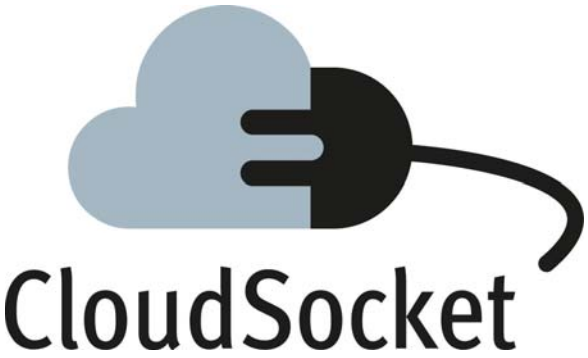


EXPLANATORY NOTES: MODELLING PROTOTYPES FOR BPAAS D3.2

Editor Name	Knut Hinkelmann
Submission Date	June 30, 2016
Version	1.0
Confidentially Level	PU

Co-funded by the Horizon 2020
Framework Programme of the European Union

Executive Summary

The output of this deliverable is a prototype which consists of the modelling method implemented in the ADOxx platform, the BPaaS Ontology and approaches for semantic lifting. The development of the prototype is based on Deliverable D3.1, which describes the specification on hybrid modelling and semantic lifting to enable Cloud Eco-System Design for the BPaaS Design Environment.

This report 1) provides a brief recap of the design environment, 2) describes the BPaaS modelling environment and the modelling method to achieve smart Business IT alignment 3) shows how the ontology has been further developed and which concepts for describing cloud services it includes, 4) introduces the semantic rules for matching semantically lifted business processes and workflows, and 5) shows the implementation, practical application, and setup and configuration of the prototype.

The BPaaS Ontology aims at supporting the semantic augmentation of the business process models and workflows, and enabling service discovery. The ontology has been developed by analysing the business scenarios and competency questions in Deliverable D3.1. In order to describe cloud services the concept of functional (business perspective) and non-functional requirements (technical perspective) has been introduced.

Then the ADOxx prototype for BPaaS Design environment has been extended. It enables the editing of several model types including business processes, workflows, service description model, organizational model, document model, decision model and KPI models. The semantics of the models is defined by the BPaaS ontology. The BPaaS Design prototype access the ontology in order to annotate the models via using an interface to a web service which has been implemented.

A first preliminary version of the prototype is available free for download from the CloudSocket webpage (cloudsocket.eu/download). The application and testing for some of the business scenarios will show strengths and limitations of the modelling framework. To this end the prototype will be developed further aiming at satisfying the requirements from the business scenarios of the use case partners.

Project Context

Workpackage	WP3: Business Process as a Service Research
Task	T3.1: BPaaS Design Environment Research
Dependencies	Based on D3.1 and WP4

Contributors and Reviewers

Contributors	Reviewers
Knut Hinkelmann (FHNW), Sabrina Kurjakovic (FHNW), Benjamin Lammel (FHNW), Emanuele Laurenzi (FHNW), Robert Woitsch (BOC)	Jürgen Jähnert (BWCON), Joaquin Iranzo Yuste (ATOS), Wilfried Utz (BOC), Kyriakos Kritikos (Forth)

Approved by: Robert Woitsch (BOC) as Coordinator

Version History

Version	Date	Authors	Sections Affected
0.1	1 June 2016	Hinkelmann	Structure
0.2	2 June 2016	Hinkelmann	Introduction
0.3	12 June 2016	Lammel	Implementation, technical description and manual for installation
0.4	13 June 2016	Laurenzi	Non-functional requirements, non-functional descriptions, Matching rules for Business-IT alignment / Ontology prototype
0.5	14 June 2016	Kurjakovic	Functional requirements / ontology prototype
0.6	14 June 2016	Hinkelmann	BP Restructuring and alignment of argumentation
0.7	22 June 2016	Kurjakovic, Laurenzi, Lammel	Executive summary, refinement of requirement and implementation sections
0.8	23 June 2016	Hinkelmann, Kurjakovic, Lammel, Laurenzi	Revisions, overview of prototypes and final chapter
0.9	27 June – 30 June	Hinkelmann, Lammel, Kurjakovic, Laurenzi	Adapted according to reviewers' feedbacks
1.0	30 June	Hinkelmann, Lammel, Kurjakovic, Laurenzi, Woitsch	Finalizing the prototype Approval

Copyright Statement – Restricted Content

Table of Content

List of Figures

List of Tables

1 INTRODUCTION

This document introduces the research contribution for the BPaaS Design Environment, which supports the alignment of Business and IT in the Cloud. In this chapter it is described how this research works for the BPaaS Design Environment and how it supports overall BPaaS lifecycle.

1.1 Project Context of this Document

The modelling prototypes for BPaaS are a first implementation of the BPaaS Design Environment specification, which is described in Deliverable D3.1 "Modelling Framework for BPaaS" (CloudSocket 2015b). The BPaaS Design Environment is part of the CloudSocket architecture as described in Deliverable D4.1 "First CloudSocket Architecture" (see Figure 1).

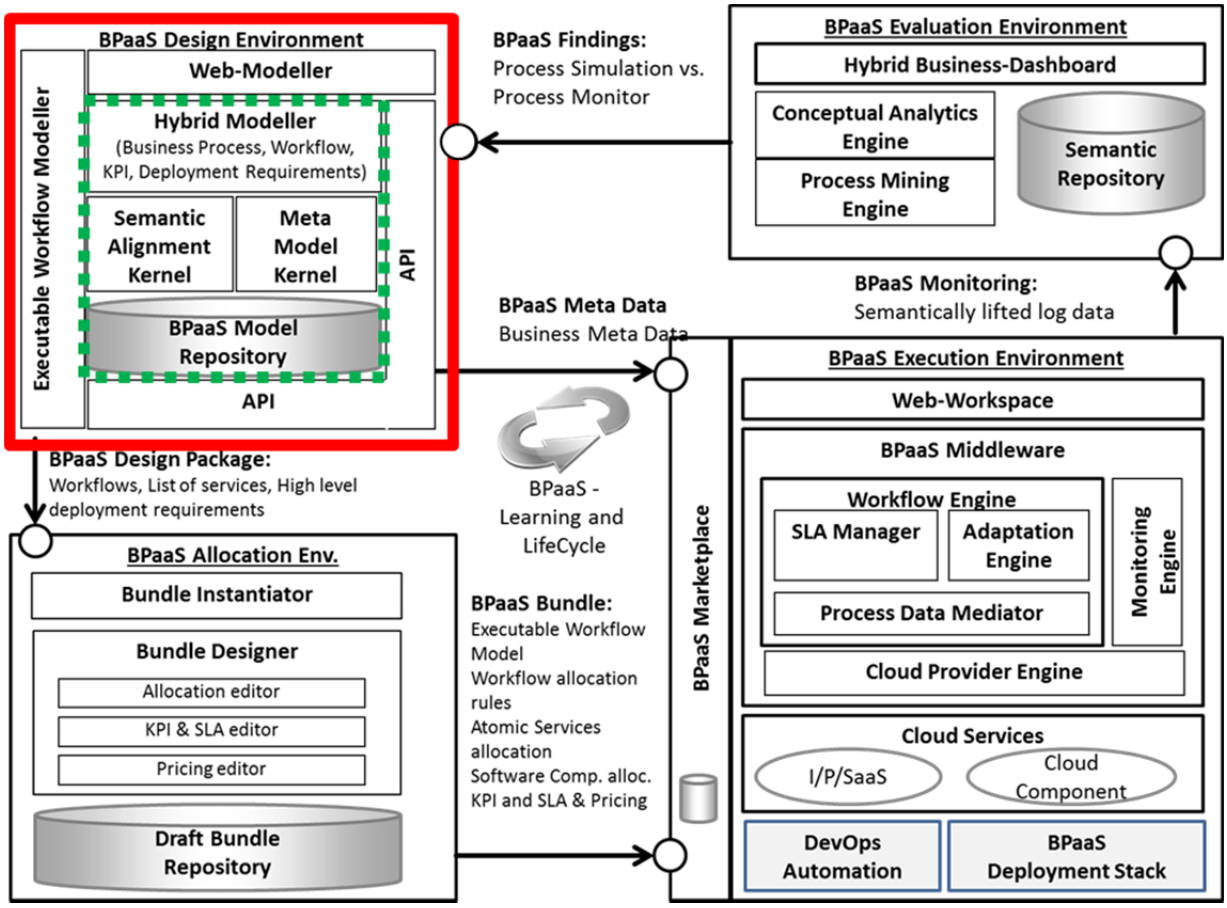


Figure 1: Initial High-level Architecture of CloudSocket (CloudSocket 2015a)

The Hybrid Modeller as part of the BPaaS Design Environment focuses on the smart alignment between business requirements and cloud offerings. Figure 2 highlights this phase of the CloudSocket approach, as it was presented in the Description of Action (CloudSocket 2014). As specified in the initial CloudSocket architecture (CloudSocket 2015a), the BPaaS Design Environment provides conceptual modelling tools for (a) designing domain specific business processes, (b) executable workflows, (c) additional description and rules for deployment as well as (d) Key Performance Indicators (KPIs). The modelling prototypes for BPaaS include:

- the user interface to graphically design domain-specific business processes and their requirements, as well as the modelling of executable workflows,

- the semantic lifting enables the semantic annotation of business process and workflow models with concepts from the BPaaS ontology,
- a semantic alignment is realized by mapping rules which identify suitable workflows that satisfy the business process functional and non-functional requirements.

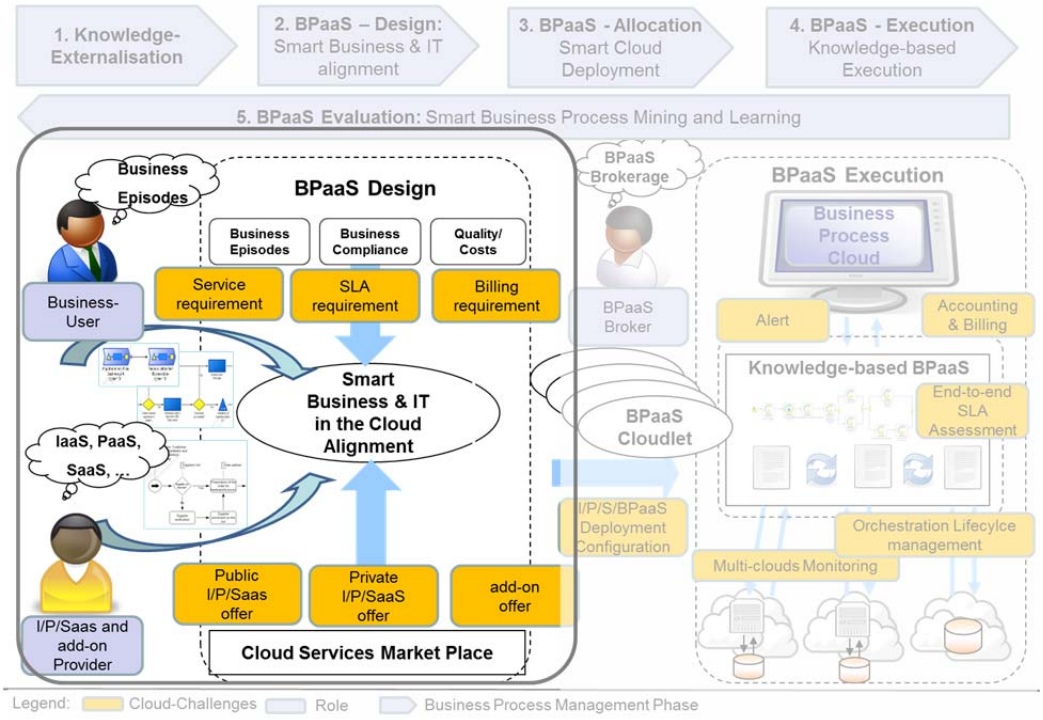


Figure 2: Focus of Business and IT-Cloud Alignment as Part of the CloudSocket Approach

1.2 Structure of the Document

Chapter 2 gives a brief overview of the modelling prototypes used to achieve alignment of business and IT in the cloud. Chapter 3 introduces the BPaaS modelling method. It depicts the models used to achieve a smart alignment of business and IT. Chapter 4 refers to the ontology concepts and describes the implementation of the functional and non-functional requirements. Chapter 5 explains the semantic lifting of the BPaaS Models. It includes a conceptual description of the semantic annotation and shows the concrete application. Chapter 6 describes how human understandable models are made machine understandable. Furthermore, it explains the implementation of the matching between business processes and workflows. Chapter 7 refers to the installation and configuration of the prototype. It includes the graphical modelling and annotation environment (ADOxx), and the setup and use of the web service for annotation. Chapter 8 concludes with a summary of the deliverable and outlook for further developments.

2 OVERVIEW OF THE MODELLING PROTOTYPES

The modelling prototypes for BPaaS are a first research implementation of the BPaaS Design Environment as described in Deliverable D3.1 (CloudSocket 2015b). This implementation integrates various modelling approaches, allowing for both human and machine interpretation of the models produced (Hinkelmann et al. 2016). It builds on the knowledge engineering for business process management as presented in (Karagiannis & Woitsch 2010), and supports informal (text), semi-formal (graphic) and formal (ontology, rules) knowledge representations.

- At the user interface, graphical notations are provided, which can be easily be understood by the CloudSocket Broker. Text elements can be used to further explain aspects for which no graphical representation is provided.
- Smart business and IT alignment requires formal knowledge representation with appropriate inference mechanisms. The knowledge base therefore contains an ontology, which defines the semantics of the modelling elements, and rules for service discovery and allocation.

Semantic lifting integrates the human-interpretable models of the interface with the machine-interpretable models (see Figure 3). It makes the semantics of graphical and textual models explicit (Kappel et al. 2006; Hrgovcic et al. 2013).

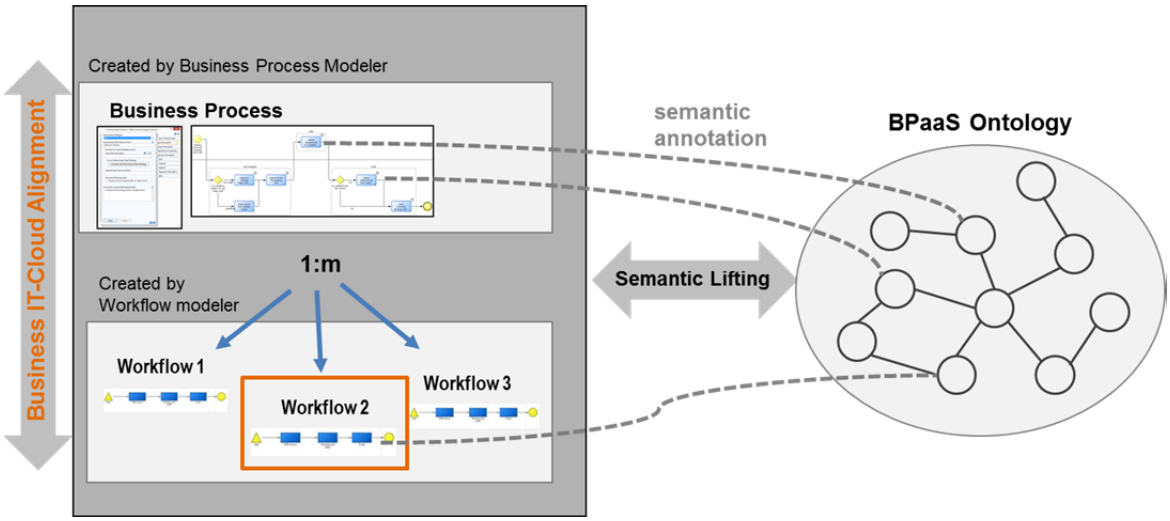


Figure 3: Semantic Lifting

Figure 4 provides a sketch of the framework for the smart business and IT alignment in the cloud. On the left-hand side there is the human-interpretable modelling environment which is implemented in ADOxx.org and referred to as BPaaS Modelling Environment. On the right hand side, there is the machine-interpretable ontological representation and the inferencing for the smart business and IT alignment.

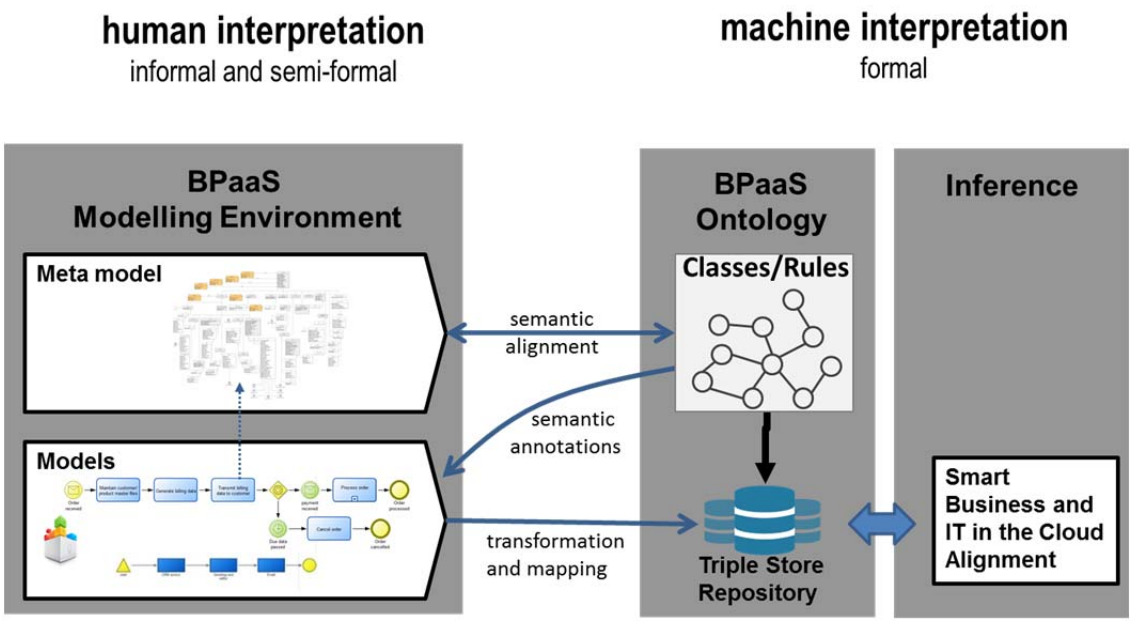


Figure 4: Overview of the BPaaS Design Environment and Smart Business IT-Cloud Alignment

The ontology defines the semantics of the meta model elements. This means in particular that it contains class definitions for the modelling elements of business processes and workflows. Furthermore, it contains class definitions, which can be used to annotate models and model elements. The facts of the knowledge base are created by transformation, which creates instances and maps them to the corresponding classes of the ontology.

