CloudSocket

CloudSocket

MODELLING FRAMEWORK FOR BPAAS

D3.1

Editor Name	Knut Hinkelmann
Submission Date	December 31, 2015
Version	1.0
Confidentially Level	PU

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

Executive Summary

This document introduces the BPaas Design Environment which supports the alignment of business and IT in the cloud. This is the first phase of the BPaaS lifecycle. Its output is the BPaaS Design Package, which is the used in the BPaaS Allocation Environment.

This is the first deliverable of work package 3 "Business Process as a Service Research". It contains the specification on hybrid modelling and semantic lifting to enable Cloud Eco-System Design. It is the basis for the development of a prototype for the BPaaS Design Environment, which is due end of June 2016 as Deliverable D3.2.

The research contributes to the development of an ontology for Business Process as a Service and the semantic lifting of graphical models.

The BPaaS Design Environment introduces the user interface to design domain specific business processes, the semantic lifting of those processes as well as specifications of executable workflows. The modeling method is implemented in the AOOxx modeling environment and consists of several model types including business process diagrams organizational model, document model, decision model, KPI models and service description model. The packaging of the business process, workflow, decision, and KPI models is done in the 'BPaaS Alignment Model'. The concept 'BPaaS Design Package' creates an export package for the BPaaS Allocation Environment.

The semantics of the models is defined by the BPaaS Ontology. The BPaaS Ontology is a cloud-specific extension of the ArchiMEO enterprise ontology. The elements of various model types are embedded in the class hierarchy of ArchiMEO. Additional classes and concepts are introduced to represent the cloud ecosystem and to allow for mapping business processes and workflow.

Semantic lifting allows integrating human-interpretable models with the machine interpretable BPaaS Ontology. Three types of semantic lifting have been developed. First, the Cloud Broker has the possibility to manually annotate process models with elements defined in the BPaaS Ontology. Seven different annotation approaches are realized. A second approach for semantic lifting is the transformation of models into a semantic format. The models are exported from the modelling tool and instances of classes defined in the BPaaS Ontology are created. The third kind of semantic lifting directly aligns the meta models with the classes defined in the BPaaS Ontology.

A first rapid prototype of the BPaaS Design Environment has already been implemented and is made available via the CloudSocket website. It consists of the model method implemented in the ADOxx tool, the BPaaS Ontology and contains approaches for semantic lifting.

Project Context

Workpackage	WP3: Business Process as a Service Research
Task	T3.1: BPaaS Design Environment Research
Dependencies	Input to D3.2, T3.2 and WP4

Contributors and Reviewers

Contributors	Reviewers
Mehmet Albayrak, Knut Hinkelmann, Kyriakos Kritikos, Sabrina Kurjakovic, Benjamin Lammel, Robert Woitsch,	Yongzheng Liang (BWCON), Emanuele Bellini (MATHEMA), Jörg Domaschka (UULM), Dimitris Plexousakis (FORTH)

Approved by: Robert Woitsch (BOC) as Coordinator

Version History

Version	Date	Authors	Sections Affected
0.1	9th November 2015 - 8th December 2015	Hinkelmann, Kurjakovic, Lammel (FHNW), Woitsch (BOC)	Initial setup: Document structure, Introduction, methodology, ontology development, modelling method development
0.2	9th December 2015	Woitsch (BOC)	
0.3	10th December 2015	Kritikos (FORTH)	State of the Art
0.4	10th December 2015	Woitsch (BOC)	Introduction, modelling method development, prototypes
0.5	11th December 2015	Hinkelmann, Lammel, Kurjakovic, Emmenegger	Ontology development,
0.6	13th December 2015	Hinkelmann et al., (FHNW)	Introduction, state of the art, methodology, ontology development
0.7	21th December 2015	Hinkelmann, Lammel, Kurjakovic (FHNW), Kritikos (FORTH)	Including feedback of reviewers State of the art
0.8	22ndth December 2015	Hinkelmann, Lammel, Kurjakovic (FHNW)	Ontology development, prototype, Methodology, overall improvements
0.9	24th December 2015	Hinkelmann, Kurjakovic, Lammel (FHNW)	Overall improvements
1.0	30th December 2015	Hinkelmann (FHNW), Dimitris Plexousakis (FORTH), Robert Woitsch (BOC), Stefan Wesner (UULM)	Finalisation and 2nd review

Copyright Statement – Restricted Content

Table of Content

List of Figures

List of Tables

1 INTRODUCTION

This document introduces the BPaaS Design Environment, which supports the alignment of Business and IT in the Cloud. In this chapter it is described how the BPaaS Design Environment fits into the CloudSocket architecture and how it supports the BPaaS lifecycle. Then the contribution to research is described and a general introduction into the structure of the BPaaS Design Environment is given.

1.1 Project Context of this Document

The BPaaS Design Environment is part of the CloudSocket architecture as described in Deliverable D4.1 "First CloudSocket Architecture" (see Figure 1). Other elements of the architecture are the BPaaS Allocation Environment, the BPaaS Execution Environment, and the BPaaS Evaluation Environment as well as the BPaaS Marketplace.

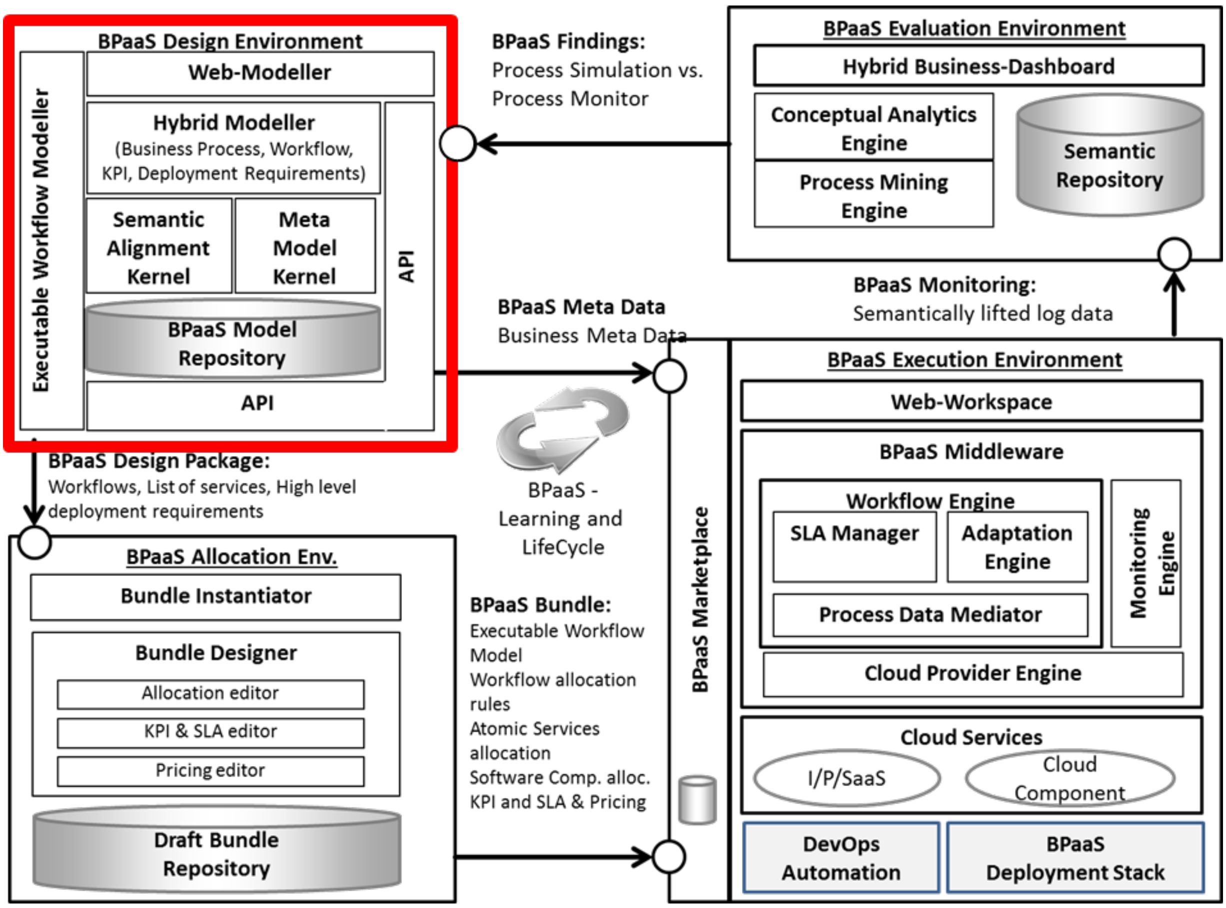

Figure 1: Initial High-level Architecture of CloudSocket (CloudSocket 2015b)

The BPaaS Design Environment introduces the user interface to design domain specific business processes, the semantic lifting of those processes as well as specifications of executable workflows. Semantic annotations are partly expressed in extensions of business process models and partly as ontologies. All this information is handed over to the BPaaS Allocation Environment.

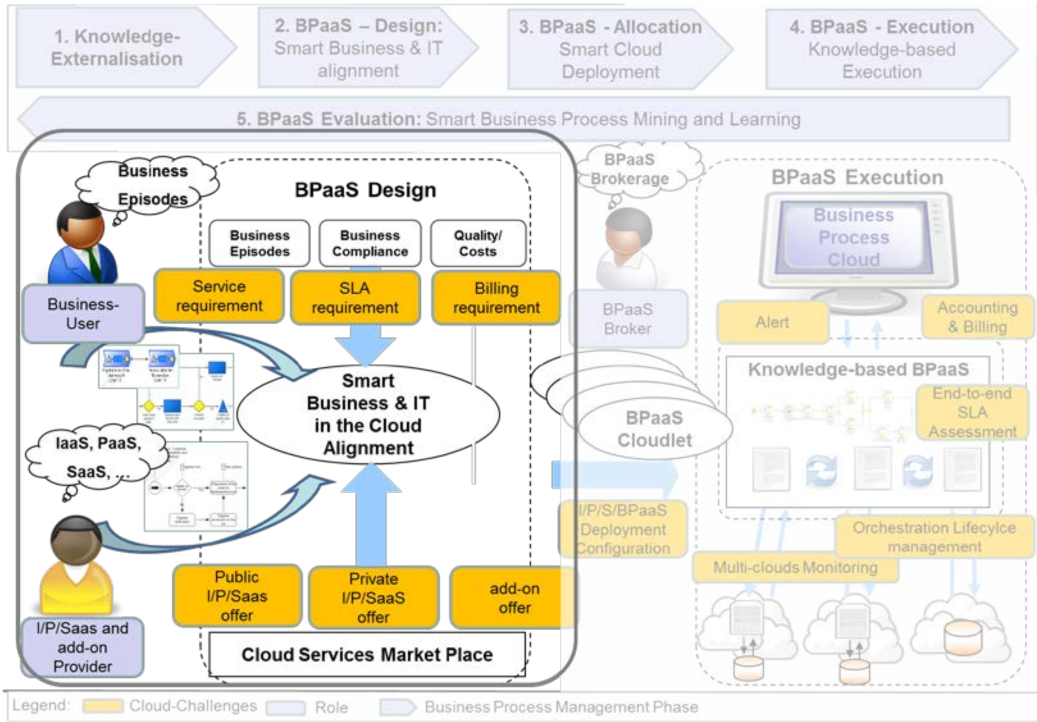

Figure 2 Focus of Business and IT-Cloud Alignment as Part of the CloudSocket Approach

The BPaaS Design Environment focuses on the smart alignment between business requests 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 First CloudSocket Architecture (CloudSocket 2015b), 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. In order to provide those different modelling tools a meta modelling platform is used, which support standard process modelling languages like BPMN (OMG 2011).

In addition, the BPaaS Design Environment has the possibility to annotate the models with an ontology. The so-called semantic lifting (see Figure 3) enables the semantic annotation of BPaaS models with ontology concepts, which are represented in formal languages like RDF (W3C 2014) and OWL (W3C 2012). Thus, the models produced are the following:

- a domain specific business process model in BPMN format and additional information such as cloud specific requirements or KPIs along with information pertaining to the description of the organization, its non-functional requirements and its main business objectives,
- a semantic lifting of the business process model typical form of a BPMN model along with RDF-based semantic annotations to business & IT ontology concepts,
- the executable workflow model in BPMN along with RDF-based semantic annotations and
- the definition of KPIs based on OWL-Q

This is the first deliverable of work package 3 "Business Process as a Service Research". It contains the specification on hybrid modelling and semantic lifting to enable Cloud Eco-System Design. It is the basis for the development of a prototype for the BPaaS Design Environment, which is due end of June 2016 as Deliverable D3.2. The prototype will show the architecture and description of process orchestration for Business Process as a Service. The use of the hybrid modelling and semantic lifting for BPaaS allocation and execution is not part of this deliverable, but will be described in Deliverable D3.3.

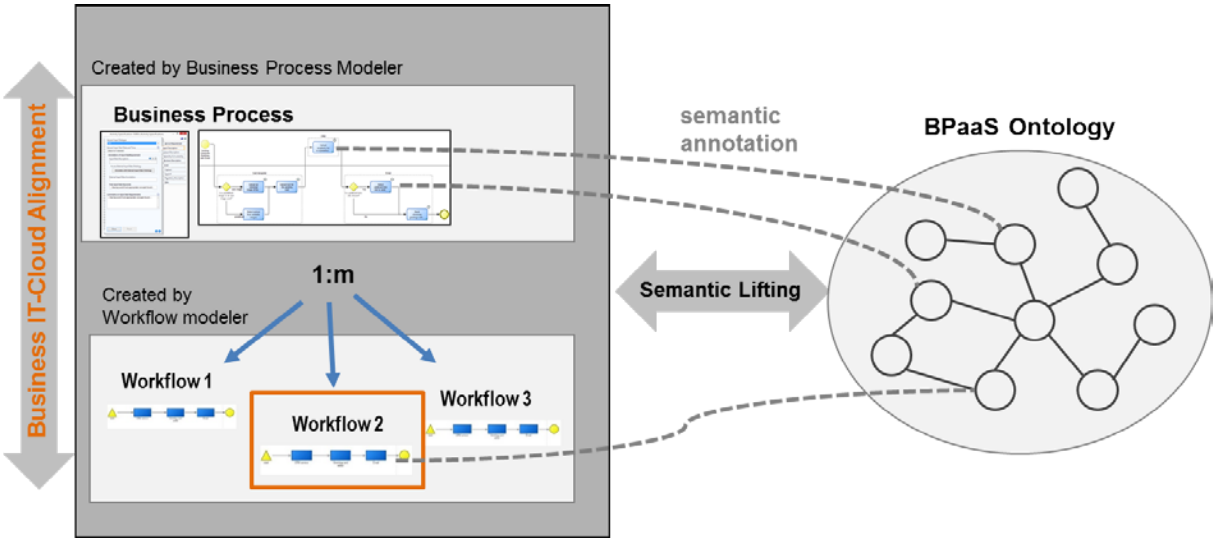

Figure 3: Semantic Lifting

1.2 Research Problem

The BPaaS Design Environment supports the smart alignment of business and IT in the cloud. In particular, this involves the identification of executable workflows for a given domain-specific business process model, which is a specific application of service discovery. Both models - business process model and executable workflow model - are modelled in the BPaaS Design Environment. They are semantically enriched using business and IT ontology concepts (see section 6.

There already exist approaches for ontology-based business service discovery (e.g. Akkermans et al. 2004, da Silva Santos et al. 2008). However, the approaches lack a detailed specification of the modelling languages and the ontology.

One research result of CloudSocket is the BPaaS ontology for business-IT in the cloud alignment (see CloudSocket 2015). In this research we explore and show, which elements an enterprise ontology should include in order to support the alignment of business and IT in the cloud. We reuse concepts well-known from business process modelling and enterprise architecture and represent them in an ontology.

A second focus of research is the semantic lifting. According to the First CloudSocket Architecture (CloudSocket 2015b), the following actors are proposed for the BPaaS Design Environment: The CloudSocket Broker is an organisation that provides a marketplace for business processes towards the CloudSocket Customers. The CloudSocket Broker can either have a business background (Business Process Designer) or a technical background (Workflow Designer). A third role is the Ontology Expert, who is responsible for the maintenance of the alignment ontology.

For a smart alignment of business and IT in the cloud, the models have to be available in a formal representation. It cannot be assumed, however, that the Business Process Designer and the Workflow Designer are able to describe the business processes and services in a formal language. Therefore our research combines human-interpretable graphical modelling with a machine-interpretable formal representation. This so-called semantic lifting is done on two levels: (a) using the transformation from semi-formal graphical models to formal ontological representations, as well as (b) by a direct manipulation of the ontological representation by the Ontology Expert.

Once the models are semantically lifted, they have a formal nature; hence they can be interpreted by machines. We hence consider the semantic enrichment not only as a necessity for smart business and IT alignment, but also as a pre-condition for further formal processing either by preparing the workflows in the cloud, or by providing smart support for machine-driven decision-making during execution.

1.3 Introduction of the BPaaS Design Environment

In this section we give a short overview of the modelling aspects of the BPaaS Design Environment in order to support the alignment of Business and IT in the Cloud. The BPaaS Design Environment integrates various modelling approaches, allowing for both human and machine interpretation (Hinkelmann et al. 2015). 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 easily be understood by the Cloud Broker. Text elements can be used to further explain aspects for which no graphical representation is provided.
- Smart business-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. It makes the semantics of graphical and textual models explicit (Kappel et al. 2006; Hrgovcic et al. 2013).

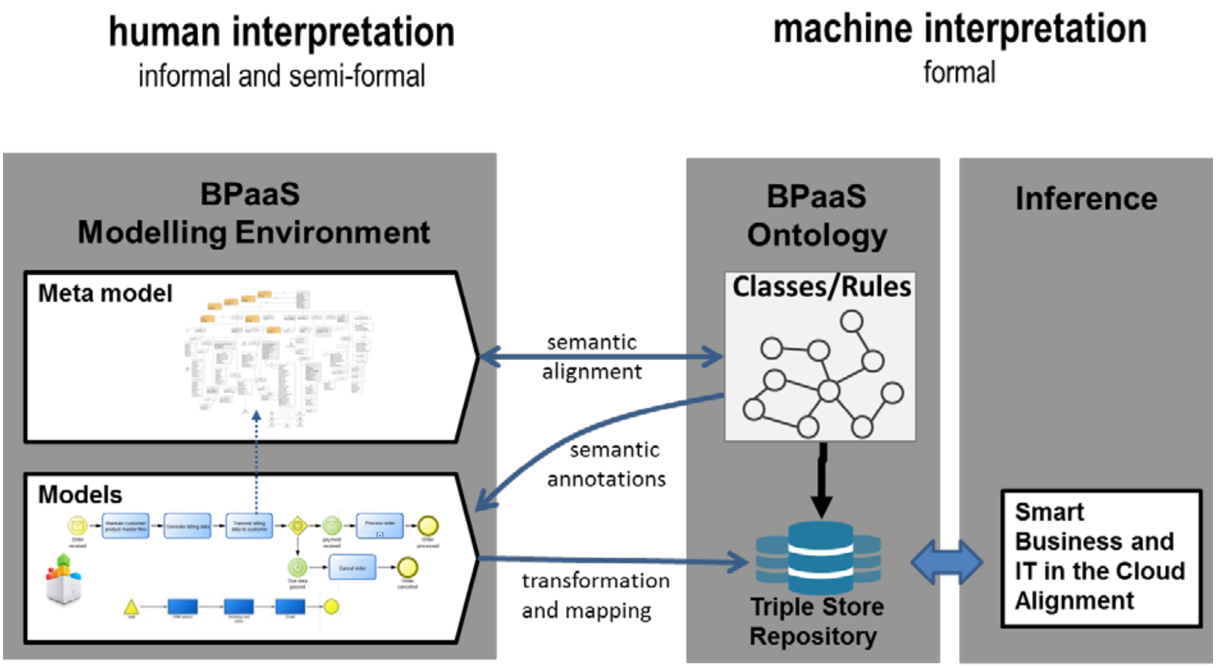

Figure 4 provide a sketch of the framework for the smart business and IT alignment in the cloud. On the left-hand side there is the human-interpretable modelling environment which is implemented in ADOxx.org. On the right there is the machine-interpretable ontological representation and the inference engine for the Business-IT in the Cloud Alignment.

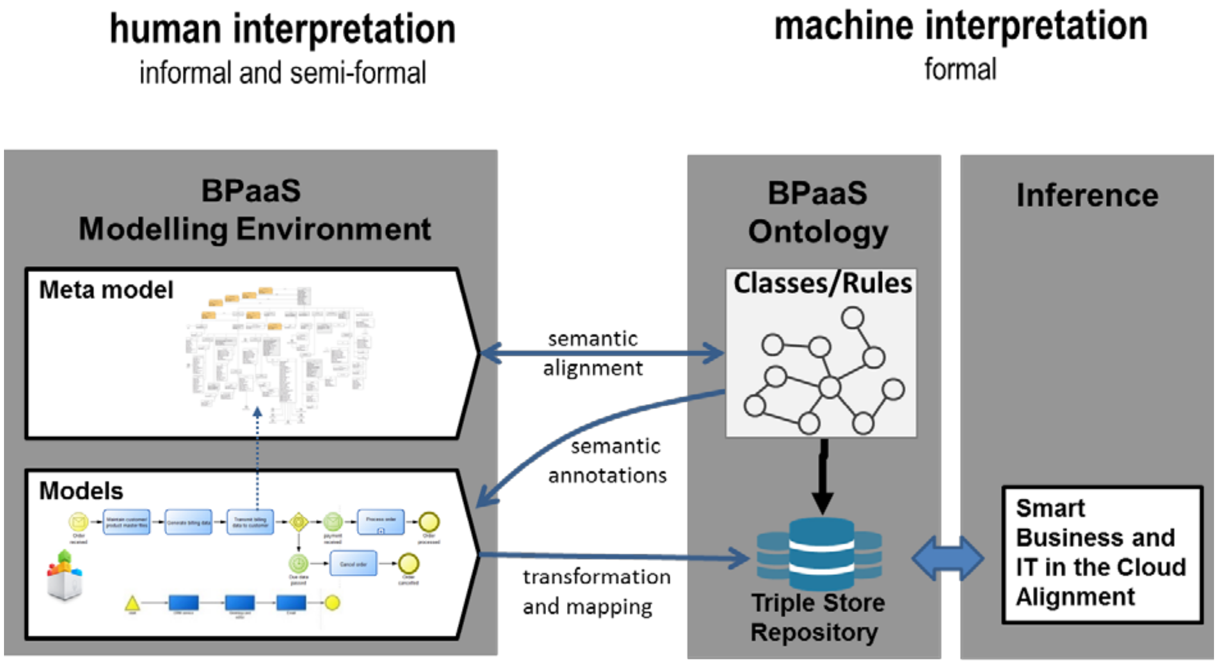

Figure 4: Overview of the BPaaS Design Environment and Smart Business IT-Cloud Alignment

The overall structure of the Smart Business IT-Cloud Alignment can thus be seen as an instantiation of a knowledge-based system (Puppe 1991). It consists of a user interface, a knowledge base and an inference component.

- The user interface interacts with the Cloud Broker. The Cloud Broker models the business process and its requirements, the workflow model and the KPIs using human-interpretable, graphical and text-based models.
- The machine-interpretable knowledge base corresponds to the semantically enriched models. It consists of two parts: The ontology is composed of class definitions and rules. The concrete facts represent the knowledge represented in the process and workflow models. Both the ontology and the facts are stored in a repository.
- The inference component can be regarded as a query system which realizes the smart business-IT alignment. It interprets the semantically enriched models and presents the answers to the user.

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 BPaaS Design has as a result the BPaaS Design Package, a specification of the workflows, services and high-level requirements (see Figure 1). It can be regarded as a three step approach:

1. Business processes and workflows are modelled using the BPaaS Modelling Environment and the semantic lifting support via the BPaaS Ontology.
    a. The workflows of the BPaaS Marketplace is modelled by a Cloud Broker, who has technical knowledge. A workflow is a composition of Cloud Services. It encompasses a respective control flow logic which indicates the ordering of the workflow tasks and the Cloud Services.

   b. The Cloud Broker uses the BPaaS Modelling Environment to model the domain-specific business process and service requirements.
2. The business process model is mapped to one or several appropriate workflow models. This workflow identification is done by the inference engine for smart business-IT in the cloud alignment using the mapping rules of the BPaaS Ontology component. This step is a special case of service discovery, where
3. Finally the BPaaS Design Package is created. It consists of the domain-specific business process and the executable workflow model which is composed of several Cloud services. with the semantic annotations, key performance indicators and additional information which is relevant for allocation and deployment.

This report has a focus on the modelling parts: The business process and workflow models, the respective semantic annotations and the rules for the business-IT alignment. The creation of the BPaaS Design Package still requires manual work by the Cloud Broker.

1.4 Structure of the Document

Chapter 2 introduces and explains the used approach for developing the BPaaS Design Environment, the used business scenarios and the resulting competency questions. Chapter 3 highlights the state of the art of enterprise modelling, semantic enrichment through semantic lifting and, service discovery and composition. Then two parallel strands are elaborated, one for each of the two modelling environments. The BPaaS modelling method is described in chapter 4. It contains the BPaaS meta model stack, the class diagrams for the model types and mechanisms for integrating the model types.The BPaaS Ontology is described in chapter 5 as a BPaaS-specific extension of the ArchiMEO enterprise ontology. Chapter 6 presents approaches for semantic lifting ranging from manual annotation to model and meta-model transformation. A first version of the prototype of the BPaaS Design Environment is explained in chapter 7. It consists of the graphical modelling and annotation environment, as well as the ontology prototype. A summary and outlook concludes as chapter Fehler! Verweisquelle konnte nicht gefunden werden..

2 METHODOLOGY

This section introduces the development methodology that is used to implement the BPaaS Design Environment starting from conceptual ideas, followed by specifications and finally concluding in the implementation.

The BPaaS Design Environment consists of two modelling components (see Figure 4):

- The BPaaS modelling environment
- The BPaaS Ontology

The BPaaS modelling environment contains the meta-model for the graphical modelling languages. The meta-models can be semantically lifted by aligning them with the concept definitions of the BPaaS ontology. Furthermore, the graphical models can be semantically annotated with concepts, which are defined in the BPaaS ontology. This means that the development of the ontology and the meta-model development have to be synchronized in the sense that the ontology contains class definitions describing the intended semantics of the elements of the graphical modelling language.

The development of such a model-based approach is supported by the so-called OMiLAB LifeCycle, which is the basis of the Agile Model Method Engineering (Karagiannis 2015) and has been developed and successfully used in the Open Models Initiative (http://www.openmodels.at). The OMiLAB LifeCycle proposes tools, phases and best practices to achieve a prototype of the modelling tool.

Figure 5 depicts in the upper part the abstract developing methodology proposed by OMiLAB and in the lower part the concrete instantiation of the project CloudSocket. First the OMiLAB abstract methodology is briefly introduced before the CloudSocket instantiation is reported.

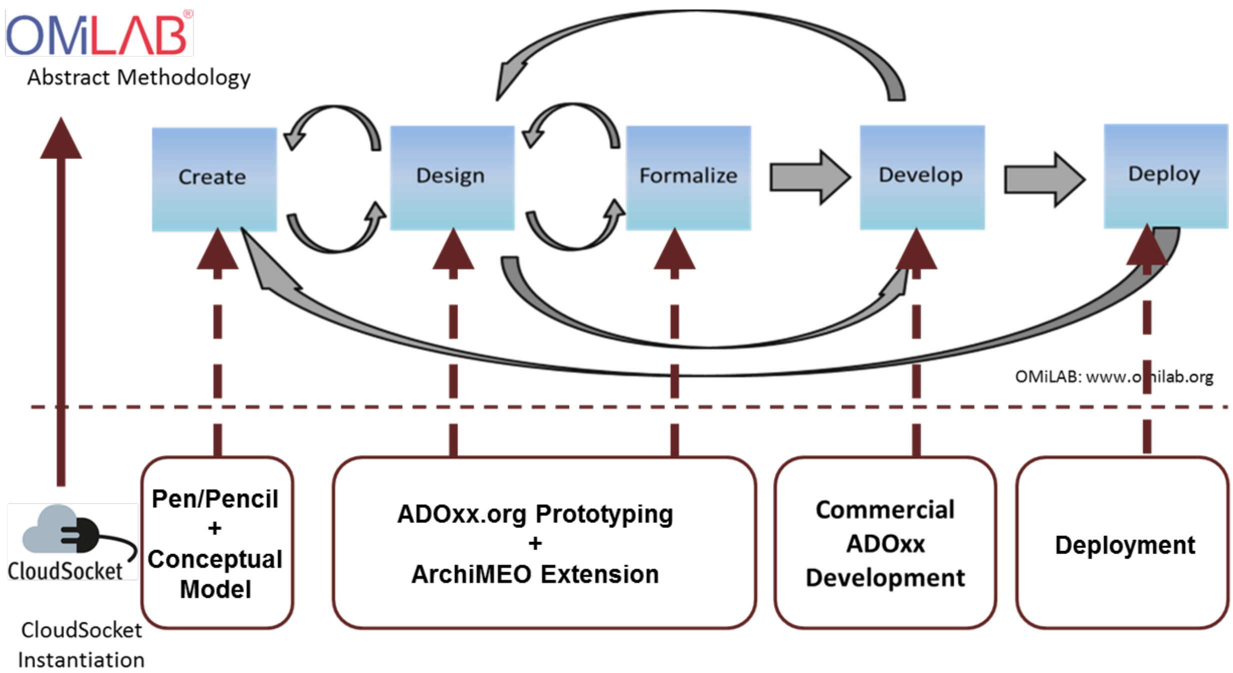

Figure 5 Modelling Prototype Development Methodology Applied in CloudSocket

The OMILAB Lifecycle consists of five phases:

1. Create Phase: In this phase the system under study, the intended application scenarios and the derived requirements are investigated. Typical support is pen and paper and common instruments of application specifications and requirement analysis. A conceptual meta model, describing the main concepts and relevant standards is recommended.
2. Design Phase: This phase specifies the modelling language with its required syntax, semantics and notation. Hence the so-called Platform Independent meta-model is specified, mechanisms and algorithms are described indicated the aimed functionality of the modelling tool.
3. Formalization Phase: The conceptual meta model must be transformed into software, hence before starting with the implementation, the Platform Independent Meta Model must be approved, if it is formally correct. This can be performed by mathematics (e.g. using FDMM (Fill et al. 2012)), by semantics (e.g. using RDF (W3C 2014)) or via rapid prototyping (e.g. using ADOxx.org - http://www.adoxx.org).
4. Development Phase: This phase transforms the platform independent meta model into a platform specific one and hence implements it into a meta model platform to realize the modelling tool.
5. Deployment Phase: This phase is concerned with the packaging, installation and deployment of the modelling tool.

2.1 Hybrid Modelling Development

This generic development methodology proposed by OMILAB is instantiated for the needs of the development of the BPaaS Design Environment in the following form:

1. Creation Phase: In this phase the domain and scope or the modelling framework are determined and the classes and the class hierarchy are defined. This phase is comparable to steps 1 to 5 of the approach for ontology development (Noy & McGuinness 2001). In CloudSocket this phase is performed by:
   - Determining the scope by analysis of business scenarios and deriving competency questions
   - State of the art surveys and literature research of existing modelling languages and ontologies in order to ensure the coverage of existing material.
   - A pragmatic design-based approach based on the results of the project plug-IT (Woitsch et al. 2009).
   - Continuous adaptation and feedback through typical collaboration instruments such as physical meetings, Internet workshops, publications, presentations and collaborative development.
2. Design Phase and Formalization Phase: Those two phases are combined using a rapid prototyping approach.
   a. In ADOxx.org rapid prototypes indicating the intension and the scope of a solution is implemented. This platform enables a quick development of prototypes, and hence enables continued feedback on the meta model design.
   b. In parallel a first prototype of the BPaaS Ontology is implemented, which extends the ArchiMEO ontology. The ontology is represented in a RDF 3.0 using the TopBraid modelling tool.
   c. The rapid prototypes are presented, discussed and feedback is provided,
   d. The meta model and ontology designs are adapted and another cycle of the rapid prototype is started.
3. Development Phase: In this phase the rapid prototype on the open and public platform ADOxx.org is transformed into the closed and commercial platform of ADOxxNP within BOC. The ontology prototype is made available for public use in the standard Turtle format.
4. Deployment Phase: This phase currently runs in parallel and deals with the Cloud-based provisioning of BPaaS Design Environment in form of SaaS.

The OMiLAB approach allows going back and forth between individual steps. There will be three iterations of rapid prototyping cycles for creation, design and formalisation. A first iteration has been done, the result of which is described in the next chapters. Chapter 4 contains the development of the modelling method. Chapter 5 describes the development of the ontology. Specific aspects of semantic lifting that are required to realize smart business and IT alignment are presented in chapter 6. In chapter 7 the current state of the prototype implementation is described.

2.2 Business Scenarios

This chapter introduces the methodology of the 'Create Phase'. We describe the approach that has been used to develop the BPaaS Ontology (chapter 5). Latter class definitions describing the intended semantics of the elements of the BPaaS modelling method (chapter 4) and the classes required for the semantic lifting. To capture the complexity of CloudSocket we draw upon the e3 value modelling method provided by Gordijn & Akkermans (2001) as starting point and guidance for the 'Create Phase. The e3 value approach "combines the rigorous approach of IT systems analysis with an economic value perspective from business sciences". The e3 modelling language provides the generic concepts that have to be present in an e-business model, such as CloudSocket.

In this regard we analysed the CloudSocket architecture - mainly for the BPaaS Design Environment - developed in Deliverable D4.1 'First CloudSocket Architecture' (CloudSocket 2015b). It describes the actions, relationships and dependencies of the CloudSocket Consumer, the CloudSocket Broker and the Cloud Provider. Based on the architecture we created the CloudSocket Eco System in the e3 value model notation (see section 3.1.3 for further information) which shows how different actors create and exchange value within e-business networks (see Figure 6).

