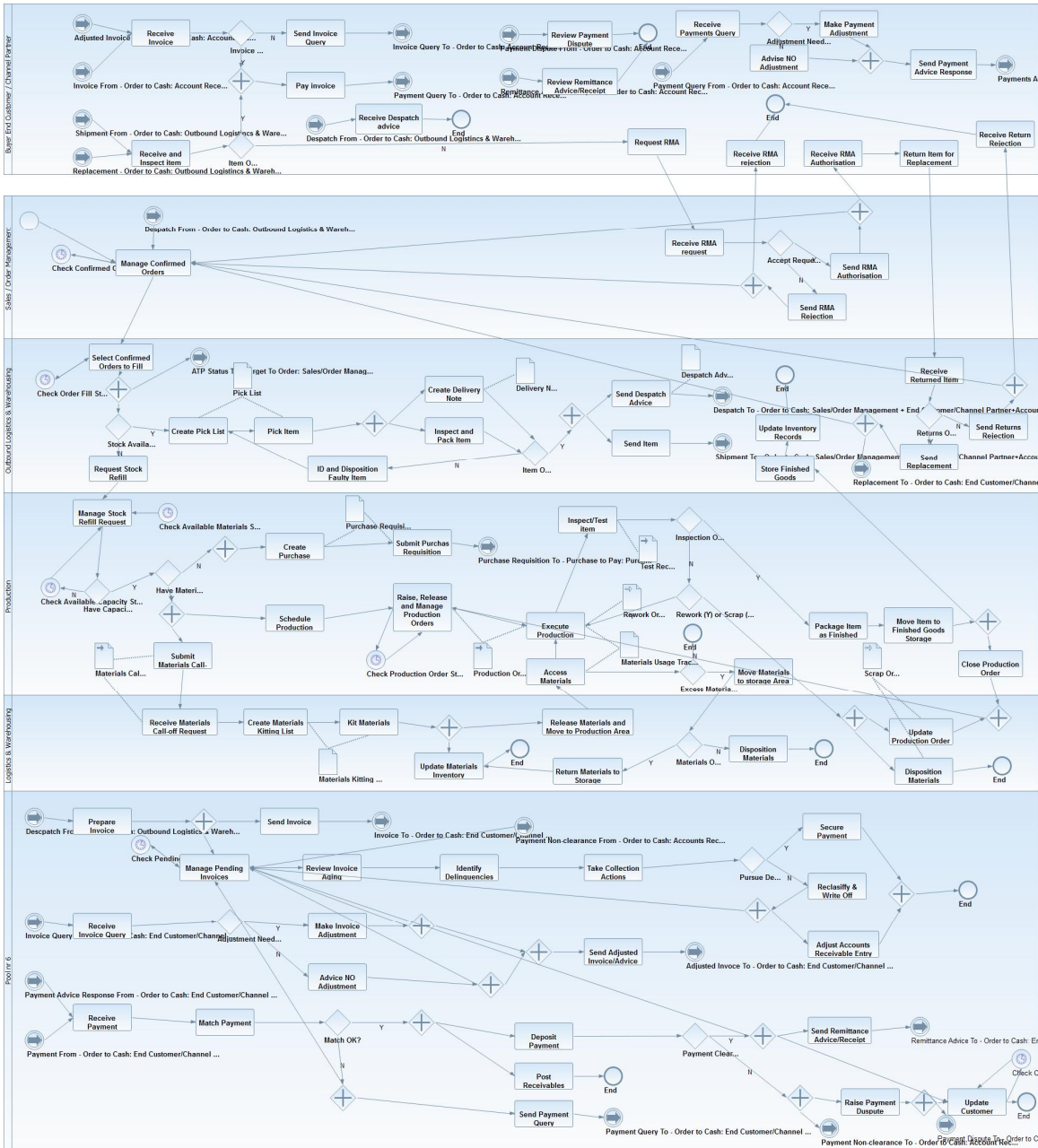
## BPMN model Test2



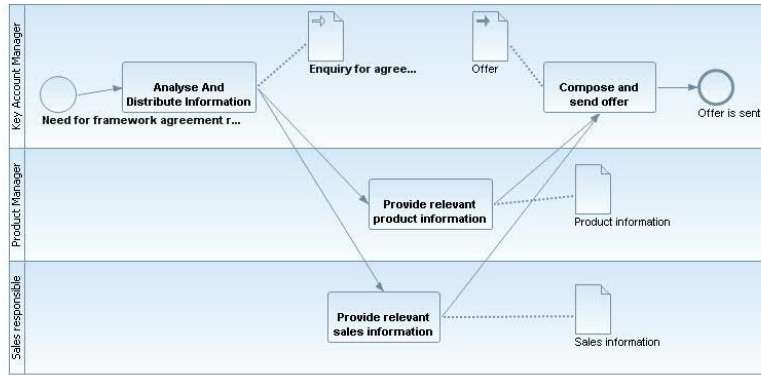
### BPMN model Test3



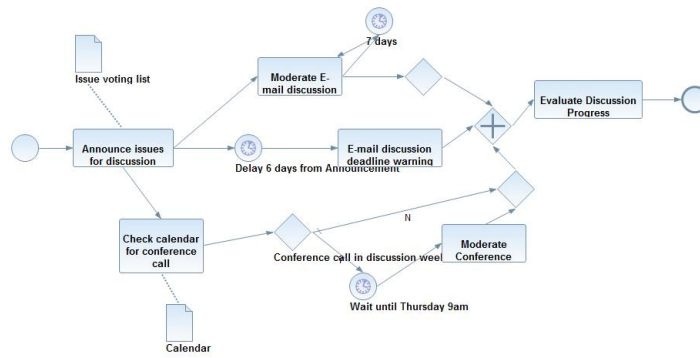
### BPMN model Test4



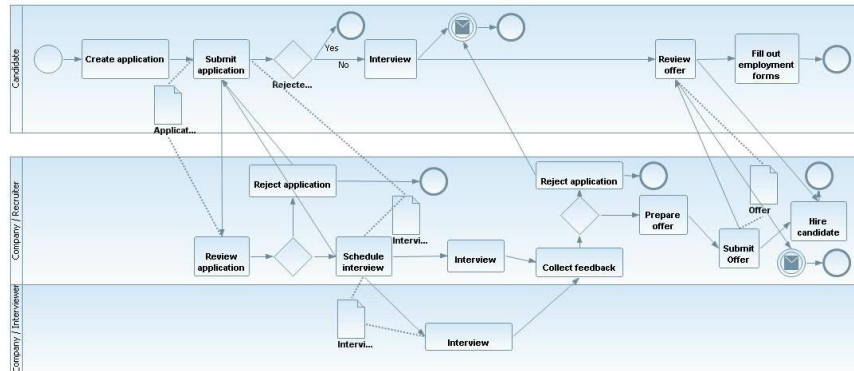