The alignment of business and IT in the Cloud, which is supported by the modelling prototypes, can be regarded as a four step approach:

1. Business processes and workflows are modelled using the BPaaS Modelling Environment and the semantic lifting support via the BPaaS Ontology. Both the business process models and workflows are annotated with a description of the functional and non-functional capabilities.
2. The information of the models is translated into a machine-interpretable representation (called triple store repository in Figure 4).
3. The business process model is mapped to one or several appropriate workflow models by comparing the business process requirements with the workflow descriptions. This workflow identification is done by the inference engine for smart business and IT in the cloud alignment using the mapping rules of the BPaaS Ontology component.
4. Finally, the BPaaS Design Package is created. It consists of the domain-specific business process and the executable workflow model, the key performance indicators and additional information which is relevant for allocation and deployment.

The prototypes have a focus on following parts: The business process and workflow models, the respective semantic annotations and the transformation and mapping rules for the business and IT alignment. The creation of the BPaaS Design Package still requires manual work by Cloud Broker experts, but could be considered to be automatized in the further development of the BPaaS design prototypes.

3 BPAAS MODELLING METHOD

In this section, we describe the modelling method for the graphical representation of business processes and workflows.

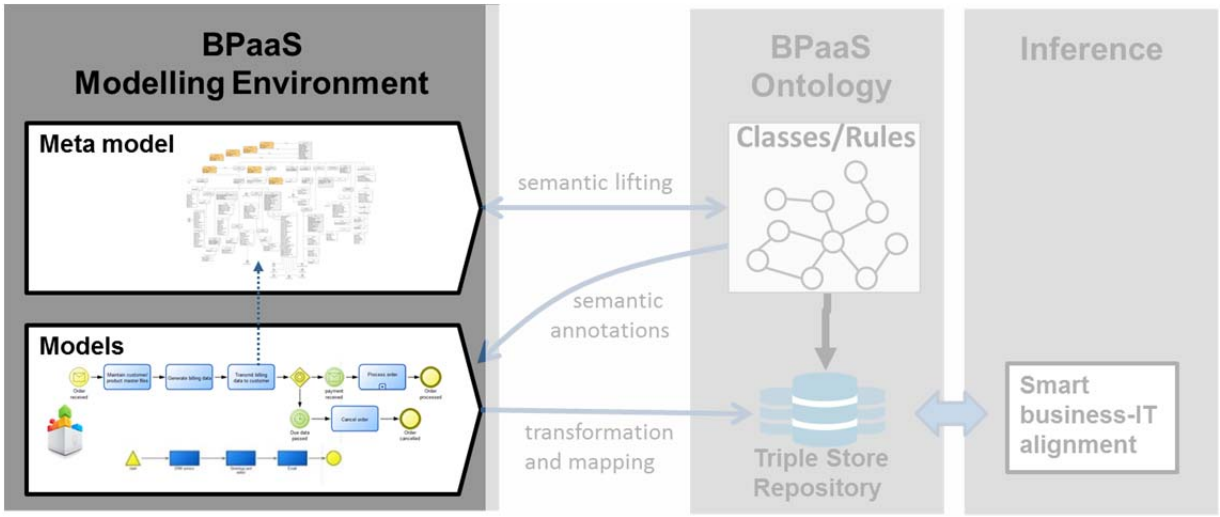


Figure 5: The modelling prototype as part of the BPaaS Design Environment

Figure 6 shows the BPaaS meta model focusing on the models that are of central importance for the semantic lifting and alignment: the domain specific business layer (business processes), the IT-Cloud relevant technical layer (workflows), as well as the model for the semantic interaction between them referred to as Service description model.

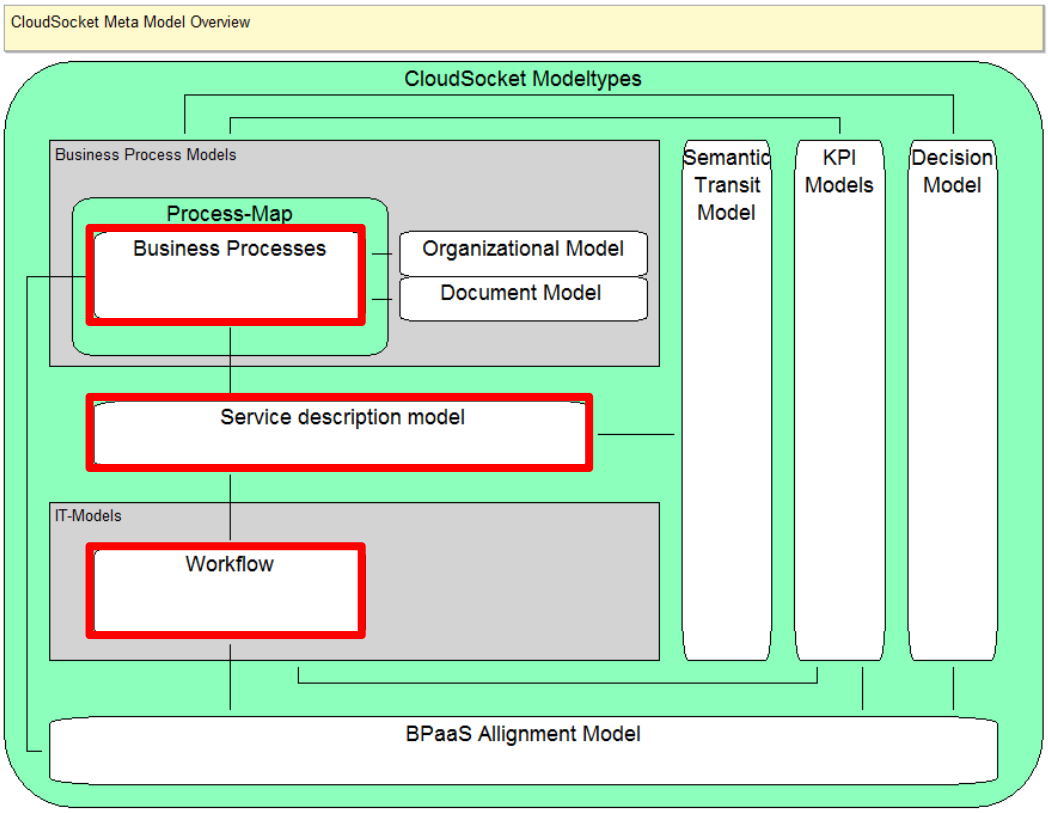


Figure 6: BPaaS Meta Model Stack

3.1 Business Process Model

For the business process modelling we use the BPMN 2.0 standard from OMG. In order to specify the requirements, we added the possibility to assign requirements to parts of a process.

This is performed by making a group for those tasks, which should be implemented by a cloud service and make a reference to a business process requirement specification. A group can include either a single task or several tasks. To explain the meta model, we show an example of a model created with it: In the process of Figure 7, there are two groups "User part" and "Social Media Execution Part". For each of these groups there can be references to Business Process Requirements specifications (see Section 3.3.1).

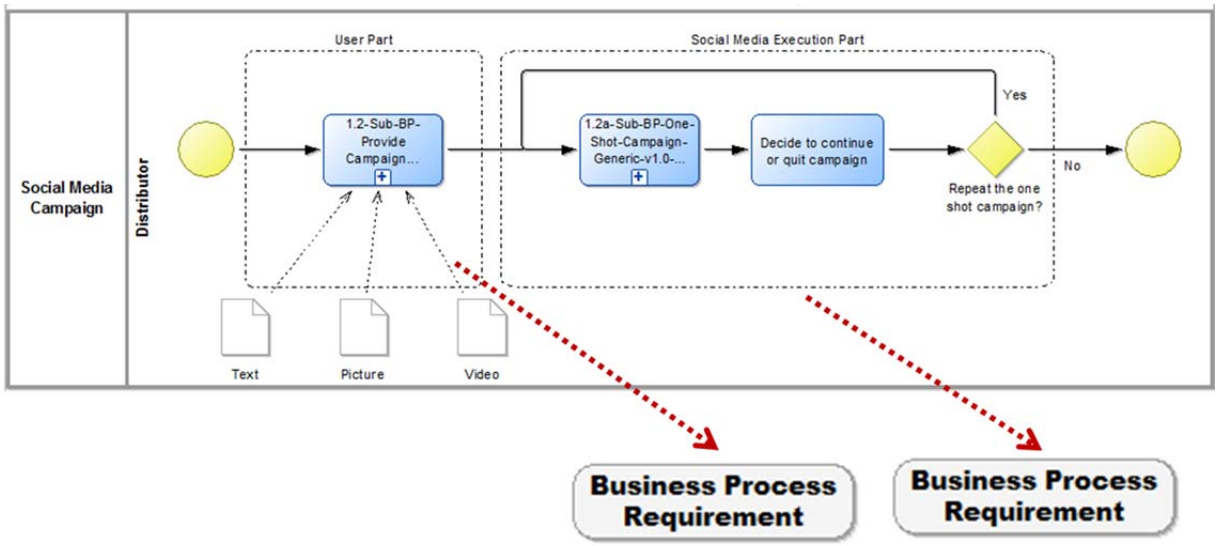


Figure 7: Business Process Model with References from Groups to Requirements

3.2 Workflow Model

A workflow model reveals technical aspects of the business process and is therefore meant to be designed and understood by technical people. In particular, in contrast to business processes, workflow models do not have requirements. Instead there is a reference to descriptions of service functionalities and non-functional properties. However, since lanes in BPMN represent actors, and in workflows the cloud services are actors because they execute the task, we provide annotation references from the lanes to the workflow descriptions (see the two dotted red lines heading to Workflow Description in Figure 8). The workflow description is an aggregation of the properties of the cloud services.

As an example, Figure 8 shows the workflow social media campaign. Like business processes workflows are expressed in the modelling language BPMN 2.0. Nevertheless, we decided to represent workflows with a separate model type, because there are different attributes and references for workflow elements.

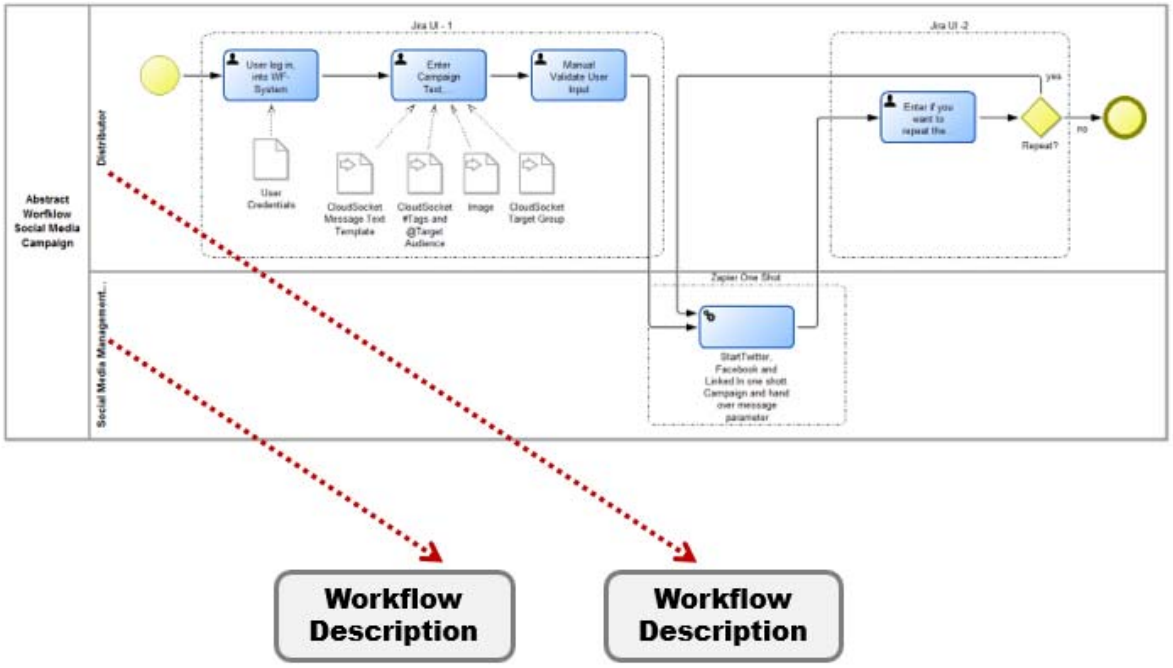


Figure 8: Abstract Workflow Model with Reference from Lane to Description

3.3 Service Description Model

The semantic interaction between the domain specific business layer (business process) and the abstract workflow including IT-relevant aspects is done by semantically lifting these two models with the 'Service Description Model'. Latter represents a cloud specific description concept. It is based on the FODA approach (Kang 1990), where each business process activity is analyzed according to IT requirements.

Within this model type each task of both processes can be semantically enriched by describing

- functional and non-functional requirements of business processes
- functional and non-functional descriptions of workflows

3.3.1 Modelling Business Process Requirements

We stick with our "Social Media Campaign" application scenario to underpin the description of the model type. Figure 9 depicts the two relations that start from the task group annotation (Business Processes model type - section 3.1) and head to the two instantiations of the Business Process Requirement specifications: "User part annotation elements" and "Social media execution part annotation elements". These two elements are modelled in a model of the type Service Description Model. In the ADOxx environment this kind of relation is called *Interref* and connects elements of different model types.

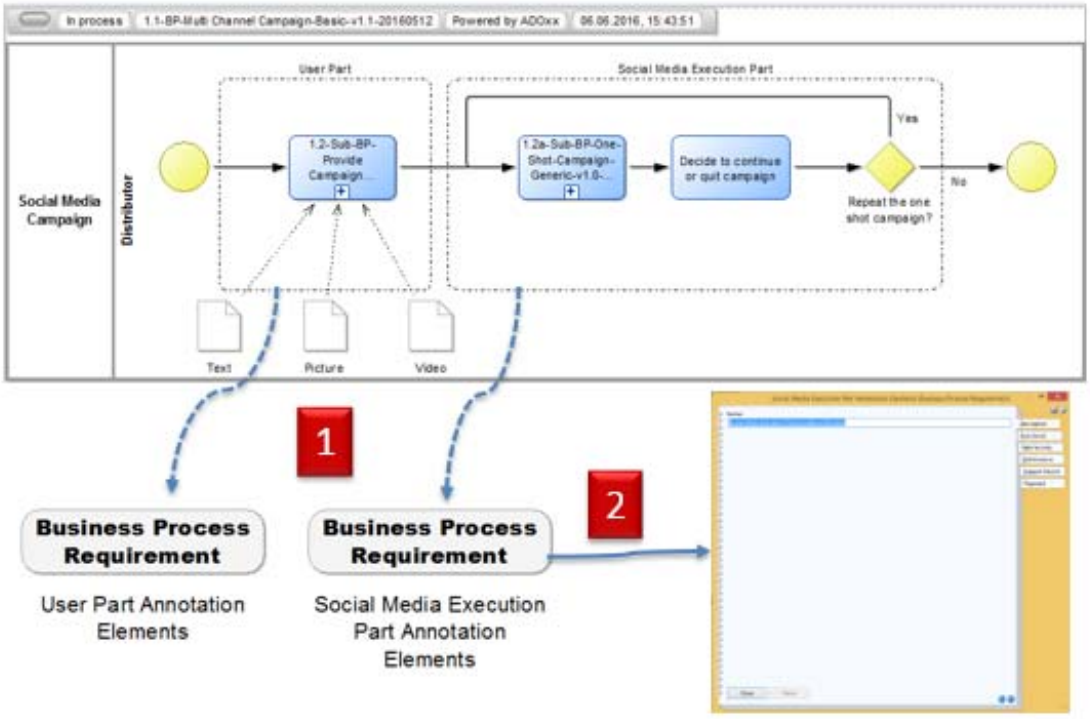


Figure 9: From Group Activity to Annotation Elements

Each Business Process Requirement instance contains:

- a description
- functional requirement specification
- non-functional requirement specifications dealing with Data Security, Performance, Support Service and Payment aspects

The requirements are specified in the notebook of the corresponding element (see bottom right window in Figure 7). Step 2 symbolizes the double click on the Business Process Requirement instance which opens such a notebook. The requirements are further elaborated in the following subsections.

3.3.1.1 Functional Requirements

The functional requirements specify the functionality of a task or a group of tasks. There are three attributes to specify these requirements (see Figure 10).

- With the first attribute, the tasks or groups of tasks are categorised by assigning hierarchies from the APQC Process Classification Framework (APQC 2014)
- Additionally, the semantics of the tasks can be specified by assigning an object and action from a predefined taxonomy.

In the following we describe the principle of the functional requirements specification.