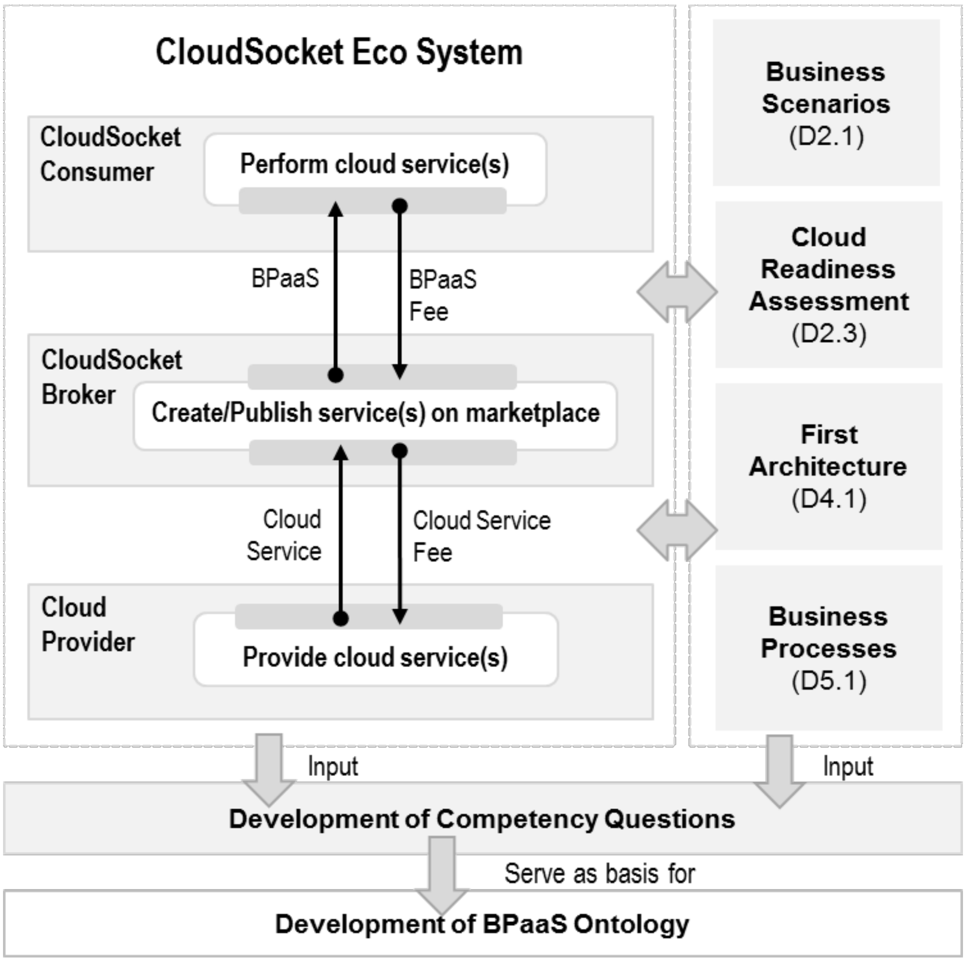

Figure 6: Development of BPaaS Ontology

The architecture defines the scope of the graphical modelling languages that have to be implemented in the BPaaS Ontology, such as BPMN, Organizational Model, Document Model, KPI Models, Workflow Model etc. (see BPaaS Meta Model Stack in section 4.2.1). For this purpose we reused, adapted and extended concepts of already available enterprise ontologies and created new models where necessary.

For the development of the BPaaS Ontology needed for the semantically lifting of the graphical languages we analysed business scenarios and developed a list of so called competency questions (see section 2.3) that the ontology should be able to answer. Business scenarios serve as a starting point, since they represent real situations as they occur in enterprises. In order to develop the competency questions we analysed the single components of the CloudSocket Eco System. Who are the involved actors? What kind of value objects is exchanged? What are the value activities? Who are the composite actors?

In order to answer these questions we draw upon project results: the business episodes developed in task T2.1 and described the Deliverable D2.1 'CloudSocket Use Case and Evaluation Criteria Analysis' (CloudSocket 2015a), the business processes defined in task T5.1 and described in Deliverable D5.1 'Initial CloudSocket Setup Report' (CloudSocket 2015c), and the cloud readiness criteria developed in task T2.3 and described in Deliverable D2.3 'Cloud Transformation Framework' (CloudSocket 2015d).

The project decided to adapt a simple scenario which guides an initial integration: the Christmas card sending scenario (see section 2.2.1). Being aware that a single process is not sufficient to determine the scope of the hybrid modelling framework, we analysed additional processes from the use case partners MATHEMA and BWCON (see section 2.2.2).

In summary it can be stated that the results of tasks T2.1, T5.1 and T2.3 enabled us to derive the relevant evaluation criteria for the development of the ontology concepts that describe the CloudSocket Eco System appropriately.

2.2.1 Sending Christmas Card Process

The Christmas Card process describes the steps for creating personalized Christmas cards with own greetings and photos and to send these card to friends and business partners. The key activities to perform the business process are following:

- Upload an individual image of the user or select picture from available images
- Upload text for the greeting card
- Upload recipients list in a predefined format
- Select preferred time slot to send emails
- Send Christmas greetings with emails

Three services which can support together the key activities are identified during the design phase, namely Card design, CRM (Customer Relationship Management), and Email. They are represented as lanes in Figure 7. In order to measure and evaluate the performance of the process, relevant KPIs (key performance indicator) were defined (see CloudSocket 2015c).

The process description and the KPIs are used as input to determine the scope of the BPaaS Modelling Method and the BPaaS Ontology. They must contain classes, attributes and relations that allow modelling of the business process and its requirements. Competency questions are used as a methodology to determine the scope (see section 2.3)

2.2.2 Business Processes from Use Case Partners

Being aware that a single process is not sufficient to determine the scope of the hybrid modelling framework, we analysed additional processes from the use case partners MATHEMA and BWCON, which were defined in Task T5.1:

- Customer support process
- Ordering process
- Human resource hiring process
- Purchasing process
- Kiosk partner management process
- Kiosk social media monitoring and response

The business processes definitions narrow down the business scenarios of Task 2.1, which describe different so called business episodes following the storytelling approach (see Deliverable D2.1 (CloudSocket 2015a)). As a result it has been derived what kind of Business Process as a Service (BPaaS) cloud solutions Cloud Consumers require with respect to functional and non-functional aspects. Task 5.1 provides concrete business processes and related key performance indicators as they might occur in small and medium enterprises (CloudSocket 2015c).

In Task T3.1 the business processes and KPIs are used to identify additional competency questions which help to determine the scope of the BPaaS modelling and BPaaS Ontology. They provided additional requirements. For example, in the human resource recruiting process (see Figure 8), file management with check-in/check-out is a requirement. Regarding security, employee records have to be stored in the EU and access has to be password protected.

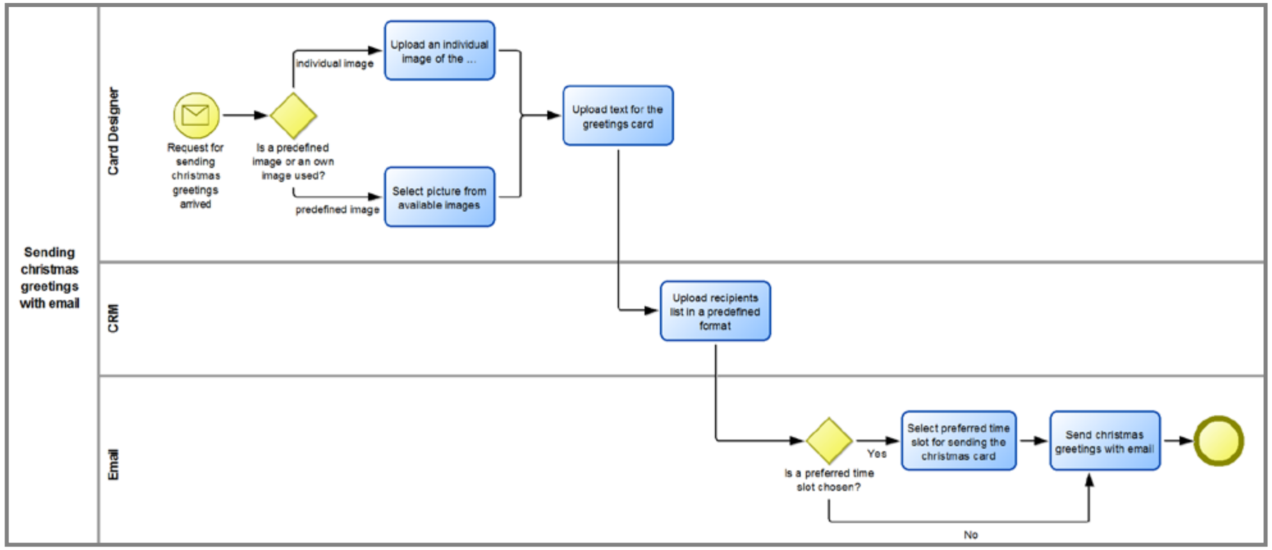

Figure 7: Sending Christmas Card BPMN Process

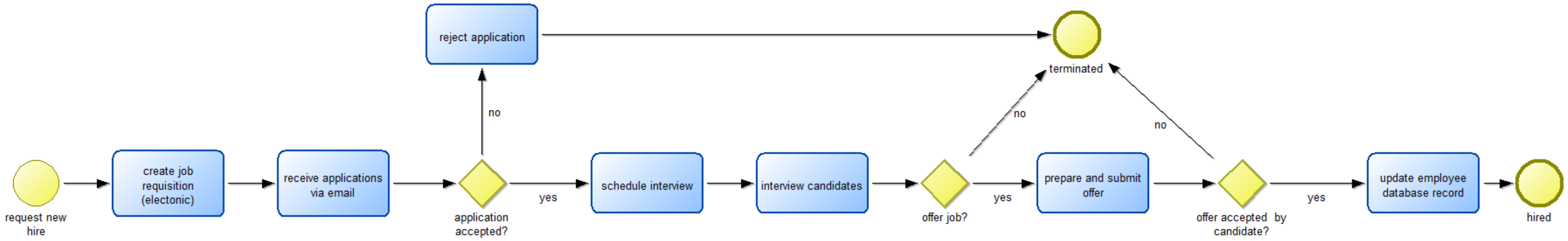

Figure 8: Human Resource Hiring BPMN Process

2.3 Determining the Scope of the Hybrid Modelling Framework

One way to determine the scope of the modelling framework is to sketch a list of questions that the system should be able to answer. These questions are called competency questions. They have been introduced by Gruninger and Fox (1994) as a method for enterprise engineering. The approach is widely known and was amongst others adopted by Leenheer & Mens (2008), De Brujin (2003) and Cardoso (2010). Competency questions are formulated in natural language and applied to the motivating scenarios. Fox et al., (1996, p. 134) describe competency questions as "benchmarks in the sense that the ontology is necessary and sufficient to represent the tasks specified by the competency questions and their solution". The goal is to develop a set of questions that serve as a basis to determine the scope of the ontology (Uschold & Gruninger 1997). Hence competency questions play an important role with respect to the intended purpose of the ontology. Four areas of competency questions have been identified that can be raised from Cloud Consumer perspective:

- General alignment questions: Questions regarding the mapping of business processes to workflows and bundles.
- Business perspective: Questions with respect to payment, contract, monitoring and support of the BPaaS and questions prospective customers might ask in order to assess the trustworthiness of the cloud service provider
- Security/legal perspective: Questions that are important with respect to security/risks/functionality or for SMEs in general or for SMEs operating in highly regulated industries
- Technical perspective: Questions with respect to data formats, platforms or implementation.

General Alignment Perspective
• Are there existing workflows and bundles for my business process? • Which parts of my business process can be served by existing workflows? • Which part of my BPs are Cloud ready (can be automated)? • Is this workflow realising my business process? • What are appropriate KPIs to measure the business process and the respective workflow? • How to develop the workflow? • What is the exact sequence of the process? • What are the exact input, output and method specification? • What is the level of possible business process adaptation? • How many workflow variants need to be realised? • What are the required deployment rules? • What are deployment constraints posed? • What skills / roles are needed to perform the workflow? • What is the needed technical infrastructure? • How is adaptive deployment realized? • What are the location requirements of the BP?

Table 1: Alignment Competency Questions

Business Perspective	
Cloud service provider	• How long is the provider on the market? • When was the company founded? • How many employees work for the provider? • Does the provider have customer references?
Location and communication	• Is the headquarters of the cloud service provider in my country? • Does the cloud service provider have a subsidiary in my country? • Does the provider offer the cloud service in my language?
Pricing/contract	• Can I pay per month? • Can I pay per year? • Can I terminate the contract whenever I want or do I need to consider deadlines? • Is there a minimum contract length (restriction) with respect to contract termination? • Can I try the service before I decide to use (to pay) for it? • When can I terminate or cancel a contract
Value added services	• Does the provider offer consulting/additional services? • Can I get support in my language? • Is there a dedicated person/employee on whom I can rely for obtaining support? • Does the provider offer support (e.g. hotline, web form)? • Can I have access to online documentation? • Is there a dedicated employee per client to have customised support? • Which kind of support is offered by which type of contract? • Who is monitoring and assessing the SLOs? • What are the qualifying conditions for the SLOs of the SLA • What are the maintenance periods for the BPaaS? • What are the penalties / rewards for an SLO? • What happens when the BPaaS service is modified?

Table 2: Business Perspective Competency Questions

Security/Legal Perspective	
Compliance	• What compliance support does the provider offer? • Is the provider compliant with FISMA, PCI DSS, HIPAA, SOX, GLBA, NERC CIP, or other regulations that are relevant to my industry? • Does the provider conduct security assessments (self- or third-party performed)? • Does the provider follow best practices and procedures for security management? • What kinds of incident management and reporting are supported by the provider? • Does the provider follow certain disaster recovery and data back-up plans
Legal aspects	• Is the provider compliant with the legal regulations in my country? • Can the provider guarantee that my data stay in my country or on a certain continent? • Can the provider guarantee the location of my data processing? • Is sensitive data (e.g. customer data) protected according to legal regulations? • Does the provider archive my data properly so that I have access if required? • Can the provider guarantee that my data stay on a specific position/location or are not processed by entities situated in undesired places

Cloud service provider security	• Is the provider certified in the field of cloud computing? • What specific key management and data protection strategies and methods are supported by the cloud provider? • What does the provider offer in terms of physical and systems security? • What are the disaster recovery and back up plans supported by the cloud provider? • How quickly can operations be restored if the main system goes down? • What reporting options/audit support is available? • What are the vulnerability detection and resolving capabilities of the cloud provider? • Does the provider conducts any kind of assessment (self-assessment or third-party one)?
Access and permission to the cloud service	• How many users can perform the cloud service? • Do the users have access to the information they are allowed to see? • Is it possible to define different permissions for different users (e.g. read, write, change…)? • Can I also give access to suppliers or customers if needed? • Can I define time slots for the access? • Is there monitoring/logging for user access available?
Restriction and scalability	• How can I ensure that an employee who leaves the company has no access to the cloud service? • If my company grows and I need more user accounts, can I extend the cloud service user accounts? • How long does it take to get additional user accounts? Can I get them quickly?

Table 3: Security/Legal Perspective Competency Questions

Technical Perspective	
Data	• What kind of input is required? • What kind of output will be the result?
Service	• How configurable is the service? • Does the service employ error handling mechanisms? • What is the abstract interface of the service (suitable for human- or machine-based interaction with the service in order to invoke it) • What are the restrictions for using this service?
Scheduling	• When does is need to start? • When does it need to be finished/completed?
Performance	• How reliable does the service needs to be? • How available does it need to be? • What is the response time? • How many users does it need to serve at the same time?
Logging	• Which parameters need to be tracked? • How long does the log file remain?

Table 4: Technical Perspective Competency Questions

3 STATE OF THE ART

3.1 Enterprise Modelling

Modelling the business processes, workflows and services in CloudSocket is part of enterprise modelling - the description and definition of the processes, structure, information and resources of an enterprise. According to Fox and Gruninger (1998) an enterprise model must supply the information and knowledge necessary to support the operations of the enterprise. Enterprise modelling techniques are developed in several fields such as business process modelling, information modelling, systems modelling, and enterprise architecture.

Models are representing part of reality or a vision in an agreed modelling language. Karagiannis and Woitsch (2010) use the term "knowledge space" to name what is represented in a model. The knowledge space represented in models is specified according to the four dimensions *form, content, interpretation*, and *use* (see Figure 9).

- The *form* represents the syntax and semantic.
- The *content* is seen as the domain in which knowledge engineering is applied.
- Depending on the intended use only a subset of the knowledge space's content might be of interest. Model types are specialized for specific uses.
- The representation of knowledge is either focused on machine *interpretation* or on human *interpretation*. In the context of *enterprise modelling,* graphical models typically are cognitively more adequate for *human interpretation. Enterprise ontologies*, on the other hand, are formal representations which can be *interpreted by machines*.

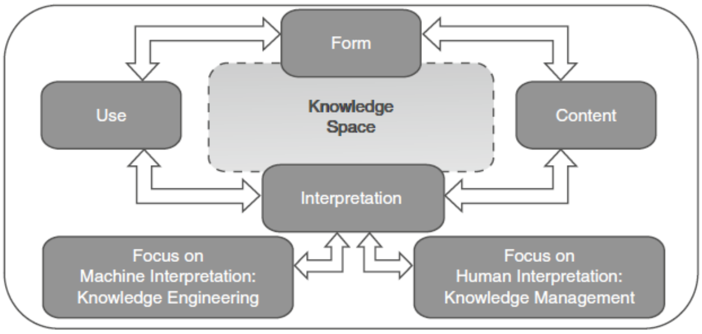

Figure 9: The Four Dimension of a Knowledge Space

3.1.1 Enterprise Architecture Descriptions

Enterprise architecture applies architecture principles and practices to guide organizations through the business, information, process, and technology changes necessary to execute their strategies. An Enterprise Architecture description contains a variety of models, which describe all relevant

- Business structures (e.g. business processes, organization structure, roles, business objects)
- IT structures (e.g. applications, data, information systems, infrastructure)
- and their relationships

Many frameworks were developed to assist in this task. Matthes (2011) points out that to date more than 50 enterprise architecture frameworks are available.

The Zachman framework is of particular interest because according to Matthes (2011) it is widespread and builds the basis for many other frameworks. The Zachman Framework is a two dimensional matrix (Zachman 2008).

Rows depict different perspectives of the role a stakeholder may take (named planner, owner, designer, builder and subcontractor), and columns represent the various aspects that should be considered. They are “different abstractions from or different ways to describe the real world” (20 p. 592). The aspects (rows) are named based on the fundamentals of communication. The interrogatives What (data), How (function), When (time), Who (people), Where (network), and Why (motivation) build the basis for the concise description of complex ideas (Zachman 2008).

Zachman gives no advice on how the enterprise architecture description should look: intersections of perspectives and aspects can be represented in models of various model types, like a data model or a process model. Those model types can in turn be represented in various languages. OMG has developed several specialized modelling languages for enterprise architecture modelling, for example Business Process Model and Notation (BPMN) (OMG 2011), Case Management Model and Notation (CMMN) (OMG 2013), and the Business Motivation Model (BMM) (OMG 2010). The purpose of these graphical modelling languages is to support communication between human stakeholders. They are not intended for machine interpretation - although there does exist execution engines for BPMN.

TOGAF is another well-known EA framework (The Open Group 2011). The overall enterprise architecture as composed of a set of closely inter-related architectures: Business Architecture, Information Systems Architecture (comprising Data Architecture and Application Architecture), and Technology (IT) Architecture (The Open Group 2012).

The ArchiMate Standard (The Open Group 2012) introduces an integrated language for describing enterprise architectures. ArchiMate fits into the TOGAF framework as it provides concepts for creating a model that correlates to its three architectures (layers). ArchiMate provides a graphical representation of its language elements based on UML class diagram but customized and limited to a small set of modelling constructs in the interest of simplicity of learning and use. The standard claims that architecture descriptions “are formal descriptions of an information system, organized in a way that supports reasoning about the structural and behavioural properties of the system and its evolution” (The Open Group 2012). However, the ArchiMate language has one shortcoming: it is intended for human interpretation and not suitable for *automatic* reasoning for two reasons. It is too coarse grained as it only contains basic concepts and relationships that serve general enterprise architecture modelling purposes (Thönssen 2013).

3.1.2 Enterprise Ontology (EO)

The purpose of ontologies in enterprise modelling is to formalize and establish the shareability, re-usability, assimilation and dissemination of information across all organizations and departments within an enterprise. A machine-understandable and interpretable architecture description would allow to answer questions like "which processes are affected by the replacement of an application or service?", "which data is required by the process?", "why did we choose this specific service?"

As shown by (Kang, Lee, Choi, & Kim 2010) and (Hinkelmann, Merelli, & Thönssen 2010) an enterprise ontology (EO) can meet this request. Describing enterprise architecture as an ontology started in the 1990s with TOVE (Fox, Barbuceanu, & Grüninger 1996), The Edinburgh Enterprise Ontology (Uschold, King, Moralee, & Zorgios 1997) and the organizational memory (Abecker, Bernardi, Hinkelmann, Kühn, & Sintek 1998). More recent work are the Context Based Enterprise Ontology (Leppänen 2005). Den Haan (Den Haan 2009) has used an enterprise ontology to realize a Model-Driven Enterprise Engineering.

In CloudSocket we build on the semantically enriched Enterprise Architecture Description (seEAD), which is logically structured into four parts (see Figure 10):

- Following Bertolazzi et al. (2001), a Top-level Ontology (TOL) comprises generic concepts like time, location and event
- The Enterprise Upper Ontology is the ontological representation of ArchiMate, based on the ArchiMate Specification (The Open Group 2012), represented in an executable ontology language as developed in the context of the semantic web (Allemang and Hendler 2011).
- The Top Level Ontology and Enterprise Upper Ontology together build the basis for the ArchiMEO ontology which adapts and enhances the ArchiMate standard by additional concepts and relations, for example to describe business processes.
- ArchiMEO can be extended by application specific ontologies, comprising specific concepts of a certain enterprise or domain.

In the APPRIS project, seEAD has been applied for building an early warning system for risks in the supply chain (Emmenegger et al. 2013). Other applications have been in the areas of contract management (Thönssen & Lutz 2012) and master-data management for heterogeneous data stores (Hinkelmann et al. 2013).

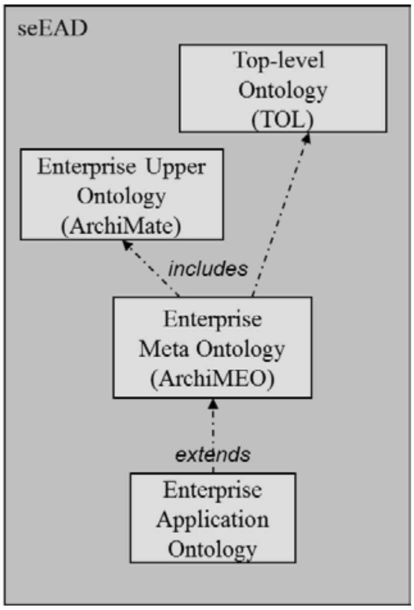

Figure 10: The ArchiMEO Structure

3.1.3 Virtual Enterprise Ontology

This section depicts the concept of virtual enterprise (also referred to as networked) ontologies which describe how different actors create and exchange value within networks. Nowadays enterprises operate in globalized, complex and highly competitive markets. This goes along with shorter planning and implementation cycles, rapid environmental changes and distributed work environments. In the era of digital transformation enterprises are forced to continuously rethink and adapt their business strategy in order to maintain their competitiveness (Wagner 2004).

This new way of doing business led towards a broad field of research with respect to intra-organizational collaboration referred to as 'virtual enterprise'. It describes a phenomenon which came up with the emergence of new technologies. Latter allow organizations to collaborate with other organizations across boarders in order to create value. In this way enterprises can take advantage of the capabilities of a network while maintaining their independency. Although the term 'virtual enterprise' lacks a clear definition there is a set of common characterizing elements (Camarinha-Matos & Afsarmanesch 1999). According Byrne (1993) "the virtual corporation is a temporary network of independent companies - suppliers, customers, even erstwhile rivals - linked by information technology to share skills, costs, and access to one another's markets. It will have neither central office nor organization chart. It will have no hierarchy, no vertical integration".

Walton & Whicker (1996) state that "the Virtual Enterprise consists of a series of co-operating 'nodes' of core competence which form into a supply chain in order to address a specific opportunity in the market place". Camarinha-Matos & Afsarmanesch (1999) combine aforementioned definitions stating that "a virtual enterprise is a temporary alliance of enterprises that came together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks".

3.1.3.1 Virtual Enterprise Taxonomy

The general definition of virtual enterprises can be applied to many different organizations. A classification helps to capture the complexity. Camarinha-Matos & Afsarmanesch (1999) proposed taxonomy for virtual enterprises as depicted in Figure 11:

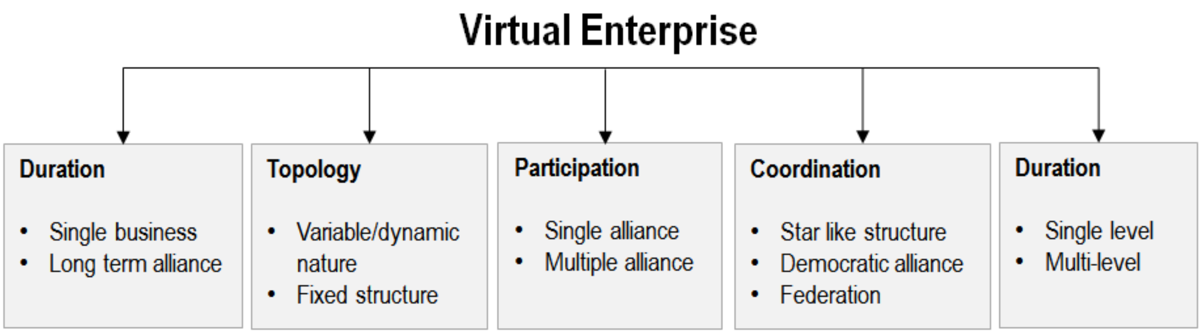

Figure 11: Virtual Enterprise Taxonomy

- Duration: Some organizations collaborate for a single opportunity, while others cooperate for a specified long term time span.

- Topology: In case of a variable/dynamic nature network partners can dynamically enter and leave the network. A fixed structure topology is given in networks with an established supply chain. There is little variation with respect to stakeholders that join or leave the network.

- Participation: In a single alliance the organization participates only in one network at the same time. If an organization participates in different networks at the same time it is referred to as multiple alliances.

- Coordination: In a star like structure there is one organization which dominates by managing all information flows within the network. For instance, a car manufacturer might be surrounded by different external suppliers. This network of suppliers is relatively fixed, since the enterprise established contracts in order to ensure constant delivery of required parts for manufacturing. In a democratic alliance all network participants cooperate on an equal basis. In some cases it requires an enterprise which is responsible for the administration, and monitoring of the organizational structure. The federation is the extreme case where organizations create a joint coordination structure. The close relationships allow organizations to benefit of joint management of resources and skills within the network.

- Visibility: The visibility described "how far, along the network, can one node see the virtual enterprise cooperation. In most cases a node only sees its direct neighbours" (single visibility). In case of multilevel visibility an organization has access to information about other nodes of the network. This information might lead to benefits for the whole network, such as better operation.

3.1.3.2 The e3 Value Ontology

Business models in the real world are limited to a textual or ad-hoc graphical representation. However, modelling of e-business models has to consider how value creation and exchange takes place in a network of different actors. The e3 value ontology takes specific properties of e-business models as well as properties of services that have a grounding in the real world into account. This allows describing how value is created, exchanged and consumed in a network with different actors. Hence the ontology can be used to express and discuss real world business cases and moreover to develop and compare different scenarios in order to support decision making.

Gordijn & Akkermans (2001) propose three distinct perspectives that are of central importance for the development of an e-business model: the business value viewpoint, the business process viewpoint and the system architecture viewpoint. The business value viewpoint describes how economic value creation, exchange and consumption take place in a network which consists of different actors. The business process viewpoint refers to the execution of the processes that allow aforementioned creation, exchange and consumption of value. The system architecture viewpoint includes all technical components that enable the execution of the process. The e3 value ontology defines a number of concepts which are explained in the following (Gordijn, Akkermans & van Vliet 2000):

- Actor: An actor is an independent entity who aims to make a profit. An actor can also be a legal entity. In an e-business model each actor should be able to make a profit.
- Value object: Objects are exchanged by actors, such as products, services, money or consumer experiences
- Value port: In order to request or to provide a value object actors use the value port. The idea of the value port is to abstract from the business process (higher level) and to depict how value objects can be integrated into the eco system of the e-business model.
- Value interface: Shows the value object that can be exchanged between two actors in return for another value object.
- Value exchange: Value ports are connected with each other through the value exchange (can also take place between multiple actors > 2)
- Value offering: This is an assembly of value exchanges.
- Market segment: Actually market segments are a marketing concept. In context of e3 value ontology the market segment describes different groups of actors that have a set of common properties.
- Composite actor: Partnership between different actors who collaborate in order to offer objects of value.
- Value activity: Activity that is performed by an actor in order to make a profit or to increase its utility

The e3 value ontology is further elaborated in section 3.3 Service Processing: Discovery and Composition.

3.1.4 Task Ontologies

The use of task ontologies and their linkage to business goals can leverage and support the semantic discovery and composition of business services and BPaaS for the following reasons: (a) it provides a way functionality can be decomposed into fine-grained levels which can then be fulfilled by technical/software services; (b) there is a mapping to business goals such that there is not only traceability but also the construction of services which are meaningful and have an impact on the business.

Task ontologies go back to the research in problem solving frameworks where tasks are connected to problems and are linked to particular implementation methods. As such, such frameworks have followed a top-down approach for solving any kind of problem which starts from the problem itself and its mapping to a task and going down to identifying concrete methods which can be used for the realisation of fine-grained tasks.

It is not the purpose of this deliverable to review the state of the art in the modelling of task ontologies and the use of respective formalisms. On the contrary, its goal is to review current task ontologies which have been used in different (business) domains in order to examine whether some of them could be adopted in the context of the CloudSocket project. The review focuses on the richness of the ontologies and the depth of the levels involves which can well be adopted towards the goal of having an integrated task ontology which could be used to capture the functional variability across different business domains. To this end, this section is separated into different sub-sections which focus on the analysis of each task ontology proposal.

3.1.4.1 APQC Process Classification Framework

APQC Process Classification Framework (APQC 2014) is a taxonomy of business process (BP) which facilitates their tracking and comparison according to performance aspects. It comprises five levels which start from a generic business process categories and go down to particular activities able to fulfil the functionality of these business processes. Each element of this taxonomy is also linked to performance metrics able to assess it for which formulas and respective units are also provided. The highest level comprises 13 BP categories related to the development of products and services, the marketing, the delivery of these products and services, the management of human capital, IT and financial resource and so on. The first 5 categories map to operational/core business processes while the rest to management and support processes.

In (Teuteberg et al. 2009), APQC is transformed into a domain ontology which is exploited in order to support the development of a framework for BP benchmarking. This framework addresses the syntactic and semantic heterogeneity problem through the use of ontologies which are exploited in order to: (a) translate language constructs from one conceptual modelling language to another one; (b) to explicate the semantics of language constructs; (c) to determine equivalent language constructs; (d) to represent general domain knowledge (APQC); (e) to map elements from one process model to elements of another one.

3.1.4.2 MIT Process Handbook

The MIT Process Handbook (MPH) (Malone 2003) comprises three kinds of contents: (a) generic models of typical business activities; (b) specific case examples; (c) frameworks for classifying the aforementioned types of knowledge. Various generic activity models have been modelled which contain in total 1400 entries. On the other hand, the case examples map to three particular domains (supply chain, hiring & e-business) and contain 570 entries, while the knowledge structuring frameworks comprises 3252 entries and contains generic verbs taxonomies as well as other types of activity categories. Each entry (e.g., "sell") comprises a description of the activity, a list of its parts (e.g., "Identify potential customers"), potential properties of the activity (e.g., location, cost, time), a list of related processes, specialisations of this activity (ways activity can be performed), categorisations of specialisations called bundles (e.g., "shell how" & "shell what" as well as example and view bundles), activity uses (i.e., as part of another activity), generalisations (e.g., "exchange" & "provide"), and trade-off tables (where specialisations of the activity are compared according to activity properties).

As it can be seen, MPH is quite extensive and contains different models and taxonomies of activities which could be exploited by the CloudSocket project. It also includes cases which could be considered as best, typical or bad practices is specific domains that can also be regarded as domain task models linked to particular business model archetypes allowing comparisons between industries. In the sequel, we shortly analyse those MPH artefacts that seem to the ones mostly related to the current research goal of the analysis.

The MPH's generic models of process activities comprise activities that can be exploited across different business domains. Four generic models are currently included, namely: (a) MIT Business Activity Model (BAM), (b) MIT Business Model Archetypes, (c) a collection of business process models defined elsewhere and (d) basic coordination process models. From these models, we actually analyse only the first one which seems more relevant to our main research goal and seems to be more detailed and structured. In fact, the models in the collection of (c) can be seen as a kind of specialisation of the top activity in the MIT BAM which can be linked to different types of groupings of the respective sub-activities. To this end, this has led to cross-referencing the contents of the models in this collection to the MIT BAM leading to a quite extensive activity categorisation scheme. We would like to highlight that in this collection of models, there exist widely-known process reference models such as SCOR.

The MIT Business Activity Model contains at its root the activity of "Produce as a Business" and contains five basic parts "Buy", "Make", "Sell", "Design" and "Manage". Each activity part has then sub-parts thus mapping in overall to taxonomy of activities of 3 levels at most. The top activity also has a specialisation called "Produce as a typical business" to cover the way business is conducted for large companies.