### BPMN model Test5



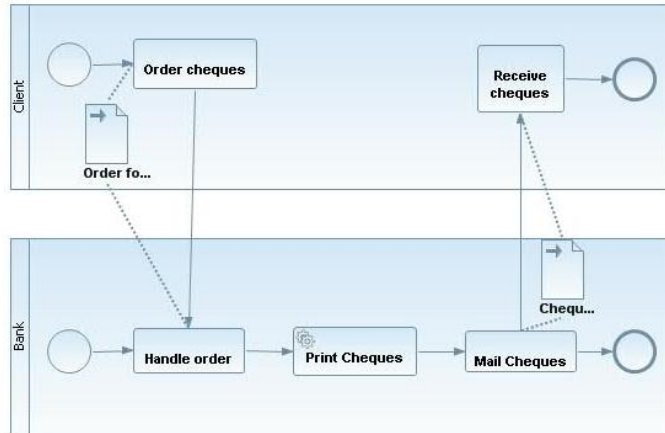
### BPMN model Test6



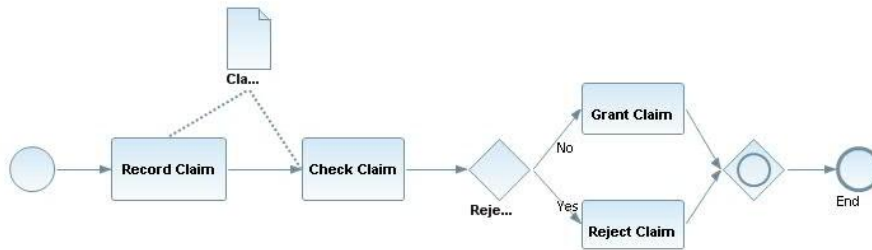
### BPMN model Test7



### BPMN model Test8



### BPMN model Test9



### BPMN model Test10