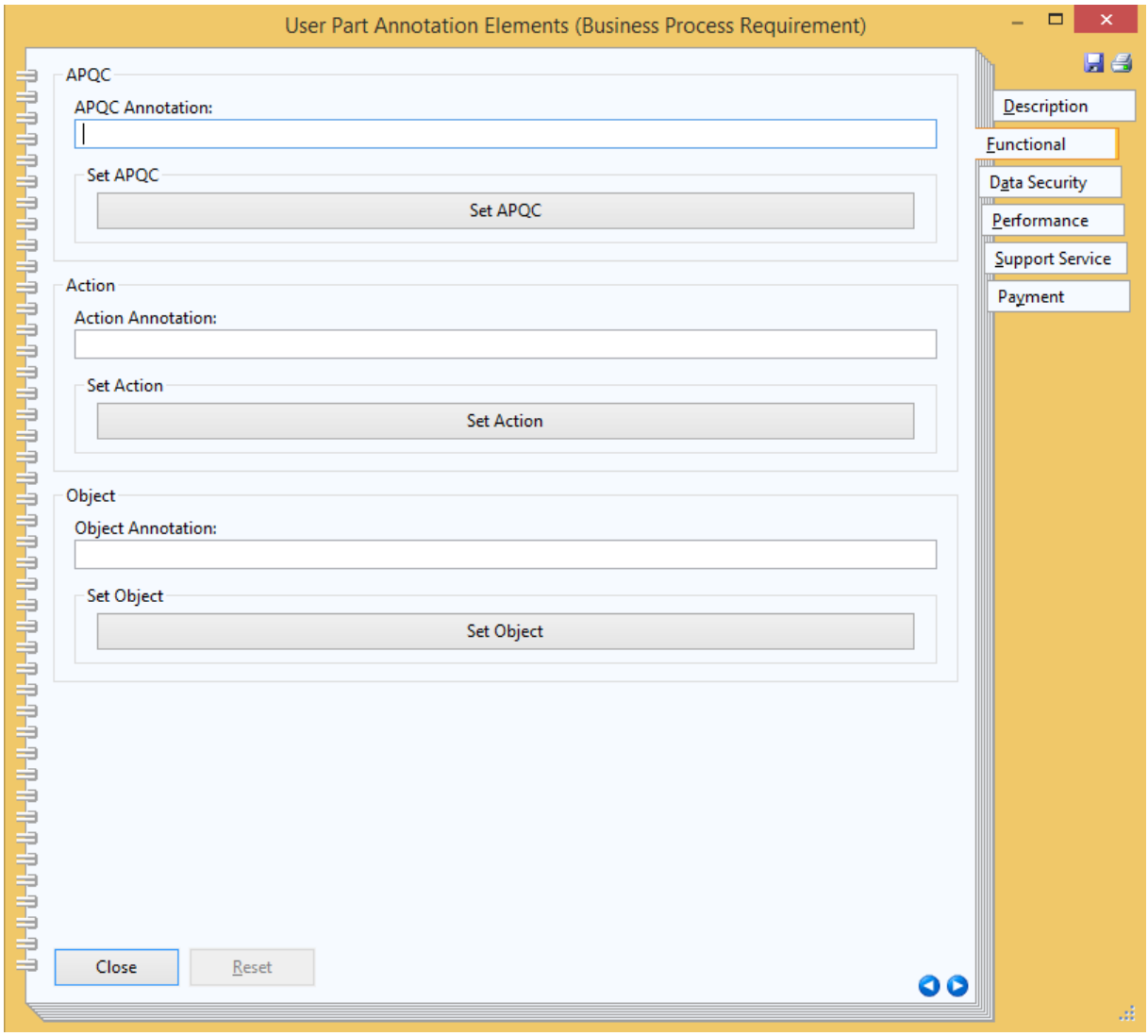


Figure 10: Screenshot to annotate functional requirements

APQC Classification

The APQC Process Classification Framework comprises five levels which start from 13 generic business process categories on top (see Table 1) and goes down to particular tasks. An annotation for a task or a group of tasks can be made on any of the five levels, depending on the granularity of the task or group of tasks.

We demonstrate the APQC annotation with the business Social Media Campaign. Figure 11 depicts the relevant snippet of the business process. The modeller can browse the APQC hierarchy from the top to the bottom until the respective suitable level is selected for making the annotations to the business process parts. It is up to the modeller to find the appropriate level in the hierarchy depending on the granularity of the business process activities.

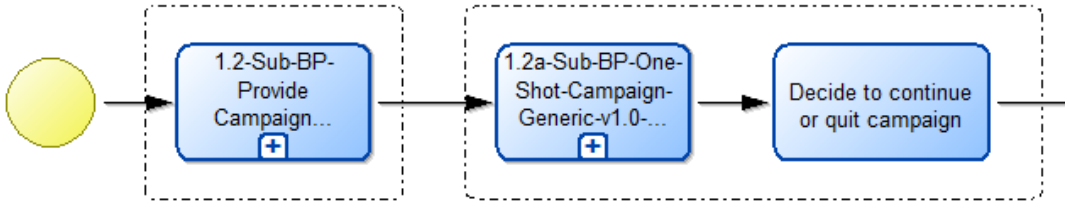


Figure 11: Social Media Campaign

In order to get started an APQC top level category is selected as depicted in Table 1. The business process belongs to the area of marketing, hence in this particular case the category "3.0 Market and Sell Products and Services" is selected. Since this top-level category is too abstract the modeller needs to go deeper in the hierarchy.

1.0	Develop Vision and Strategy
2.0	Develop and Manage Products and Services
3.0	Market and Sell Products and Services
4.0	Deliver Physical Products
5.0	Deliver Services
6.0	Manage Customer Service
7.0	Develop and Manage Human Capital
8.0	Manage Information Technology (IT)
9.0	Manage Financial Resources
10.0	Acquire, Construct, and Manage Assets
11.0	Manage Enterprise Risk, Compliance, Remediation, and Resiliency
12.0	Manage External Relationships
13.0	Develop and Manage Business Capabilities

Table 1: APQC Top Level Categories

In case of the Social Media Campaign it makes sense to select "3.3 Develop and manage marketing plans" (Table 2) and subsequently "3.3.4 Develop and manage promotional activities" as depicted in Table 3.

3.1	Understand markets, customers, and capabilities
3.2	Develop marketing strategy
3.3.	Develop and manage marketing plans
3.4	Develop sales strategy
3.5	Develop and manage sales plans

Table 2: APQC Process Group

3.3.4	Develop and manage promotional activities
3.3.4.1	Define promotional concepts and objectives
3.3.4.2	Develop marketing messages
3.3.4.3	Define target audience
3.3.4.4	Plan and test promotional activities
3.3.4.5	Execute promotional activities
…	…

Table 3: APQC Process and Activities

The fourth level seems to be appropriate in order to enter the respective annotations as shown in Figure 12. The task 1.2-Sub-BP-Provide Campaign Content is annotated with "3.3.4.2 Develop marketing messages". The second group is annotated with "3.3.4.5 Execute promotional activities".

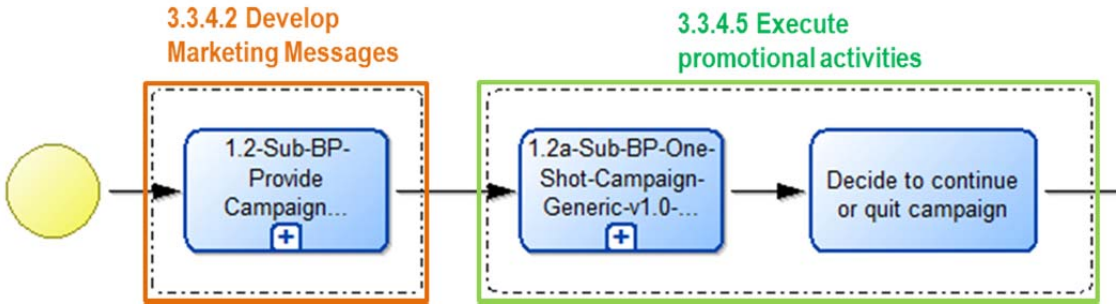


Figure 12: Annotated Social Media Campaign Business Process

In the prototype the navigation through the levels of the APQC classification framework is supported by referencing the BPaaS Ontology. When the user clicks the button "Set APQC" (see Figure 10), the top level categories are retrieved from the ontology and presented to the user (see Figure 13).

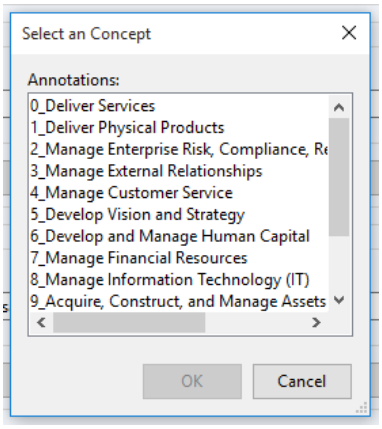


Figure 13: Top Level APQC Classification Retrieved from the BPaaS Ontology

If after selecting the appropriate category, the user decides to go further into detail, another request is sent to the ontology to select the subcategories of the selected category. The details of the ontology and the interface between the modelling environment and the ontology for the semantic annotation are explained in sections 4.2 and 5.2, respectively.

Functional Description

In addition to APQC we introduced the Functional Description Ontology. It consists of an action and an object. This specification is derived from the convention that activities should have names of the form VERB-NOUN (Silver 2011). The verb corresponds to the action and the noun to the object.

While a modeller is free to choose any names for a task, our modelling framework supports the consistent specification of actions and objects by a predefined taxonomy. The taxonomy is represented in the BPaaS Ontology (see section 4.2.2).

The taxonomy allows adding additional information to the business processes or workflows. The taxonomy can be used to represent a domain specific semantic augmentation of the APQC ontology. Hence elements of the object/action taxonomy can be linked to one or several APQC categories. For instance, the APQC category "3.3.4.2 Develop marketing messages" represents the content of a marketing message. By analysing typical online and offline marketing processes we identified a set of marketing message objects and corresponding actions. We applied the same procedure to other APQC categories, such as "3.3.4.5 Execute promotional activities". Additionally, the taxonomy allows specifying semantics of activities for which no appropriate APQC classification can be found.

Figure 14 and Figure 15 show snippets of the Object taxonomy.

Figure 14: Snippet of Object Taxonomy Develop marketing messages

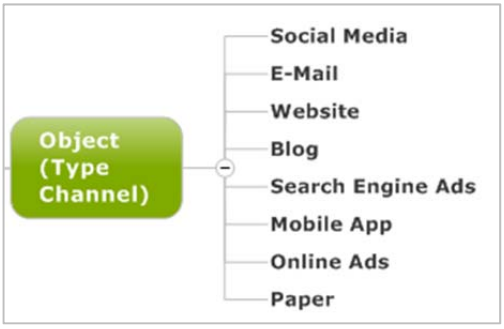


Figure 15: Snippet of Object Taxonomy Execute promotional activities

For the demonstration of the object/action annotation we again use the business process social media campaign. In addition to the general APQC classification, we want to specify that the process allows uploading content (image and text) and publishing it on Facebook and Twitter. Hence, the campaign content is annotated with the objects “Image” and “Text” and the action “Upload”. The execution of the campaign is annotated with the object “Facebook” and “Twitter”. The action is “publish”.

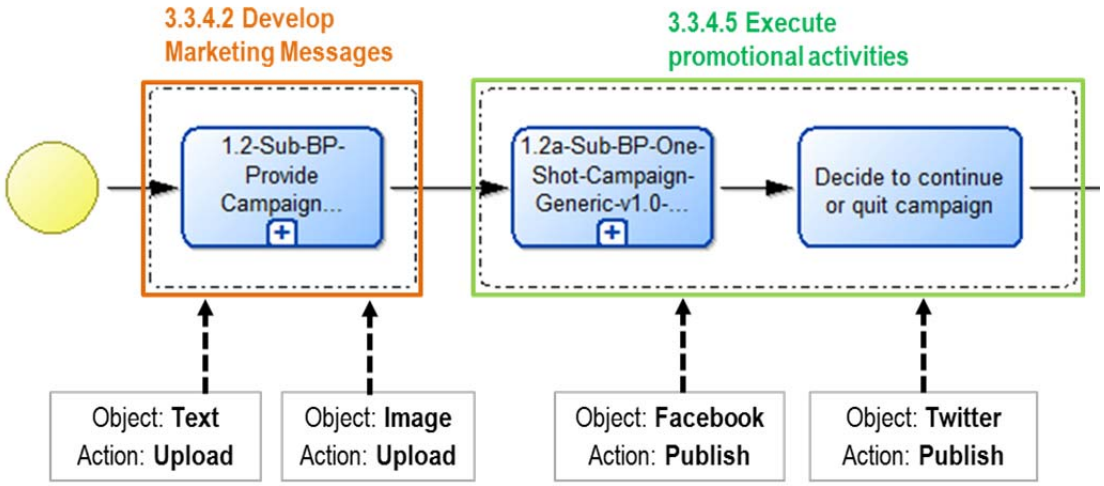


Figure 16: Annotation of Social Media Campaign with Object/Action Taxonomy

As in the case of providing for the APQ-based annotations, the modeller is supported in identifying the appropriate actions and objects by providing a reference to the BPaaS Ontology.

3.3.1.2 Non-functional Requirements

The non-functional requirements and specifications were determined from the Cloud Service Level Agreement Standardisation Guidelines (C-SIG SLA 2014). As already mentioned in the Deliverable D3.1 (CloudSocket 2015b), these guidelines are an outcome of the European 2020 initiative “Digital Agenda for Europe” and has been published in order to standardize and streamline the terminologies and understanding of Cloud Service Level Agreements.

However, the challenge to consider non-functional requirements for the business-speaking world only, required much more than mere guidelines. The representation of functional requirements is done in an iterative approach. While assuming that the business user would not have an IT background, we consulted existing cloud services in marketplace. After expert consolidations within the consortium, there was a workshop prioritizing the concepts. Hence a tight cooperation with the allocation partner FHOSTER was established. The outcome was an initial set of attributes grouped into four categories (represented as *chapters* in the notebook of the modelling environment - see Figure 17), which will be improved iteratively when we gather experiences by using the prototype:

1. Performance:
    - Downtime - refers to the time in a defined period the service is not available. For the business layer we proposed *downtime in minutes per month,* e.g. the service can be down for 500 minutes each month.
    - Media type – refers to the types of media that the user would like to upload in the business process, e.g. uploading video. In this prototype version we consider a fixed size for each type (e.g. for video its 5GB).
    - Number of process executions – refers to the number of times a process will be executed in a defined period. In this version we consider a fixed time expressed in years, e.g. the user enters 50 marketing campaign per year.
    - Number of simultaneous users – refers to the target for the maxium number of separate cloud service customer users that can be using the cloud service at one time, e.g. 5 simultaneous users.
    - Response Time – refers to the time interval between a cloud service customer initiated event (stimulus) and a cloud service provider initiated event in response to that stimulus. For the business layer we proposed three levels of response time – high (<=30 seconds), medium (<=3 minutes and >30seconds) and low (>3 minutes). The mapping among levels and response time expressed in time is done via matching rules (see section 6.2.1).
2. Data Security:
    - Backup retention time – refers to the length of time in which the cloud service provider will retain backup copies of the cloud service customer. Our solution enables the business user to enter values in terms of number of weeks, months or years.
    - Backup frequency - refers to the period of time between complete backups of cloud service customer data. In our solution the business user can perform an exclusive selection among the the following values: at least daily, at least weekly, at least monthly, at least yearly.
    - Backup restore time - refers to the committed time taken to restore cloud service costumer data from backup. In our solution the business user can enter the restore time expressed in minutes.
    - Data location - refers to the location of processed and stored data. Location can be either continent (e.g. Europe rather than Asia), or nation (Switzerland rather than Spain), or even more detailed like regions or cities. In order to keep it simple for the business user, our solution enables the user to choose one data location. This will then be compared to both processed data and stored data location that are annotated in the workflow. E.g. if the user chooses Europe as data location, all workflows that have stored data location and processed data location from European countries will be retrieved. *(Related matching rule not implemented yet)*
    - Data encryption level – relates to the level of the discipline that embodies principles, means and methods for the transformation of data in order to hide its information content, prevent its undetected modification and or prevent its unauthorized use. In the business level layer, the user can choose among a high, medium or low level. Similar to response time, for each level we assigned one or more encryption types. Encryption type appear in the workflow leayer only, E.g. the encryption types Sha-256, Blowflish are considered high.

3. Service Support:
    - Support availability - it refers to the time period an interface is made available by the cloud service provider to handle issues and queries raised by the cloud service customer. E.g. support in weekend, support Monday to Saturday, support Monday to Friday, support 24/7, start hour and end hour of requested support or support not required at all.
4. Payment:
    - Payment plan - it refers to the preferred payment plan(s) the user would like to deal with, e.g. a free of charge plan, monthly fee, prepaid annual plan, try free first and/or a customizable plan.
    - Target Market – it refers to addressed user target of the cloud service. The user can choose the target market he/she fits in (a.k.a. user target), e.g. Businesses, Schools and Universities, growing teams, small teams, individuals, or not specific target.

These elements together with the related chapters were added in the Business Process Requirement notebook. Figure 17 shows one example of how the *performance chapter* looks like in the ADOxx modelling environment.