To enable the sophisticated browsing of the MPH entries, it was evident that all entries must be part of the specialisation hierarchy. To this end, the MPH has developed an extensive classification structure for the specialisation hierarchy which can be used for the classification of any kind of activity irrespectively of whether it concerns business aspects or not. This classification structure relies on the use of a generic verbs hierarchy which is finally linked at the lowest level to the MIT BAM model. The top node in this hierarchy maps to the activity of "Act" which is then further classified into 8 generic verbs: "create", "modify", "preserve", "destroy", "combine", "separate", "decide", "manage" & "unclassified". The first four activities can occur for any kind of object, while the next two can act on multitudes of objects. The seventh and eighth activities are informational activities which could be included in the first action partitioning but have been mapped to a new partition in order to highlight their significance to business. The final activity is a placeholder for other types of activities that cannot be included in the first three partitions. To further illustrate the levels involved, consider the case of the "sell" activity. To reach the second level hierarchy, we need to go up three levels involving the activities of exchange (1st), move (2nd) and modify (3rd).

3.1.4.3 The Value Reference Model (VRM)

The Value Reference Model (VRM) has been developed by the Value Chain Group (2015). The goal of the VRM is to support and to enable enterprises to integrate four domains of central importance for their business: enterprise management, product development, supply chain integration and customer relationship management. In order to support an integrated value chain the VRM provides unified process reference architecture. The key elements of this architecture consist of inputs/outputs, metrics and best practices. The VRM is separated in a three level process hierarchy: plan, govern, and execute ad depicted in figure

The Govern level addresses decision based processes and provides required policies, rules and procedures that allow implementing the plan and execute processes. The Plan level provides processes which allow appropriate execution by matching organizational objectives with available resources. The Execute level is composed of processes which are performed in order to meet customer needs.

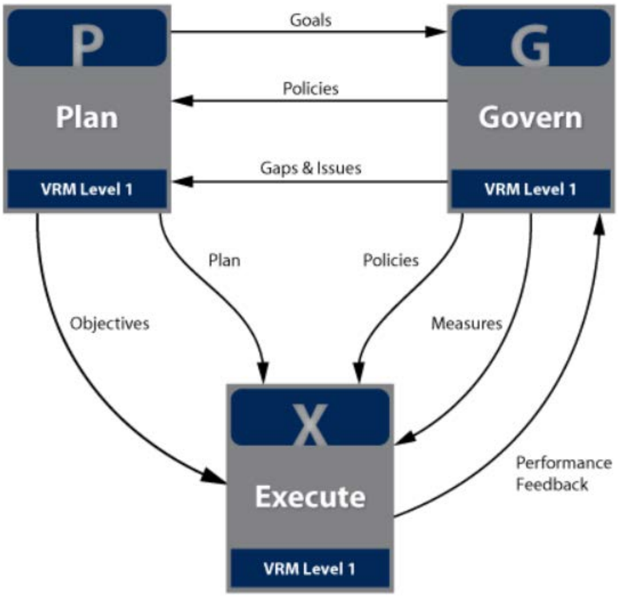

Figure 12: Value Reference Model Levels

3.1.4.4 XEROX Management Model

XEROX has created a reference model (Barr 1998) for its operational business processes which maps to a two-level hierarchy and includes activities like "Time to Market" and "Integrated Supply Chain". In total, 51 activities are captured in this model.

3.1.4.5 RosettaNet

It is an industry-driven e-business process standard which includes the specification of inter-company public processes in the forms of Partner Interface Processes. It includes 50 PIPs. There have been already approaches like the one in (Haller et al. 2008) which has attempted to semantically enrich it. In the aforementioned approach, the respective ontology produced comprises two main levels where the first one maps to 8 clusters and the second level into segments. For instance, the cluster "Order Management" is detailed into 4 segments including the process of PIP3A "Quote and Order Entry".

3.1.4.6 Products & Services Taxonomies

There exist various taxonomies focusing on the classification of products and services. As a service can be considered as a kind of component able to fulfil the functionality of a task, such taxonomies can be regarded as relevant towards our analysis goal. However, the mixing of potential domain entities, like products, and respective offered functionalities/services has to be taken with care if such taxonomies are to be exploited. These taxonomies include UNSPSC, eCl@ss and eOTD which follow a horizontal approach spanning different industries. UNSPSC comprises 4 levels mapping to segments, families, classes and actual products/services. eCl@ss also follows a 4 level classification system with a different coding scheme. In addition, for each class in the hierarchy, eCl@ss includes an application class which involves the description of certain properties to be used to describe the respective product/service. eOTD is ECCMA's Open Technical Dictionary which includes a language-independent database of concepts and respective terms, definitions and images linked to individuals, organisations, locations, goods, services, processes, rules and regulations. However, in contrast to the other two taxonomies, eOTD does not include a class hierarchy (or even class-property relationships) so it is actually a flat taxonomy of different types of artefacts. Based on these classifications, various approaches have been proposed to ontologise them, including those in Hepp (2005) and Klein (2002). The approach in Hepp (2005) seems quite sophisticated as it promises to ontologise any kind of product/service classification system by being able to deal with the respective heterogeneity involved.

3.1.4.7 SCOR

SCOR has been developed by the Supply Chain council as a domain-specific business process reference model across industries for supply chain management. This reference model comprises three main levels: (a) process type linked to the scope of the respective business process where 5 process types have been identified: plan, source, make, deliver and return; (b) configuration level explicating the core process categories; (c) process element level decomposing process categories into concrete process elements which are described according to their functionality, input, output, performance metrics, best practices, system capabilities and tools. The last level actually maps to the domain activities that can be used to define concrete business processes. In this respect, this level actually represents a kind of task ontology which comprises all possible activities/tasks that can be involved in the supply chain domain. Through the addition of the higher levels, we then have a task taxonomy where the activities at the lowest level can be categorised and partitioned into certain groups.

The SCOR reference model has been originally informally described. However, there have been approaches which have migrated it into the semantic space. It is worth mentioning two of these approaches. The first approach, relying on a particular PhD thesis (Lin 2008), focuses on extending SCOR towards generalising existing elements into 3A concepts (Activity, Artefact, Actor-Role) defined in the General Process Ontology (GPO). A goal ontology is also exploited for which hard goals are derived from process elements and their I/O while general soft goals are modelled mapping to performance attributes of Reliability, Responsiveness, Flexibility, Cost and Assets from which domain-specific goals are derived from the metrics of these attributes. In (Zdravkovć et al. 2011), a framework for semantic enrichment of reference model has been proposed which relies on a set of tools that are able to transform reference models along with the information of domain ontologies into a set of semantic models covering also application ontologies. This framework has been applied in the case of SCOR where it has been mapped to a KOS knowledge organisation system which is represented in a computerised language (OWL).

3.1.4.8 Domain-specific task ontologies

Various task ontologies have been proposed for particular domains. The scope here is not to provide an exhaustive review of all possible approaches but just an indication of the existence and possible richness of those ontologies in their corresponding domain in order to exemplify that the goal of selecting and enriching an existing task ontology for a certain domain is not hard to be achieved.

A task ontology for the tourism domain is proposed in (Park et al. 2012). This ontology is organised according to the 6 main top concepts in the domain which are always associated to the user goal (e.g., accommodation). Then, each such concept is associated to generic activities that can be performed on/for it. For instance, in case of accommodation, there are 6 activities that can be performed, such as search, reserve, move, and compare. Next, in a second level and again for a specific concept, an activity is further drilled down into more concrete activities that also involve more concrete domain concepts (either related to a manipulation of an object by the activity or to some particular aspect like location or time). For instance, the search for accommodation can be mapped to searching for a hotel, condominium, or lodging nearby or in a specific location (either here, a zone or a city). For map-based mobile guides, a specific typology of tasks is proposed in (von Hunolstein and Zipf 2003). This typology maps to two levels of tasks where the first level comprises 6 entries and the second level 2 or 3 entries which further discriminate the type of concrete functionality that is offered. Each entry is also associated to a communication goal or purpose. For instance, the top entry of Locator is mapped to 3 sub-tasks related to the type of positioning (e.g., own, others, or other objects) and is also related to particular purposes (e.g., orientation, navigation & overview).

In (Albrecht 1996), a set of 20 GIS operations grouped into 6 categories, catering mainly for the user perspective, were proposed compiled from 144 GIS analytical operations and functions from different GIS systems.

In the medical domain, two approaches have been spotted. In (Battisto et al. 2009), after performing a tasks analysis from human factor research in order to identify nursing capabilities and respective activities, a set of 10 top-level activities was derived including activities like administering medication and patient assessment. Each top-level activity is then associated to specific examples of more concrete activities. For instance, in the case of administering medication, the more concrete activities mapped are related to retrieving medications, preparing medications, documenting administration of medication, administering medication and monitoring intravenous pump. In (Shahar et al. 1998), a set of tasks which can be exploited for the support of skeletal plans execution was proposed. This set includes tasks like the verification, validation and modification of an executing plan.

3.1.5 Basis for Ontology Development

There are different kinds of research results that help to identify cloud service providers and/or cloud services. This section discusses the foundation for developing the ontology, classing and attributing the elements. We have made use of two different views.

1. Classification model for Infrastructure as a Service (Repschläger et al. 2011)
2. Cloud Service Level Agreement Standardisation Guidelines

3.1.5.1 Classification model for Infrastructure as a Service

The classification model is based on Repschläger et al. (2011), and is built on six so-called target dimensions (1st level). The dimensions reflect the general objectives of cloud computing and include:

- Flexibility
- Costs
- Scope & Performance
- IT Security
- Reliability and Trustworthiness
- Service & Cloud Management

Each target dimension contains abstract classification criteria (2nd level). These criteria are based on Repschläger et al. (2011) elements for Infrastructure as a Service and include for the dimension of costs

- Price model
- Service charging

Each of these abstract classification criteria contains operative classification criteria (3rd level), which is for example:

(1st Level) (2nd Level) (3rd Level)

- Costs
  - Price Model
    - Price class
    - Price resilience
    - Price options
    - Price transparency
    - Payment options
    - Payment method.

The classification model is shown in Figure 13. The benefit of this model is to structure the topics and elements according to the cloud dimensions and to identify the requirements and KPIs for the service and the provider more easily. Even the model is designed for Infrastructure as a Service, it provides also valuable input for cloud services in general and Business Processes as a Service.

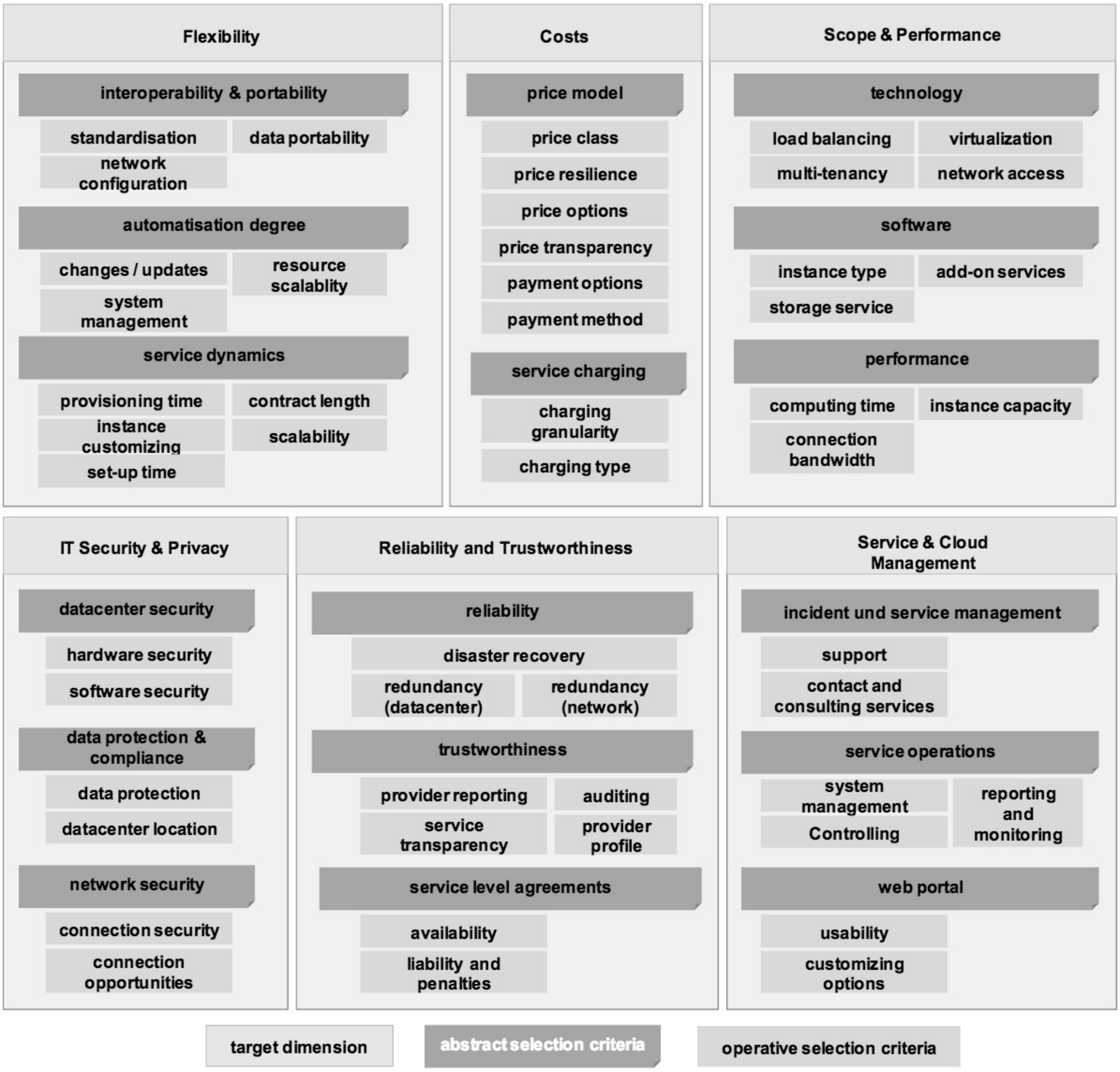

Figure 13: Cloud Service Provider Selection Model

3.1.5.2 Cloud Service Level Agreement Guideline

As an outcome of the European 2020 initiative “Digital Agenda for Europe”, the Cloud Service Level Agreement Standardisation Guidelines have been published in order to standardize and streamline the terminologies and understanding of Cloud Service Level Agreements (C-SIG SLA 2014).

The guideline is kept neutral from a technological and business model perspective of the cloud service in order to make it applicable for the wide range of cloud services on the market. It also considers standards and guidelines such as European Network and Information Security Agency (ENISA), National Institute of Standards and Technology (NIST), International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). It provides a founded basis for describing “Cloud Essential Characteristics”, which are:

a. Broad Network Access
b. Measured Service
c. Multi-tenancy
d. On-demand self-service
e. Rapid elasticity and scalability
f. Resource pooling.

The structure of the Service Level Agreement guideline is based on Service Level Objectives (SLO) that have been grouped into four major categories:

- Performance SLOs
- Security SLOs
- Data Management SLOs
- Personal Data Protection SLOs

Figure 14 shows the graphical structure of the Service Level Agreement.

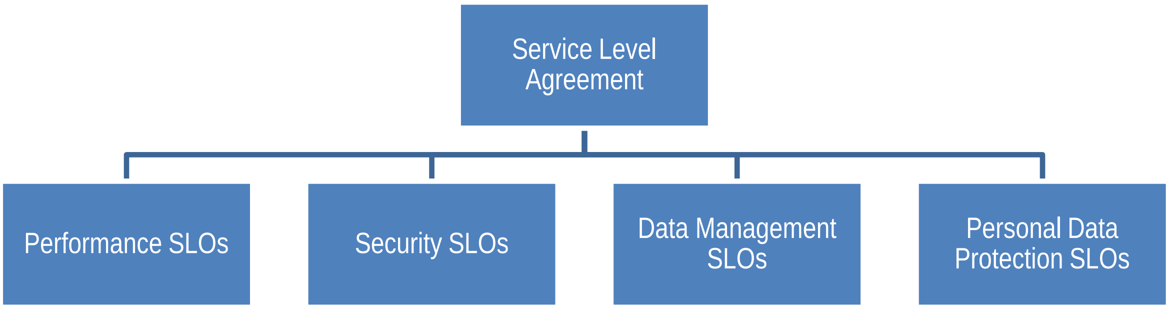

Figure 14: Service Level Agreement and Service Level Objective Structure

These definitions, attributes and objectives have been used as a basis for developing the ontology and updated with other attributes that we defined to be essential for the purpose of the CloudSocket project. The following subchapters describe the content of the service level objects in a roughly way. Some of the objectives have a high degree of detail that can just partly be specified by the annotating role. These will be questioned by the ontology in a more appropriate way and translated into respective objective.

3.1.5.3 Performance Service Level Objectives

The performance SLO describe the objectives that are related to the performance of the cloud service, usually between the cloud service customer and the cloud service provider, which contain the following:

- Availability
- Response Time
- Capacity
- Capability Indicators
- Support
- Reversibility and the Termination Process

Availability can be defined by the “Level of Uptime”, “Percentage of Successful Requests” and the “Percentage of timely service provisioning requests”. The “Level of Uptime” can be calculated as follows: Uptime = Total Possible Available Time – (Total Downtime – Maintenance Downtime)

The “Percentage of Successful Requests” and “Percentage of timely service provisioning requests” describe objectives for error-free requests and requests that have been tackled in the respective time period. The Response Time covers the “Average response time” and the “Maximum response time” that are allowed for the service to take in order to react and/or interact with the user or other services. The Capacity objective is described by the “Number of simultaneous connections”, “Number of simultaneous cloud service users”, “Maximum resource capacity” and the “Service Throughput”. Maximum resource capacity targets the resources such as data storage, memory, number of CPU cores, etc. The Service Throughput refers to objectives like “Requests per minute”. The Capability Indicators sub objectives include the “External connectivity” such as services that are seen as external from the respective cloud service perspective (e.g. CRM system for getting the customer data). Support covers the related objectives like Support hours, support responsiveness and resolution time. The last objective "Reversibility and the Termination Process" is described by the data retrieval period, the data retention period and the residual data retention.

3.1.5.4 Security Service Level Objectives

The Security SLO describes objectives for improving assurance and transparency of the cloud service. It contains the following sup objectives:

- Service Reliability
- Authentication & Authorization
- Cryptography
- Security Incident Management and Reporting
- Logging and Monitoring
- Auditing and security verification
- Vulnerability Management
- Governance
    - Service Changes.

Service Reliability is described by the level of redundancy, to make sure that the service switches over to another instance in case of failure. It also contains the service reliability that "describes the ability of the cloud service to perform its function correctly and without failure over some specified period".

The Authentication & Authorization objective covers:

- User authentication and identity assurance level that can be based on standards like NIST SP 800-63 (Electronic Authentication Guidelines), ISO/IEC 29115 (Entity Authentication Assurance Framework) or the Kantara Initiative's Identity Assurance Framework (IAF)
- Authentication mechanisms that are supported by the Cloud Service Provider (CSP)
- Mean time required to revoke user access
- User access storage protection to protect the user access credentials
- Third party authentication support for third party authentication.

Cryptography covers the "Cryptographic brute force resistance" objective which might be aligned to the ECRYPT II security level recommendations (Smart 2012) or the FIPS PUB 140-2 recommendation (NIST 2001). The "Key access control policy" objective deals with strength of the access protection. "Cryptographic hardware module protection level" deals with the protection using dedicated hardware modules. Security Incident Management and Reporting is described by the "percentage of timely incident reports", the "percentage of timely incident responses" and the "percentage of timely incident resolutions".

The Logging and Monitoring objectives include "logging parameters", "log access availability" and the "logs retention period". For auditing and security verification, the applicable certifications of the service are critical and consider, beside the certification standard and certifying body also values like expiration date and renewal period. The Vulnerability Management includes the "percentage of timely vulnerability corrections" and the "percentage of timely vulnerability reports", as well as, "reports of vulnerability corrections". For the governance of the service, the service changes and the "cloud service change reporting notifications" are central for keeping track of the service evolution and the changes that might impact the stakeholders or other services that need to cooperate.

3.1.5.5 Data Management Service Level Objectives

The Data Management service objectives include the following objectives:

- Data Classification
- Cloud Service Customer Data Mirroring, Backup & Restore
- Data Lifecycle
- Data Portability.

The Data Classification objectives contains a specification of the data that is processed by the service by making use of the following categories:

- cloud service customer data
- cloud service provider data
- cloud service derived data

The "cloud service customer data use by the provider" objective "describes stated policy for any intended use of cloud service customer data". The cloud service derived data use specifies what data is generated by the service including the use and the rights. Cloud Service Customer Data Mirroring, Backup & Restore deals with mirroring, frequency and the backup and recovery mechanism, which might be also related to the geographical location of the cloud service provider's data centres. Beside these objectives, it deals also with the "Data Backup Frequency", "Backup Retention Time", "Maximum Data Restoration time" and the "Percentage of Successful Data Restorations".

The Data Lifecycle objectives include "Data deletion type" for defining the weak or strong deletion of the data, the percentage of timely effective deletions and the percentage of tested storage retrievability. Data Portability makes sure that the data portability format and the data portability interface can be specified in case of changing the provider.

3.1.5.6 Personal Data Protection Service Level Objectives

The Personal Data Protection categories is meant to be for cloud services that have act as data processor, and include the following objectives:

- Codes of conduct, standards and certification mechanisms
- Purpose specification
- Data minimization
- Openness, transparency and notice
- Accountability
- Geographical location of cloud service customer data.

Codes of conduct, standards and certification mechanisms objectives describe which standards and certifications are applicable when dealing with personal data. The purpose specification makes sure that both parties agree on the purpose of processing, so that the data is not used for other purposes or interests. Data minimization objectives deal with the temporary data retention period and the cloud service customer data retention period. Openness, transparency and notice refer to the involvement of subcontractors that also deal with the data, if the cloud services makes use of it, and the categorization of data (e.g. health-related, financial, etc.). The accountability includes the agreed consequences, if a breach of the agreed policy happens. Geographical location of cloud service customer data are defined by the geolocation list where the data can be processed and the selection where it is allowed for being processed.

3.1.6 Meta-modelling

A well-known approach in providing concepts and instruments for both (a) the appropriate representation of the layers with concept models as well as (b) the realization of weaving mechanisms is conceptual modelling. Meta modelling (Karagiannis and Höfferer 2006) is introduced as a realization approach to develop domain-specific IT-supported concept modelling. Based on Strahringer (1996), Karagiannis and Kühn (2002) a layered approach for conceptual modelling is used.

Meta models can be specified with a meta modelling language that is derived from a meta meta model. In the following the most prominent meta meta models based on (Kern et al. 2012) are mentioned: (a) Ecore from the Eclipse platform (Budinsky et al. 2004), (b) GOPRR from MetaEdit+ Platform (Kelly and Tolvanen 2008) and (c) MS DSL Tools and MS Visio (Cook et al. 2007). Additionally, the following meta meta models have been introduced: (d) MOF (OMG 2015a), which is realized on different UML Profile platforms (e) ADOxx based on the equally named platform (ADOxx 2015), (f) Obeo Designer on Eclipse (Obeo 2015) and (g) Generic Model Environment GME (ISI 2008).

Besides the technical functionality, the provision of a model repository as well as the flexible adaptation approach, the ADOxx platform is collaboratively developed via the ADOxx.org community with more than 700 developers world-wide and more than 3000 stakeholders. Hence, in order to guarantee sustainability after the project period, the whole conceptual implementation is performed on the open collaborative CloudSocket development space on ADOxx.org (http://www.adoxx.org).

Conceptual models are a commodity in expressing processes and can be defined by any of the aforementioned meta models. There is a plethora of different process notations such as but not limited to BPMN (OMG 2011), IDEF (KBSI 1995), BPMS (Karagiannis et al. 1996), UML (OMG 2009), BPAL (De Nicola et al. 2010), CMMN (OMG 2013) or DMN (OMG 2015b). Each preferable depending on the aspects that have to be described. In CloudSocket the well-known BPMN, CMMN and DMN formats have been selected to describe the domain specific business processes.

The Enterprise modelling framework of Zachman (Zachman 2008) is used as the basic skeleton for business and IT alignment as it identifies not only the different layers from business down to IT but also different aspects from process to human workers such as information and organization.

In the plugIT project (Woitsch et al. 2009) the Business and IT alignment modelling language WIKI (http://plug-it-project.eu/plugITwiki) has been developed based on extended aspects of the Zachman framework in the form of data, knowledge, processes, people, organisation, application, products and motivation. Strategy, business, system and technology are the perspectives that span a matrix of modelling languages, each with a list of different modelling languages (see Figure 15).

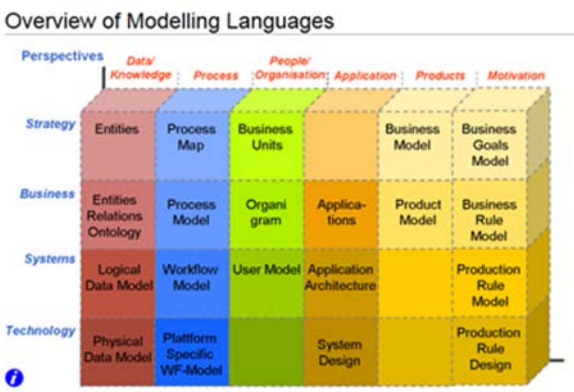

Figure 15 Modelling Languages for Business and IT

Other enterprise models like TOGAF (The Open Group 2011) and Computer Integrated Manufacturing Open System Architecture CIMOSA (Kosanke 1995) are mentioned for completeness reasons, although CloudSocket builds on Zachman as a classification framework.

Modelling for Cloud-based applications is a rather new research topic. Preliminary ideas are proposed in (Mohagheghi et al. 2010), as well as by the projects MODAClouds (MODAClouds 2015) and PaaSage (PaaSage 2015). As a result the current CloudML (Ferry et al. 2014), (Brandtzæg et al. 2012) can be considered which has been further developed into the CAMEL multi-domain language in the context of the PaaSage project covering many more aspects apart from deployment, such as security, monitoring, providers and requirements.

The orchestration or choreography part of BPMN is used to define the executable workflows; hence the same notation is used to define different aspects of a business process. ADOxx is used to develop the BPaaS Design Environment in order to (a) realize a hybrid modelling tool that can model both aspects – the domain specific business process as well as the cloud specific technical workflow, (b) enable semantic lifting of business processes and workflows to enable smart business and IT alignment and finally (c) introduce an alternative way of modelling by introducing a text-based modelling tool like XText (Eclipse 2015) but within a powerful meta model environment.

3.2 Semantic Enrichment through Semantic Lifting

3.2.1 Introduction into Semantic Lifting of Meta Models

"Lifting" is seen as the act or process of rising or raising to a higher position (The Free Dictionary n.d.). Here the higher position is seen as the enriched semantic information that provides an additional meaning to the model. "Semantic Lifting" refers to the process of associating a model content items with suitable semantic objects as metadata to turn "unstructured" – or in our situation "semi-formalized" – content items into semantic knowledge resources (IKS 2012).

"Semantic Lifting of Business Processes" aims at representing a business process in an ontology modelling framework that is not specific to a particular business process language or ontology language (De Nicola et al. 2008). Based on the above definition, our understanding of "semantic lifting" of business process models to support business and IT-Cloud alignment is the process of raising business process and other related models with suitable semantic objects to a higher formal representation (Woitsch et al. 2013).

Lautenbacher et al. (2008) indicate the main benefits for semantically annotating business processes which are the following: (a) semantic search in business process models; (b) enhanced validation of business process models; (c) automatic business process execution; (d) better re-use of process fragments; (e) replacement of process fragments; (f) auto-creation, adaptation and auto-completion of business process models; (g) realisation of collaborative business processes through automatic integration of partner business processes.

Semantic Lifting of meta models is a special way of meta model merging, as a domain specific meta model is merged with a semantic meta model. Interpreting semantic models as well as conceptual meta models as independent meta models, lead to the conclusion that merging patters for meta models can be applied. Kühn (2004) summarizes prominent meta model merging patters:

1. Reference pattern, where two meta models are complementary and should not or cannot be changed. . There exist different ways to enable a reference depending whether the meta model can or cannot be adapted.
    a. In case the meta model is compiled into a software tool, and not source files or appropriate development know how is available, the reference needs to be realized using the exiting meta models and existing functionality.
    b. In case the meta model can be adapted, it is possible to reference with appropriate tool support.

    Although the latter solution is more user friendly, it requires an implementation that considers such a reference from one meta model (e.g. BPMN) to another meta model (e.g. RDF).
2. Transformation pattern, where two meta models are in principle complementary but part of one meta model correspond or can be created out of parts of the other meta model.
    a. In case of a complete transformation, there are transformation rules that transforms models from one meta model (e.g. business process model using BPMS) into models that conform to the other meta model (e.g. workflow model using BPMN). The underlying algorithm is called "graph-rewriting" able to parse a model, transforms and re-write it into a translated context.
    b. In case of a proxy transformation, the models from one meta model are integrated into another model of the second meta model, but act as proxy models. Although this means that the proxy models are redundant with the original modelling tool, the user interface as well as the user support may be better, when the models are kept within one application.
3. Use or aggregation pattern where parts of one meta model are used within another meta model or are aggregated along with the other meta-model into a new meta model, respectively.

a. In case of use, a particular aspect of one meta model is required by another meta model; hence such aspect is re-used (e.g. KPI or DMN in BPaaS). This is similar to the way software libraries are used, when one library can be included as part of another one when a specific functionality from the former is required for fulfilling the functionality of the latter.

b. In case of aggregation, the new meta model derived comprises the individual meta-models (or their parts mapping to the required aspects),

4. Merge and extension patterns of meta models constitute a further evolution of aforementioned use and aggregation pattern, by integrating closely related meta models. The challenge is to develop a new or extended meta model that still keeps the same concepts of the original meta model, so the original behaviour and coverage is not disturbed. Reflecting the complexity of each meta model the resulting meta model may be inappropriate to be handled by a user, as being more complex than the original ones, but provides a holistic overview.

The BPaaS meta model applies several of aforementioned merging patterns to merge and extend BPMN, DMN, KPIs and additional meta models for organizational descriptions. Transformation patterns are applied to define graph re-writing from business processes to workflows. Reference patterns are applied for the semantic enrichment of the models.

3.2.2 Different Techniques for Semantic Lifting of Meta Models

The enrichment of meta models with semantic maps to two main aspects:

- Semantic nature of the meta model mapping to the structure and the available concepts that are defined in the meta model. Some meta models already include semantics semantic while for others their content has to be semantically-lifted or enriched.
- Annotation responsibility in the process of semantically lifting where intelligent decisions are required while annotating, which can be performed (mapping to the respective responsibility) by humans via intellectual and manual annotations, or by machines that follow pre-defined mapping rules.

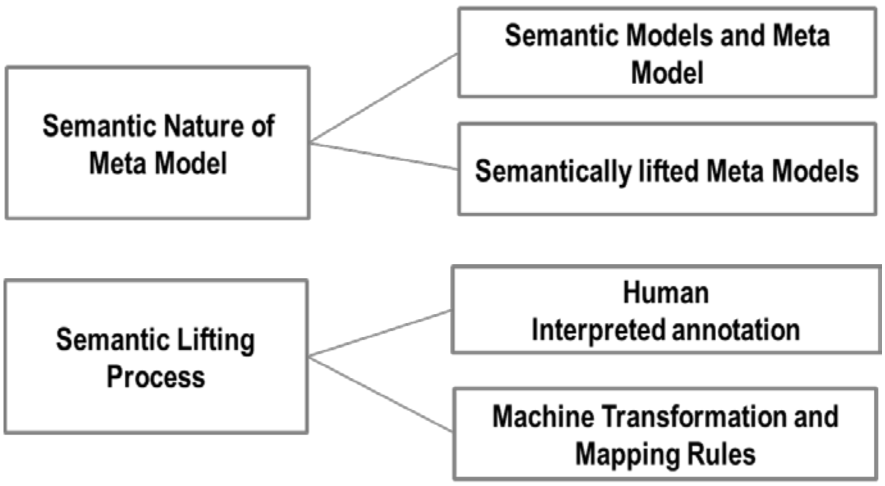

Figure 16 Realization Types of Semantic Lifting

Figure 16 introduces a topology of the different semantic lifting approaches.

In the following the nature of the meta model is elaborated, whereas the meta models can be distinguished with respect to their level of formality. Formal models that include semantic expressions are considered as “semantic models or meta models”, whereas semi-formal models without semantic expressions are considered as “semantically lifted meta models” that require additional external ontologies.

3.2.3 Semantic Nature of Models

This section introduces the insights of Semantic Models, Meta Models and how those models can be enriched using annotations in order to add semantic information.

3.2.3.1 Semantic Models and Meta Models

Sematic Models and Meta Models provide sufficient semantic expressions to semantically lift (annotate) models and the meta model.

a. OWL-WS: A Workflow Ontology for Dynamic Grid Service Composition exploits OWL concepts enabling the description of semantic workflow graphs (Beco et al. 2005). Hence the semantic is already in the meta model by starting from OWL as a meta model.
b. SAWSDL: Semantic annotations for WSDL is semantic annotation of the Web Service Description Language, and hence enriches WSDL with semantic concepts that are available for annotation (LSDIS 2005).
c. WSMF: The Web Service Modelling Framework WSMF (Fensel and Bussler 2002) is a semantically based modelling framework to describe Web-Services. Annotation is therefore possible with available semantic concepts.
d. A Graphical Language (Jinho, Sujeong and Mooknun 2012) to Integrate Process Algebra and State Machine Views for Specification and Verification of Distributed Real-Time Systems. Formal models such as process algebra can be mapped to semantical concepts benefiting from the operational semantics that is provided by a directed graph.
e. In (Thomas & Fellmann 2007), a proposal for the semantic extension of Event-Driven Process Chains (EPC)s has been suggested where the semantics of individual model elements are identified through the concepts of an ontology by following a multi-level approach comprising three main levels: (a) ontology; (b) meta-data; (c) model. Through this approach, the meta-model of the EPC-based BP is enhanced in order to allow for the inclusion of semantic annotations such that the respective existing tools and frameworks can still work with the basic EPC description but also the corresponding meta-data are captured which can then be exploited in order to perform various reasoning tasks.
f. A context-based process semantic annotation model (CPSAM) is proposed in (Mturi 2015) which captures five main perspectives: (a) functional, (b) behavioural, (c) organisational, (d) informational and (e) context (mapping to the modelling of process relationships, the process environment and the process goals). This annotation model can then be exploited to provide annotations for existing business processes in well-known formats.
g. In (Belhajjame & Brambilla 2009) a semantic BP annotation approach is being proposed which allows annotating BPs with so-called *abstract business processes* or tasks mapping to concepts of a specific ontology which has been constructed via a collection of concrete BPs.
h. In the context of the SUPER project (SUPER 2009), an ontology stack for BP modelling has been developed (Dimitrov et al. 2007) comprising the following sets of ontologies:
   - an upper process ontology (UPO) modelling generic concepts, such as tasks, goals and conditions
   - a business process modelling ontology (BPMO) providing abstractions over existing BP modelling languages
   - ontologised versions for parts of existing BP modelling languages (BPMN, EPC & BPEL). In case of the semantic extension to BPEL, this is enhanced through the exploitation of the WSMO ontology through references to WSMO goals in order to enable goal-oriented discovery, mediation and execution of services.

   Through this ontology stack, models defined via existing BP modelling languages are semantically lifted through references to domain ontologies, WSMO goals and semantic constraints.

3.2.3.2 Semantically lifted Meta Models

For semantically lifted Meta Models, external semantics are used to enrich the meta model, hence aforementioned reference or use patterns will be applied.

a. Lifting Metamodels to Ontologies: A Step to the Semantic Integration of Modelling Languages (Kappel et al. 2006). Meta models are enriched with semantic annotations to ontology concepts to provide smart semantic-based algorithms.
b. Meta Models in Action is a collection of every available usage of meta models and introduce the possibility to enrich meta model and models with external ontology concepts (Karagiannis and Höfferer 2006).
c. An semantic annotation framework for business processes (BPs) has been proposed in (Lin 2008). This framework enables four types of semantic annotations: (a) profile annotations on basic descriptions of process models; (b) meta-model annotation on BP modelling languages; (c) model annotations on the contents of BP models; (d) goal annotations on the intentions of BP model owners.
d. Salah et al. (2014) attempt to address the workflow interoperability through proposing two different approaches depending on the type of targeted environment. For homogeneous environments, the authors propose the decoration of workflow models with lexical and structural annotations. For heterogeneous environments, they propose a common semantic annotation structure to annotate process models at the following levels: (a) meta-models, (b) content of models, (c) model profiles and goals for semantic discovery, (d) basic aspects of models, such as the informational one. In this respect, different types of ontologies are employed to perform the annotation, including workflow, domain-specific and profile and goal ontologies. The proposed approaches are agnostic to the respective ontology representation language to be exploited.

3.2.4 Semantic Lifting

Semantic lifting process is the second aspect beside the semantic nature of the meta model. The process, hence the sequence of steps that are executed to enrich a semi-formal meta model with formal semantic can either be performed by a domain expert or by smart technology.

3.2.4.1 Human Interpreted Annotation

Human Interpreted Annotation is the intellectual task of a domain expert, who manually annotates the model with semantic concepts.

a. Semantic Lifting of Business Process Models (De Nicola et al. 2008) is the use of ontological concepts to enrich business process models. The core of this approach lies on an enriched OWL language called OPAL allowing the specification of semantic business process objects.
b. Business and IT Alignment: The IT-Socket (Woitsch et al. 2009) uses reference patterns to introduce external ontology concepts into business process models to support semantic supported alignment. The core of this work maps to a meta model approach and a manual lifting with external ontology concepts.

3.2.4.2 Machine Transformation and Mapping Rules

Machine based transformation and mapping rules enable the automatic semantic annotation according to pre-defined rules. We provide particular examples for this in the following which map to particular annotation/transformation approaches

a. Mappings, Maps and Tables: Towards Formal Semantics for Associations in UML2 (Diskin and Dingel 2006). Software description with UML are often a candidate to apply automatic mapping as the envisioned intension of software description with UML is to be so formal that a Model Driven approach can be applied.
b. Merging Models with the Epsilon Merging Language (EML) (Kolovos et al. 2006). This sample represents the family of formal algorithm languages.
c. Rule-based approaches like: A Rule-Based Approach to the Semantic Lifting of Model Differences in the Context of Model Versioning (Kehrer et al. 2011); Model-driven Rule-based Mediation in XML Data Exchange (Liao et al. 2010) apply rules for automatic annotation.
d. Semantically Lifting with a proprietary solution SiLift (Kehrer, et al. 2012) is applied to identify the evolution of models.
e. A Model Transformation Approach Based on Homomorphic Mappings between UML Activity Diagrams and BPEL4WS Specifications of Grid Service Workflows (Hlaoui et al. 2011).
f. Semantic Lifting of Unstructured Data Based on NLP Inference of Annotations (Marinchev 2012).
g. Leopold et al. (2015) propose an automatic technique for semantic business annotation via concepts from the PCF taxonomy. The novelty of the technique proposed is that it combines semantic similarity measurement with probabilistic optimisation. In particular, different similarity types between BP model elements and the taxonomy concepts as well as the distance between the taxonomy concepts are used to assist in the matching performed according to Markov logic formalisation. In addition, similarity is measured via a corpus-based approach of second-order similarity.
h. Tonella & Di Francescomarino (2010) propose a semantic annotation approach which can be used to construct a domain ontology skeleton or to extend an existing ontology as well as propose annotation suggestions to the business expert for BPMN elements. This approach relies on the use of Natural Language Processing (NLP) techniques (linguistic analysis) operating over the labels of BPMN elements as well as on a similarity calculation algorithm (based on an information content similarity measure) which operates over the ontology constructed and the elements of a BPMN model.

The aforementioned list of different lifting processes indicates the different approaches and possibilities to semantically enrich meta models.

The benefits of human-interpreted annotation

- are of higher quality when performed from the domain expert,
- are more flexible as the domain expert can abstract and adapt to the situation,
- do not require complex modelling and no learning curve of the software,
- easy to start with.

On the other hand, the benefits of an automated annotation approach are:

- once a transformation has been learned, masses of data can be annotated in a minimum of time,
- the annotations are predictable and traceable, hence ensure stable outcomes,
- no domain expert is required, once the system is installed,
- efficient when it is running.

Aforementioned discussion already proposes the methodology by starting with manual annotation to quickly gain confidence with business and IT-Cloud alignment before extracting the semantic annotations into rules that can be executed automatically.

The goal is not to exchange the human expert with smart rules, but to support a human with smart assistance which is able to propose annotations, check consistencies or provide warnings in case of unstable annotations.

3.3 Service Processing: Discovery and Composition

Alignment of Business and IT in the Cloud can be regarded as a special case for service discovery and composition. In the BPaaS Design Environment the business process is mapped to a workflow. A workflow model is composed of tasks with Cloud Services assigned to them. Thus, the identification of workflows, which realize a business process can be regarded as a special case for service discovery. While a workflow consists of several tasks, several services have to be composed in order to realize a workflow.

3.3.1 Service Discovery

In the context of discovering and re-using services for producing various design products, like composite services, applications or business processes, the respective work can be classified based on the type of service to be discovered. It can also be further classified based on the description aspect of focus, i.e., functional and non-functional. Moreover, further classification can be performed based on the type of technology and respective formalism employed into ontology-based and non-semantic approaches (e.g., Information Retrieval (IR), constraint solving). In the following, we first review the related work and then we present a respective table of summary which provides an overview of an approach according to these three classification aspects.

3.3.1.1 Software Service Discovery

Functional software service matchmaking mainly operates over the service input and output (I/O). The respective approaches proposed exploit either Information Retrieval (Dong et al. 2004) or ontology-based techniques (Paolucci et al. 2002) or a combination of both (Klusch et al. 2009; Plebani & Pernici 2009) to perform the service I/O matching. The latter two approach types cater for better accuracy but do not take into account service behaviour. In this way, they will never exhibit a perfect accuracy level. To overcome this latter issue, few approaches (Sycara et al. 2002; Keller et al. 2004) have employed behaviour-based service matching by relying on full input-output-precondition-effects (IOPE) service profiles. While the accuracy levels exhibited are indeed higher, the main drawback of these approaches is that they rely on full IOPE service profiles which do not exist in reality and require additional modelling effort by the service provider.

Non-functional software service matching approaches can be categorized as constraint-based, ontology-based or mixed. Constraint-based approaches (Cortés et al. 2005) express the quality-based service description as a constraint model and then employ certain constraint solving techniques and matching metrics (e.g., subsumption (Cortés et al. 2005)) to perform the respective service matchmaking. Such approaches assume that the quality-based service description re-uses terms that have been defined in a common quality term repository. Ontology-based approaches (Zhou et al. 2004) exploit ontologies to semantically describe service quality capabilities and requirements and utilize subsumption reasoning to infer whether the quality capabilities match the respective requirements posed. Such approaches have the drawback that can only process unary quality-based service descriptions comprising just one quality term per constraint. Finally, mixed approaches (Kritikos & Plexousakis 2014) combine the best techniques from the aforementioned approach categories. They rely on ontologies not only to semantically describe the service quality description but also to semantically validate it. Then they align these descriptions based on their quality terms (e.g., quality metrics) and finally transform the aligned descriptions into constraint problems so as to use the service matchmaking techniques of the first approach category. The latter category of approaches exhibits better accuracy than the others and is also able to cope with n-ary quality service specifications (i.e., multiple terms per constraint can be involved).

3.3.1.2 Cloud Service Discovery

In (Ruiz-Alvarez & Humphrey 2011) an approach focusing on cloud storage service discovery has been proposed where cloud storage requirements and capabilities are expressed via a semi-formal language. The framework in (Garg et al. 2011) relies on the Analytical Hierarchy Process (AHP) (Saaty 1980) to produce the weights to each quality term and then ranks cloud services as a weighted average on their performance with respect to these quality terms. In (Zeng et al. 2009) Wordnet is exploited to assess the similarity of concepts in the I/O of cloud service specifications and thus produce an overall ranking for each service advertisement. Buyya et al. (2010) propose a federated cloud environment able to match user quality requirements to cloud services. D' Andria et al., (2012) propose a PaaS matchmaking and selection framework where PaaS are filtered according to user requirements and then ranked based on the number of user preferences satisfied. Cloud blueprint matchmaking is addressed in (García-Gómez et al. 2012) where based on an initial composite blueprint document crossing different cloud levels, user-requirements are matched against cloud service offerings and the matchings are used to enhance the cloud blueprint document to make it suitable for cloud service selection and subsequent application deployment and execution. In (Zhang et al. 2012) an ontology-based recommendation system is proposed focusing on infrastructural services but covering both functional and non-functional aspects. They also propose a particular ontology able to cover the modelling of infrastructural services and their relations. An ontology-based cloud service discovery system also focusing on infrastructural services has been proposed in (Kang and Sim 2010). This system calculates three different types of reasoning to perform the cloud service matching: concept similarity and object and data property similarity. An ontology-based QoS-aware cloud service discovery approach for cloud deployment of appliances is proposed in (Dastjerdi et al. 2010) which relies on WSMO for the description of the IaaS offerings. A novel SaaS selection algorithm is proposed in (Saouli et al. 2014) which takes into account functional and QoS aspects and relies on the new concept of existence degree.

3.3.1.3 Business Service Discovery

A business service is considered as a business activity, part of an organisation's business model which maps to particular intangible outcomes (Baida et al. 2004). It is executed on behalf of another organisation, it involves the transfer of value and it is linked to the satisfaction of certain business goals. A business service is actually realised by a software service, thus linking the business goals to the respective IT realisation able to satisfy them. In this context, few approaches have focused on the discovery of business services. The approach in (Kritikos et al. 2013) is able to semantically match business services according to their I/O and quality requirements by employing existing functional and non-functional (software) service matchmaking techniques and enabling them to operate on a higher-level. The method proposed by Akkermans et al. (2004) focuses on semantic business service composition into so-called bundles which relies on the user's requirements/goals and satisfies all types of constraints posed (inherent and function-based). Obviously, as service discovery is part of service composition, a first step in the composition process is to discover services which can potentially be used in combination to satisfy user requirements. To this end, the service discovery algorithm employed in this approach seems to rely on semantic I/O matching. Finally, the semantic approach in (da Silva Santos et al. 2008) relies on goal-oriented service discovery comprising three main steps: (a) goal matching between user and domain goals; (b) mapping goals to a set of tasks; (c) matching services to required tasks. Steps (b) and (c) seem, though, to require that the respective mapping/matching knowledge is already in place. This means that a lot of manual effort is required in order to guarantee an accurate service discovery and subsequent composition result. It should be highlighted that the approaches involved in business service discovery are mainly semantic. This totally satisfies the use of ontologies in order to bridge the business to IT gap as this is the approach followed also in CloudSocket.

Approach	Service Type	Description Aspect	Semantic
Dong et al. 2004	Software service	functional (I/O)	no
Paolucci et al. 2002	Software service	functional (I/O)	yes

Approach	Service Type	Description Aspect	Semantic
Klusch et al. 2009	Software service	functional (I/O)	yes
Plebali & Pernici 2009	Software service	functional (I/O)	yes
Sycara et al. 2002	Software service	functional (behaviour)	yes
Keller et al. 2004	Software service	functional (behaviour)	yes
Cortés et al. 2005	Software service	non-functional	no
Zhou et al. 2004	Software service	non-functional	yes
Kritikos & Plexousakis 2014	Software service	non-functional	yes
Ruiz-Alvarez & Humphrey 2011	Cloud storage service	functional & non-functional	no
Garg et al. 2011	Cloud service	non-functional	no
Zeng et al. 2009	SaaS	functional (I/O)	no
Buyya et al. 2010	Cloud service	non-functional	no
D'Andria et al. 2012	PaaS	functional & non-functional	yes
García-Gómez et al. 2012	Cloud service	functional & non-functional	no
Zhang et al. 2012	IaaS	functional & non-functional	yes
Kang and Sim 2010	IaaS	functional	yes
Dastjerdi et al. 2010	IaaS	non-functional	yes
Saouli et al. 2014	SaaS	functional & non-functional	no
Kritikos et al. 2013	Business service	functional & non-functional	yes
Akkermans et al. 2004	Business service	functional and non-functional (only for composition)	yes
da Santos Silva et al. 2008	Business service	functional	yes

Figure 17: Service Discovery Approaches

3.3.2 Service Composition

As far as service composition is concerned, the same categorisation criteria as in service discovery also apply, i.e., service type, aspect and semantics support. Similarly, we first perform the analysis based on the service type and then we provide a specific comparison table for providing an overview of the analysis results with respect to the three categorisation criteria.

3.3.2.1 Software Service Composition

Service composition approaches are either automatic or semi-automatic. Automatic approaches usually focus on the functional aspect and employ Artificial Intelligence (AI) (Bertoli et al. 2010) or model-driven (Brogi et al. 2008) techniques in order to automatically produce a specific composition plan. On the other hand, semi-automatic approaches are non-functional. They rely on a particular abstract composite service plan which they attempt to concretise based on the alternative set of services mapped to each task in this plan. An exception to the aforementioned pattern is the approach in Sohrabi & McIlraith (2010) which can produce a concrete service composition plan by also considering the user's qualitative preferences but is not capable of handling all possible quality parameter types.

Most of the non-functional service composition approaches follow either a statistical (Canfora et al. 2005) or path-based approach (Ardagna & Pernici 2007) leading to an over-simplification or a pessimistic view of the problem where all possible execution paths should satisfy the user requirements. Some approaches employ a heuristic (Yu et al. 2007) or a QoS decomposition (Alrifai et al. 2009) approach to achieve better performance levels but sacrificing optimality. Moreover, almost all approaches regard quality-based service offerings as simple QoS parameter values. This is quite unrealistic, especially for services operating in quite dynamic environments. Moreover, the approaches fail to produce any result for over-constrained end-user requirements. One approach which resolves most of the above issues was proposed in (Ferreira et al. 2009).

3.3.2.2 Cloud Service Composition

Complete cloud service composition is harder than software service selection as it involves composing different types of services with different characteristics at different and inter-dependent levels. However, most cloud service composition approaches focus on just one cloud service type. Those which do consider additional types, either solve a limited case of the complete problem or a slightly different problem by also neglecting all possible user requirement types, such as security, quality & location requirements.

The approaches in SaaS composition can be separated into: (a) semantic (Zeng et al. 2009), (b) heuristic-based (Kofler et al. 2010), (c) multi-tenant SaaS-based (He et al. 2012), (d) variability and multi-criteria decision making-based (Wittern et al. 2012) and (e) aspect-based focusing, e.g., on network latency and the availability of multiple SaaS service instances (Klein et al. 2012).

Gutierrez-Garcia et al. (2013) propose a self-organizing agent-based cloud service composition method which exploits distributed problem solving techniques by also relying on the contract-net protocol. This method can produce vertical, horizontal, one-time and persistent service compositions. It is able to handle both SaaS and IaaS services. However, this method seems to cover only functional and cost requirements.

The hierarchical quality model in Karim et al. (2013) goes from user requirements down to the capabilities of IaaS services. It is then used to rank service candidates across the different cloud levels. However, the ranking algorithm accompanying this model works on a different problem type where first the end-user requires SaaS services and then the providers of these services have to find suitable IaaS offerings to host these services. This algorithm does not also take into account location and high-level security requirements. Moreover, it seems to work only for sequential abstract service plans.

3.3.2.3 Business Service Composition

Akkermans et al. (2004) propose a semantic business service composition approach which comprises three main steps: (a) construction of service bundles based on the user requirements posed; (b) application of functions to extend or restrict the bundles (e.g., the use of one service requires using another or excluding another service); (c) applying constraints over service properties to filter the bundles that have been created.

In da Santos Silva et al. (2008) the semantic business service composition approach comprises the following steps: (a) matching a user to domain goal; (b) identifying the set of tasks which realise the domain goal; (c) obtaining services which realise the task functionality. In case that (b) or (c) fails, this means that either task or service composition must be performed. In case of (b), the domain goal is decomposed until the level where tasks exist for the respective sub-goals such that the combination of sub-goals mapping to tasks realising them can lead to the satisfaction of this goal. In case of (c), a service composition based on I/O-based planning is employed in order to fulfil the respective task's functionality.

Ramel et al. (2010) propose a goal-oriented business service design framework which starts from the user requirement and goes down until the level of the business service realisation in terms of software services. While this framework considers all service aspects, it lacks the appropriate automation level as most of the tasks seem to be manually performed with some assistance from the framework.

Rolland and Kaabi (2010) propose a business service composition framework and a set of guidelines for eliciting services and their composition. While this approach seems to be quite sophisticated, it exhibits particular drawbacks which include the non-appropriate handling of non-functional requirements and the lack of automation support especially in term of the discovery of software and business services.

Lo & Yu (2007) propose a SOA-based design methodology which starts from a business model and goes down to identifying potential IT services. This methodology relies on the framework as a basis for modelling and analysis as well as on a reference business model and service repository. The methodology seems to stay at a high-level and does not provide appropriate automation support in the design of SOA based systems. In addition, it seems to mainly focus on generic business services which are required in common business scenarios.

Kritikos et al. (2013) propose a semantic business service design methodology which covers all possible design scenarios and is able to both compose business and software services. In this way, it can go from the original user goals to the respective software services realising the business service fulfilling these goals. Functional business and software service composition relies on I/O-based service planning while non-functional business/software service concretisation relies on extending the service concretisation approach in (Ferreira et al. 2009) by considering both all possible alternative service plans as well as their potential realisations and attempts to best satisfy all user requirements posed.

As a business services can be required as a part (activity) or a whole business process, the work in Lapuchnian et al. (2007) can be regarded also relevant. This work exploits goals models to address the variability in the business domain. Through automated goal analysis, user qualitative preferences are used for the selection of the best business process composition candidate among all possible ones. This work does not address the realisation of a business process. In addition, the consideration of just qualitative preferences does not allow for a complete assessment over the performance of BP composition alternatives and does not also enable the propagation of the performance levels from the business process component to the whole business process composition level.

Approach	Service Type	Service Aspect	Ontology-based
Bertoli et al. 2010	Software service	Functional	no
Brogi et al. 2008	Software service	Functional	no
Sohrabi and McIlraith 2010	Software service	Functional & non-functional	no
Canfora et al. 2005	Software service	Non-functional	no
Ardagna & Pernici 2007	Software service	Non-functional	no
Yu et al. 2007	Software service	Non-functional	no
Alrifai et al. 2009	Software service	Non-functional	no
Ferreira et al. 2009	Software service	Non-functional	no
Zeng et al. 2009	Cloud service	Non-functional	yes
Kofler et al. 2010	Cloud service	Non-functional	no
He et al. 2012	SaaS	Non-functional	no
Wittern et al. 2012	Cloud service	Functional & non-functional	no
Klein et al. 2012	SaaS	Non	no
Gutierrez-Garcia et al. 2013	SaaS, IaaS	Functional (& cost)	no
Karim et al. 2013	SaaS, IaaS	Non-functional	no
Akkermans et al. 2004	business	Functional & non-functional	yes
da Santos Silva 2008	business	Functional	yes
Ramel et al. 2010	business, software	Functional & non-functional	no (ext. -> yes)
Rolland and Kaabi 2010	business, software	Functional & non-functional	yes
Lo & Yu 2007	business, software	Functional	no
Lapuchnian et al. 2007	business/BP	Functional & non-functional	no
Kritikos et al. 2013	business, software	Functional & non-functional	yes

Figure 18: Service Composition Approaches

4 BPAAS MODELLING METHOD DEVELOPMENT

This chapter explains the BPaaS modelling environment, which realizes the human-interpretable graphical and textual models. It is the result of the first three phases of the OMiLAB LifeCycle (create, design, formalize - see chapter 2). This chapter describes how the concepts for business and IT alignment can be realized on the platform ADOxx (2015).

Conceptual modelling is a knowledge representation with the aim to observe relevant parts of the real world. Such conceptual models gained commodity, as their simplified view enables to focus on the relevant aspects and thanks to the abstraction enables IT-based support like visualization, queries, simulation and transformation.

The technical framework to implement a modelling tool that supports the business and IT alignment is described by introducing:

(a) so-called vertical alignment that links domain specific business processes, decision models and Key Performance Indicator descriptions with corresponding IT-relevant workflows, deployment rules and Service Level Objectives;
(b) so-called horizontal alignment that compares either domain specific business processes, business rules or Key Performance Indicators as well as IT-related workflows, deployment rules or Service Level Objectives with each other;
(c) a new concept to support business and IT alignment, the “task requirement description” that specifies in more detail the IT-relevant aspect of a task within a domain-specific business process as well as

The technical framework consists of the:

(i) development methodology, which is the adaptive development of meta models (section 4.1);

(ii) underlying implementation paradigm, which is the meta modelling approach (section 4.2);

(iii) development environment, which is the ADOxx meta modelling platform (section 4.2.1).

4.1 The BPaaS Modelling Method

The term “model” has an extremely ambiguous nature and hence is interpreted with the meaning discussed in the feasibility study of the Open Models Laboratory (Karagiannis, Grossmann & Höfferer 2008), where a model is “a representation of either reality or vision” (Whitten, Bentley & Dittman 2004), that are created “for some certain purpose” (OMG 2003). Conceptual models belong to the family of linguistic models that use an available set of pre-defined descriptions to create a model, and enrich the pure textual models (such as mathematical formula) with diagrammatic notations.

Model can therefore be used in four different ways:

i. they act as a specification of desired target, reduce complexity, allow a structured approach and due to a common understanding support a participative creation. (Whiteman, Huff & Presley 1997)
ii. they target semi-automatic implementation of software like in the context of workflow orchestration or model driven architecture (MDA). Models are therefore seen to provide “execution support” (Kokol 1993).
iii. they are well suited for documentation especially for human interpretation and hence support knowledge management tasks.
iv. they evaluate current status against modeled targeted goals.

Targeting business and IT alignment with conceptual models, means that pre-defined diagrammatic concepts with a specific meaning are used in order to (1) specify, (2) support execution, (3) represent knowledge or (4) evaluate the business and IT-alignment.

The research community Open Models Initiative Laboratory (OMiLAB) proposes a generic modelling method specification framework (Karagiannis & Kühn 2002) that identifies all relevant parts that need to be considered for conceptual modelling. The generic framework introduced in Figure 19 enables the specification of conceptual models.

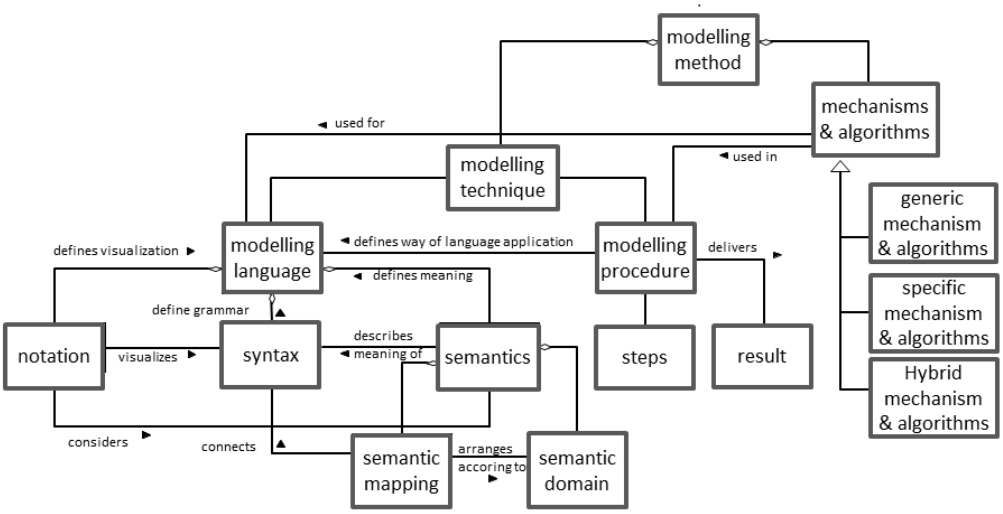

Figure 19: Modelling Method Framework Based on Karagiannis & Kühn (2002)

The framework considers three building blocks: (1) the modelling language that is most prominently associated with conceptual models, as available concepts to be used for such models are pre-defined according their semantic, their syntax and their graphical notation, (2) the modelling procedure defines the stepwise usage of the modelling language and hence is not always available, this means there are modelling languages that have not a pre-defined way of usage but leave the modeller freedom during model construction, (3) mechanisms and algorithms enable the computer-based processing of models and hence provide an IT support for the aforementioned modelling scenarios – specification, execution support, knowledge representation and evaluation.

Those three main building blocks are composed to achieve different levels in form of a modelling technique or modelling method of a concept model approach. Although there is a discussion on the different terms, it helps to classify the different approaches. The traditional Entity Relationship (ER) diagram for example has a modelling language, a modelling procedure and algorithms that enable the transformation from model into a relational database schema. UML in contrast has an expressive modelling language but no modelling procedure explaining the stepwise approach how to create a model. OWL for example defines its concepts in form of a modelling language and provides extensive algorithms for ontology inferences, but does not provide a procedure how to define a model.

All conceptual model approaches, hence also the CloudSocket approach, can be described with aforementioned framework.

4.2 The Meta Model Approach as an Implementation Approach

The meta model approach aims to simplify the development of conceptual modelling solutions, by introducing abstractions, which realize common aspects of conceptual modelling. Hence meta models realize common aspects, which can be used – without re-implementing them, through inheritance.

Based on Strahringer (1996), Karagiannis & Höfferer (2006) and Kühn (2004) Figure 20 introduces the different layers of the meta model approach.

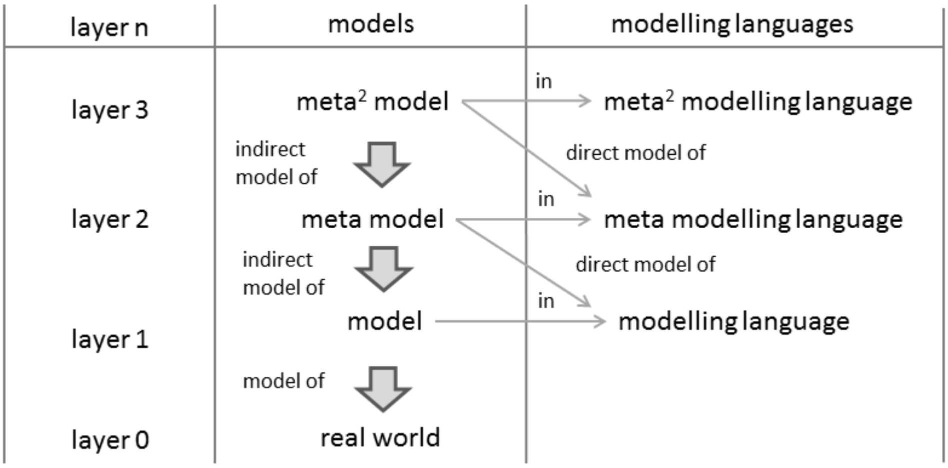

Figure 20: Meta Model Layers

Meta Model approaches have been analysed in (Karagiannis & Höfferer 2006) and can be distinguished in (i) domain, (ii) design – macro and micro level – as well as (iii) integration. Samples for macro-level design are: (a) Business Rules (Herbst 1996), (b) Decision Support, (Aalst, Weske & Grünbauer 2005), (c) Business process and - Workflow Management (Chiu, Li & Karlapalem 1999), (Rolland, Souveyet & Moreno 1995), (d) Information systems: Web-based Systems (Nikolaidou & Anagnostopoulos 2005), Agent-oriented Systems (Wagner 2003), as well as (e) Data Processing (Vassiliadis, Simitsis, Georgantas, Terrovitits & Skiadopoulos 2005). Sample for micro-level design in different domains are: (a) Knowledge representation (Mylopoulos, Borgida, Jarke & Koubarakis 1990), (b) Data Processing mit TIGUKAT (Peters & Ozsu 1993), FORM (Kim & Park 1997) or ULD (Bowers & Delcambre 2006). Meta Models for integration and interoperability are (a) Object Modelling Framework (Hillegersberg & Kumar 1999), (b) Situational Method Engineering (Brinkkemper, Saeki & Harmsen 1999), (c) integrated requirement analysis (Nissen & Jarke 1999) or (d) Data processing schema mapping (Zaniolo & Melkanoff 1982), (Cheung & Hsu 1996).

All aforementioned meta models can be specified with a meta modelling language that is derived from an meta meta model. In the following the most prominent meta meta models based on (Kern, Hummel & Küne 2012) are mentioned: (a) Ecore from the Eclipse platform (Budinsky, Steinberg, Merks, Ellersick & Grose 2004), (b) GOPRR from MetaEdit+ Platform (Kelly 2008) and (c) MS DSL Tools and MS Visio (Cook, Jones, Kent & Wills 2007). Additionally the following meta meta models are introduced: (d) MOF (OMG 2013), which is realised on different UML Profile platforms (e) ADOxx based on the equally named platform ADOxx (ADOxx.org 2013), (f) Obeo Designer on Eclipse (Obeo 2013) and (g) Generic Model Environment GME (GME 2013).

Relevant parts of the real world – in our case business and IT alignment – are seen as layer 0. For this purpose a set of concepts – in form of a modelling language -, a stepwise procedure – in form of modelling procedure – and a set of software functionality – in form of mechanisms and algorithms - are provided to enable the creation of a model on layer 1. The modelling language is understood as the meta model, as it is a model of the concepts available for the model. This meta model is for example defined in a meta model language like ALL (ADOxx

Library Language (ADOxx.org 2013)). The specification of the meta model can again be defined by a model – the so called meta meta model or as a synonym meta2model.

4.2.1 BPaaS Meta Model

The BPaaS meta model is concerned with the (a) domain specific business models and the corresponding (b) the IT-Cloud relevant technical models.

4.2.2 The BPaaS Meta Model: Class Diagram

The BPaaS meta model defines the (a) domain specific business layer and (b) the IT-Cloud relevant technical layer, as well as the semantic interaction between them.