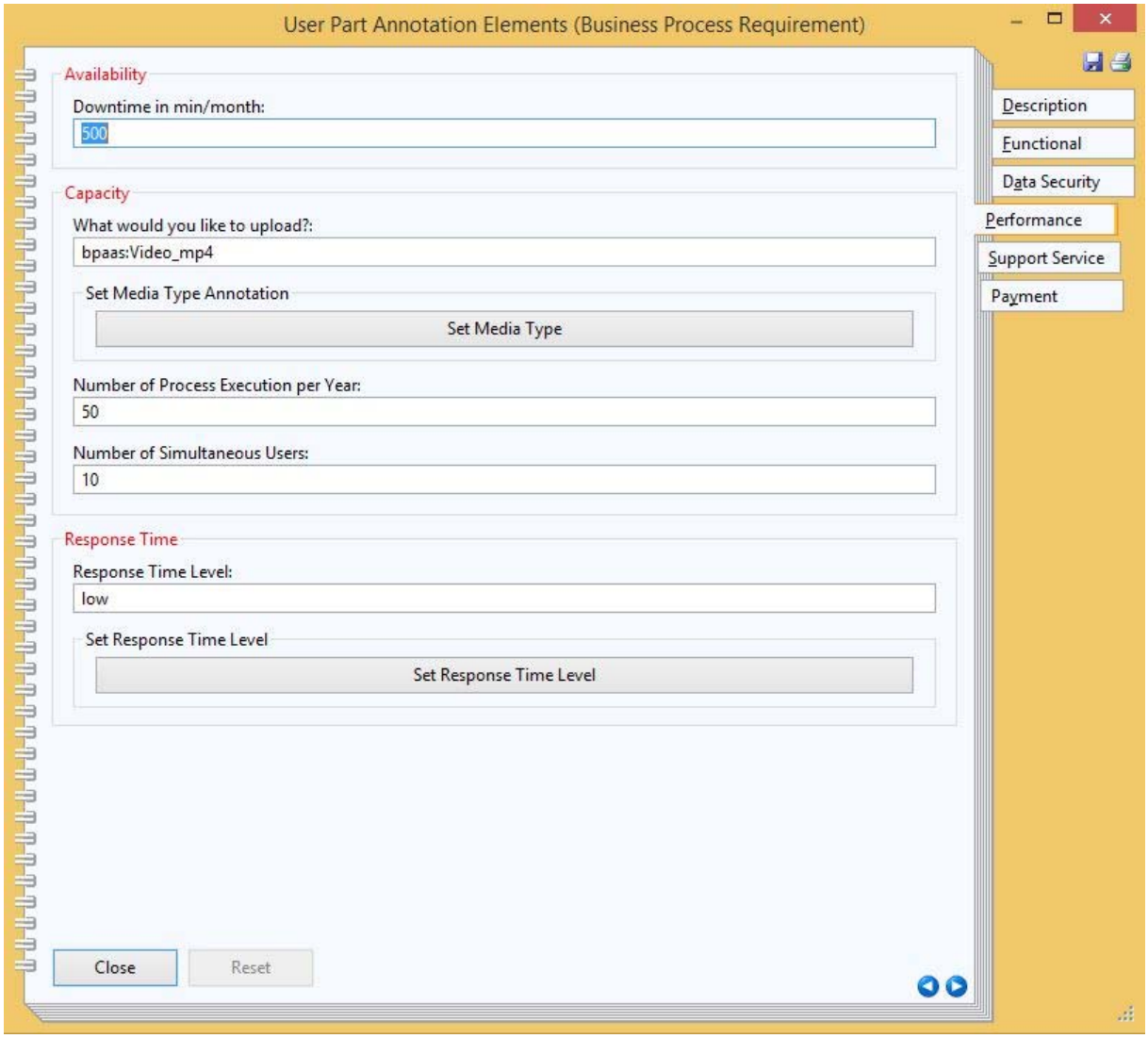


Figure 17: Performance Chapter in the Business Layer

Some elements like downtime in minutes per month are entered manually by the user, while others like media type are selected from a list of values defined in the BPaaS Ontology. The list appears after clicking on the related button (e.g. Set Media Type button in Figure 17). This approach recalls the semantic lifting technique

described in Chapter 5.2, where the human-interpretable models are enriched with semantics in order to support business and IT alignment with smart technology.

3.3.2 Modelling Workflow Descriptions

While in the business layer we have requirements, in the workflow layer we talk about descriptions. In fact, if a workflow matches with a business process means that the annotated requirements of the former are satisfied by the annotated descriptions of the latter. The Service Description Model enables the user to express functional and non-functional properties of the workflow layer.

3.3.2.1 Functional Workflow Description

The annotation of the functional description used for the workflow is the same as for the business process (see section 3.3.1.1).

3.3.2.2 Non-functional Workflow Description

It is assumed that the business process requirements are specified with attributes that can be declared by a business user, while the workflow requirements reflect a more technical language. Therefore on the workflow level, some of the non-functional descriptions are different from the business process requirements.

The mapping between non-functional business process requirements and non-functional workflow descriptions is done by matching rules, which are part of the smart business and IT alignment (see section 6.2). The matching is not always a 1:1 correspondence. It can happen that one or more elements in the business process requirements can be mapped to one or more non-functional descriptions, hence we claim that their relationship is *n:m.*

Following we describe the list of non-functional descriptions and how they relate to the requirements in the business layer (see section 3.3.1.2).

1. Performance
   - Percentage of uptime per month - describes the time in a defined period the service was available, over the total possible available time, expressed as a percentage.
   - Capacity – it is expressed in terms of both hardware and network. For hardware capacity we consider *the maximum available data storage*.
     For network capacity, capacity is classified as the *maximum simultaneous connections* and *maximum simultaneous service users,* e.g. 4000 maximum simultaneous connections and 2000 service users connected.
   - Response time is measured as *max average response time* that refers to the time that is required between the requests from the stakeholder to the system and the receipt, elaboration, execution and response of the request. It is expressed in milliseconds, i.e. 500 ms.
2. Data Security:
   - Backup management (retention, frequency and restore time) are the same as in the business layer. The relation with the related requirement for each of them is *1:1, e.g.* an access period description with value *daily* won't match with an access period requirement with value *weekly*. The workflow user can choose more than one the backup frequency as it is assumed it's driven by a backup strategy.
     Additionally, backup has the following technical information:
     - *Restoration Time* - expresses the time required in order to revert to a previous situation from an existing backup, i.e. 15 minutes.
   - Type of Encryption – refers to which kind of encryption is applied in the memorization of data e.g. SHA-1 (medium level in the encryption level of the business layer) or SHA-256 (high level in the encryption level of the business layer). In this example, the mapping will choose only workflows with encryption SHA-256, if the business requires high encryption.

- Processed, and stored data location - refer to the location where data are processed, archived and stored. e.g. continent *(e.g. Europe rather than Asia), or nation (Switzerland rather than Spain), or even more detailed like regions or cities.* The processed data location can be used to identify which laws are applied for the execution of data.

3. Service Support:
   - Service support availability, the same as in the business layer (see above). If the value in the business layer is covered by the description, there is a match. E.g. a description of 24/7 would cover a service support requirement Monday to Friday.
4. Payment
   - Payment plan, the same as in the business layer (see above). The relation with the related requirement is *1:1,* e.g. a description with value *monthly fee* would match with a requirement with value *monthly fee.*
   - Target Market, the same as in the business layer (see above). The relation with the related requirement is *1:1,* e.g. a requirement annotation element with value *Schools and Universities* would match with the the description annotation element with value *School and University.*

Figure 18 depicts the *performance chapter* of the workflow description requirement that relates to the lane distributor (see first lane of Figure 8). In this application scenario all element values are inherited by the service Jira. In chapter 6 we describe how element values of the business layer can match with element values of the workflow layer via rules.

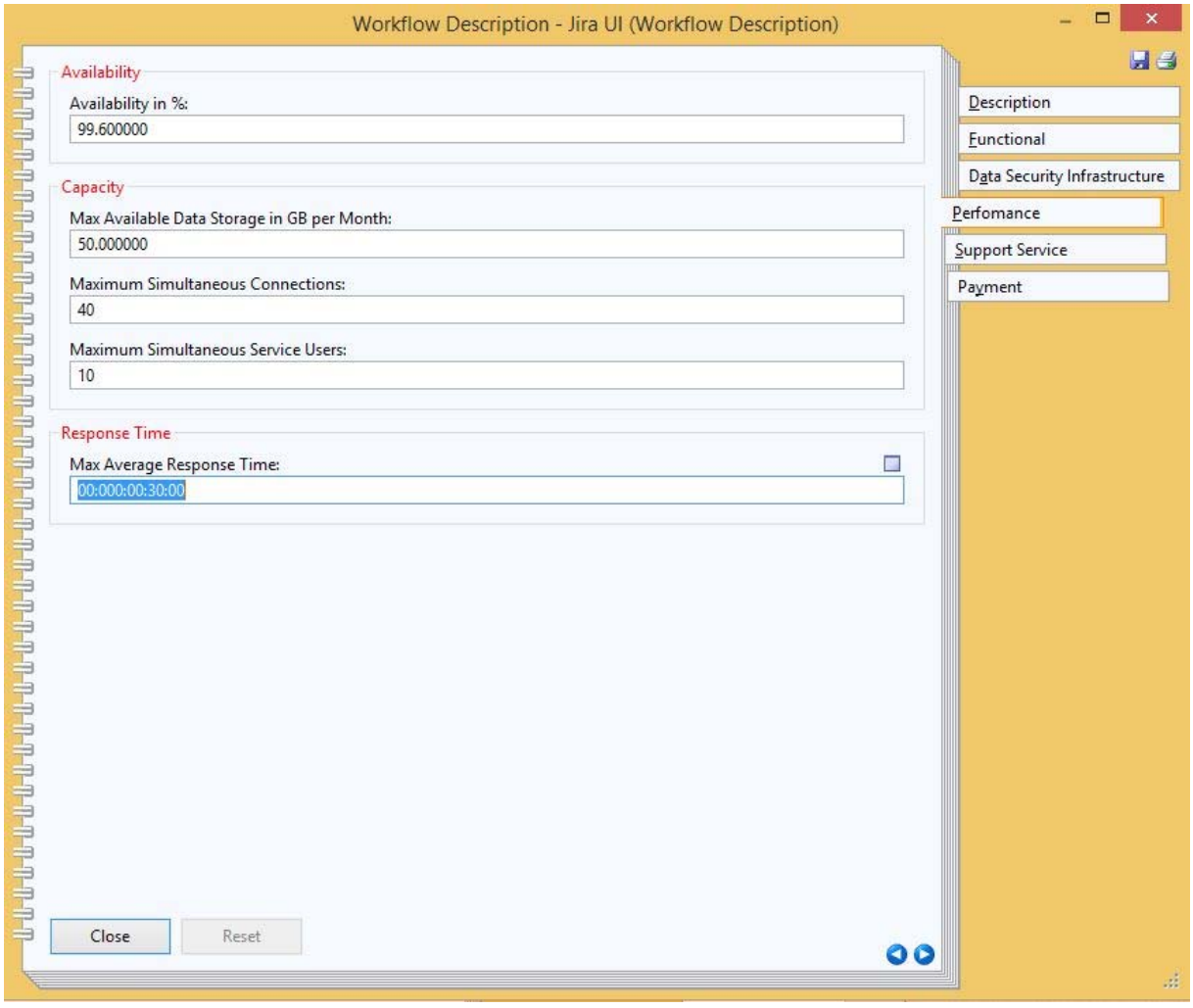


Figure 18: Performance Chapter in the Workflow Layer

4 ONTOLOGY PROTOTYPE

This chapter describes the prototype for the BPaaS Ontology (see Figure 19). The rationale and main concepts of the BPaaS Ontology have been described in Deliverable D3.1 (CloudSocket 2015b). Section 4.1 gives an overview of the BPaaS Ontology and its structure. Section 4.2 and 4.3 then focuses on the concepts and ontologies for describing the cloud service requirements.

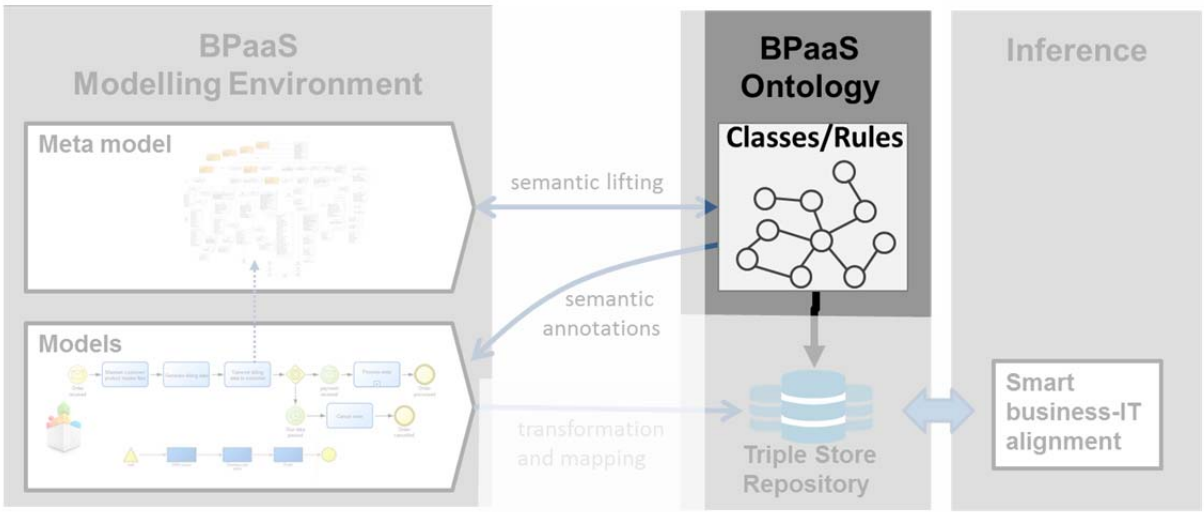


Figure 19: The BPaaS Ontology as part of the BPaaS Design Environment

4.1 BPaaS Ontology

Figure 20 depicts the BPaaS Ontology, which is an extension of the ArchiMEO ontology. The BPaaS Ontology includes all concepts required to describe cloud specific requirements in order to achieve a smart alignment of business and IT. The cloud-specific extensions were determined from the analysis of the business scenarios and the competency questions and are describe in detail in Deliverable D3.1 (CloudSocket 2015b).

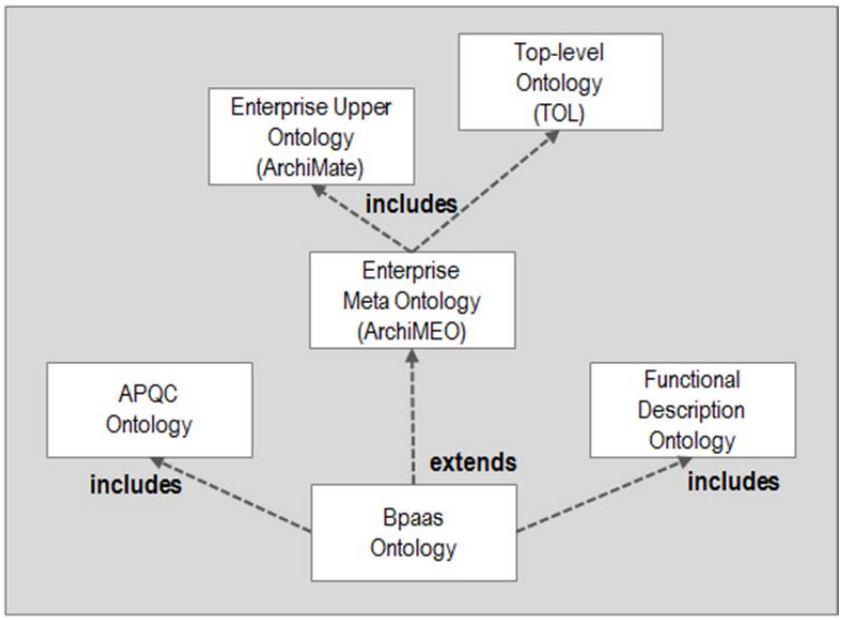


Figure 20: Extending ArchiMEO with the BPaaS Ontology

In order to support the semantic augmentation of the business process models and workflows, we introduce the concept of functional and non-functional requirements. While functional requirements describe what a system should do (user perspective), the non-functional requirements describe how the system should behave (technical perspective). For instance, security, availability or scalability are non-functional requirements. However, the requirement “conducting a marketing campaign” is a functional requirement since it explains the functionality. The BPaaS Ontology includes both functional and non-functional requirements. Moreover, it is extended by the APQC ontology and functional Business Process description ontology which are used in context of the functional requirements. The APQC ontology reflects the structure of the APQC Process Classification Framework (see section 4.2.1), which is complemented by the functional business process description ontology (see section 4.2.2).

The classes of the conceptual model for BPaaS are integrated in the ArchiMEO ontology using the new namespace "bpaas". This is done by identifying the already existing classes in ArchiMEO, which correspond to the classes of the conceptual model. These classes are extended with relations and properties identified in the conceptual model. If for a class in the conceptual model no corresponding class is available in ArchiMEO, a new class is created and integrated in the appropriate part of the ArchiMEO class hierarchy.

4.2 Functional Requirements Ontology

The semantics of functional requirements are represented by two ontologies that complement each other: the APQC ontology and the Functional Business Process ontology (object/action). Both ontologies are used for the annotation of single activities (tasks) and/or groups of activities (tasks) as depicted in Figure 21.

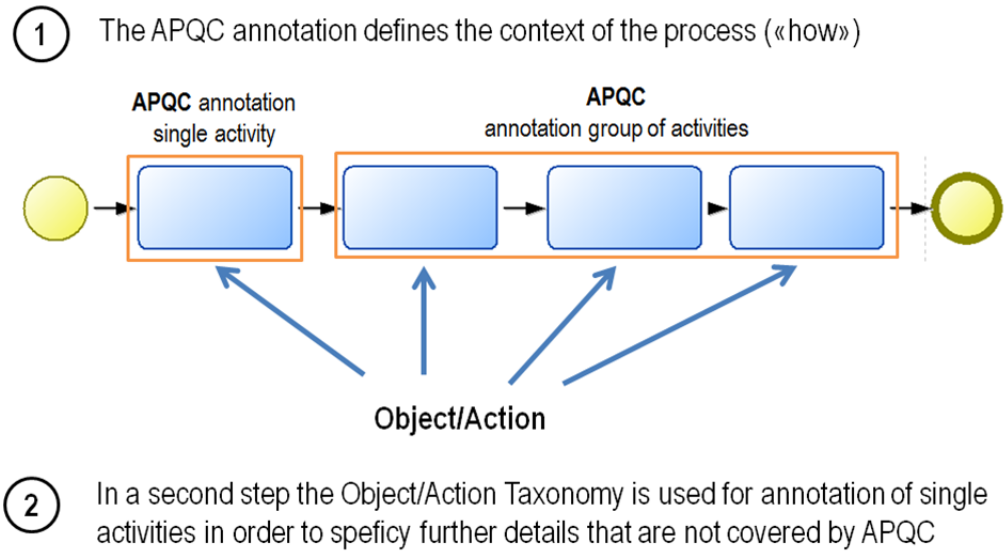


Figure 21: Functional Requirements Annotation

4.2.1 APQC Ontology

The APQC Process Classification Framework comprises five levels which start from 13 generic business process categories on top and go down to particular tasks (see section 3.3.1.1). Figure 22 depicts the ontological representation of APQC.

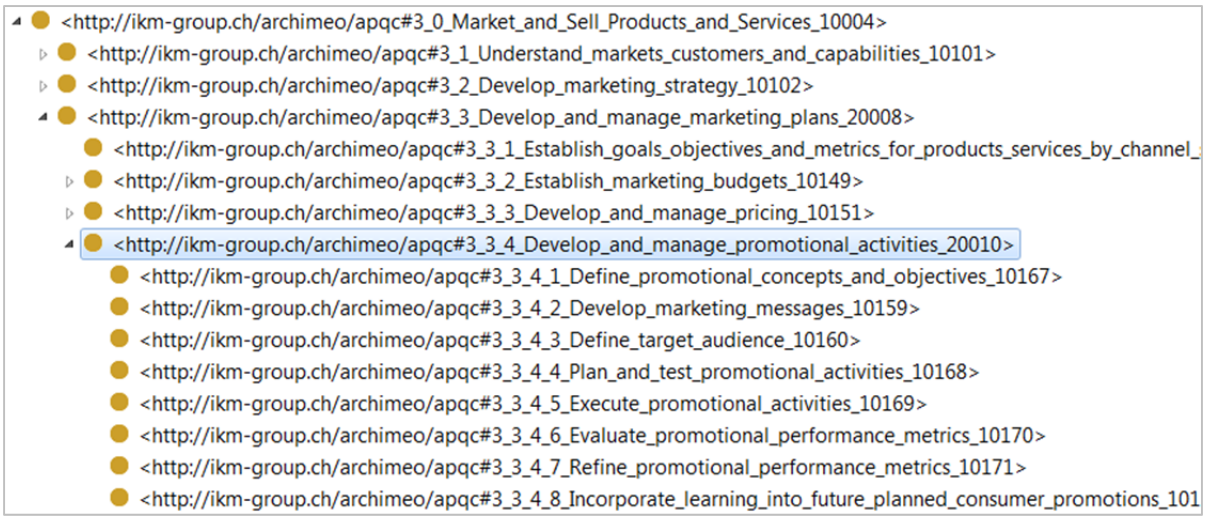


Figure 22: APQC Ontology

4.2.2 Functional Description Ontology

In addition to APQC we introduced the Functional Description Ontology. It consists of an action and object taxonomy and allows adding additional information to the business processes. Figure 23 and Figure 24 depict the ontological representation of the object/action taxonomy.

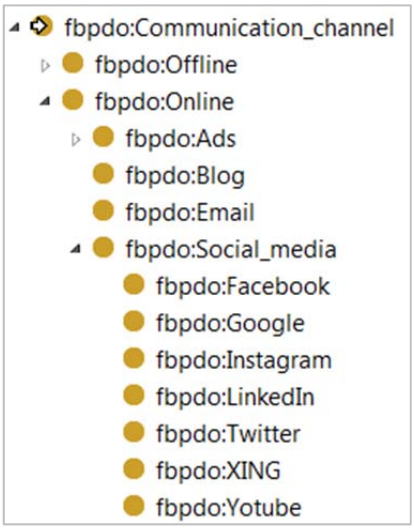


Figure 23: Snippet of Object Taxonomy

Figure 24: Snippet of Action Taxonomy

4.3 Integrating Non-functional Elements in BPaaS Ontology

Since the non-functional requirements are mainly based on the Cloud Service Level Agreement Guidelines (C-SIG SLA 2014), we built on the meta-model of the business that is described in the Deliverable D3.1 (CloudSocket 2015b) and partially depicted in Figure 25. In order to semantically enrich the non-functional requirements, we extended the Business Model Motivation Meta Model already integrated in our ArchiMEO ontology. A requirement is generally seen as a motivation element, therefore we created a sub-class of the concept Motivation Element - the “Business Process Requirement” class (see Figure 25).

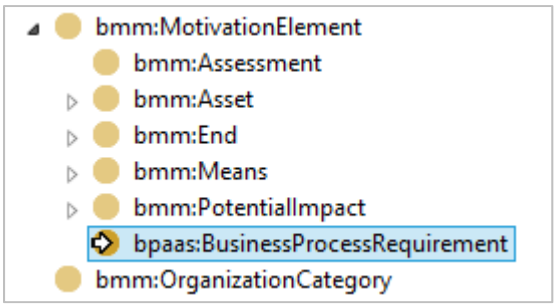


Figure 25: Business Process Requirement Class

Workflow descriptions on the other side reflect annotation elements of an application services. Therefore we created the class "Workflow Description" as sub-class of the "Application Service" class (see Figure 26).

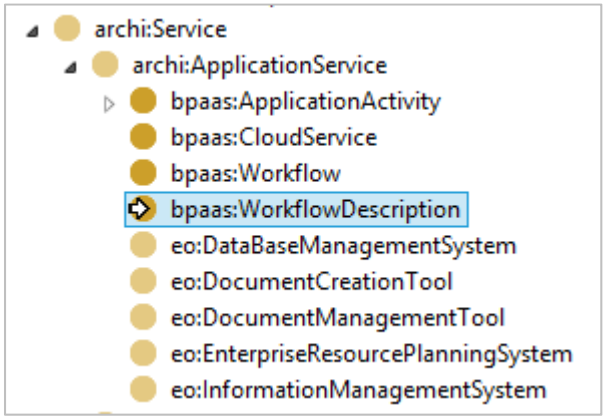


Figure 26: Workflow Description Class

All the annotation elements listed in section 3 are in the form of either object properties or data type properties. More in detail, all those elements that are inserted manually like downtime per minutes or number of process execution were added as datatype property, while those who are selected via web service were modelled as object properties, e.g. media type or the encryption level (see Figure 27 and Figure 28), where some properties of the two above mentioned classes are depicted, respectively).

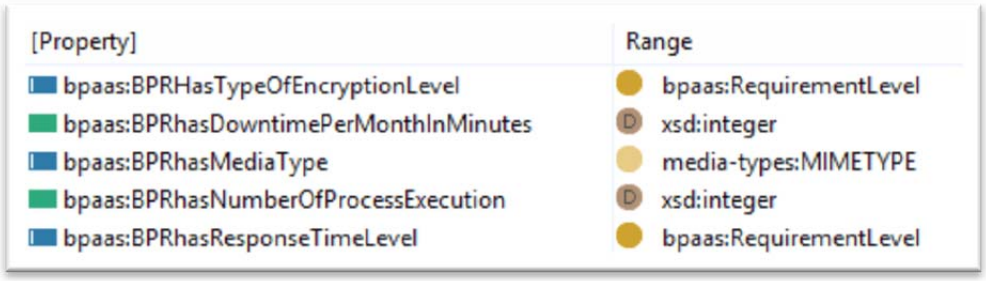


Figure 27: Some Properties of the Class "Business Process Requirement"

[Property]	Range
bpaas:WFDhasDataProcessingLocation	top:Location
bpaas:WFDhasDataStorage	xsd:decimal
bpaas:WFDhasDataStorageLocation	top:Location
bpaas:WFDhasTypeOfEncryption	bpaas:Encryption
bpaas:WFDhasTypeOfRaid	bpaas:TypeOfRaid

Figure 28: Some Properties of the Class "Workflow Description"

Object properties point to classes that are populated by instances called from the web service. For example, the Requirement Level class has the three instances – high, medium and low. Having an object property in the ontology enables the aforementioned semantic alignment by means of web service call. In order to better describe this approach, let's take into account the example in Figure 29. When the user clicks on the button to set an annotation element value, the web service is called (step 1). Next, the web service returns the appropriate class, e.g. MediaType. At this point, the user can navigate through the instances of the class and choose the wanted one, e.g. Video mp4 (step 2). As soon as the user choses the annotation value, it will appear on the related textbox (step 3).

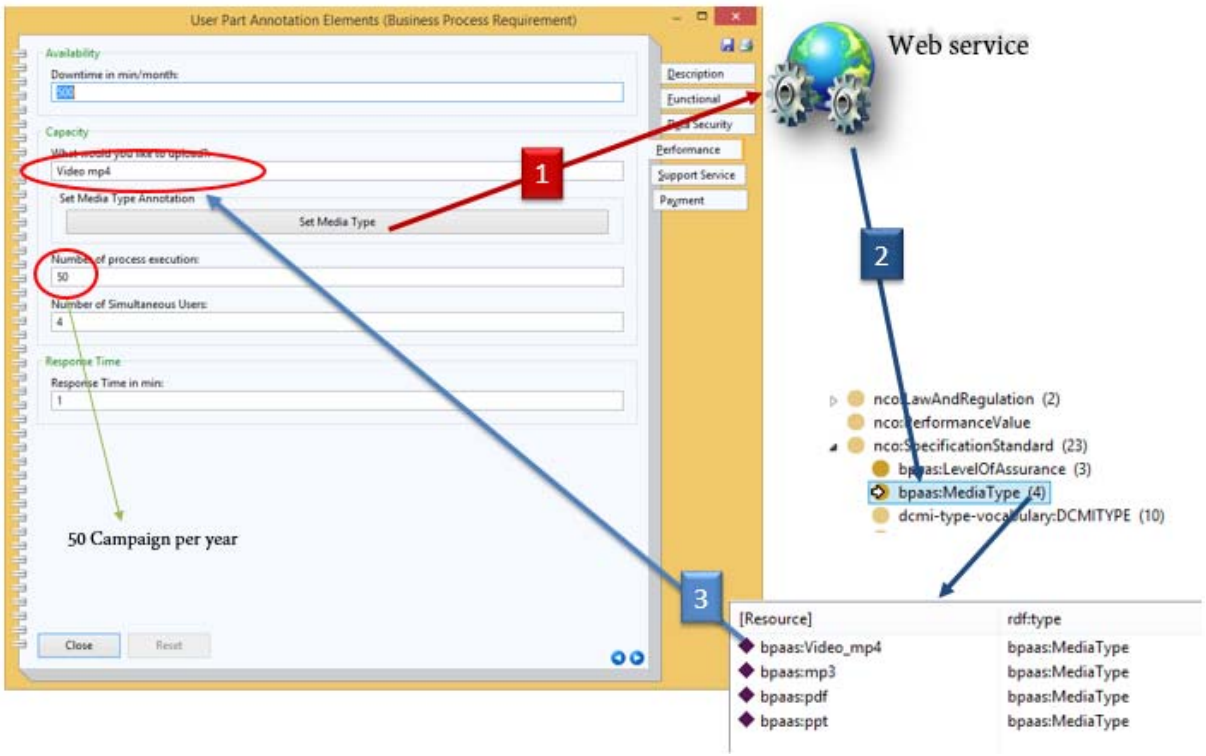


Figure 29: Semantic Alignment from the Modelling Environment

Details of the alignment implementation are described in section 5.2.

5 SEMANTIC LIFTING OF BPAAS MODELS

Merging meta models and ontologies enable the introduction of smart semantic technologies when using conceptual meta models like business process or workflow models. In CloudSocket the human-interpretable models are enriched with semantics in order to support business and IT alignment with smart technology. This paragraph describes the overall process, split into the parts

- Semantic alignment of meta models
- Semantic annotation
- Transformation

First an introduction is provided into the topic, before starting with the elaboration of the different technical solutions.

5.1 Semantic Alignment of Meta Models

Models are human interpretable but difficult for systems to interpret and process. Therefore, models follow a defined modelling language that defines the (1) notation, (2) syntax, and (3) semantics (Karagiannis & Kühn 2002). The meta model contains all of these information, which is represented in structures (taxonomy), classes, attributes and relations. Figure 30 shows the placement in the overall picture.

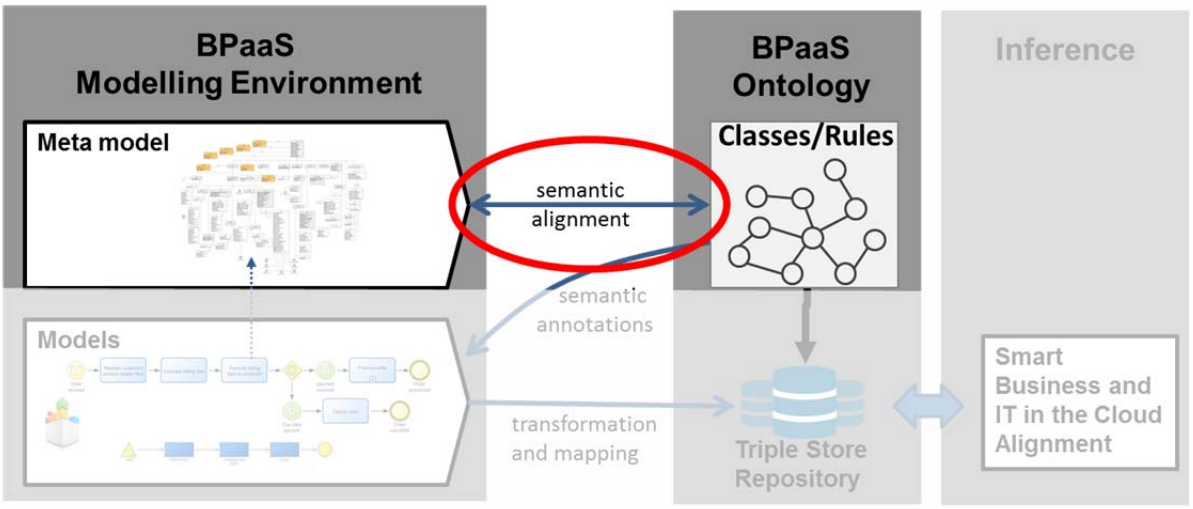


Figure 30: Semantic Alignment on the Meta-Model Layer

In the context of CloudSocket, the ontology has been extended according to the application scenario of Business Processes and Workflows. As described in chapter 4, ArchiMEO (Enterprise Meta Ontology) has been extended by the Business Process as a Service Ontology. The Business Process as a Service (BPaaS) Ontology contains concepts and relations that can express the meta model and vice versa.

The ontologies can be publicly accessed using the following links:

- APQC: https://github.com/BPaaSModelling/APQC-Ontology/blob/master/apqc.ttl
- BPAAS: https://github.com/BPaaSModelling/BPaaS-Ontology/blob/master/bpaas.ttl
- Functional Business Process Description: https://github.com/BPaaSModelling/Functional-Business-Process-Description-Ontology

5.2 Semantic Annotations

Semantic annotations allow the human modeller to categorize and classify elements by adding semantic. The overall context of this mechanism is outlined in Figure 31.

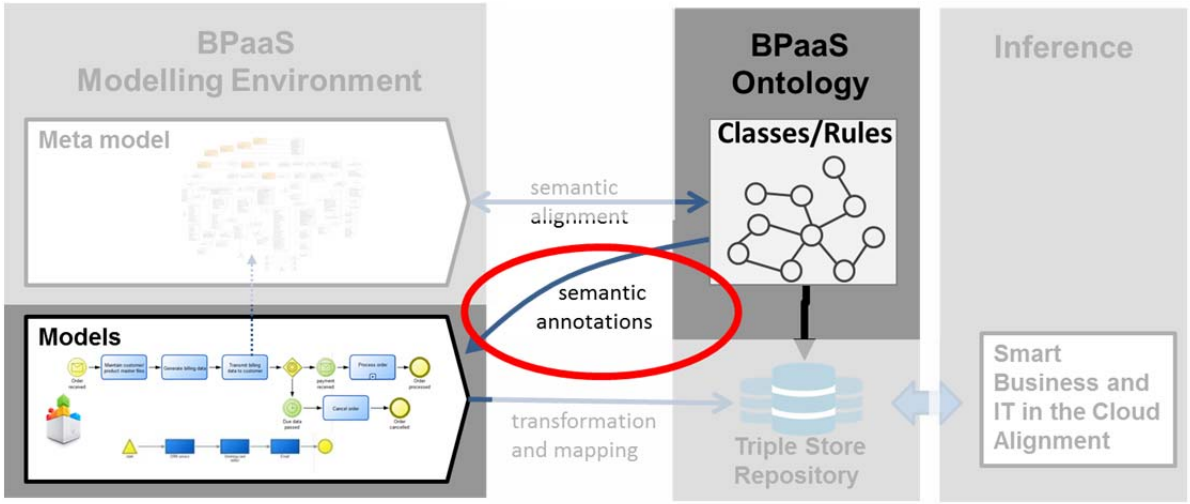


Figure 31: Semantic Lifting of Models by Semantic Annotations

There are seven different ways of implementing semantic lifting for weaving between the different modelling layers as described in Deliverable D3.1 (CloudSocket 2015b). In the prototype we implemented the deep integration with web service.

5.2.1 Conceptual Description

The web service creates and maintains the link to the different ontologies. The modelling environment calls the web service for a concept by providing the context and the web service returns the resulting classes and instances, depending on the context. The communication flow is shown in Figure 32.