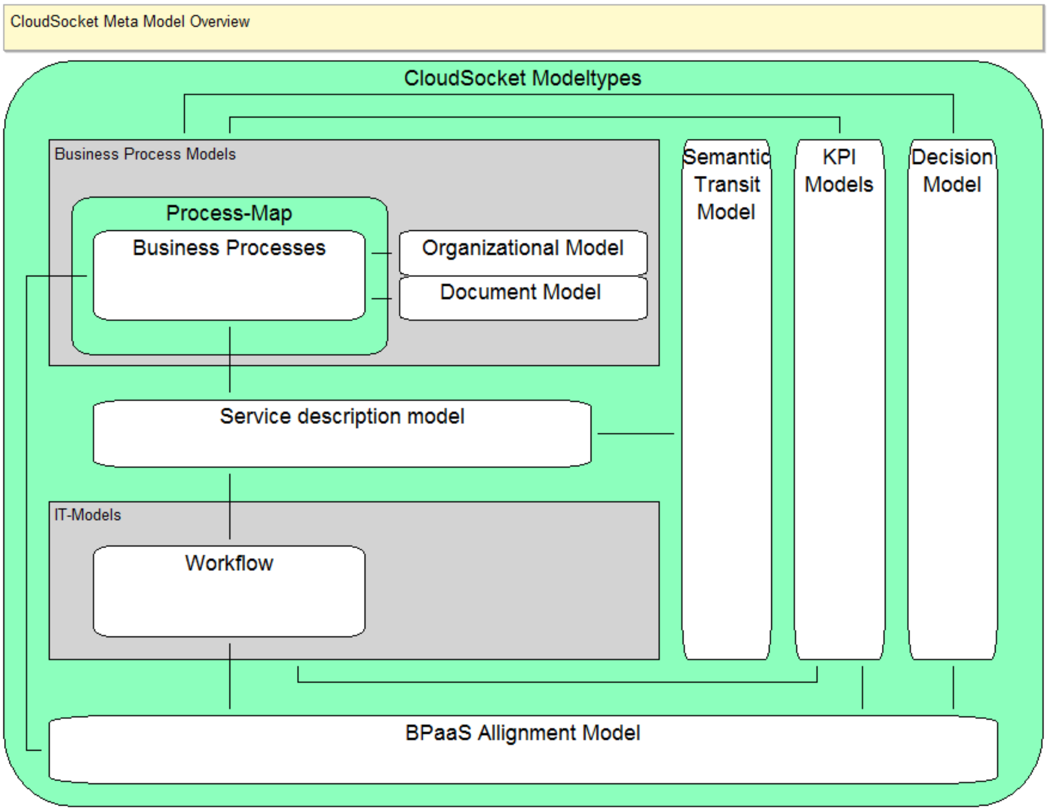

Figure 21: BPaaS Meta Model Stack

The overview of model types of the business layer is represented in the model stack in Figure 21. As shown in Figure xx in the domain specific business layer there are the following model types:

i. Process map model
   Process maps are used to describe an overview of the business process models. The class diagram of this model type is described in Figure 22.

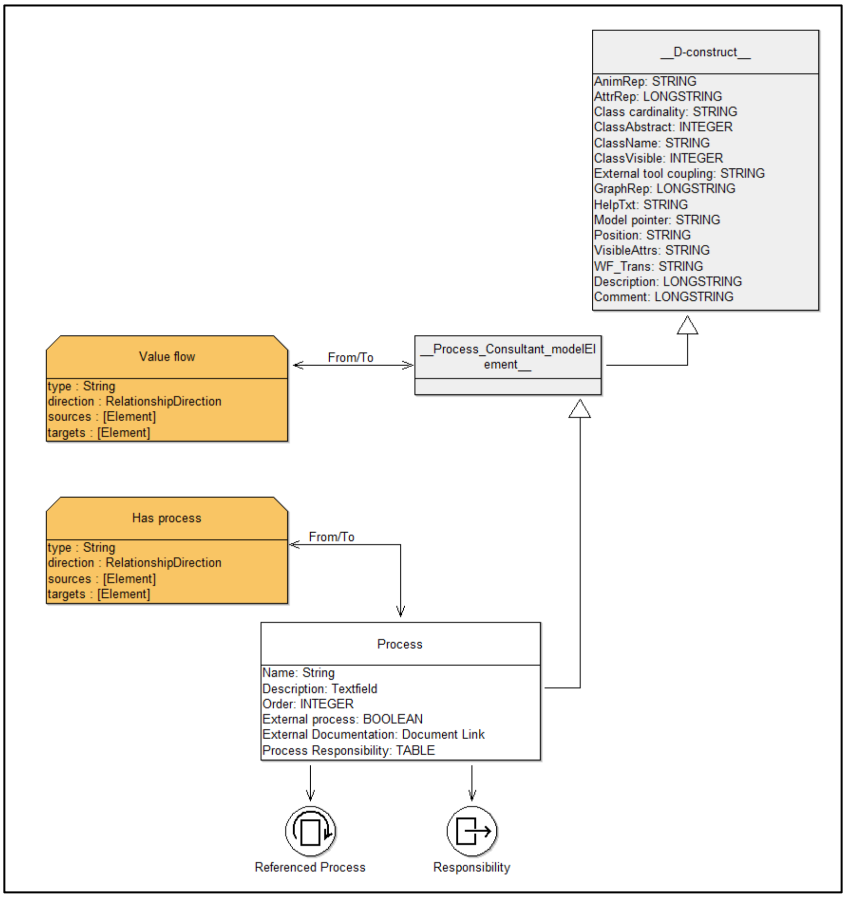

Figure 22: Class Diagram of Business Process Models

ii. Business process model

The Business Process Model and Notation (BPMN 2.0) is a standardization for designing processes in an enterprise. The modelling language of the BPaaS' business processes is a subset of BPMN 2.0 that can be linked to a (a) service description model (b) decision model, and to a (c) key performance indicator models. The subset is selected based on practical experience of the types of concepts used for modelling such procedures and is described in Figure 23.

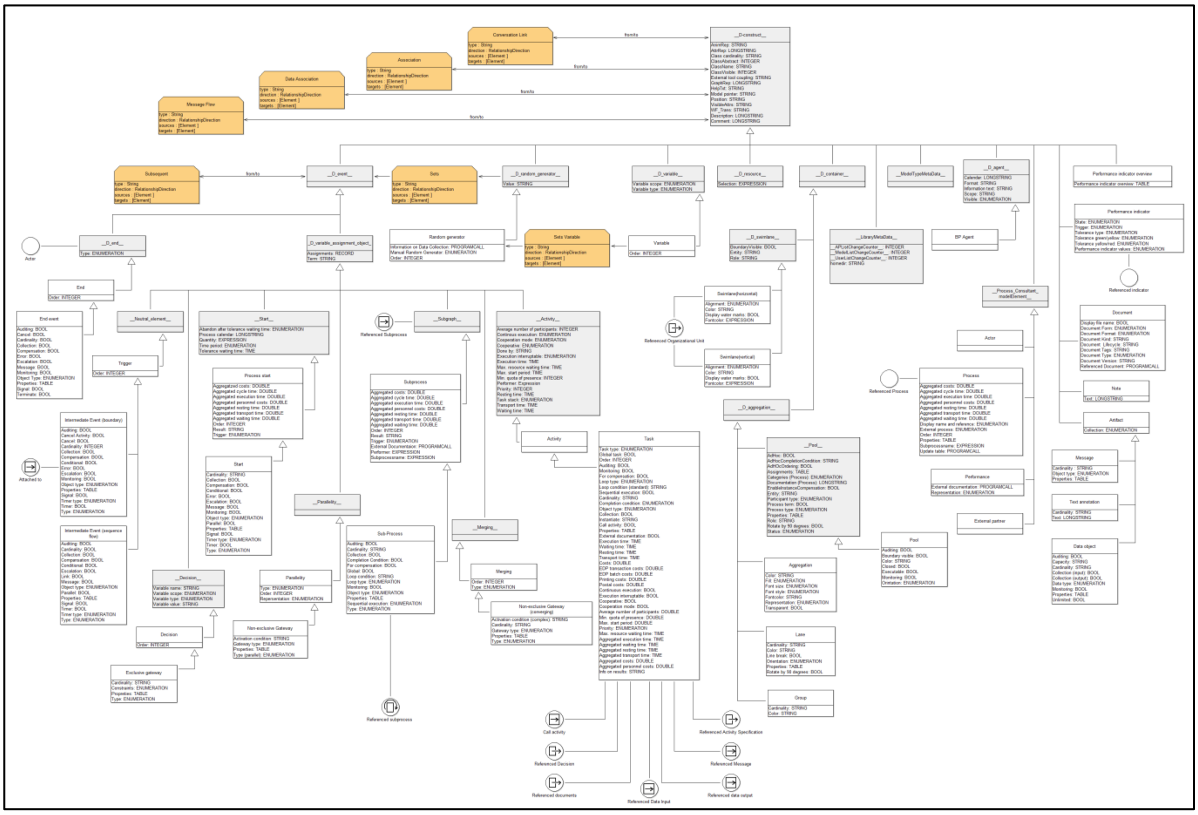

Figure 23: Business Process Modelling Language Class Diagram

iii. Organizational model

Organizational models describe the structure of an organization (organizational chart). The structure of the organizational charts can be built hierarchically using organizational sub models to e.g. illustrate a detailed structure of a working environment. The class diagram of the organizational structure model is described in Figure 24.

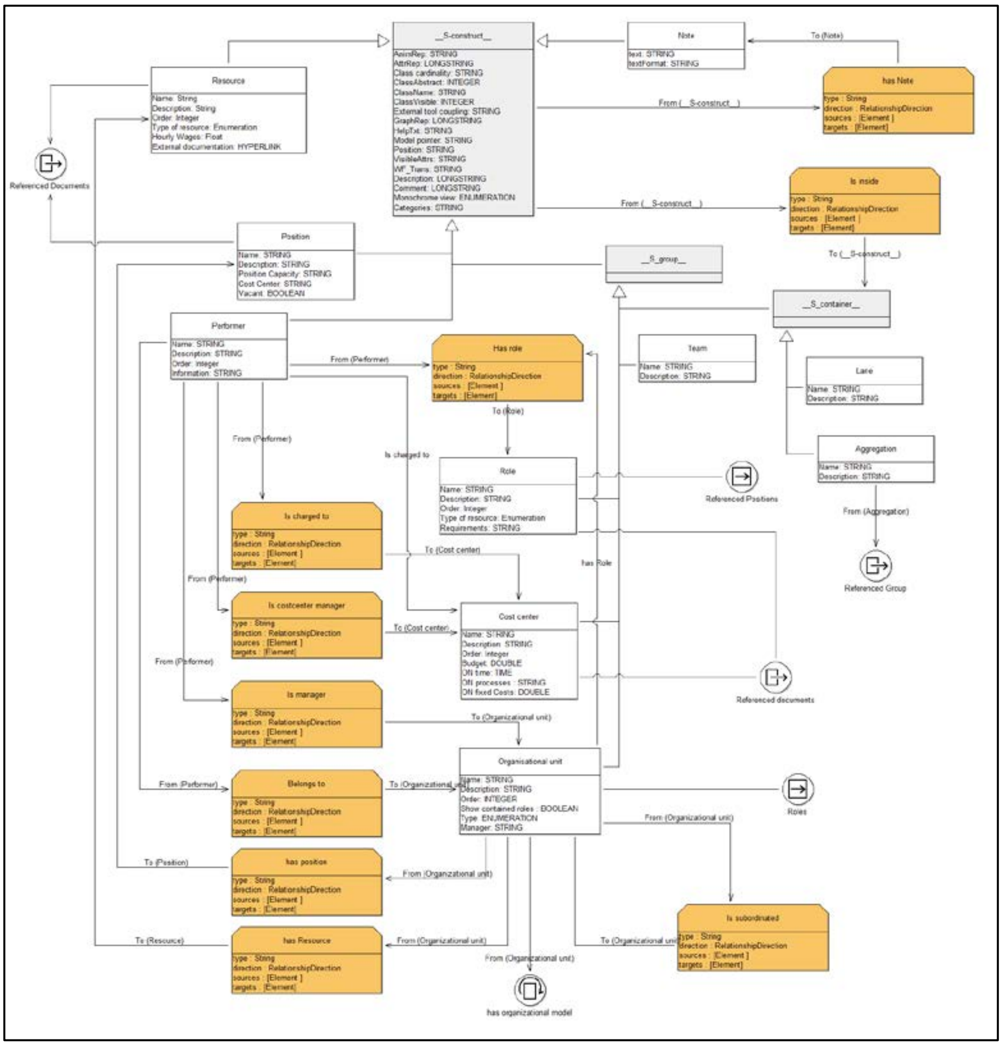

Figure 24: Organizational Structure Model Class Diagram

iv. Document model

Document models contain documents (templates), which are utilized in the processes (input, output to activities etc.). Document models can be built hierarchically using document sub models to e.g. illustrate a detailed structure of documents. The class diagram of the document model is described in Figure 25.

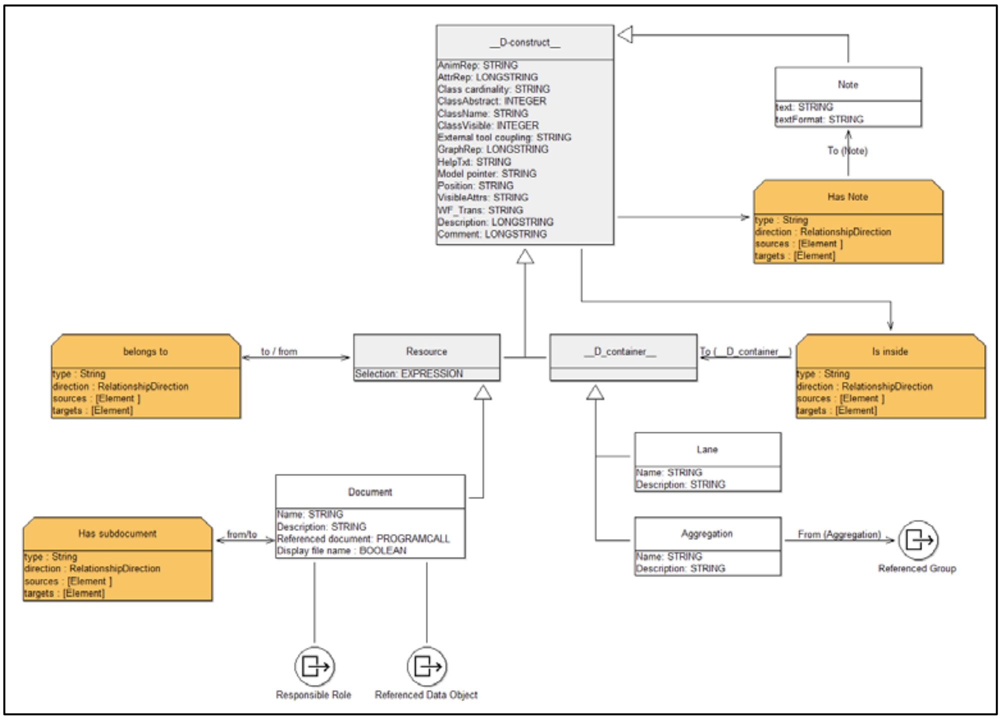

Figure 25: Document Model Class Diagram

In the IT-Cloud relevant technical layer the workflows are described with BPMN 2.0 (see above) language including technical details that are defined within the execution engine of the workflow. The semantically interaction between the domain specific business layer and the IT-Cloud technical layer is done by semantically lifting this two models with the ‘Service Description Model’. Within this model type each task of both processes can be semantically enriched by describing the requirements derived from the business process for Cloud Services considering beside the Service Requirements (a) Functional-, (b) Input- (c) Output- (d) Non-Functional- (e) Business-, and (f) Regulatory description dimensions. The concept ‘Activity description’ is linked with the task elements of the BPMN through the weaving principle. The class diagram of the ‘Service Description Model’ is shown in Figure 26.

The semantically enrichment of each concept can be done in three different ways:

Description in natural text.

(a) Textual annotation to defined ontologies.
(b) Semantic lifting with a set of concepts of the ‘Semantic Transit Model’.
(c) The class diagram of this model type is described in Figure 27

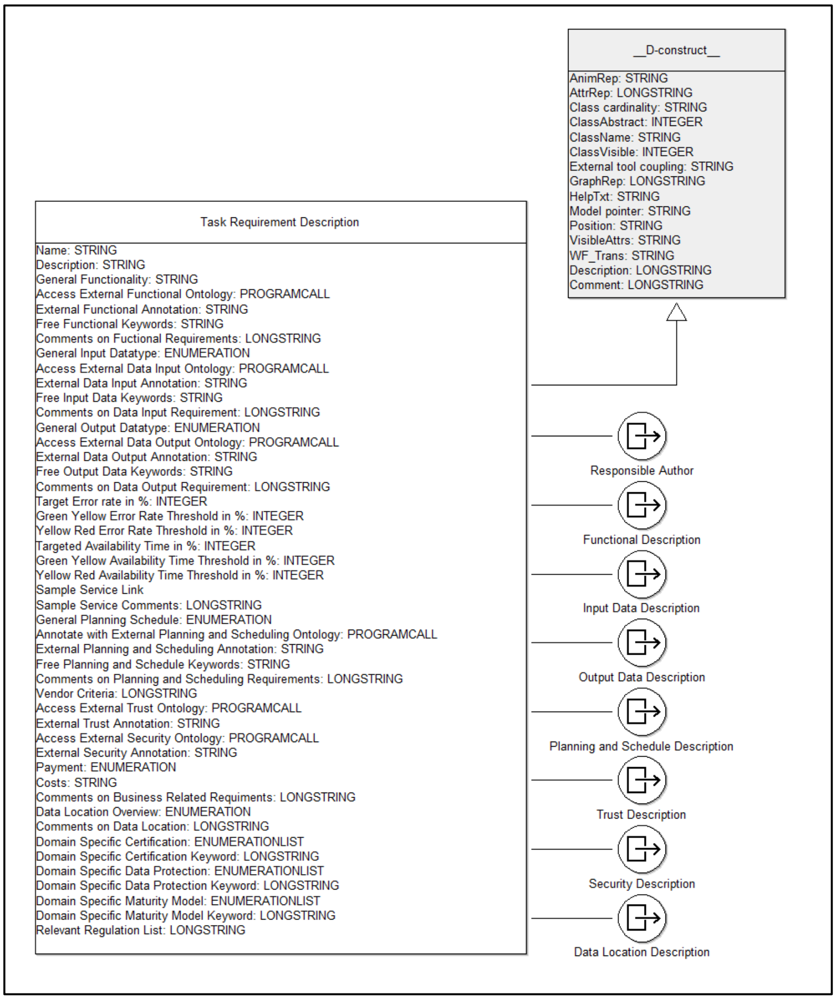

Figure 26: Service Description Model Class Diagram

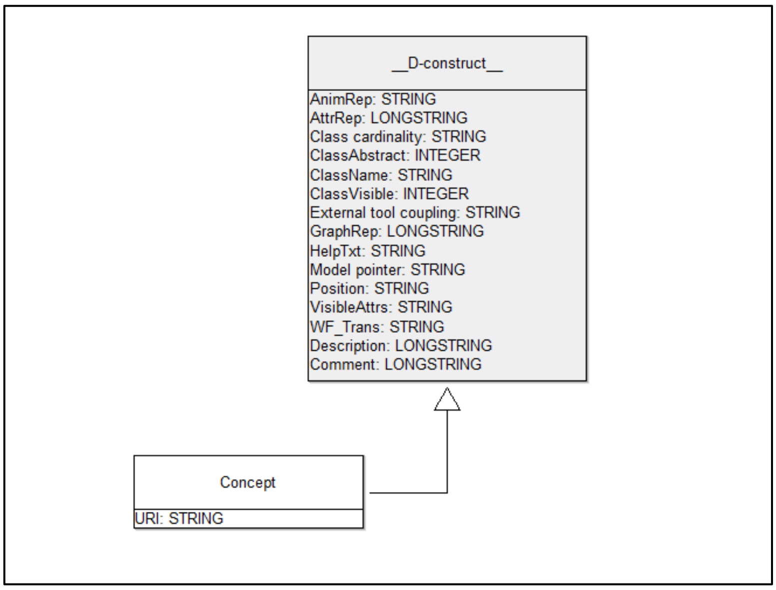

Figure 27: Semantic Transit Model Class Diagram

For both layers, business and IT, the links to the KPI and the decision models are done with the weaving concept. The KPI definitions consist of two model types:

(a) KPI-Cause and Effect Model

Within the cause and effect model the definition of the Strategic and Operational goals are done. These concepts can be quantified by performance indicators. The Class diagram of the Cause and Effect Model is shown in Figure 28.

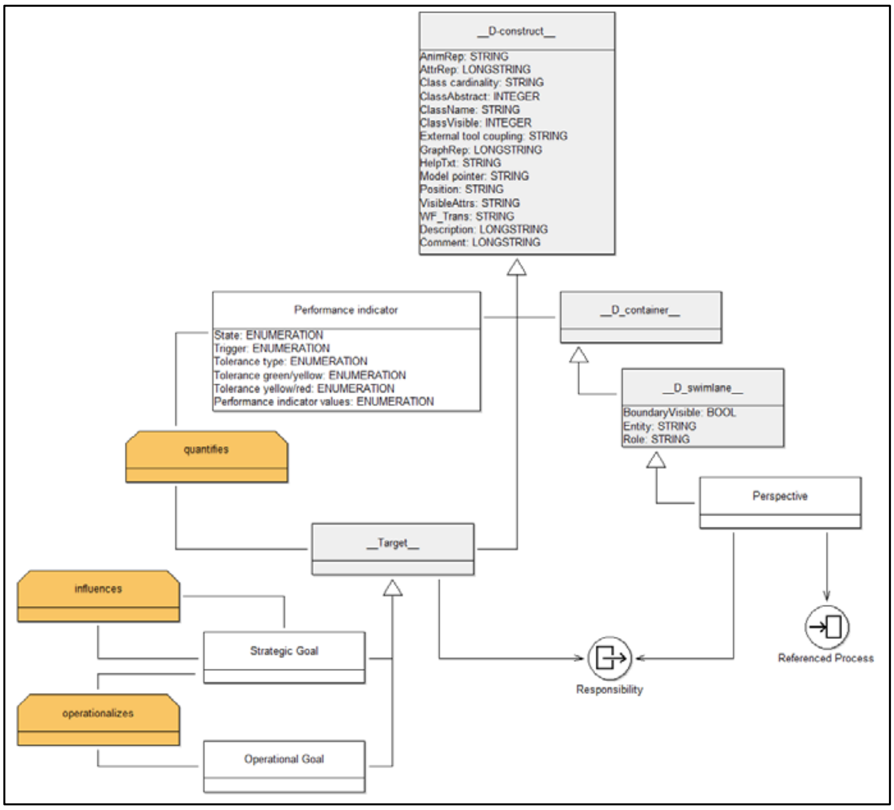

Figure 28: KPI-Cause and Effect Model

(b) KPI Indicator Model

The indicator models contain all relevant information concerning indicators. In this model type one can define the formulas of the indicator calculation as well as the data sources needed for the operational data access, e.g. databases or Excel tables. The class diagram of the KPI indicator models are shown in Figure 29.

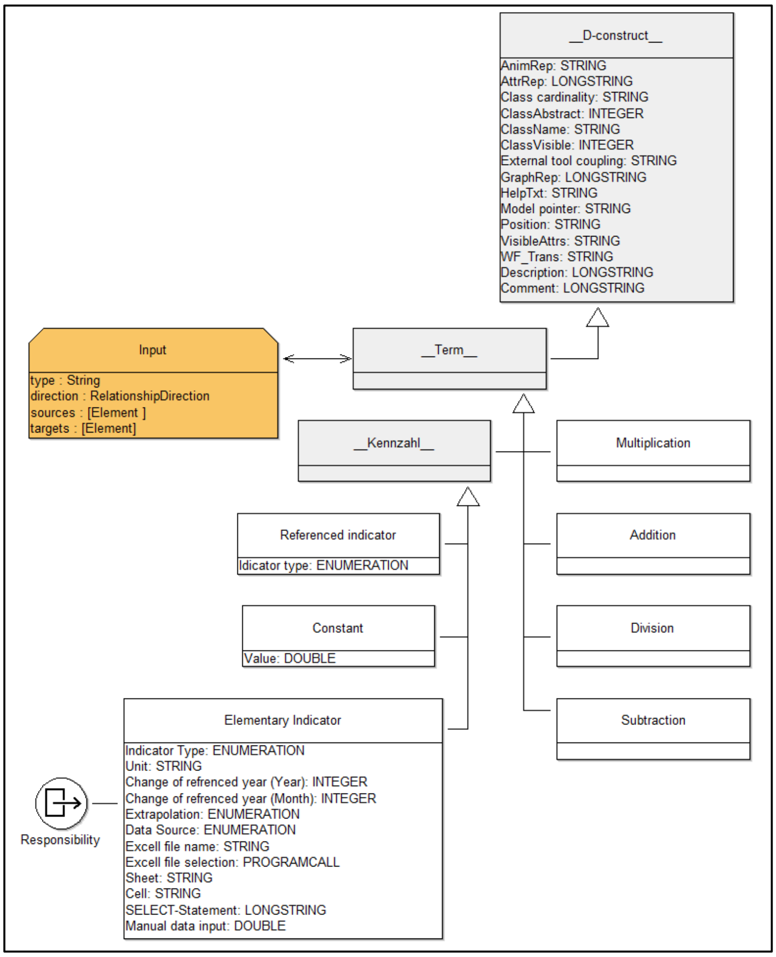

Figure 29: KPI-Indicator Model Class Diagram

The second link to the business processes beside the KPI models are the link to the decision models. This model type gives you the possibility to design your business decisions. The aim is to enable business users (e.g. analysts, technical developers) to comprehend the decisions that have been defined. The class diagram of the decision models is shown in Figure 30.

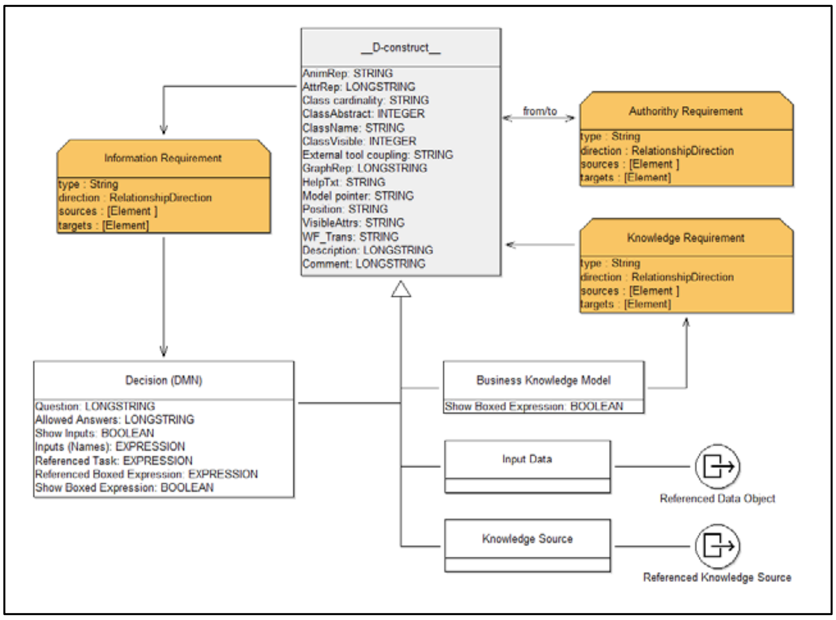

Figure 30: Decision Model Class Diagram

The packaging of the business process, workflow, decision, and KPI models is done in the 'BPaaS Alignment Model' and is done with the weaving concept. The concept 'BPaaS Design Package' within this model type is used as a 'pool' and collects all the models and creates an export package. This package contains the following model exports:

i. Business process export as a BPMN file.
ii. Business process as a portable network graphics (PNG)
iii. Workflow export as a BPMN file.
iv. Workflow as a portable network graphics (PNG).
v. Key performance indicators export as a cockpitXML export.
vi. Key performance indicators as a portable network graphics (PNG).
vii. Decision model export as a DMN file.
viii. Decision model as a portable network graphics (PNG).

The class diagram of the 'BPaaS Alignment Model' is shown in Figure 31.

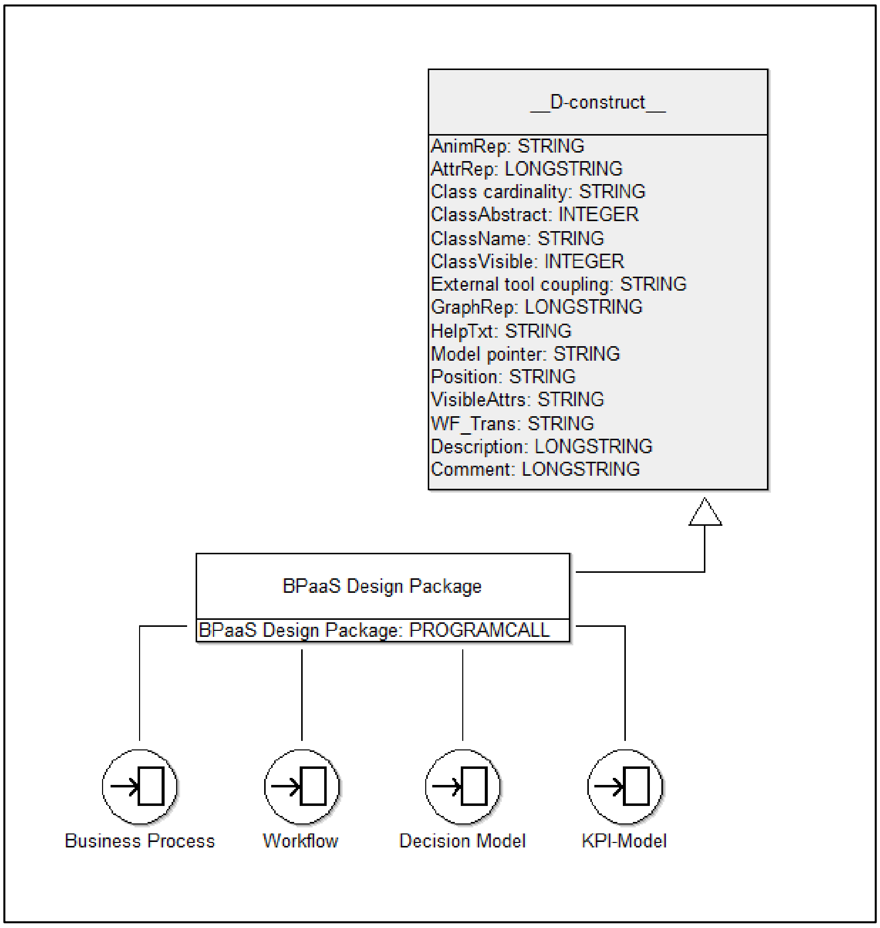

Figure 31: BPaaS Alignment Model Class Diagram

4.2.3 The Formal Description of the BPaaS Meta Model

The meta model overview is provided in FDMM form (Fill et al. 2012).

A meta-model is a tuple MM = $\langle \mathbf{MT}, \preccurlyeq, \mathrm{domain}, \mathrm{range}, \mathrm{card} \rangle$ where MT is the set of the defined model types, i.e. for i=1,…,m we have MT={$\mathrm{MT_1}, \mathrm{MT_2}, \dots, \mathrm{MT_m}$}.

The $\mathbf{MT_i}$'s (i=1,..,m) are themselves tuples $\mathbf{MT_i}$=$\langle \mathrm{O_i^T}, \mathrm{D_i^T}, \mathrm{A_i} \rangle$, where:

- $\mathrm{O_i^T}$ is the set of object types or classes,
- $\mathrm{D_i^T}$ is the set of data types, and
- $\mathrm{A_i}$ is the set of the attributes.

In CloudSocket we use the following model types:

- MT_1 Business Process Diagram (BPD defined in BPMN),
- MT_2 Company Map (CM supporting BPMN)
- MT_3 Data and Document Model (DDM supporting BPMN)
- MT_4 Working Environment Model (WE supporting BPMN)
- MT_5 Decision Requirement Diagram (DRD defined in DMN),
- MT_6 KPI- Cause and Effect Model (CEM)
- MT_7 KPI-Indicators Model (INDM)
- MT_8 Service Description Model (SDM)
- MT_9 Semantic Transit Model (STM using RDF)
- MT_{10} BPaaS Alignment Model (BPAM)

To introduce the relevant parts of FDMM used in our CloudSocket definition we introduce:

MT={BPD, CM, DDM, WE, DRD, CEM, INDM, SDM, STM, BPAM}.

$\preccurlyeq$ defines an ordering on $\mathrm{O^T}$. Let $\mathrm{o_1^t}, \mathrm{o_2^t} \in \mathrm{O^T}$ we say $\mathrm{o_1^t}$ is subclass of $\mathrm{o_2^t}$, if $\mathrm{o_1^t} \leq \mathrm{o_2^t}$

The domain is a function with domain: A→ $\mathrm{P(O^T)}$

The range maps an attribute to the power set of all pairs of classes and model types, all data types, and all model types.

$$\text{range: } \mathrm{A} \rightarrow \mathrm{P}(\cup_j(\mathrm{O_j^T} \times \{\mathrm{MT_j}\}) \cup \mathrm{D^T} \cup \mathrm{MT})$$

The cardinality function

$$\text{card : } \mathrm{O^T} \times \mathrm{A} \rightarrow \mathrm{P}(\mathbb{N}_0^+ \times (\mathbb{N}_0^+ \cup \{\infty\}))$$

For details on the modelling language, please refer to the CloudSocket development space on ADOxx.org.