Figure 32: Modelling Environment - Web Service Communication

The modelling environment receives at the beginning the link to the web server. While the human models the diagram, the properties can be set. Properties that need to receive a conceptual annotation, provide the possibility to make a call to the web service (1). The web service receives the request containing contextual information and queries the ontology accordingly (2). The result set is returned (3) and being processed by the web service according to the interface description to the modelling environment. Once the processing is done, the web service returns the enriched result set (4) to the modelling environment that makes sure that the annotation is placed at the right place in the meta data of the modelling element.

5.2.2 Concrete Application

Figure 33 shows a notebook of the Business Process Requirement Element. The ADOxx notebook is the user interface for maintaining the meta data of modelling elements. This requirement receives functional annotations, such as the relational APQC process classification. Since the user might not know the APQC process names by

heart, the user clicks on the “Set APQC” button and triggers the web service request. The request includes the context, which is in this case APQC.

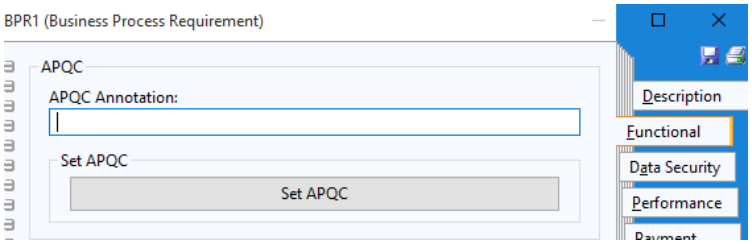


Figure 33: Model Element Notebook - Unannotated

The web service receives the request and queries the ontologies for sub-concepts (first level) of “AmericanProductivityAndQualityCenter” at the namespace “apqc”. The result set of the query is converted into the interface requirements of ADOxx and returned to the modelling environment.

ADOxx receives the results and converts it into the selection box. The user can select the item that fits to the requirement that is being specified. The selection is outlined in Figure 34.

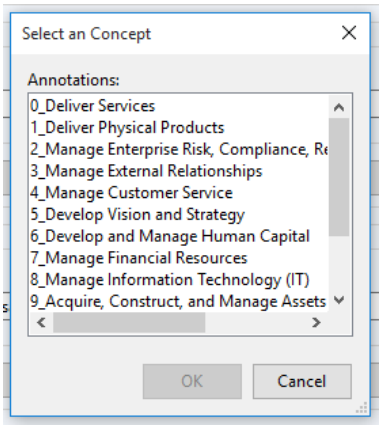


Figure 34: Selection Box Showing the Results of the APQC Web Service Request

After the user selected the best fitting APQC categorization, the modelling environment asks the user for querying the next level. This question is shown Figure 35.

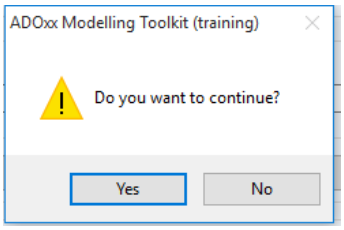


Figure 35: Multi Level Selection User Dialog

If the user decides to increase the granularity of the annotation, “yes” needs to be clicked, if not, the user can stop on the current level by choosing “no”.

If yes is clicked, the modelling environment calls the web service again and takes the current selection as new context.

If no is selected, the latest selection is transferred as value into the "APQC Annotation" attribute of the Business Process Requirement notebook. The annotation result is shown in Figure 36.

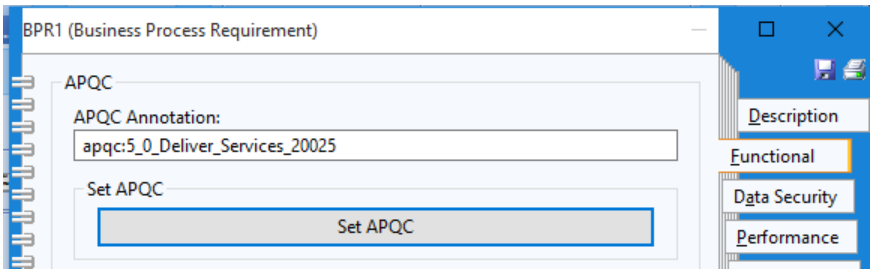


Figure 36: Model Element Notebook - Annotated

While the selection is focussed on human interpretation, the annotation is technical by showing the category, name of the process and the process ID.

6 SMART BUSINESS AND IT ALIGNMENT

6.1 Model Transformation

In order to enable smart business and IT alignment, a transformation is implemented, which creates a formal representation of the graphical and script-based models. In the overall picture, this part is described as transformation and mapping (Figure 37).

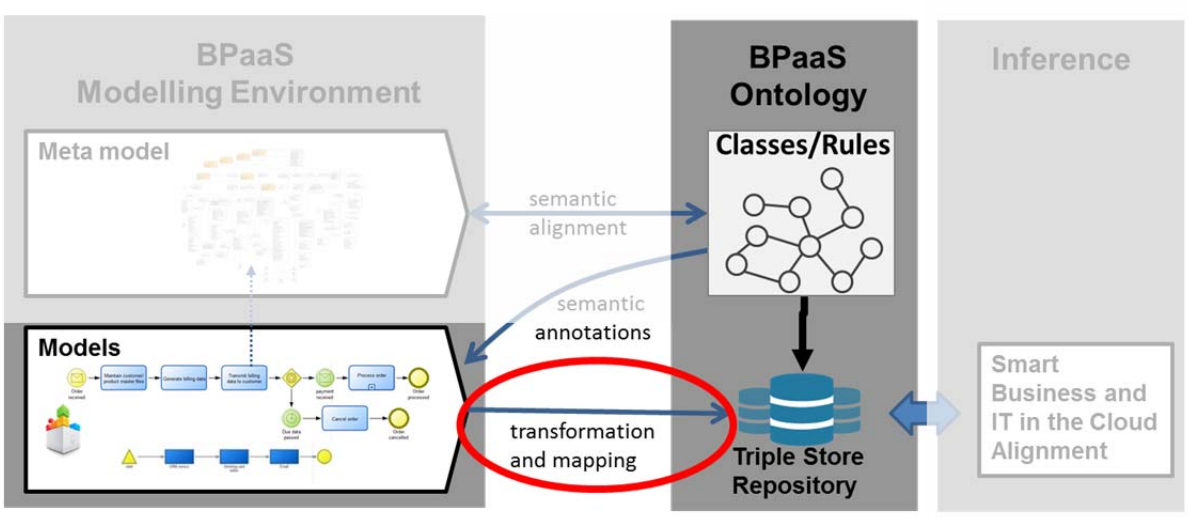


Figure 37: Transformation and Mapping

6.1.1 Conceptual Description

ADOxx.org exports the annotated models into an Extensible Markup Language (XML) formatted description. The EXtensible Stylesheet Language (XSLT) engine transforms the XML and creates instances of classes and relations among the instances defined in the BPaaS ontology. These instances together with the class definitions are added to a triple store, where the instances can be queried and rules (inferences) can be applied.

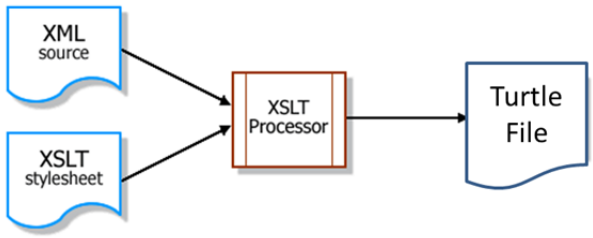


Figure 38: Transformation Concept

Figure 38 shows the conceptual setup of the transformation and includes the following four elements for the transformation:

1. XML Source:
2. XSLT stylesheet
3. XSLT Processor
4. Turtle file

The XML source contains data in a semi-structured format.

The XSLT stylesheet contains templates and descriptions for parsing and processing the XML source. It is structured according to the export format.

The XSLT processor is the engine that applies the stylesheet against the XML source and outputs the results in a new file. In this case it's a turtle file format.

6.1.2 Concrete Application

The used mechanism is adapted from the transformation approach of the LearnPAd project (Emmenegger et al. 2016) as shown in Figure 39.

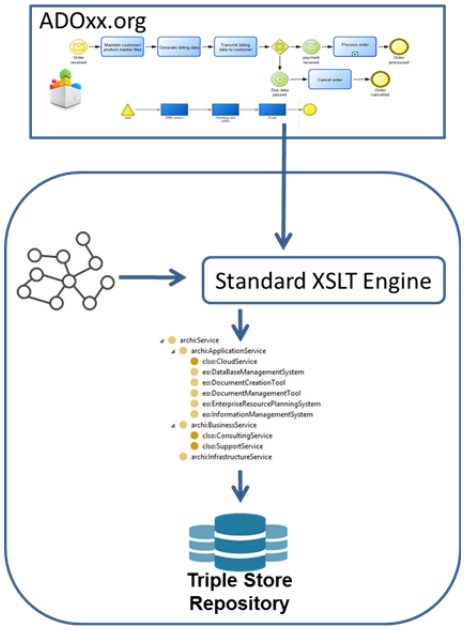


Figure 39: Transformation into Ontology

The ADOxx XML source file contains all the objects and relations that have been modelled and annotated using the web service. The XSLT engine applies the XSLT stylesheet on the XML source file and outputs the data based on the taxonomy and concepts of the ontology in a turtle file format. This file and format is compatible with triple stores that allow further processing.

Taking the annotation example of the previous paragraph, the XSLT stylesheet parses the XML source file for the Business Process Requirements Model. Figure 40 shows the underlying parsing statement.

```
…
<xsl:apply-templates
        select="//MODEL[@modeltype='Business Process Requirements Model']"
        mode="BPRequirementsModel" />
…
```

Figure 40: XSLT Model Type Parsing

The parsing point jumps to the defined model which is indicated to the name "BPRequirementsModel". Since the model type contains just one object, it parses the Business Process Requirement using the "BusinessProcessRequirement" template, as shown Figure 41.

```
<xsl:template match="MODEL" mode="BPRequirementsModel">
<xsl:apply-templates        select=".//INSTANCE[@class='Business Process Requirement']"
```

```
                    mode="BusinessProcessRequirement" />
</xsl:template>
```

Figure 41: XSLT Instance Parsing

The pointer arrives at the "BusinessProcessRequirement" template and starts parsing the single attributes and meta data. A basic example is given in Figure 42.

```
<xsl:template match="INSTANCE" mode="BusinessProcessRequirement">
data:<xsl:value-of select="@id"/>
  rdf:type bpaas:BusinessProcessRequirement ;
  rdfs:label "<xsl:value-of select="@name"/>"^^xsd:string ;
  <xsl:if test="./ATTRIBUTE[@name='Downtime in min/month'] !="">
  bpaas:hasDowntimePerMonthInMin:<xsl:value-of select="./ATTRIBUTE[@name='Downtime in min/month']"/> ;
</xsl:if>
<xsl:if test="./ATTRIBUTE[@name='APQC Annotation'] !="">
  rdf:type <xsl:value-of select="./ATTRIBUTE[@name='APQC Annotation']"/> ;
</xsl:if>
.
</xsl:template>
```

Figure 42: XSLT Attribute and Meta Data Parsing

"data" represents the namespace of the "bpaas" data and receives the object ID given by ADOxx. Since this element is of the type Business Process Requirement, it gets associated to the concept "bpaas:BusinessProcessRequirement". The label receives the name that was given during the modelling process. There are two different handlings of annotations.

- Annotation without web service
- Annotation with web service

The annotation without the web service is used for number values and Booleans. Figure 42 outlines the example with the attribute name "Downtime in min/month". It receives just an integer value, defined in the meta model as well as in the ontology. The sounding IF-clause makes sure that a value has been entered. If no value has been entered, this attribute and the ontological association will be ignored in order to prevent corrupted data sets.

The annotation with the web service is used for example for the APQC annotation. The stored meta data is extracted the same way as for the name "Downtime in min/month".

The result of the transformation is shown in Figure 43.

```
data:obj.126998
  rdf:type bpaas:BusinessProcessRequirement ;
  rdfs:label "BPR1"^^xsd:string ;
  bpaas:hasDowntimePerMonthInMin 2 ;
  rdf:type apqc:5_0_Deliver_Services_20025 ;
.
```

Figure 43: Example Transformation Output

The same applies to the other elements such as workflow descriptions.

6.2 Smart Business and IT Alignment (Matching)

The smart business and IT alignment in the cloud identifies appropriate workflows for business processes based on the specification of the business process requirements and the workflow descriptions. It relies on a set of mapping rules. The rules and the implementation of the mapping are defined in the following subsections.

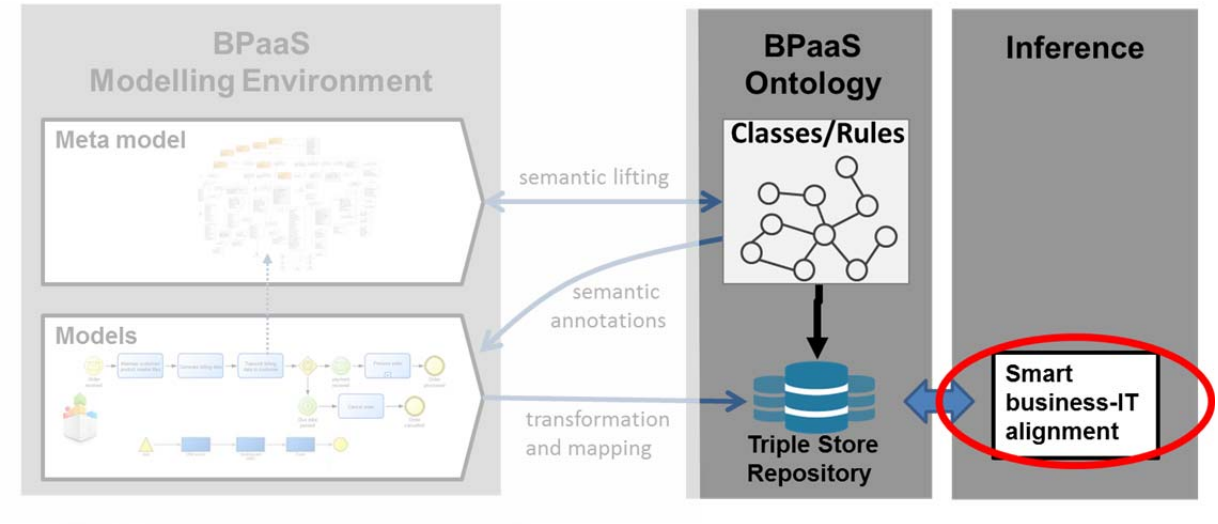


Figure 44: The Smart Business-IT Alignment prototype

6.2.1 Rules for Business and IT in the Cloud Alignment

In order to provide smart business and IT alignment in the cloud we distinguish between two categories of rules that differ from their degree of complexity, and one query. Rules are used to map business language to the technical one, while the query is used to compare business process requirements with workflow descriptions. The first category of rules is used to convert one element value that resides in the business layer to an element value that resides in the workflow layer. For instance, the value of "downtime in minutes per month" shown above in Figure 7, is converted in "Availability in %" by means of the following SPIN rule:

```
CONSTRUCT {
   ?req bpaas:hasAvailabilityInPercent ?availability .
}
WHERE {
   ?req a bpaas:BusinessProcessRequirement .
   ?req bpaas:hasDowntimePerMonthInMin ?val .
   BIND ((100 - (?val / ((24 * 30) * 100))) AS ?availability) .
}
```

The value of availability in % is then compared to the value of Availability in % (see value in Figure 18) described in each workflow by means of a query. The same mechanism applies on other annotation elements and the query simply includes the comparison of all these elements.

The second category of rules that embeds more complexity is used to make calculation among different annotation elements of the business layer such that the result can be again used for comparison purposes. For instance, given the media type, its size and the number of process executions (business language), we calculate the storage capacity of the cloud service in GB, which is expressed in technical language. The adopted SPIN rule is as follows:

```
CONSTRUCT {
   ?bpr bpaas:BPRHasDataStorage ?totalsize .
}
WHERE {
   ?bpr a bpaas:BusinessProcessRequirement .
   ?bpr bpaas:BPRHasMediaType ?video .
   ?video bpaas:mediaTypeHasSize ?mediasize .
   ?bpr bpaas:BPRHasNumberOfProcessExecution ?num .
   BIND ((?num * ?mediasize) AS ?totalsize) .
}
```

Thus, by multiplying the number of process executions (e.g. 50 campaign per year) with the size of the chosen media type (e.g. video mp4 of 5 GB), we infer the required value for data storage, i.e. 250 GB. The latter is then compared to the maximum available data storage annotated in the workflow description. Figure 18 shows a maximum available data storage of 500 GB, which is well covered by our inferred value, i.e. match succeeded. However, not all the inferred values are to be less than the target value for a match to succeed. Depending on the annotation element, some value should be exactly the same or greater than the value in the workflow description.

Finally, for the proper execution of the SPIN-rules, the BPaaS ontology has been further adapted, e.g. the previous example required adding of the datatype property “bpaas:mediaTypeHasSize”.

In the next version of the prototype we intend to express matching rules via DMN for human interpretability purposes. These will then be translated into SPIN rules.

6.2.2 Implementation of the Smart Business IT Alignment

The matching of the business process requirements and the workflow descriptions has been realized in an external application, due to the use of semantic technologies. The BPaaS-Annotation-Matcher loads the required ontologies and combines them to one single model. The same applies to the transformed model data. Once, all the data has been loaded, the rules (section 6.2.1) and the inferencing are applied.

In order to explain the matching algorithm, the availability example is used.

In the business process requirements, the modeller can enter the downtime per month in minutes, while the availability of a workflow is expressed in percentage. The rule applies the formula

(1-(MINUTES)/(60*24*30))*100.

For the example of two minutes, the formula calculates the following result

(1-(2)/(60*24*30))*100 = 99.99537037.