4.3 BPaaS Mechanisms and Algorithms

4.3.1.1 Horizontal BPMN and DMN & KPI Weaving

Weaving is a modelling technique were different model types are linked with each other. Additionally to the MTs defined or supporting BPMN (OMG 2011), we have to add

a) cloud specific extensions to define deployment rules. The deployment rules are defined on domain specific business process level using the standard DMN ().
b) primary and sub-goals, that are defined by the Cause-Effect model CEM as a cockpit, where the data of this target and current values can be defined in the KPI Indicator model INDM. These data can be read from excel files, databases or defined manually within the elementary indicator concept. The Cause-Effect model type CEM is defined as follows:

$$MT_{CEM} = \langle O_{CEM}^{T}, D_{CEM}^{T}, A_{CEM} \rangle$$

Where

$$\boldsymbol{O}_{CEM}^{T} = \{Perspective, Strategic\ goal, Operational\ goal, Performance\ indicator, operationalizes, influences, quantifies\},$$

$$\boldsymbol{D}_{CEM}^{T} = \{String, Double, Integer, Enum_{PER}, Enum_{ITYPE}, Enum_{DS}, Enum_{LT}\}$$

$$Enum_{PER} = \{Day, Week, Month, Quarter, Half-year, Year\}$$

$$Enum_{ITYPE} = \{Absolute, Relative\}$$

$$Enum_{DS} = \{Database, Excel\ file, Manual\ data\ input\}$$

$$Enum_{LT} = \{Less\ is\ better, More\ is\ better, Two-sided\}$$

$$\boldsymbol{A}_{CEM} = \{State, Periodicity, Change\ of\ refereced\ date\ (year), Change\ of\ refereced\ date\ (month), Indicator\ Type, Unit, Limit\ Type, Threshold\ green/yellow, Threshold\ yellow/red, Data\ source\}$$

The formalism of the Indicators model type, that acts as data container for the Cause and Effect Models is defined as follows:

$$MT_{IND} = \langle O_{IND}^{T}, D_{IND}^{T}, A_{IND} \rangle$$

Where,

$$\boldsymbol{O}_{IND}^{T} = \{Referenced\ indicator, Elementary\ indicator, Constant, Multiplication, Addition, Division, Subtraction\}$$

$$\boldsymbol{D}_{IND}^{T} = \{String, Integer, Double, Longstring, Enum_{Extrpol}, Enum_{DS}\}$$

$$Enum_{Extrpol} = Extrapolate\ to\ end\ of\ the\ month, Fetch\ value\ from\ the\ database$$

$$Enum_{DS} = \{Database, Excel\ file, Manual\ data\ input\}$$

$$\boldsymbol{A}_{IND} = \{Name, Indicator\ type, Value, Unit, Change\ of\ refrenced\ year\ (Year), Change\ of\ refrenced\ year\ (Month), Extrapolation, Data\ Source, Excell\ file\ name, Excell\ file\ selection, Sheet, Cell, SELECT-Statement, Manual\ data\ input\}$$

Hence a horizontal weaving mechanism from BPD to DRD is implemented in the following way:

$\mathrm{MT}_{\mathrm{BPD}_{\mathrm{spec}}} = \{\mathrm{MT}_{\mathrm{BPD}_{\mathrm{spec}}}, \preccurlyeq, \mathrm{domain}, \mathrm{range}, \mathrm{card}\},$

$$\mathrm{O}^{\mathrm{T}}_{\mathrm{BPD}} := \mathrm{O}^{\mathrm{T}}_{\mathrm{BPD\_spec}}$$

$$\mathrm{A}_{\mathrm{BPD}} \coloneqq \{\mathrm{A}_{\mathrm{BP}_{\mathrm{spec}}}, \text{Referenced Decisions-from}, \text{Referenced Decisions-to}\}$$

Currently the deployment rules are expressed as “decisions” following the DMN specification. It is expected, that support and simplifications will be needed to ease the description of deployment relevant rules on such high level.

4.3.1.2 Vertical BPMN cloud-specific Enrichment

In order to support the vertical alignment between layer I – business processes – and layer II – workflows – the Business Process Diagram (BPD) model type is extended with a cloud-specific description concept.

This new concept is initially named Service Description (SD) is introduced. This concept improves the communication between the business process designer and a workflow engineer. It is based on the FODA approach (Kang 1990), where each business process activity is analyzed according IT requirements. The class ‘Service Description’ contains attributes which describes the requirements derived from the business process for cloud services considering (a) technical, (b) domain, and (c) business dimensions.

Those attributes are in text format to allow free description of requirements. The expectation is that the free text format can partly be transformed into a semi-formal representation after experience and user feedback. A semantic lifting of those requirements is foreseen to partly automate the business and IT alignment.

Fist the Attribute of the BPD is extended with:

$$\mathrm{A}_{\mathrm{BPD}} \coloneqq \{\mathrm{A}_{\mathrm{BP}_{\mathrm{spec}}}, \text{Referenced Service Description-from}, \text{Referenced Service Description-to}\}$$

We have introduced a new model type including the class “Activity Description” so that

$$MT_{SDM} = \langle O^T_{SDM}, D^T_{SDM}, A_{SDM} \rangle \text{ where},$$

$$O^T_{SDM} = \{Activity\ Descripion\},$$

$$\boldsymbol{D}^T_{SDM} = \{String, Longstring, \boldsymbol{Enum}_{GF}, \boldsymbol{Enum}_{GID}, \boldsymbol{Enum}_{GOD}, \boldsymbol{Enum}_{GPS}, \boldsymbol{Enum}_{VC}, \boldsymbol{Enum}_{PAY}, \boldsymbol{Enum}_{DLO}, \boldsymbol{Enum}_{DSC}\}$$

$$\boldsymbol{Enum}_{GF} = \{not\ applicable, Apply\ Rule, Manual, Receive, Transform, Send, Store, Wait, Update, Create, Assign, Others\}$$

$$\boldsymbol{Enum}_{GID} = \{Text, Number, Date, Picture, File, Geo\ Data, User\ Data, Weblink, Configuration\ Format, Others\}$$

$$\boldsymbol{Enum}_{GOD} = \{Text, Number, Date, Picture, File, Geo\ Data, User\ Data, Weblink, Configuration\ Format, Others\}$$

$$\boldsymbol{Enum}_{GPS} = \{on-demand, no-specific\ time\ slot, Calendar\ specific\ time\ slots, Season\ specific\ time\ slot, Day\ specific\ time\ slot, Other\ time\ slot\}$$

$$\boldsymbol{Enum}_{VC} = \{Avoid\ Vendor\ Lock, Trust, Security, Helpdesk\ available, Onsite\ Visits\ possible, Maintenance\ available, Training\ offered\}$$

$$\boldsymbol{Enum}_{PAY} = \{Not - Decided - Yet, Pay - Per - Process, Pay - On - Demand, Pay - Monthly, Pay - Three - Monthly, Pay - Bi - Yearly, Pay - Yearly, Other\}$$

$$\boldsymbol{Enum}_{DLO} = \{World - Wide, Europe, National, Regional, Local, Other\}$$

$$\boldsymbol{Enum}_{DSC} = \{ITIL, ISO\ 27001, BASEL\ III, other\}$$

$$\boldsymbol{A}_{SDM} = \{Name, Description, General\ Functionality, Access\ External\ Functional\ Ontology, External\ Functional\ Annotation, Free\ Functional\ Keywords, Comments\ on\ Fuctional\ Requirements, General\ Input\ Datatype, Access\ External\ Data\ Input\ Ontology, External\ Data\ Input\ Annotation, Free\ Input\ Data\ Keywords, Comments\ on\ Data\ Input\ Requirement, General\ Output\ Datatype, Access\ External\ Data\ Output\ Ontology, External\ Data\ Output\ Annotation, Free\ Output\ Data\ Keywords, Comments\ on\ Data\ Output\ Requirement, Target\ Error\ rate\ in\ \%, Green\ Yellow\ Error\ Rate\ Threshold\ in\ \%, Yellow\ Red\ Error\ Rate\ Threshold\ in\ \%, Targeted\ Availability\ Time\ in\ \%, Green\ Yellow\ Availability\ Time\ Threshold\ in\ \%, Yellow\ Red\ Availability\ Time\ Threshold\ in\ \%, Sample\ Service\ Link, Sample\ Service\ Comments, General\ Planning\ Schedule, Annotate\ with\ External\ Planning\ and\ Scheduling\ Ontology, External\ Planning\ and\ Scheduling\ Annotation, Free\ Planning\ and\ Schedule\ Keywords, Comments\ on\ Planning\ and\ Scheduling\ Requirements, Vendor\ Criteria, Access\ External\ Trust\ Ontology, External\ Trust\ Annotation, Access\ External\ Security\ Ontology, External\ Security\ Annotation, Payment, Costs, Comments\ on\ Business\ Related\ Requiments, Data\ Location\ Overview, Comments\ on\ Data\ Location, Domain\ Specific\ Certification, Domain\ Specific\ Certification\ Keyword, Domain\ Specific\ Data\ Protection, Domain\ Specific\ Data\ Protection\ Keyword, Domain\ Specific\ Maturity\ Model, Domain\ Specific\ Maturity\ Model\ Keyword, Relevant\ Regulation\ List\}$$

Attribute attachments:

$$\forall \text{ attr } \in \{\text{A}_{SDM}\}: \text{domain(attr)} = \{Activity\ Description\},$$

$$\forall \text{ attr } \in \{\text{A}_{SDM}\}: \text{ range(attr)= rg} \in \{String, Longstring, \boldsymbol{Enum}_{GF}, \boldsymbol{Enum}_{GID}, \boldsymbol{Enum}_{GOD}, \boldsymbol{Enum}_{GPS}, \boldsymbol{Enum}_{VC}, \boldsymbol{Enum}_{PAY}, \boldsymbol{Enum}_{DLO}, \boldsymbol{Enum}_{DSC}\}$$

$$\forall \text{ attr } \in \{\text{A}_{SDM}\}: card(\text{Activity Description}, \text{attr}) = \langle 1,1 \rangle,$$

Currently the vertical weaving from business process to workflow is performed due to intellectual manual modelling, it is expected that semantic support can be introduced, when aforementioned attributes are semantically enriched.

4.3.1.3 Vertical Alignment with Semantic Lifting

There are seven different ways of implementing semantic lifting for weaving between the different modelling layers.

Here we refer to the so-called Semantic Transit Model Type (STM) as MT_7. This implementation enables to semantically lift any object of the business process or workflow models with a set of concepts in the Semantic Transit Model, which have been imported from the ontology.

Hence, the Semantic Transit Model Type is defined as:

$MT_{ST} \coloneqq \{O_{ST}, D_{ST}, A_{ST}\}$, where

$$O_{ST} \coloneqq \{Concept\}$$

$$D_{ST} \coloneqq \{String\}$$

$$A_{ST} \coloneqq \{\, Name, URI, Referenced\ concept\text{-}from, referenced\ concept\text{-}to\}$$

In order to enable the semantic lifting with a reference from any object in any model type, we use the super class of all objects:

$$\forall\ class\ x\ \in \{O^T_{BPD}, O^T_{CD}, O^T_{CM}, O^T_{DDM}, O^T_{WE}, O^T_{DRD}, O^T_{STM}\}$$
$$\exists\ class\ \boldsymbol{super}\text{: super} \geq x$$

and define that link in form of:

domain($Referenced\ concept\text{-}from$)={ $Referenced\ concept$}

range($Referenced\ concept\text{-}from$)= {super}

card($Referenced\ concept\text{-}from = \langle m, \mathrm{n}\rangle$)   for m,n$\in N$

This semantic lifting via a MTSTM provides a tool support via the references – implemented as so-called INTEREFS in ADOxx – but does not need a full established Ontology Management System (OMS) interaction.

Hence, it is expected that the interaction with the OMS will be introduced on a later stage of the project, in order to support also the semantic lifting in tight interaction with an OMS.

Aforementioned results can be downloaded on ADOxx.org (http://www.adoxx.org).

4.3.1.4 BPaaS Alignment Model as a Packaging Role

The packaging of the business process, workflow, decision, and KPI models is done in the 'BPaaS Alignment Model' and is done with the weaving concept. The concept 'BPaaS Design Package' within this model type is used as a 'pool' and collects all the models and creates an export package.

The formalism of this model type is as follows:

$$\boldsymbol{M}_{BPAM} = \langle \boldsymbol{O}^T_{BPAM}, \boldsymbol{D}^T_{BPAM}, \boldsymbol{A}_{BPAM}\rangle\text{, where}$$
$$\boldsymbol{O}^T_{BPAM} = \{BPaaS\ Design\ Package\}$$

$$\boldsymbol{D}^T_{BPAM} = \{String\}$$

$$\boldsymbol{A}_{BPAM} = \{Name, Business\ Process - from, Business\ Process - to, Workflow - from, Workflow - to, Decision\ Model - from, Decision\ Model - to, KPI - Model - from, KPI - Model - to\}$$

The weaving concepts are defined as follows:

$$domain(Business\ Process - from) = \{Business\ Process\}$$

$$range(Business\ Process - from) = \{BPaaS\ Design\ Package\}$$

$$card(Business\ Process, Business\ Process - from) = \langle 1,1 \rangle$$

$$domain(Business\ Process - to) = \{Business\ Process\}$$

$$range(Business\ Process - to) = \{M_{BPM}\}$$

$$card(Business\ Process, Business\ Process - to) = \langle 1,1 \rangle$$

$$domain(Workflow - from) = \{Workflow\}$$

$$range(Workflow - from) = \{BPaaS\ Design\ Package\}$$

$$card(Workflow, Workflow - from) = \langle 1,1 \rangle$$

$$domain(Workflow - to) = \{Workflow\}$$

$$range(Workflow - to) = \{M_{BPM}\}$$

$$card(Workflow, Workflow - to) = \langle 1,1 \rangle$$

$$domain(Decision\ Model - from) = \{Decision\ Model\}$$

$$range(Decision\ Model - from) = \{BPaaS\ Design\ Package\}$$

$$card(Decision\ Model, Decision\ Model - from) = \langle 1,1 \rangle$$

$$domain(Decision\ Model - to) = \{Decision\ Model\}$$

$$range(Decision\ Model - to) = \{M_{\mathrm{DRD}}\ \}$$

$$card(Decision\ Model, Decision\ Model - to) = \langle 1,1 \rangle$$

$$domain(KPI - Model - from) = \{KPI - Model\}$$

$$range(KPI - Model - from) = \{BPaaS\ Design\ Package\}$$

$$card(KPI - Model, KPI - Model - from) = \langle 1,1 \rangle$$

$$domain(KPI - Model - to) = \{KPI - Model\}$$

$$range(KPI - Model - to) = \{M_{CEM}\}$$

$$card(KPI - Model, KPI - Model - to) = \langle 1,1 \rangle$$

5 BPAAS ONTOLOGY DEVELOPMENT

This chapter describes the prototype for the BPaaS Ontology, which is implemented as an extension of the ArchiMEO enterprise ontology (see section 3.1.2 and Figure 32). This chapter has a focus on the cloud-specific extensions, which are needed for smart alignment of business and IT in the cloud. The cloud-specific extensions were determined from the analysis of the business scenarios (section 2.2) and the competency questions (section 2.3). In order enable semantic lifting of the BPaaS modelling method, it is taken care that the BPaaS Ontology is consistent with the modelling method as described in section 4.

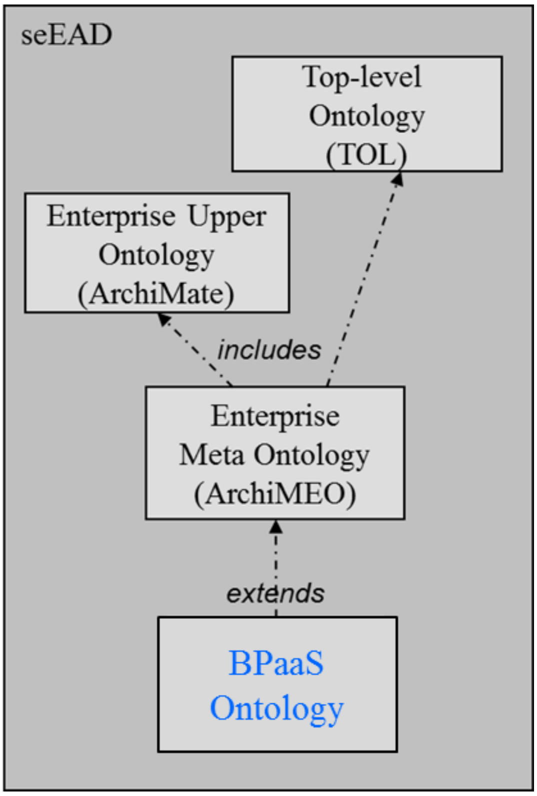

Figure 32: Extending ArchiMEO with the BPaaS Ontology

5.1 Scope of the BPaaS Ontology

According to the CloudSocket architecture, which is described in Deliverable D4.1 (CloudSocket 2015b), the focus of the BPaaS Design environment is on Business Process and Workflow Modelling. Business Process and Workflow Modelling correspond to the top two layers of the enterprise architecture framework according to TOGAF and ArchiMate (see Figure 33). Since the ArchiMEO ontology covers all three enterprise architecture layers, the semantics for the BPaaS Design Environment can be modelled as an extension of the ArchiMEO.

It should be noted that according to the CloudSocket architecture (see Figure 1) the design of Service Bundles is not part of the BPaaS Design Environment and thus is not covered by the BPaaS Ontology as described in this report.

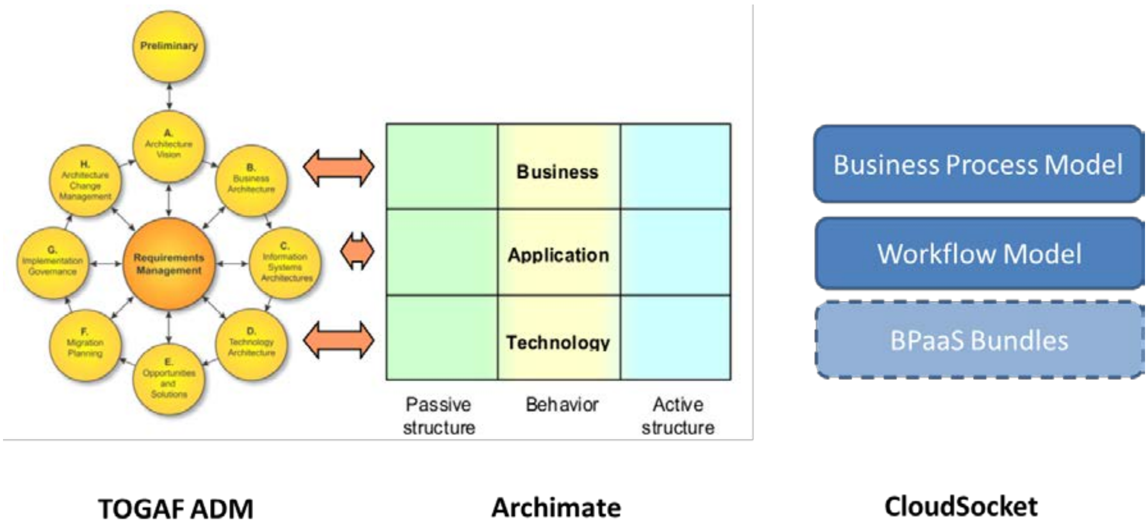

Figure 33 Layers of the CloudSocket Architecture

Semantic lifting allows integrating human-interpretable models with machine interpretable models. The human-interpretable models are represented in the modelling tool ADOxx.org. The corresponding model types explained in section 4.2.1 and visualized in the BPaas Meta Model Stack of Figure 21.

- Business Process Model: Business Processes (modelled in BPMN), organisational structure model, and document model
- Workflow Model: Technical extensions of BPMN business process models

The two layers are linked using several other model types: Service description model, Decison Model, KPI Models.

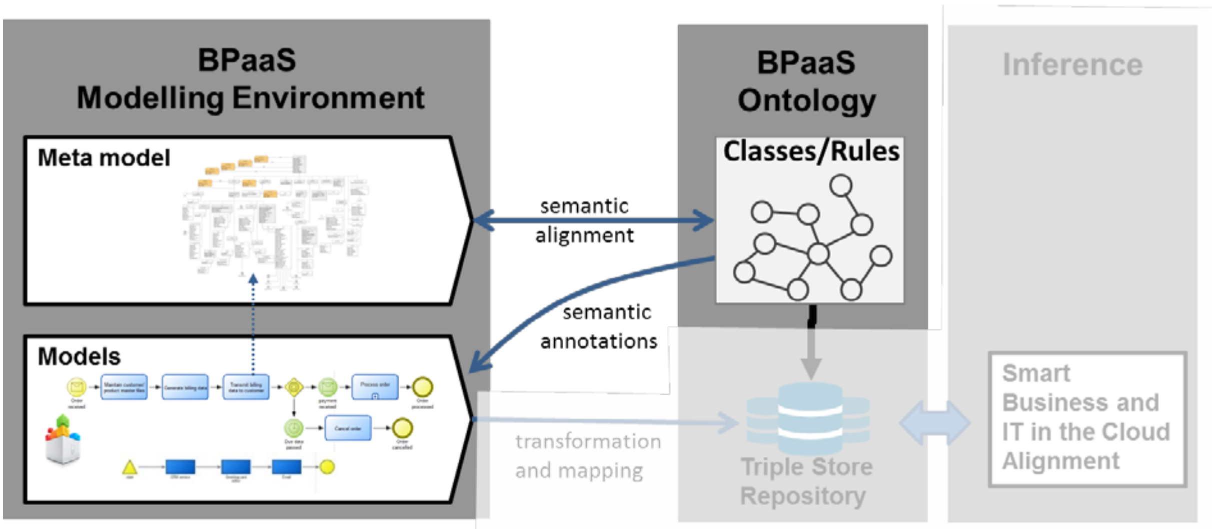

Figure 34: Business Process Semantic Lifting

For the semantic lifting of these model types, ArchiMEO contains classes which represent the modelling elements of these modelling languages. As an example, Figure 35 shows the ArchiMEO classes of BPMN modelling elements.

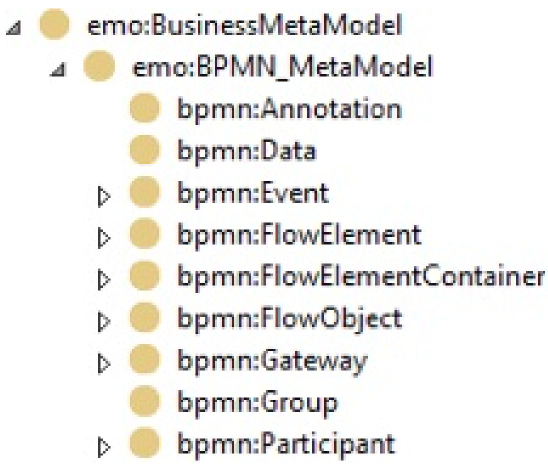

Figure 35 ArchiMEO classes of BPMN modelling elements

The elements of these modelling languages are embedded in the concepts coming from ArchiMate. For example, a BPMN activity is represented as a subclass of a Business Activitiy, which itself is a subclass of a Behaviour element in ArchiMate. The namespaces in Figure 36 indicate the sources of the classes ("archi" stands for "ArchiMate" and "bpmn" stands for ""BPMN"").

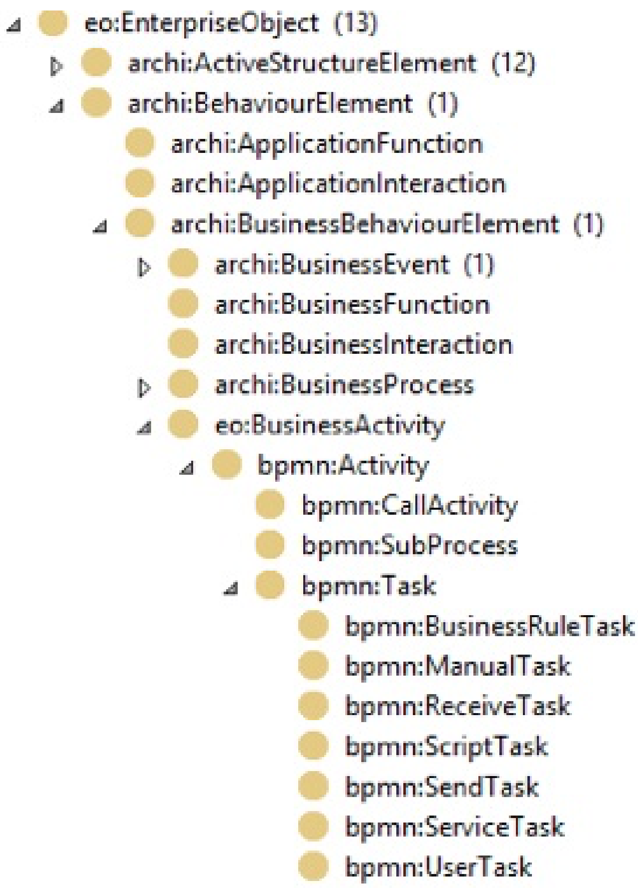

Figure 36 BPMN elements embedded in the ArchiMEO ontology

5.2 Concepts for Business Process as a Service

In this section we describe the concepts for BPaaS Ontology and their integration in the ArchiMEO ontology. The class diagrams represent the conceptual model for the BPaaS Ontology.

The classes of the conceptual model for BPaaS are then integrated in the ArchiMEO ontology using the new namespace "clso". 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.

5.2.1 Task Ontology

To specify the functionality of a process task, each task in a process model can be annotated with a task category which is defined in the task ontology. As shown in section 3.1.4, there exist several task ontologies. For the BPaaS Ontology we chose the APQC Process Classification Framework (APQC 2014), because it is the most used process framework in the world. It creates a common language for organizations to communicate and define work processes comprehensively and without redundancies (APQC 2015).. Organizations are using it to support benchmarking, manage content, and perform other important performance management activities. There are also industry-specific Process Classification Frameworks, e.g. for banking, insurance, consumer products and many more. The APQC Process Classification Framework comprises five levels which start from 13 generic business process categories and go down to particular tasks (see Figure 37).

Figure 37: Hierarchy of APQC's Process Classification Framework (APQC 2014)

Teuteberg et al. (2009) transformed the APQC Process Classification Framework into a domain ontology. We adopted this transformation to the latest version of the PCF. The leads to five concepts ("Category," "Process Group," "Process," "Activity" and "Task") and individuals for each entry of the PCF.

Each element of this taxonomy is also linked to performance metrics, for which formulas and respective units are provided.

Each tasks stands for a dedicated activity to cover. This functionality can be annotated using the BPMN ontology (see Figure 38). This ontology already contains the separation of Call Activity, Sub Process and Task that can be used for a detailed classification. The class itself takes activity grouping such as Apply Rule; Manual; Receive;

Transform; Send; Store; Wait; Update; Create; Assign; Others. These basic groupings help to identify the needed service bases on the functionality of the activity.

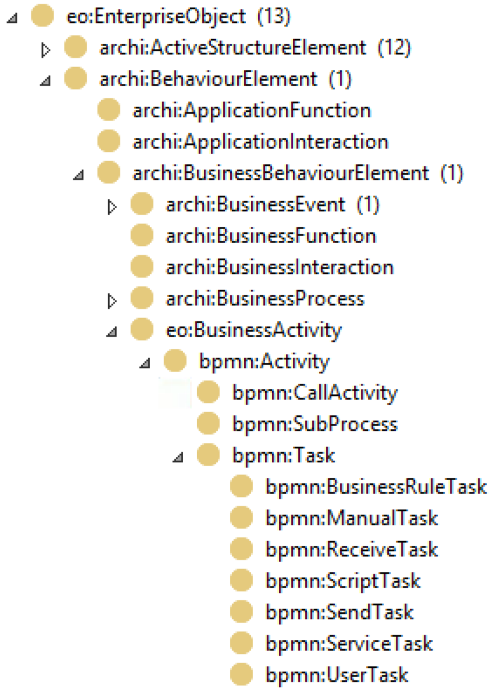

Figure 38: Ontology - Activity

5.2.2 Business Perspective

The business perspective aims to describe the relevant criteria related to the buying/decision making process of the service, such as are payment, contract/pricing and additional services. As reflected the business perspective competency questions, a potential CloudSocket Consumer wants to know whether the Cloud Provider is trustworthy.

In general, trust can be regarded as a basic expectation regarding the behaviour of an interaction partner. “Trust in cloud computing and technology is fundamentally as subjective as its counterpart”. Hence the concept of trust is manifold and difficult to capture (Meixner & Buettner 2012). In order to narrow down the concept of trust with respect to the business perspective we introduce the information economic theory. Latter serves as framework which allows deriving the relevant criteria from business perspective.

“One of the ways of probing customer behaviour with regard to services is to build a probable cause and effect relationship between service characteristics and their effect on customer search and buying behaviour” (Verma 2012). The information economy theory provides a model for information search in situations where the consumer is uncertain about the service quality. Starting from a buyer perspective, the SEC framework states that each service has three types of attributes: search attributes, experience attributes (Nelson 1972, 1974) and credence attributes (Darby & Karni 1973) as depicted Figure 39. Products and services might have one, two or all three types of aforementioned attributes (Srinivasan & Till 2002).

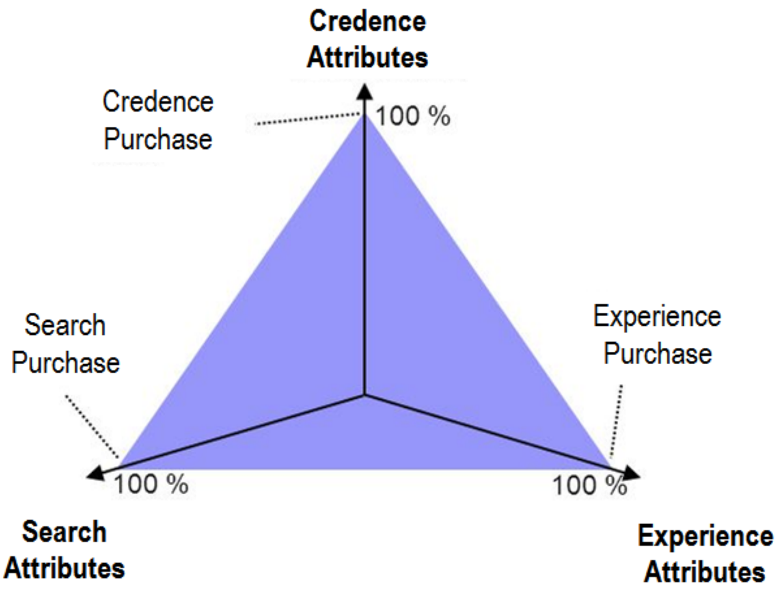

Figure 39: Information Economic Theory

Search attributes are the ones that can be evaluated before the consumer buys a product, because the information is publicly available. For instance, decision makers can evaluate the website of a vendor, the price of services or physical attributes of a good. Experience attributes can only be evaluated after the product or service has been purchased. For instance, if consumers have already bought products or services from a specific vendor they know that they can rely on this enterprise. Here the brand of a product or the reputation of an enterprise plays an important role, since consumers assume a consistent quality of products and services (Nelson 1973, 1974).

Credence products are very difficult to evaluate, even after several purchases. The buyer can not verify credence attributes before or after buying the service, but he has to trust the information provided by the supplier. Compared to other services a credence service is more likely to be customized to the needs of customers. Hence the consumer bears higher risks because it is more difficult to compare with other services. Because the consumers can't confidently evaluate, they have the tendency to rely on word of mouth (testimonials), reputation of the brand name, the price or the quality of service (Darby & Karni 1973).

From Business Process as a Service perspective the search attributes are of central importance. The search attributes have been analysed in order to develop the concepts in the ontology that are relevant from CloudSocket buyer perspective. Figure 40 depicts the overall conceptual model for the business perspective. The individual parts of the model are described in the following chapters.

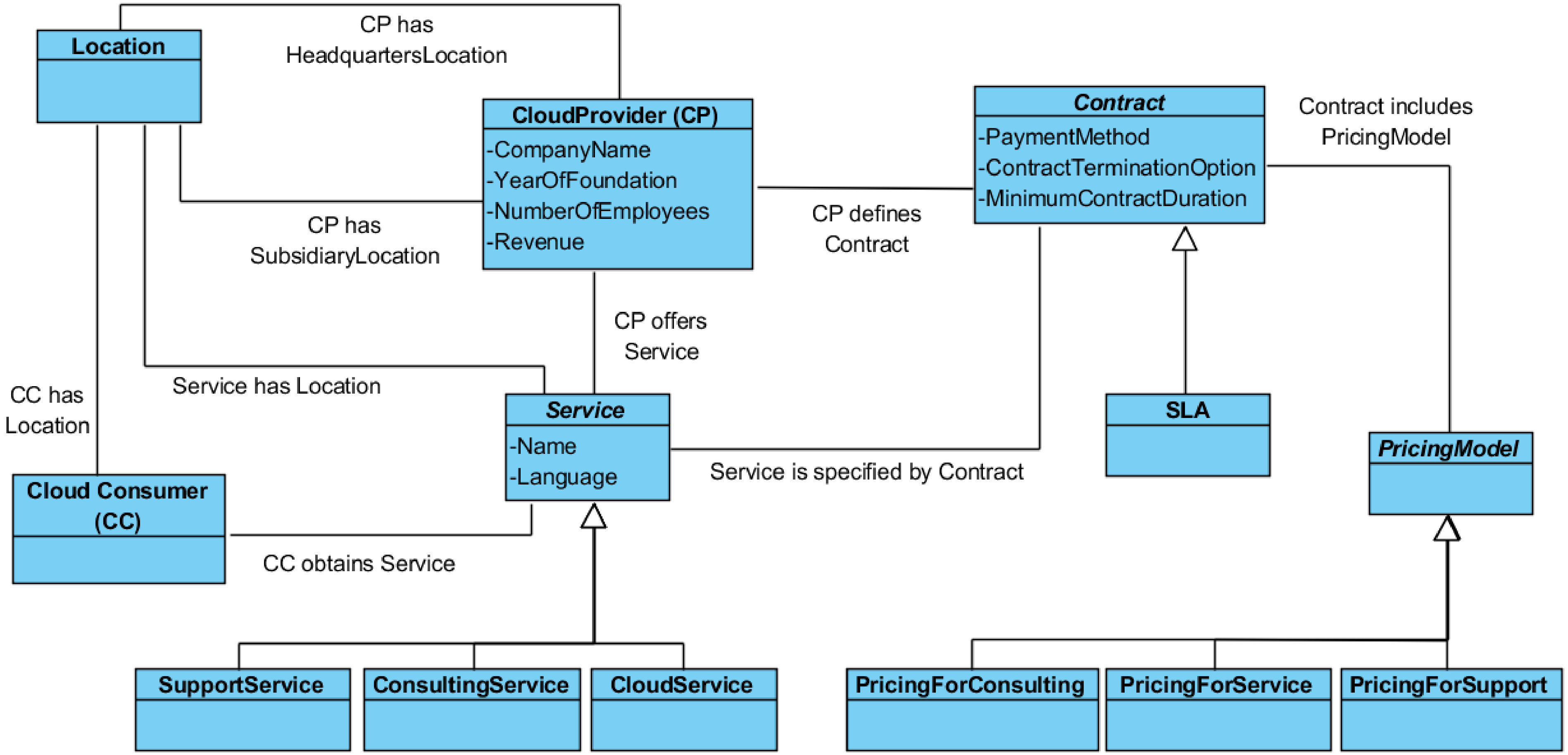

Figure 40: Business Perspective Meta Model

5.2.2.1 Cloud Provider

In order to specify how established the Cloud Provider is we use a pragmatic approach. We presume that there are several search attributes (as described in section 5.2.2) that provide the required information to evaluate whether the Cloud Provider is an established enterprise. For instance, the foundation date of the company is a criterion which shows how long the company 'survived' on the market. We start from the assumption that the longer the company exists (or the division that offers Cloud Services), the higher is the possibility that it 'survives' in the near future. Furthermore the number of employees and the turnover has been defined as relevant criteria.

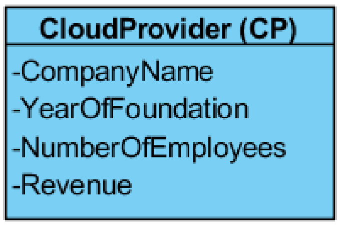

Figure 41: Cloud Provider

Figure 42 shows that Cloud Provider and Cloud Consumer are modelled as subclasses of the ArchiMate Business Role in the ArchiMEO ontology.

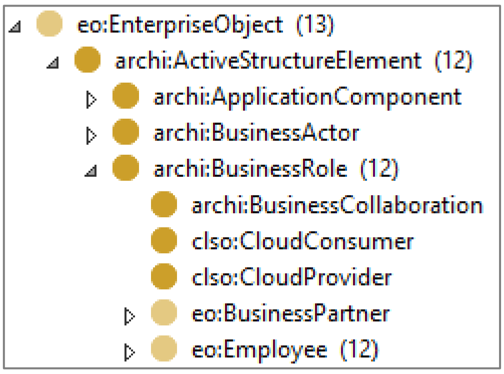

Figure 42: Cloud Consumer and Cloud Provider in ArchiMEO Ontology

5.2.2.2 Cloud Provider Offers Service

Figure 43 shows that a Cloud Provider offers one or more Cloud Service at different levels (PaaS, IaaS, SaaS, and BPaaS). These levels can reach from enabling the creation of virtual machines to host software components to the direct SaaS or BPaaS call to realize certain functionalities. In general, a service has a relationship to the concept location, since a Cloud Provider might restrict the offered service to certain locations. Furthermore, the Cloud Provider offers services which complement the Cloud Service. An additional service can be either a support service or a consulting service. Furthermore there are different types of consulting services, such as on site consulting or online training. A support service ensures that the Cloud Consumer gets support for the offered service, for instance, hotline or online support.

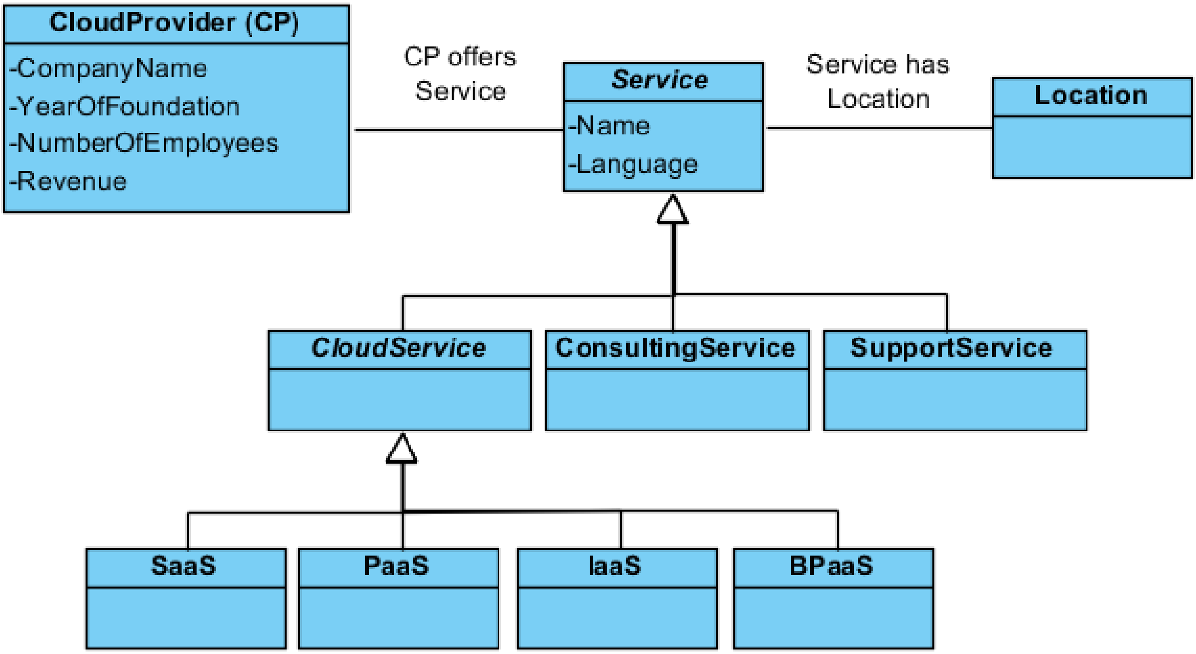

Figure 43: Cloud Provider offers Service

ArchiMate has services on business, application and technology layer. As can be seen in Figure 44, the CloudService is modelled as a service on the application layer, while ConsultingService and SupportService are regarded as services on the business layer.

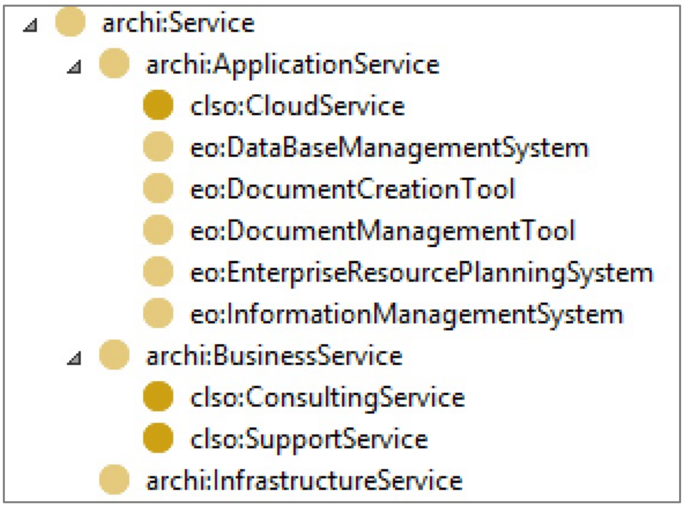

Figure 44: Services in ArchiMEO Ontology

5.2.2.3 Cloud Provider Has Location

The Cloud Provider can have different types of locations. From business perspective the headquarters and subsidiaries are relevant. A small Cloud Provider might have only one location. A big enterprise, for instance, has a headquarters and in addition several subsidiaries in different countries over the world. The competency questions revealed that this differentiation is relevant with respect to Cloud Consumer needs. Some Consumers want to obtain services from a provider who is located in their country or who has at least a subsidiary in their country or region.

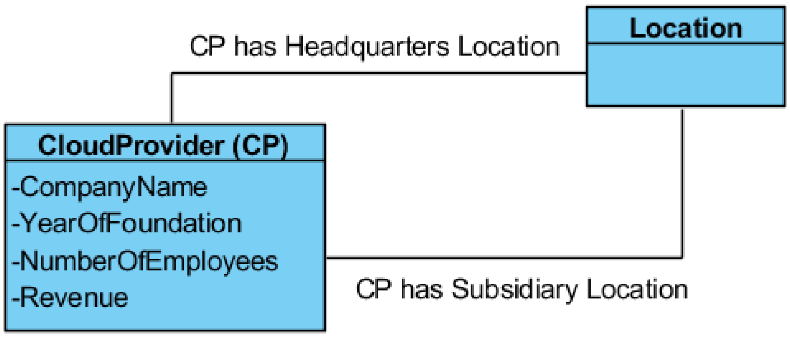

Figure 45: Cloud Provider Has Location

For the location of the Cloud Service the corresponding concept of the top level ontology can be reused (see Figure 46).

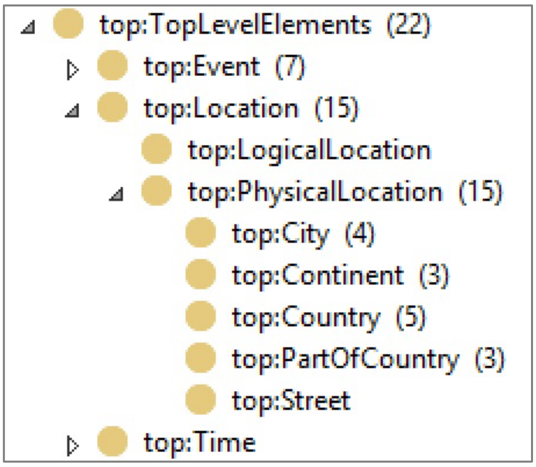

Figure 46: Location Concepts in ArchiMEO Ontology

5.2.2.4 Cloud Provider Defines Contract

The Cloud Provider defines the contract for the offered Cloud Service as shown in Figure 47. The payment method, depicted in the concept of the contract, refers to the kind of payment, such as credit card or invoice. The contract termination option describes when the contract can be terminated (e.g. at any time or with 30 days period of cancellation). Some Cloud Providers define a minimum contract length depending on the service. For instance, a Cloud Service has to be obtained for minimum one year. In this case an enterprise is obliged to pay the Cloud Service for one year before the contract can be cancelled.

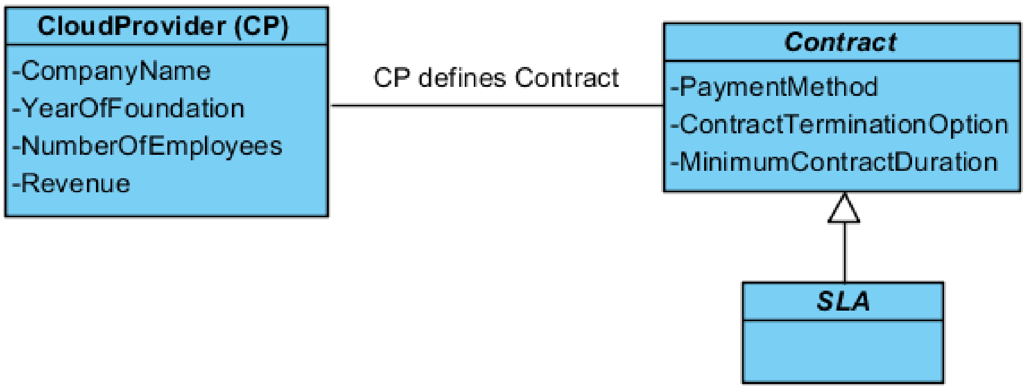

Figure 47: Cloud Provider Defines Contract

The pricing model describes how the Cloud Service is charged. Pricing models in the cloud are complex and often a result of competitive behaviour as well as co-operation between different actors of the cloud ecosystem (Laatikainen, Ojala & Mazhelis 2013). The pricing model for CloudSocket has three sub concepts as depicted in Figure 48. The first one is PricingForService. On a high level one can distinguish between two cloud pricing models: subscription and pay-as-you-go-model. The subscription is a sort of ‘bundle’. The Cloud Provider offers a defined set of functionalities that is charged, for instance, on monthly basis/per user or on yearly basis/per user. The pay-as-you-go model allows to charge by what is used. Enterprises select and pay only for the memory, storage, CPU etc. they need (Al-Roomi, Al-Ebrahim, Buqrais & Ahmad 2013). Amazon is an example for a pay-as-you-go pricing model. Customers pay for compute resources on an hourly basis from the time they launch it until they terminate it. However, for data storage and transfer they pay on a gigabyte basis.

Furthermore we distinguish between PricingForConsulting and PricingForSupport. Depending on the offered Cloud Service the Cloud Provider can define if and what kind of consulting services and/or are included. The example of the Cloud Provider Salesforce (2014) shows that depending on the Cloud Service different pricing models for consulting and/or support might apply. Salesforce offers for the Cloud Service ‘sales’ different editions and hence pricing models. The ‘unlimited edition’ includes 24/7 toll-free-support and online training. The ‘enterprise edition’ is cheaper, because it does not include aforementioned support and online training. Here additional fees apply.

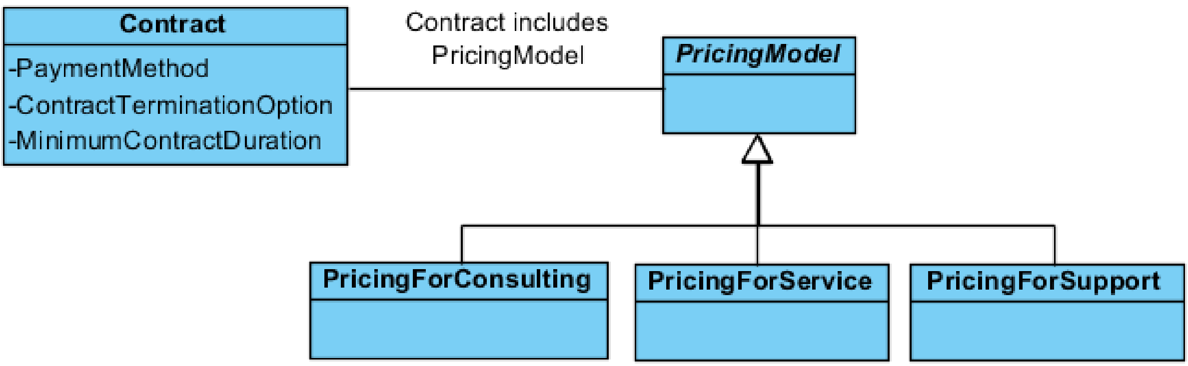

Figure 48: Contract Includes Pricing Model

The Service Level Agreement and Pricing Model are represented as a subclasses of the already existing concept Contract (see Figure 49).

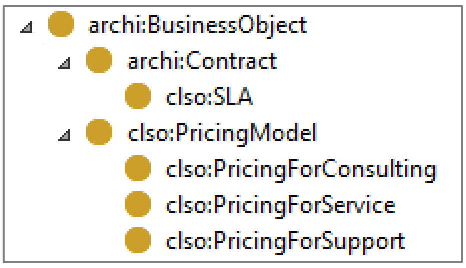

Figure 49: SLA and Pricing in ArchiMEO Ontology

5.2.3 Security/Legal Perspective

One of the major challenges companies face with respect to processing data in the cloud, is assessing the security risks. For example, for performing a business process like the one of the Christmas card use case there might be some data that require fine grained control to enforce protection (e.g. the personal data of the recipient or the image, which may be bound by copyright), whereas other data can be public or widespread (electronic Christmas card). As outlined in section 3.1.5 Basis for Ontology Development and reflected in the competency questions there is a set of security and legal requirements that are of central importance with respect to Cloud Services. This chapter introduces the concepts in the ontology required to specify these requirements.

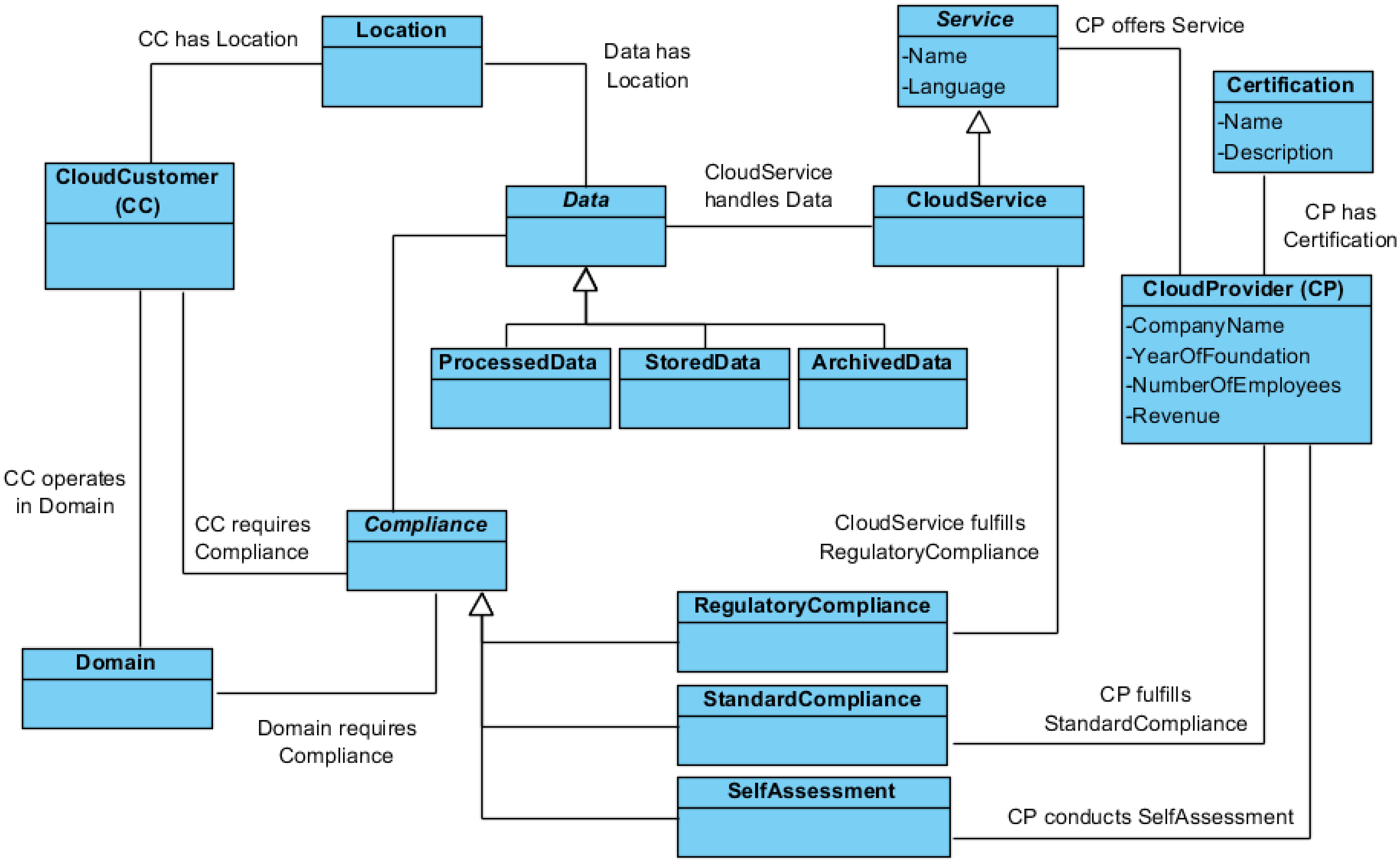

Figure 50: Security/Legal Perspective Meta Model

5.2.3.1 Customer/Domain Requires Compliance

Compliance requirements of enterprises are driven by two factors. On one hand the enterprise can operate in a specific domain, such as banking or insurance. In this case the Cloud Service and/or the Cloud Provider are subject to different regulations. On the other hand there might be requirements which arise due to company internal or external regulations. A decision of the management board, for instance, is an internal regulation. However, external regulations are prescribed by the law of the corresponding country.

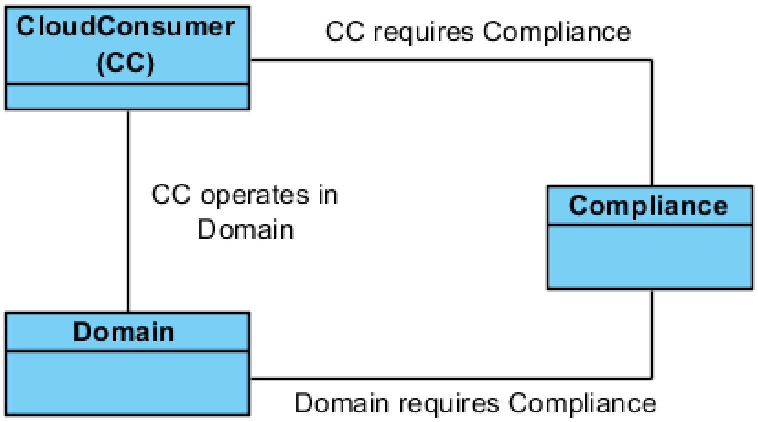

Figure 51: Cloud Consumer/Domain Requires Compliance

5.2.3.2 Regulatory Compliance, Standard Compliance and Self-Assessment

In CloudSocket we distinguish between three types of compliance: regulatory compliance (Eurocloud 2012), standard compliance (Meixner & Buettner 2012), and self-assessment. Regulatory compliance describes the rules that have to be considered in order to comply with laws of the corresponding country. Hence companies are obliged to follow laws related to data protection, accounting/tax regulations and industry specific regulations. Most regulations are related to data management. The second type - standard compliance - provides best practice recommendations. The goal is to ensure that an enterprise meets the requirements of accepted best practices in its domain or industry. The self-assessment describes the assessment through the Cloud Provider itself.

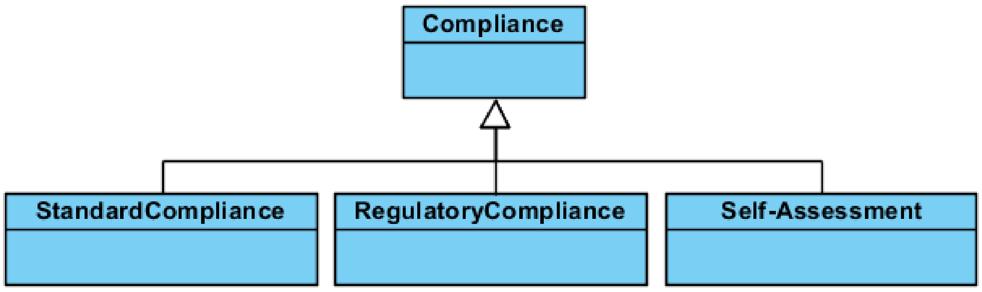

Figure 52: Standard Compliance, Regulatory Compliance, and Self-Assessment

Figure 53: Compliance in ArchiMEO Ontology

5.2.3.3 Cloud Service Handles Data

A Cloud Service can process, store and archive data as depicted in Figure 54. Personal Data (e.g. customer) which are processed by Cloud Services can be related to particular regulations at the country of the origin of data. Such regulations impose that particular types of data should not be stored outside the country.

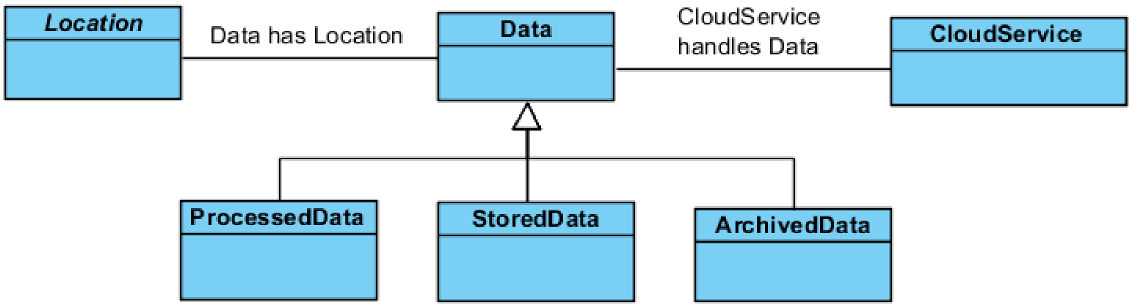

Figure 54: Cloud Service Handles Data

The ontology maintains the processed, stored and archived data attributes within the data object class of the passive structure element of the ArchiMEO ontology (see Figure 55).

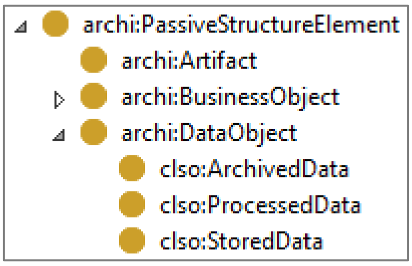

Figure 55: Data in the ArchiMEO Ontology

In order to annotate the data inputs and outputs of activities, the ontology makes use of the BPMN:Data class (see Figure 56). The user can annotate which format or kind the data is to be expected, e.g. Text; Number; Date; Picture; File; Geo Data; User Data; Weblink; Configuration Format.

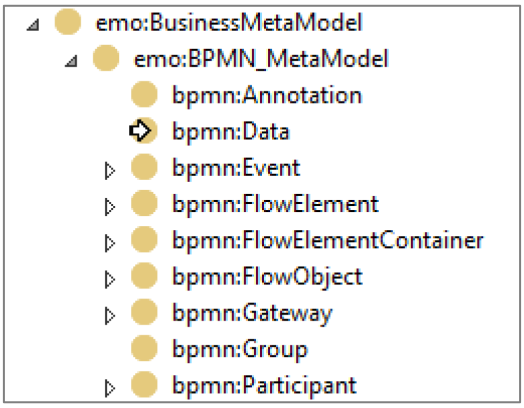

Figure 56: BPMN Data in ArchiMEO Ontology

5.2.3.4 Cloud Service Fulfils Regulatory Compliance Regulations

Since the Cloud Service handles the data, it is necessary to ensure that the handling of data is compliant with the law of the country in which the Cloud Consumer is located. According to Swiss law, for instance, personal data is exposed to risk if it is processed in a country which doesn't comply with Swiss laws. The laws in most European countries are compliant with Swiss law. However, only a few countries outside Europe can offer the same degree of protection. If a SME from Europe doesn't ensure that personal data is stored appropriately it might violate the privacy of its stakeholders and hence break the laws (Eurocloud 2012).

Other regulations describe how long the data has to be archived. In Austria, for example, specific accounting information has to be kept for seven years. Moreover in case of a judicial procedure the enterprise has to keep required data as long as necessary. In specific cases regulations impose that data has to be stored for up to 22 years (e.g. information about land and buildings, § 6 Abs. 1 Z 9 lit. a UStG) (Eurocloud 2012).

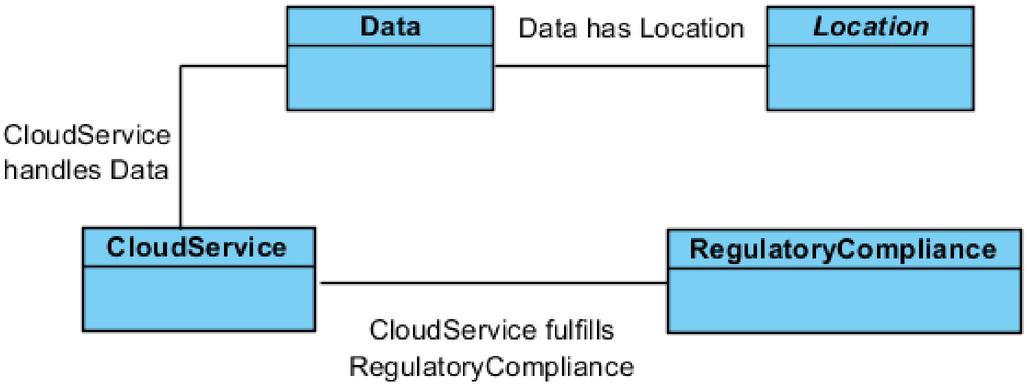

Figure 57: Cloud Service Fulfils Regulatory Compliance

5.2.3.5 Cloud Provider Fulfils Standard Compliance Regulations

Besides regulatory compliance there are different standard compliances which provide best practice recommendations on information security management. The International Electrotechnical Commission (IEC) and the International Organization for Standardization (ISO), for instance, developed a set of information security standards, namely the 27000 series. Latter is applicable for different size of enterprises and therefore also for Cloud Provider. The series ISO/IEC 27001 defines the requirements which have to be considered with respect to the management of information security within the context of the overall risk of a business organization (Meixner & Buettner 2012). Apart from the ISO standard, there is a plethora of other standards. For instance, the Payment Card Industry Data Security Standard (PCI DSS) is required by organizations that handle credit card data from the major credit card suppliers, such as Visa, MasterCard or American Express etc. (PCI 2015).

Figure 58: Cloud Provider Fulfils Standard Compliance

5.2.3.6 Cloud Provider Conducts Self-Assessment

The self-assessment is conducted by the Cloud Provider itself on a voluntary basis. The required frameworks for the self-assessment are provided by cloud computing initiatives/organizations, such as the Cloud Controls Matrix (CCM) form the Cloud Security Alliance (2015). The CCM framework enables Cloud Provider to conduct a self-assessment and to demonstrate its compliance with security standards by submitting a publicly available report. The goal of the CMA is to strengthen security information controls, to provide Cloud Provider with fundamental security principles and hence to support Cloud Consumer in assessing their security risks. The self-assessment provides Cloud Providers with the possibility to promote transparency and to provide visibility into its security practices.

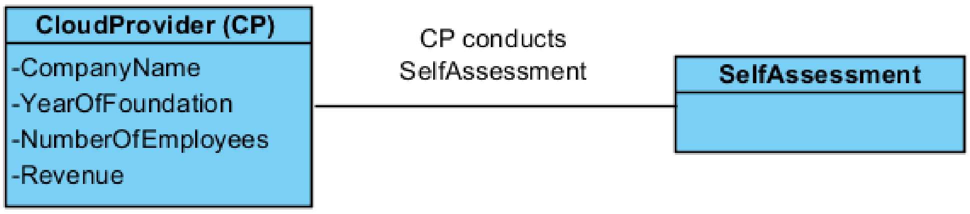

Figure 59: Cloud Provider Conducts Self-Assessment

5.2.3.7 Cloud Provider Has Certification

In contrast to the self-assessment in the previous chapter a cloud certification is conducted and awarded by an authorized certification body. A cloud specific certification takes the complexity of cloud environments into consideration by assessing several dimensions (security, data security, compliance etc.). Cloud specific certifications emerged in the last years and are still in development, for example the 'EuroCloud Star Audit' certification (eurocloud-staraudit.eu). Nevertheless we start from the assumption that they will develop further and consider possible cloud specific certifications as relevant evaluation criteria.

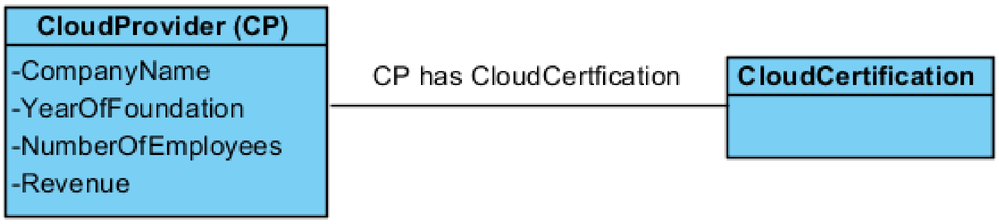

Figure 60: Cloud Provider Has Certification

5.2.3.8 Roles and Access Rights

An enterprise has different stakeholders, such as employees, suppliers or customers. The concept of the role allows defining how many users require access to the Cloud Service. Stakeholder are - amongst others - employees of an enterprise who access the Cloud Service in order to perform daily tasks. Apart from the employees there are further stakeholders that have to be considered, such as suppliers and customers. Due to technological advancement and globalization enterprises produce products and deliver services by collaborating in a network consisting of different stakeholders. Aforementioned development is referred to as 'sharing economy' or 'networked economy'. Hence the meta model depicted in Figure 61is based on the assumption that in future enterprises are going to exploit the potential of Cloud Services for collaboration and value creation with its stakeholders.

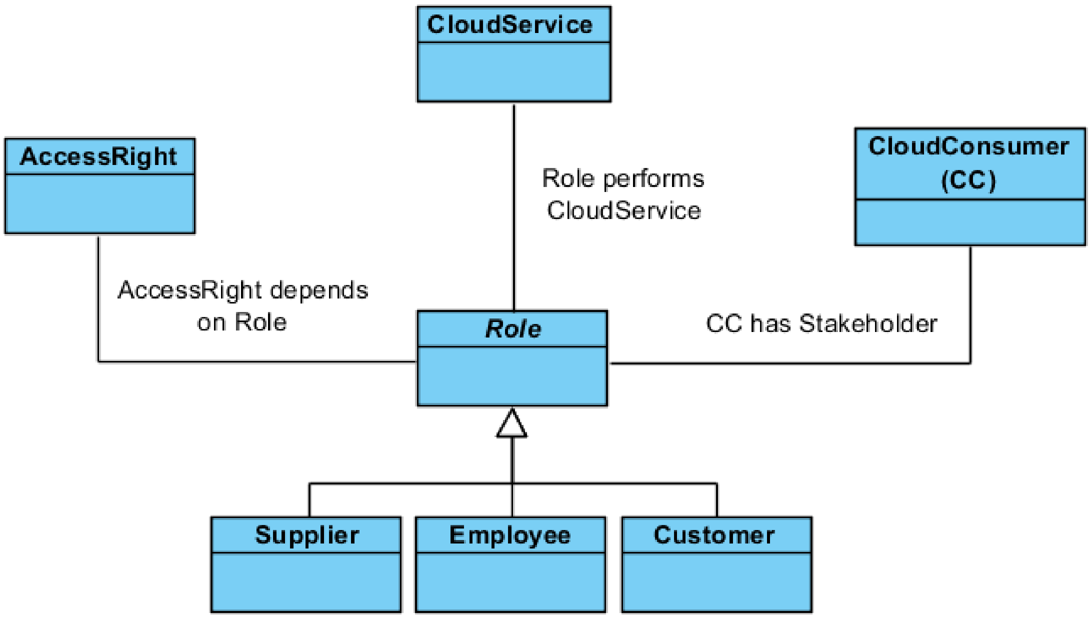

Figure 61: Role Performs Cloud Service

Each stakeholder has a role, which explicates whether this stakeholder will have access to the Cloud Service and in which way. In particular, a role can map to specific access rights with respect to the accessing of a Cloud Service. For instance, the Cloud Provider Salesforce (2014) offers customer relationship management solutions. Salesforce allows determining what information an employee, who uses the Cloud Service, can see. A sales manager, for example, is in charge of the overall revenue for a specific country. This person might need access to the customer data of all cities of the corresponding county. This allows him to monitor the overall sales performance. However, a sales representative is normally responsible for a defined territory. Therefore this person might have only access to the data of the territory he is in charge of.