This is added to the business requirement object.

The matching is executed, based on the attributes of the elements. The user selects a Business Process (Figure 45).

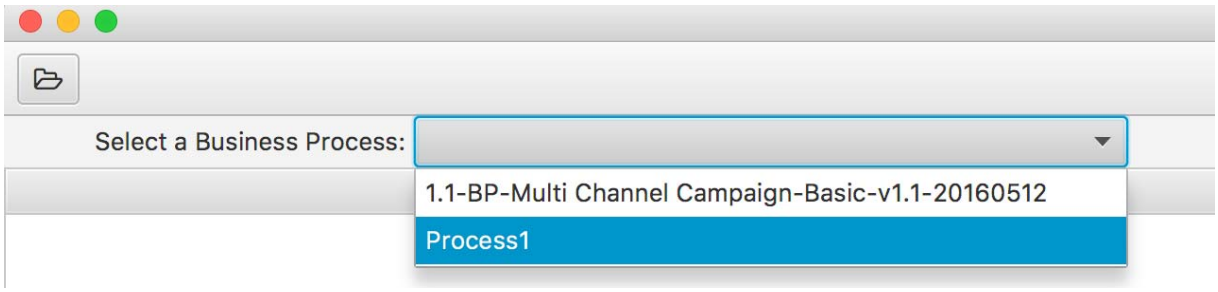


Figure 45: Selecting a Business Process

The matching algorithm queries the associated Business Process Requirements of the process and gathers their predicates and objects. Figure 46 shows an inferenced business process requirement. The applied rule added the predicate “bpaas:hasAvailabilityInPercent” having the integer object “99.99537037”.

```
data:obj.126998
  rdf:type bpaas:BusinessProcessRequirement;
```

```
  rdf:type apqc:5_0_Deliver_Services_20025;
  rdf:type fbpdo:ContentMessage;
  rdf:type fbpdo:Post ;
  rdfs:label "BPR1" ;
  bpaas:hasAvailabilityInPercent 99.99537037;
  bpaas:hasDowntimePerMonthInMin  30 ;
.
```

Figure 46: Transformed and inferenced Business Process Requirement

In order to match the business process requirement with the workflow description (Figure 47), the associated predicates and objects are used. Since the Business Process Requirement might list some predicates and objects that should not be compared (i.e. label, bpaas:BusinessProcessRequirement, etc.) we implemented a blacklist, where all the negligible predicates and objects are listed.

```
data:obj. 127006
  rdf:type bpaas: WorkflowDescription;
  rdf:type apqc:5_0_Deliver_Services_20025;
  rdf:type fbpdo:ContentMessage;
  rdf:type fbpdo:Post ;
  rdfs:label " WFD1" ;
  bpaas:hasAvailabilityInPercent 99.9999;
  bpaas:hasDowntimePerMonthInMin 30 ;
.
```

Figure 47: Transformed Workflow Description

For the properties that are described by number values, such as hasAvailabilityInPercent, matching rules do not look for exactly the same value but make a comparison to identify what the value is greater or less then the comparing object. For this reason, we implemented a filter attribute list that can be easily configured. The configuration vector is as follows:

("http://ikm-group.ch/archimeo/bpaas#hasAvailabilityInPercent","?availabilityValue","?availabilityValue >= {0}")

The first value is the URI of the property. Since we make use of different namespaces, the whole URI needs to be described. The second value is an attribute name and the third value is the comparison that is requested. The placeholder {0} will be automatically filled by the matching algorithm.

```
SELECT ?wfd ?name
        WHERE {<http://cloudsocket.eu/data#mod.126878>
                bpaas:hasReferencedWorkflowDescription ?wfd .
                ?wfd rdf:type apqc:5_0_Deliver_Services_20025 .
                ?wfd rdf:type fbpdo:ContentMessage .
                ?wfd rdf:type fbpdo:Post .
                ?wfd hasAvailabilityInPercent ?availabilityValue .
                ?wfd rdfs:label ?name .
        FILTER(?availabilityValue >= 99.99537037)
}
```

Figure 48: Generated SPARQL Query

The generated SPARQL query, shown in Figure 48 queries all workflow descriptions that are related to the workflow data:mod.126878 for the listed predicates and objects.

Figure 49 shows the results of comparing “Process 1” to two workflows in the repository. Business Process Requirement 1 matches with the workflow description 1, which is associated to both workflows.

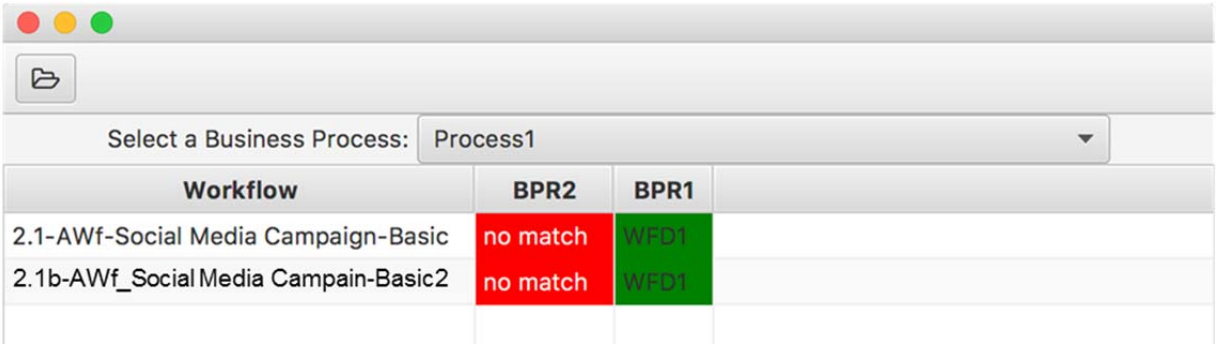


Figure 49: Prototype Showing Matching Results

7 PROTOTYPE ENVIRONMENT AND SETUP

The prototype environment is split into the following parts:

1. ADOxx Modelling Environment
    a. Modelling Library
    b. Models
    c. wget for web service calls
2. BPaaS-Annotation WebService
3. BPaaS-Annotation-Matcher

This chapter outlines the installation and setup of the environment.

7.1 Prerequisites

The prototype environment has been tested on MAC OSX 10.11.15 and Windows 10 Education, Version 10.0.10240. The recommended setup is in a Windows environment.

The following prerequisites need to be fulfilled in order test the prototype:

- Java 1.8
- Unfiltered and direct (no proxy) Internet connection

7.2 ADOxx Modelling Environment

In order to setup the environment, the ADOxx platform can be downloaded from https://www.adoxx.org/live/download. Request a free installation code[1] and follow the instructions on ADOxx "New installation Guide"[2].

7.2.1 WGET installation

The GNU Wget utility is used for downloading network data. In this prototype it is responsible for the communication with the web service.

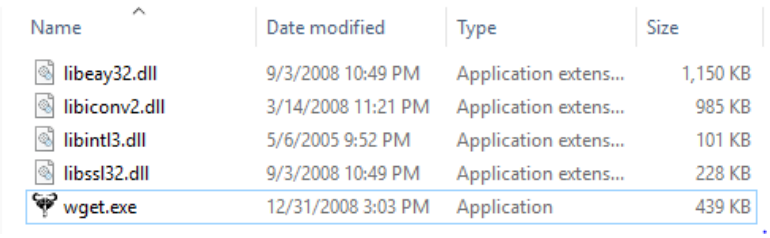
For Windows: browse to http://gnuwin32.sourceforge.net/packages/wget.htm and download the binaries for your operating system.

Run the executable installation file and follow the installation wizard.

After the installation has been finished, browse to the installation directory. Standard installation directory: "C:\Program Files (x86)\GnuWin32\bin".

Figure 50 shows the files that should be appear in the binaries folder. Copy all of those files directly into the user temp. This user temp is in Windows 10 located as follows: "C:\Users\<USERNAME>\AppData\Local\Temp".

[1] https://www.adoxx.org/live/installation-code
[2] https://www.adoxx.org/live/installation-guide-15-new



Name	Date modified	Type	Size
libeay32.dll	9/3/2008 10:49 PM	Application extens...	1,150 KB
libiconv2.dll	3/14/2008 11:21 PM	Application extens...	985 KB
libintl3.dll	5/6/2005 9:52 PM	Application extens...	101 KB
libssl32.dll	9/3/2008 10:49 PM	Application extens...	228 KB
wget.exe	12/31/2008 3:03 PM	Application	439 KB

Figure 50: Wget binary files

7.3 BPaaS Annotation Web Service Setup and Use

This paragraph describes the installation of the BPaaS annotation web service and how to test it.

7.3.1 Installation

Download the following files from https://github.com/BPaaSModelling/BPaaS-Annotation-WebService/releases/tag/1.0 and store them in the same folder locally.

- BPaaS-Annotation-WebService-0.0.1-SNAPSHOT.war
- webapp-runner.jar

In the following, the example C:\Users\Ben\Downloads> is used.

7.3.2 Test

In order to start the web service, open a command prompt and browse to the folder where the files have been saved to.

e.g. cd C:\Users\Ben\Downloads

The web service can be started with the following command:

java -jar webapp-runner.jar BPaaS-Annotation-WebService-0.0.1-SNAPSHOT.war

Figure 51 shows the command prompt output when the web service is started. The socket will be opened on the localhost port 8080.

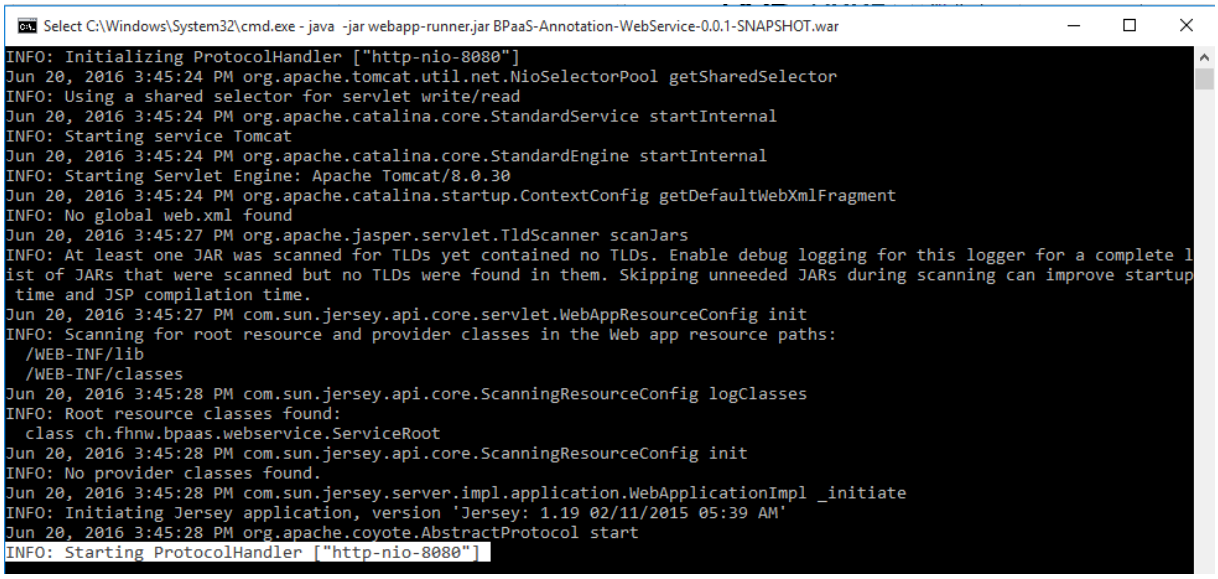


Figure 51: Web Service Command Prompt Output

In order to test the web service, open the browser and browse to:

http://localhost:8080

If the browser returns a “Hello!”, the web service works fine. Please note that the first request usually takes longer then the following ones. The reason for this is that the web service downloads the ontologies on the first request.

The web service is setup now.

7.3.3 Import the Prototype Library

Once the platform has been installed, two icons will appear on the desktop, as shown in Figure 52.

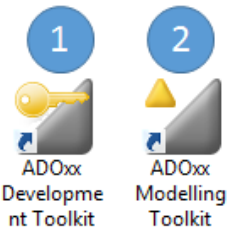


Figure 52: ADOxx Desktop Shortcuts

(1) ADOxx Development Toolkit
(2) ADOxx Modelling Toolkit

In order to import the CloudSocket Library, start the “ADOxx Development Toolkit”.

Default Username:	Admin (case-sensitive)
Password	password

The environment will start in user management mode. Click on “Library Management” as shown in Figure 53- (1) and open the settings dialog (2).

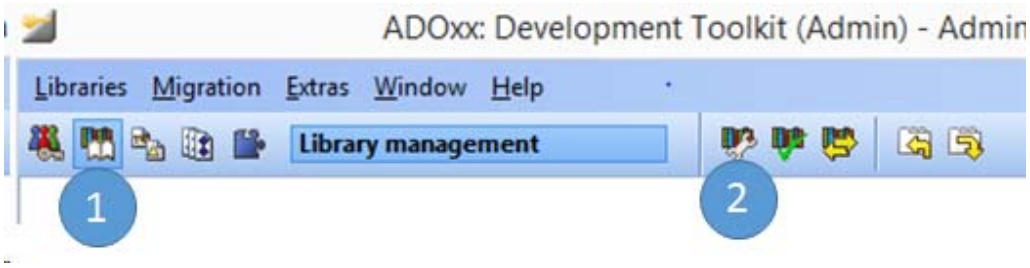


Figure 53: ADOxx Development Toolkit Library Management

The library management dialog will appear as shown in Figure 54. Click on management (1) and select import (2).

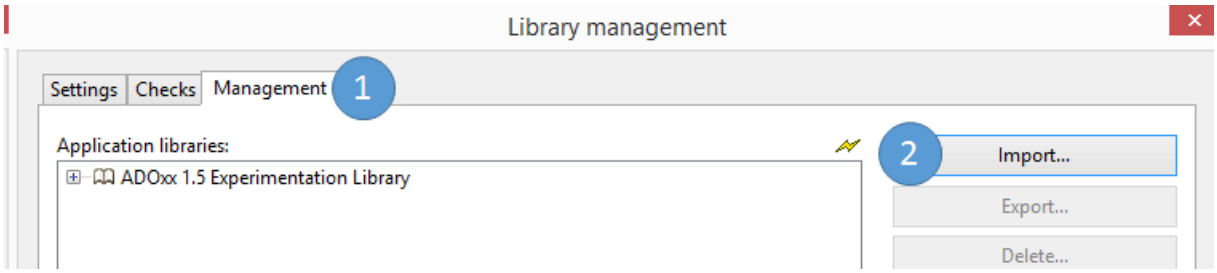


Figure 54: ADOxx Development Toolkit Import

Download the library file Meta model_Library_Prototype_v_1.0.abl from https://github.com/BPaaSModelling/Modelling-Library/releases/tag/1.0. In ADOxx, click on import and browse to the location of the downloaded file. Select the ADL file and click import.

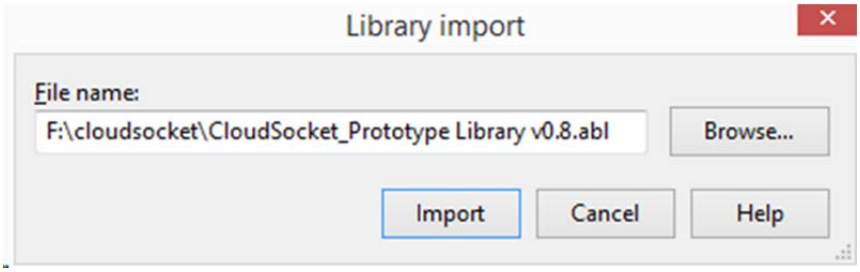


Figure 55: ADOxx Development Toolkit – Browse for Library File for Import

Once the import process finished click on yes for creating a default model group (Figure 56).

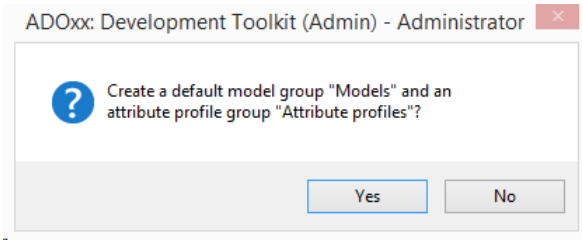


Figure 56: ADOxx Development Toolkit - Create Default Model Group

If the import has been successful, the report will look like in Figure 57.

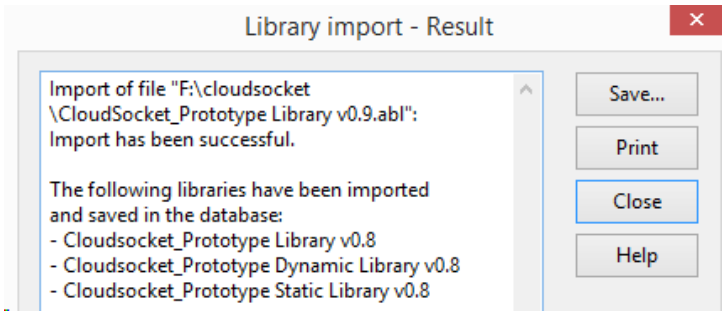


Figure 57: ADOxx Development Toolkit - Library Import Result

Create a new user and make the imported library. Click on User management (1) and select User List (Figure 58).

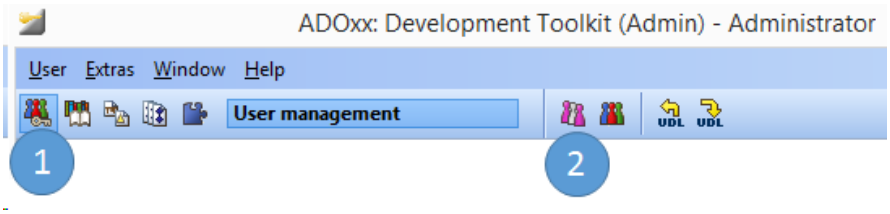


Figure 58: ADOxx Development Toolkit - User Management

The User management – User list dialog will appear as shown in Figure 59.

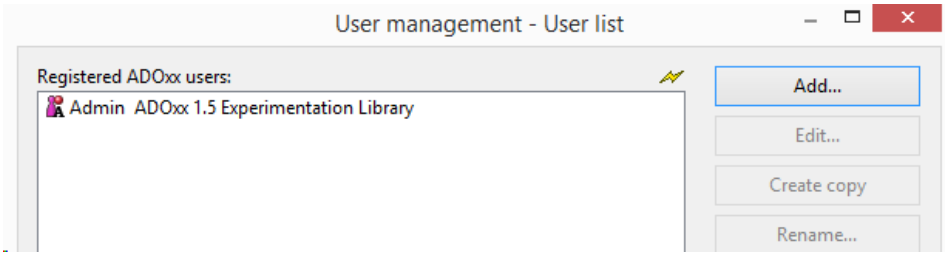


Figure 59: ADOxx Development Toolkit - User List

Enter the user name and password. Select the imported library from the dropdown box (1) and select user group (2).

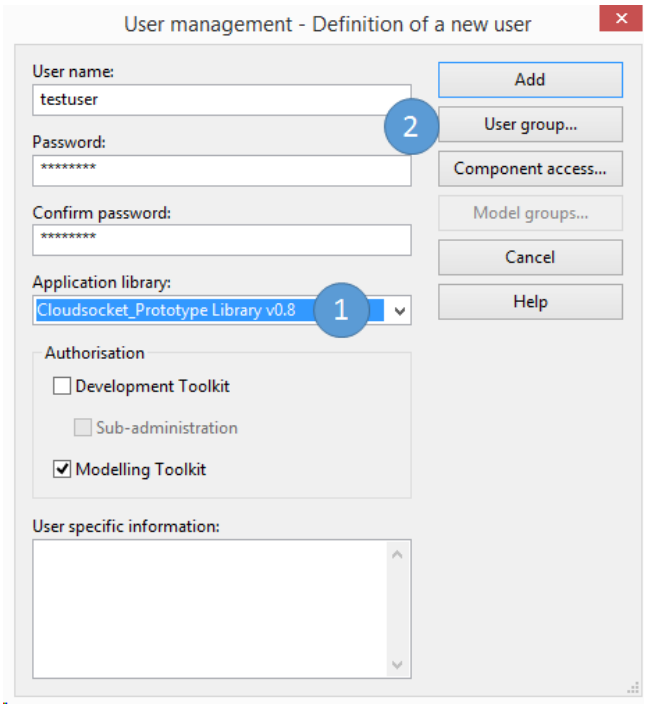


Figure 60: ADOxx Development Toolkit – Create a new User

Assign the user to a user group and click ok.

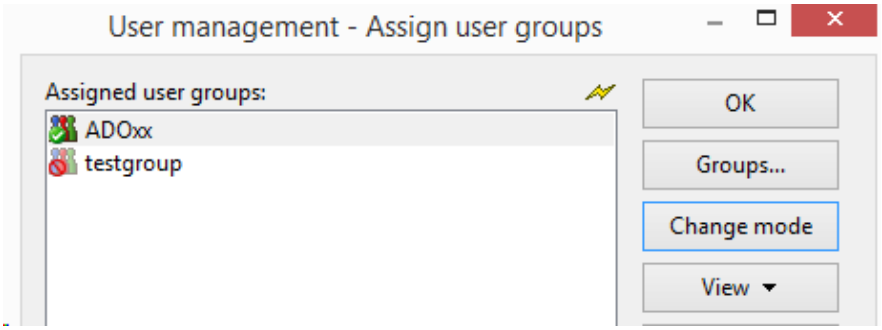


Figure 61: ADOxx Development Toolkit – Assign User Groups

The user will be created.

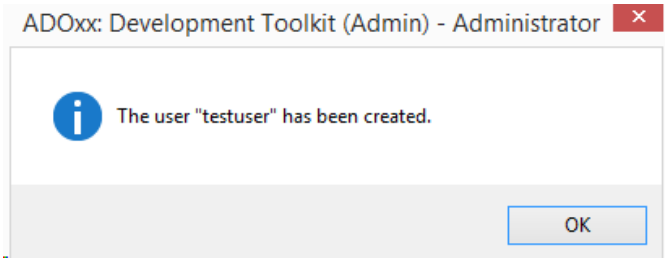


Figure 62: ADOxx Development Toolkit - User Management Dialog

7.3.4 Start the Modelling Environment

The next step will show how to make use of the library in the modelling toolkit. Open the ADOxx Modelling Toolkit (2).

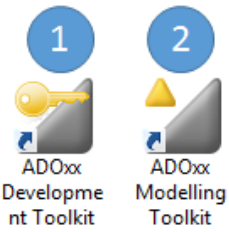


Figure 63: ADOxx Desktop Shortcuts

Click on yes at the user account control dialog.

Enter the user name and password of the newly created user, or the user that has been assigned to the library. Make sure that the correct database is selected (has been usually created and named during the installation process).

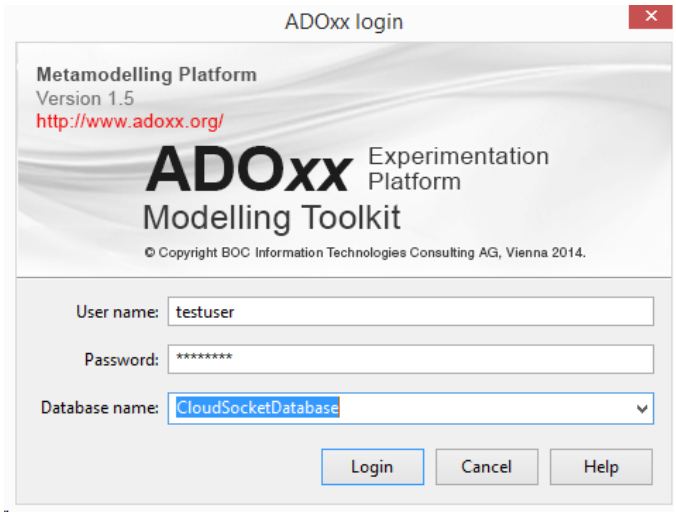


Figure 64: ADOxx Login

In order to set the link to the web service for annotation, select tools and choose “Set Annotation Service End Point” (Figure 65)

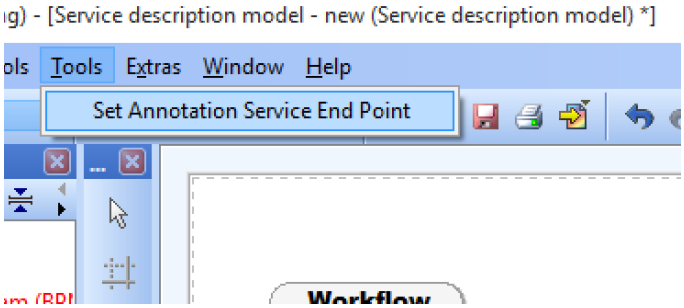


Figure 65: Set Annotation End Point

A message box appears which requests to enter the URL to the web service. If the web service has been started as described above, enter http://localhost:8080/. Please note the slash after the port numbers is important.

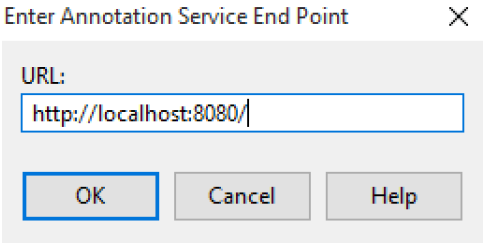


Figure 66: Enter End Point URL

The modelling environment confirms the registration of the annotation endpoint by showing the dialog (Figure 67).

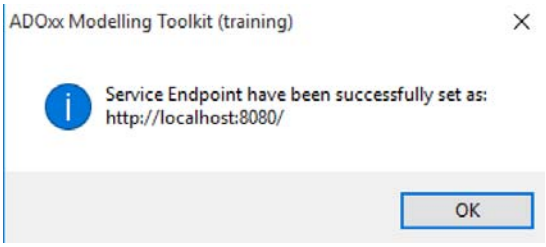


Figure 67: Confirmation Dialog

Create a new Business process diagram (BPMN 2.0) model (1), provide a name (2) and a version number (3).

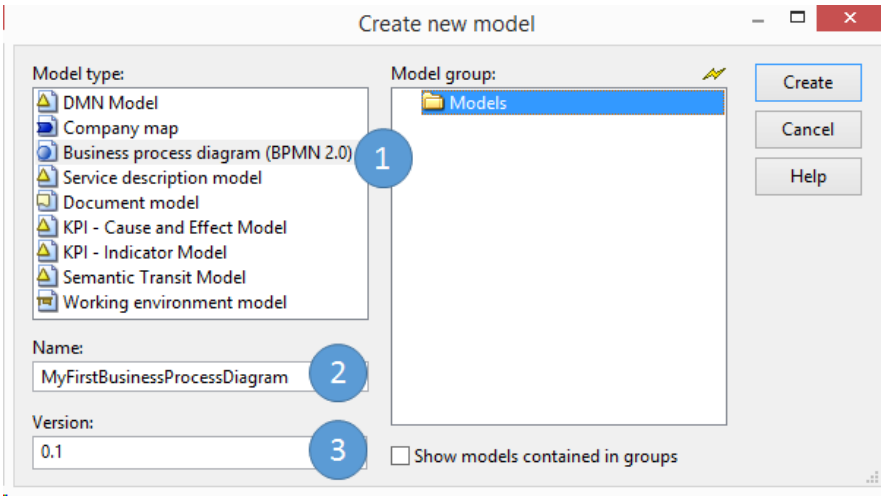


Figure 68: ADOxx Modelling Toolkit - Create Model

In order to make use of the annotation, create a Service description model and select either the workflow description or the business process requirement.

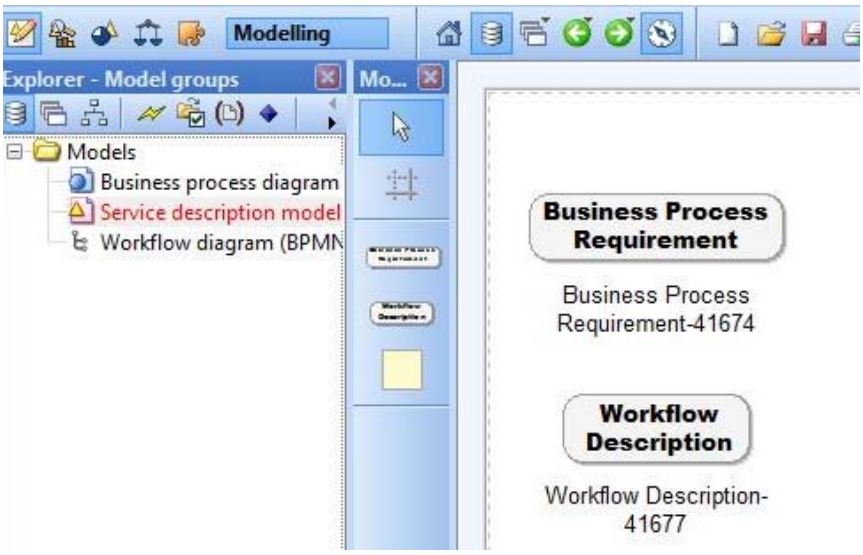


Figure 69: BPaaS Design Prototype – Service description model, Workflow Description and Business Process Requirement

7.3.5 Download and start of the Matching Environment

Browse to https://github.com/BPaaSModelling/BPaaS-Annotation-Matcher/releases and download the BPaaS-Annotation-Matcher-0.0.1-SNAPSHOT.jar file.

Usually jar files can be opened by doubled clicking on it. If it does not work, open the command line, browse to the location where the file is stored and perform the following command

java -jar BPaaS-Annotation-Matcher-0.0.1-SNAPSHOT.jar

Please note, that the application needs some time to load, since it is downloading the ontologies instantly from the internet.

Click on the open icon in order to load the ADOxx XML. Browse to the location of the file and make sure that the XML is in the same location as the “adoxml31.dtd” file that will come with the export.

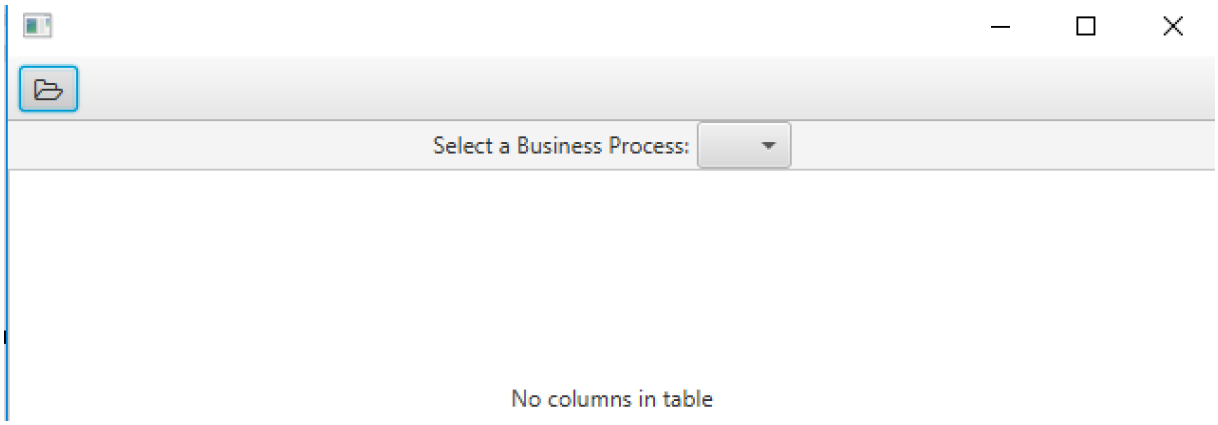


Figure 70 Selecting the XML file exported from ADOxx

The Matcher parses the export and does its semantical processing. You can choose a business process and the matching will be performed automatically, according to the Requirements and Descriptions in the modelling.

8 CONCLUSIONS AND FUTURE WORK

This chapter gives a short summary of the deliverable and an outlook to the further prototype development.

8.1 Summary

This report describes the modelling prototypes realizing the BPaaS Design Environment. The prototypes implement the graphical modelling environment (section 3), the BPaaS Ontology (section 4), the semantic lifting of graphical and ontological concepts (section 5) and the implementation of the smart business and IT alignment (section 6).

The prototype is available free for download from the CloudSocket webpage.

8.2 Future Work

The prototype will be evaluated with different processes as described in Deliverable D5.2 (CloudSocket 2016) and enhanced for additional usage scenarios. For example, the action and object taxonomy will be refined in order to cover broader application areas and to increase usability of annotation. Based on experiences in specifying additional cloud services, the non-functional requirements and workflow descriptions will be adapted to cover all aspects of SLA-relevant specifications.

Moreover, we want to foster the allocation phase. can also mention here that we want to foster the service discovery functionalities, for example in the definition of the executables workflows or the allocation phase.

9 REFERENCES

APQC (2014) Process Classification Framework Version 6.1.1. Available at http://www.apqc.org/pcf [Access December 23, 2015]

CloudSocket (2015a). First CloudSocket Architecture D4.1, August 2015.

Cloudsocket (2015b) Modelling Framework for BPaaS D3.1. December 2015.

C-SIG SLA (2014) Cloud Service Level Agreement Standardisation Guidelines. C-SIG on Service Level Agreement. Available at https://ec.europa.eu/digital-agenda/en/news/cloud-service-level-agreement-standardisation-guidelines [Accdess December 30, 2015).

Emmenegger, S., Hinkelmann, K., Laurenzi, E., Thönssen, B., Witschel, H. F., & Zhang, C. (2016). Workplace Learning - Providing Recommendations of Experts and Learning Resources in a Context-sensitive and Personalized Manner. In MODELSWARD 2016, Special Session on Learning Modeling in Complex Organizations. Rome.

Hinkelmann, K., Gerber, A., Karagiannis, D., Thoenssen, B., van der Merwe, A., & Woitsch, R. (2015). A new paradigm for the continuous alignment of business and IT: Combining enterprise architecture modelling and enterprise ontology. Computers in Industry. doi:10.1016/j.compind.2015.07.009

Hrgovcic, V., Karagiannis, D., & Woitsch, R. (2013). Conceptual Modeling of the Organisational Aspects for Distributed Applications: The Semantic Lifting Approach. In COMPSACW 2013, IEEE 37th Annual Computer Software and Applications Conference Workshops (pp. 145–150). IEEE. doi:10.1109/COMPSACW.2013.17

Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., & Peterson, A. S. (1990). Feature-oriented domain analysis (FODA) feasibility study (No. CMU/SEI-90-TR-21). Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.

Kappel, G., Kapsammer, E., Kargl, H., Kramler, G., Reiter, T., Retschitzegger, W., Schwinger, W. and Wimmer, M. (2006). Lifting Meta models to Ontologies: A Step to the Semantic Integration of Modeling Languages. In O. Nierstrasz, J. Whittle, D. Harel, & G. Reggio (Eds.), Model Driven Engineering Languages and Systems, Proceedings of the 9th International Conference, MoDELS 2006, LNCS 4199., pp. 528–542. Bernlin Heidelberg: Springer-Verlag.

Karagiannis, D., & Woitsch, R. (2010). Knowledge Engineering in Business Process Management. In Handbook on Business Process Management 2 (pp. 463–485). Berlin Heidelberg: Springer.

Silver, B. (2011). *BPMN Method and Style*, Second Edition (Second Edition). Aptos, CA: Cody-Cassidy Press.