The different Roles are represented as subclasses of the already existing concept Business Partner and Employee (see Figure 62).

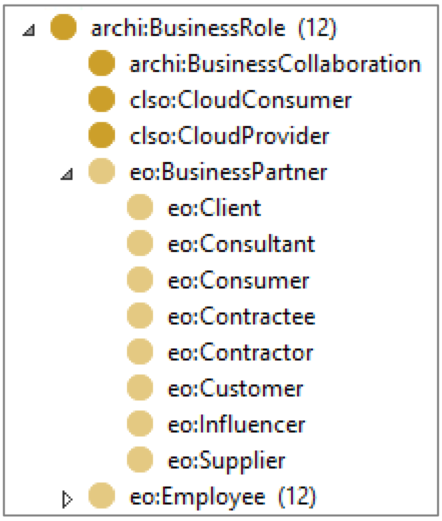

Figure 62: Role Concept in ArchiMEO Ontology

5.2.4 Workflow Perspective

The workflow perspective covers attributes that are required for selecting a workflow for a business process and setting up appropriate services. The requirements need to be gathered on alevel appropriate for a Cloud Broker.The meta model tries to close the gap between Business and IT by requesting information that a business user can provide and transform them into more technical ones.

As already indicated in in the business perspective chapter, a service handles data. The respective data can be classified using several types. Figure 63 shows the graphical relation.

Figure 63: Service Handles Data

In order to categorize the date, we distinguish among the following types that can be selected: Text, Number, Date, Picture, File, Geo Data, User Data, Web link, Configuration Format, and Others. The corresponding ontology has been already introduces in Figure 56.

Services may also have relations for the start and end time of the processing. Taking the Christmas greetings card process as an example, the end date has to be before Christmas. This can be specified by defining an end date and time. Figure 64 shows the corresponding meta model.

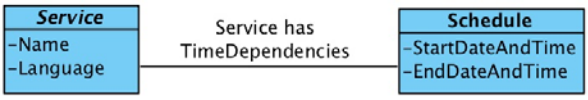

Figure 64: Service Has Time Dependencies

For the ontology, we make use of the top ontology that contains the time class. This time class maintains the relevant subclasses for specifying the start and end dates and times (see Figure 65). Concepts like hour, minute time zone, calendar day and month make it possible to define dedicated points in time.

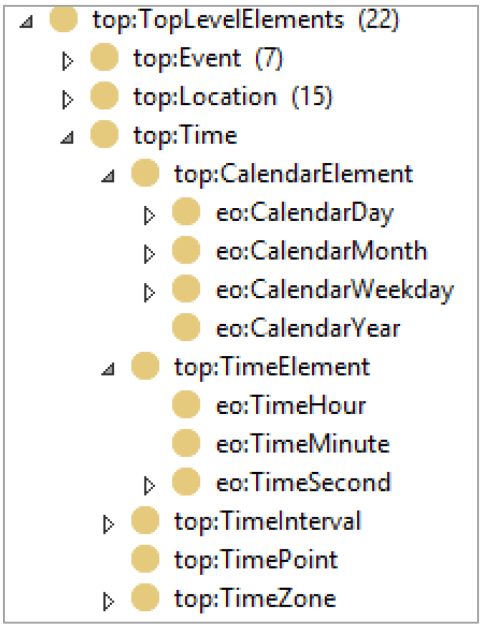

Figure 65: Ontology – Scheduling Elements

Based on the importance of the business process or the data that is processed, the Cloud Service may require performance and security objectives such as availability, reliability, security etc. In reference to the Cloud Service Level Agreement Standardisation Guidelines objectives can be attached and specified. Figure 66 shows the meta model for selected performance and security service objectives and relates the Cloud Service to Workflow and Business Process.

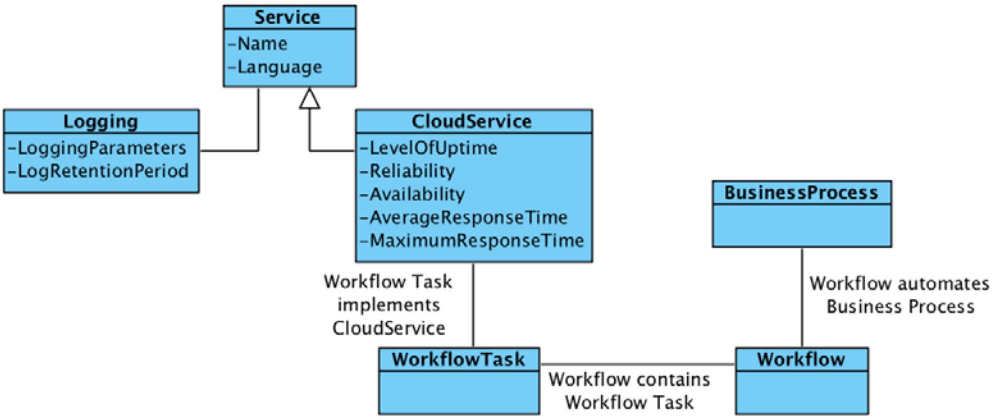

Figure 66: Cloud Service

In the ontology, the Workflow is represented as a subclass of an ArchiMate Behaviour Element and related to a Business Process via the relation "Workflow automates Business Processes". This is consistent with the definition of a Workflow as given by the Workflow Management Coalition (WfMC 1999). The ontology has been further extended by the service level agreement as a subclass of the contract class of the existing ArchiMate ontology (see Figure 67). This example shows also the embedded logging and monitoring that has been specified in the meta model above.

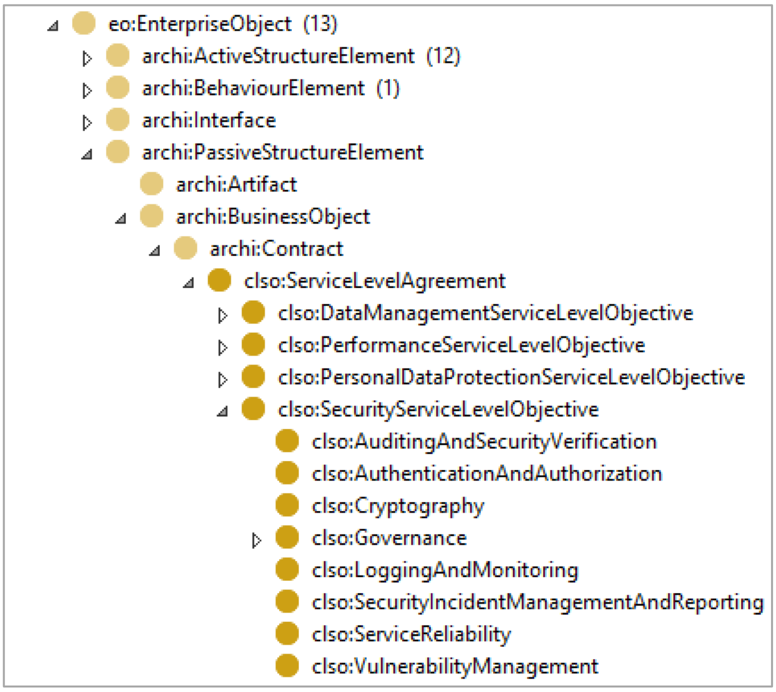

Figure 67: Ontology - Service Level Agreement – Logging and Monitoring

In terms of performance elements such as availability, capability Indicators, capacity, response time, etc, these have been places as a subclass of the service level agreement. Same applies to the security service level objectives such as cryptography, governance, service reliability, vulnerability management and many more. The graphical representation is shown in Figure 68.

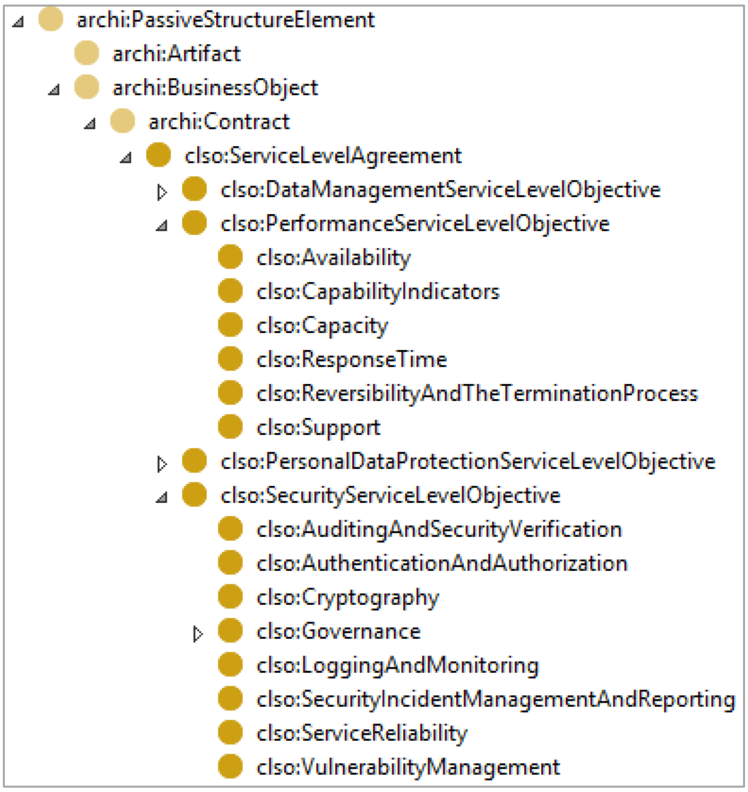

Figure 68: SLA and SLO in ArchiMEO Ontology

5.3 Rules for Business and IT in the Cloud Alignment

The semantic lifting of business process and workflow models services the purpose to support the alignment of business and IT in the cloud. The content of the models are combined with the class definitions of the BPaaS Ontology. The inference component for Smart Business and IT in the Cloud Alignment contains queries and rules to answer the competency questions. These can be complex decisions like:

- Are there existing workflows for my business process?
- Which parts of my business process can be solved by existing workflows?
- Is this workflow realising my business process?

or more simple questions like

- Does the pricing model of the service allow payments per month?
- Does the provider offer consulting services?

The alignment is based on inference rules to propose workflows, services, cloud provider etc. that satisfy the requirements specified in the service description model, which itself refers to business process models and workflow models.

The inference engine applies the inferencing rules based on the SPARQL Inferencing Notation (SPIN), a W3C specification submission (Knublauch et al. 2011). Basically its collection of RDF vocabularies enables the use of the SPARQL Query Language for RDF (W3C 2013) to define inference rules on Semantic Web models.

This is a simple query which collects all Cloud providers which offer consulting services:

```
SELECT DISTINCT ?cloudprovider ?service
WHERE {
    ?cloudprovider clso:CPoffersService ?service.
    ?service rdfs:type clso:Consultingservice .
}
```

SPIN allows to link class definitions with SPARQL queries to capture constraints and rules and formalize the expected behaviour of those classes. This is used in the BPaaS Design Environment to specify mapping rules between elements. For example, it can be used to derive requirements for the location of data depending on the type of data.

IF data contains personal data
THEN location of data is EU

The prototype of the BPaaS Design Environment (due in June 2016) will contain concrete rules relevant for the christmas card process and business scenarios (see section 2.2)

The specification of the queries and rules requires competence in ontology engineering. This cannot be expected from the Cloud Broker. However, by using the Decision Model Notation and Decision Tables a more user friendly can be created. This is explained in section 5.4.

5.4 Decision Models and Rules

Decision Model is a model type of the BPaaS modelling method. It is realized using the Decision Model and Notation standard developed by the OMG (OMG 2015b). A DMN model consists of two parts:

- the Decision Requirements Diagram
- The Decision Logic

The Decision Model Class Diagram of Figure 30 shows the model elements for the decision requirements diagram. The elements can be mapped to classes in the ontology. The input data is mapped to the data object, the knowledge source is a special kind of business object. According to (Ross & Lam 2011, p 152f) decision is a special kind of task and thus modelled as a subclass of Activity in the BPaaS Ontology.

The decision logic is associated to the busienss knowledge element. It can be represented as decision tables or scripts.The role of decisions in the BPaaS Design Environment is two-fold.

- First, business decisions can be associated to (business rules) tasks
- Second, the decisions can be used to guide the alignment of business and IT in the cloud.

5.4.1 Decision model for Business Decisions and Workflow Scripting

The typical use of Decision Models is to represent the logic for business decisions (see Figure 69). In this case, the decision logic can represent business decision criteria to guide the process flow or to make business decisions, e.g.

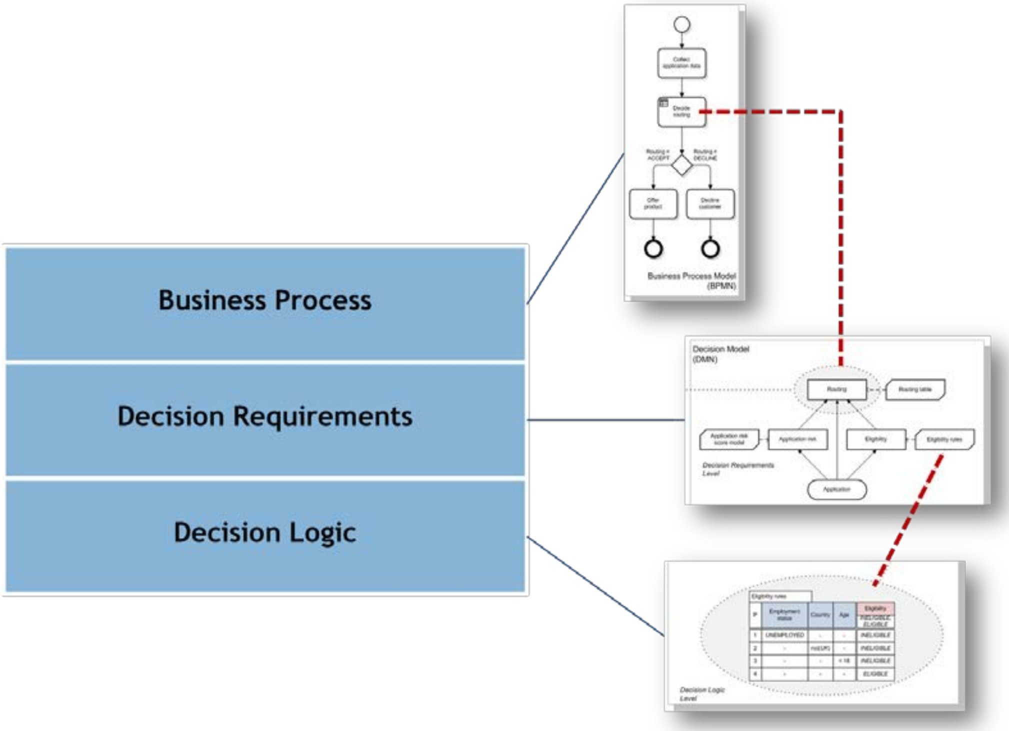

Figure 69 Main concepts of the Decision Model and their relation to Business Processes (Coenen 2013)

The decision logic can be translated into executable scripts. DMN defines the friendly enough expression language (FEEL) for the purpose of giving standard executable semantics to many kinds of expressions in decision model (OMG 2015, p 79ff). The combination of FEEL and the semantically enhanced workflow model allow to create workflow scripts which can be executed in the cloud, if appropriate cloud services are available.

5.4.2 Decision Models for Business IT-Cloud Alignment and Service Discovery

Decision models can be used by the Cloud Broker to define decision criteria for service discovery. In this way the Cloud Broker can adapt ontology-based service (see section 3.3.1).

In this case, the decision model is translated into rules which operate on the models themselves. The conditions of the rules refer to elements of the business process model and workflow model. They are useful, if service selection depends on combination of several criteria. For example, the decision model can be used to express that the location of data depends on the type of data and the type of customer:

IF data contains personal data AND customer is of class gold customer
THEN location of data is EU

This rule uses the classes data, location and customer, which are defined in the ArchiMEO and BPaaS ontology. Personal data and gold customer refer to instances of concrete process models and models.

This integration of Modeling Method and BPaaS Ontology has not yet been implemented. It is planned for the next cycle of the prototype development.

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

6.1 Human Interpreted Annotations

Semantic annotations allow for the human modeller to add semantics to the modelling element while creating the graphical models (Figure 70).

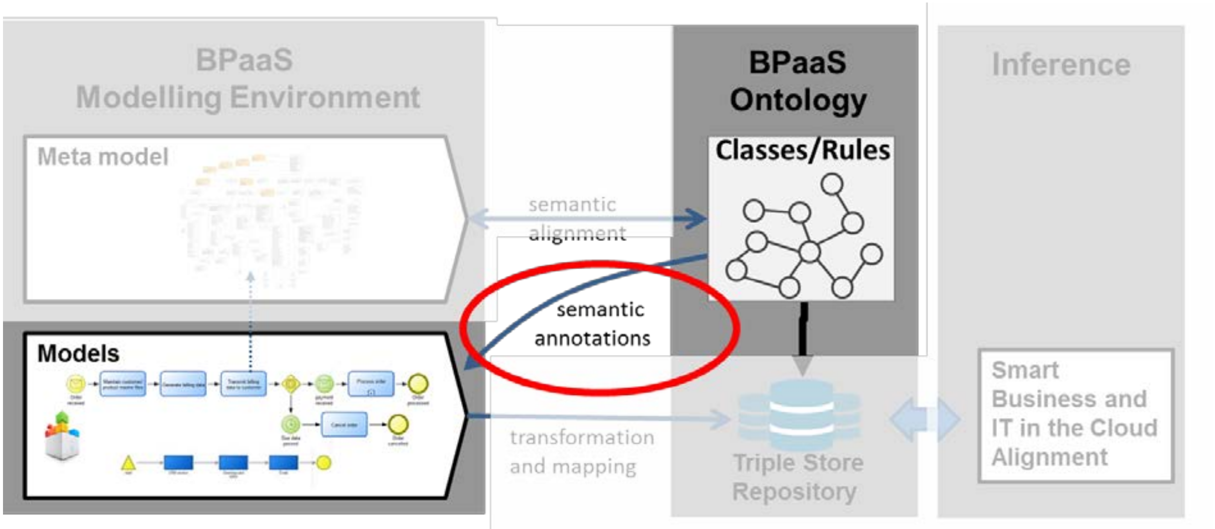

Figure 70 Semantic Lifting of Models by Semantic Annotations

There are seven different ways of implementing semantic lifting for weaving between the different modelling layers. In the following the different types for human performed semantic lifting are explained.

6.1.1 Non-Supported direct linkage: Using a Text Attribute for Manual Entries

In this scenario the semantic concept id is manually copied in an existing text attribute. Hence, there is a direct linkage between the source object and the targeted semantic concept. As it has been manually copied, it is understood as a "non-supported" linkage. It can be applied, in case the used meta models cannot be changed, hence an existing text attribute is used. In case small adaptations of the meta model are possible, it is recommended to at least create a text attribute that indicates the semantic lifting.

One example of this textual attribute that has the link information of the ontology concept is the description attribute of the class 'Activity Specification' in the 'Service description model' model type. The attribute attachment of these attributes is shown below for the "Free Input Data Keywords" attribute and is equivalent for all non-supported linkage attributes.

$$\mathrm{domain}(Free\ Input\ Data\ Keywords) = (Activity\ Description)$$
$$range(Free\ Input\ Data\ Keywords) = (String)$$
$$card(Free\ Input\ Data\ Keywords) = \langle 1,1 \rangle$$

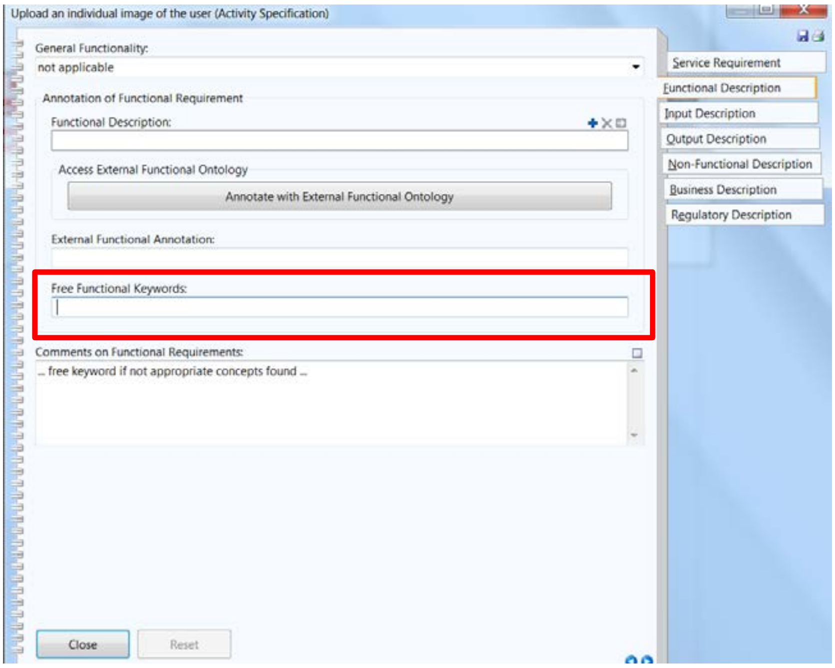

Figure 71 "Non-Supported" Linkage Example

6.1.2 Supported pre-defined direct linkage: Using a static Flat List

In this scenario, parts of the ontology are implemented as part of the meta model. Selection of concepts can be realized in form of a pre-defined flat Enumeration list that can only be changed via an update of the meta model. Such enumeration lists can be realized in form of a drop-down list, check box buttons or radio buttons depending on the appropriateness.

The selected concepts can either be interpreted as keywords, names of concepts or as mapping information for concept ids.

$$\forall Enumeration\ Lists\ \boldsymbol{enum_j}: \boldsymbol{enum_j} \in D_i^T$$

The attribute assignment of the attribute “General Functionality” is as follows and is equivalent for all $\boldsymbol{enum_j} \in D_{SDM}^T$.

$$\mathrm{domain}(General\ Functionaltity) = (Activity\ Description)$$
$$range(General\ Functionaltity) = (\boldsymbol{enum}_{GEN})$$
$$card(General\ Functionaltity) = \langle 1{,}1 \rangle$$

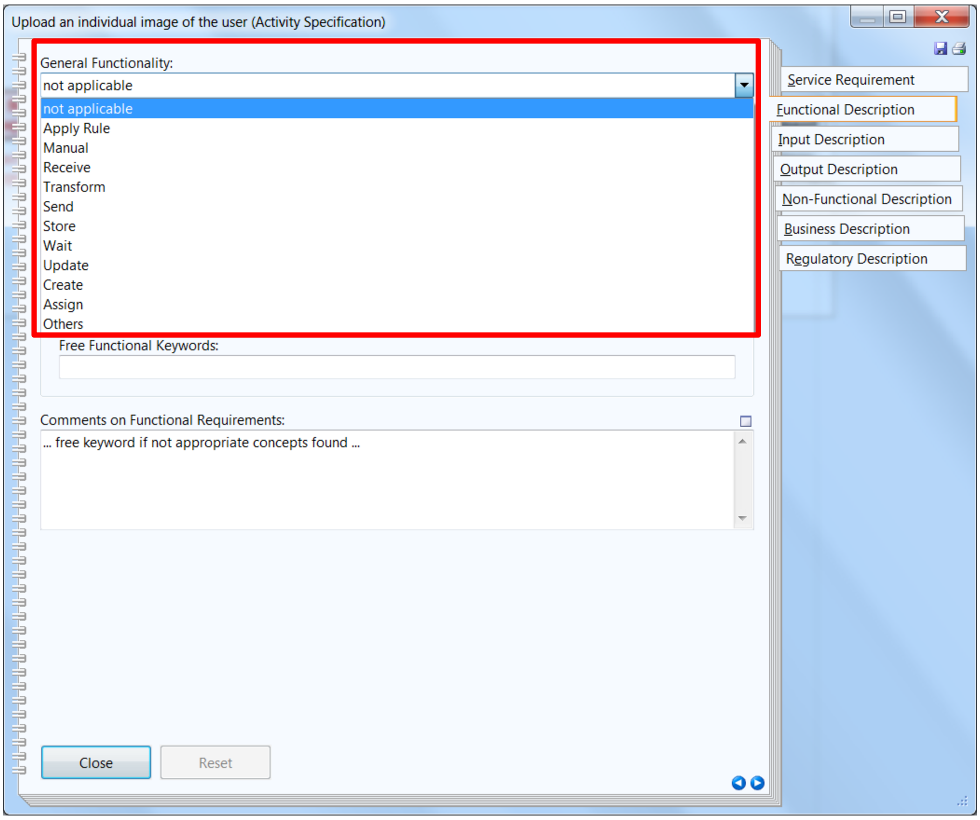

Figure 72: Enumeration List Example

6.1.3 Supported Direct Linkage: Using a Semantic Tunnel for a flexible Enumeration List

In this scenario there is a tool support to select semantic concepts from the target model form objects of the source model. The source and target model are likely in two different tools like a modelling tool for the source and an ontology system for the target. The linkage is established via a pre-defined configuration between those two tools, which is named a "semantic tunnel". This tunnel can be two Web-Services that are configured in such a way, that the target tool can be queried to list all relevant semantic concepts out of the source tool.

Depending on the software platform there are different ways to implement those tunnels. In case of using ADOxx, context specific attribute that are necessary to correctly use the "semantic tunnel" are configuration settings in hidden attributes, whereas the Web-Service invocation is performed with scripts as a functional enrichment of the user interface.

The formalism of the ADOxx case in FDMM language is shown by the example "Access External Functional Ontology" and is equivalent for all attributes $a_i \in D_{SDM}$.

$$\text{domain}(\text{Access External Functional Ontology}) = (\mathit{Activity\ Description})$$
$$\text{range}(\text{Access External Functional Ontology}) = (\mathit{String})$$
$$\text{card}(\text{Access External Functional Ontology}) = \langle 1,1 \rangle$$

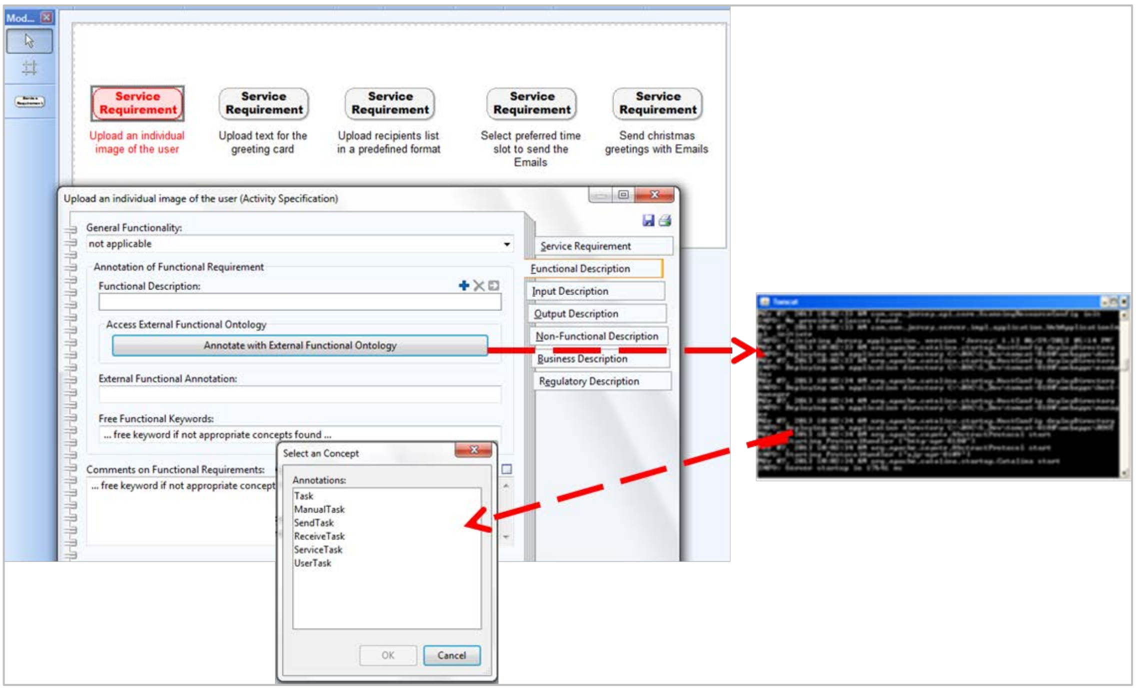

Figure 73 Semantic Tunnel for a flexible Enumeration List Example

6.1.4 Indirect Linkage: Using Semantic Transit Models

In this scenario the relevant concepts of the semantic target model are copied in a so-called "Semantic Transit Model" into the source model environment, in order to simply the handling of semantic concept selection within the source model environment. In case of using ADOxx, a new model type called semantic will be added that enables the representation of semantic concepts out of an ontology system. Once the concepts had been copied into the source model environment, typical weaving mechanisms – like Inter Model References (INTERREF) - that are supported by the modelling environment can be used.

Additionally to the aforementioned changes of the meta model, mechanisms need to be in place to handle the redundancy – like notification mechanisms - with the ontology system.

Hence, the Semantic Transit Model Type is defined as:

$MT_{ST} := \{O_{ST}, D_{ST}, A_{ST}\}$, where

$$O_{ST} := \{Concept\}$$

$$D_{ST} := \{String\}$$

$$A_{ST} := \{\,Name, URI, Referenced\ concept\text{-}from, referenced\ concept\text{-}to\}$$

In order to enable the semantic lifting with a reference from any object in any model type, we use the super class of all objects:

$$\forall\ class\ x\ \in \{\boldsymbol{O}_{BPD}^{T}, \boldsymbol{O}_{CD}^{T}, \boldsymbol{O}_{CM}^{T}, \boldsymbol{O}_{DDM}^{T}, \boldsymbol{O}_{WE}^{T}, \boldsymbol{O}_{DRD}^{T}, \boldsymbol{O}_{STM}^{T}\}$$
$$\exists\ class\ \boldsymbol{super}: \text{super} \geq x$$

and define that link in form of:

$$\text{domain}(Referenced\ concept\text{-}from)=\{\ Referenced\ concept\}$$

$$\text{range}(Referenced\ concept\text{-}from)=\{\text{super}\}$$

$$\text{card}(Referenced\ concept\text{-}from = \langle m, \text{n}\rangle) \quad \text{for m,n} \in N$$

This semantic lifting via a MTSTM provides a tool support via the references – implemented as so-called INTEREFS in ADOxx – but does not need a full established Ontology Management System (OMS) interaction. Hence, it is expected that the interaction with the OMS will be introduced on a later stage of the project, in order to support also the semantic lifting in tight interaction with an OMS.

Aforementioned results can be downloaded on ADOxx.org [30].

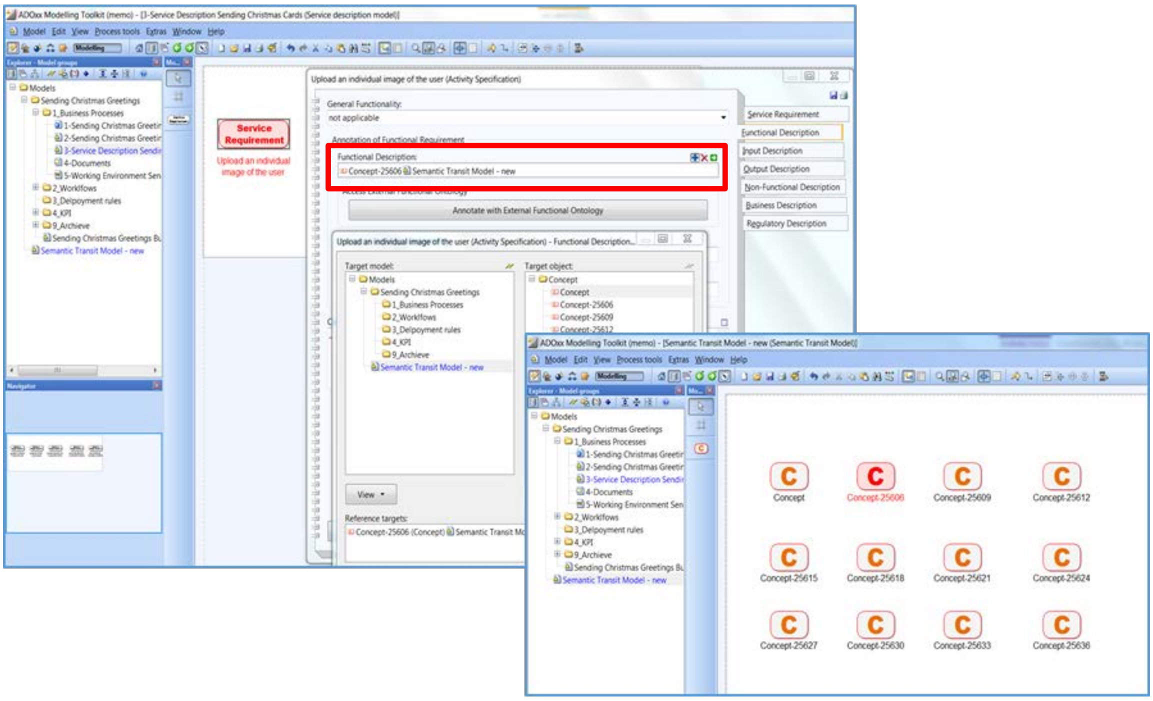

Figure 74: Using Semantic Transit Models

6.1.5 Direct and Indirect Linkage: Combination of Semantic Tunnel and Transit Model

This scenario combines the direct linkage – using the semantic tunnel – as well as the indirect linkage – using the Semantic Transit. The reason for this combination is to reduce the effort in redundancy management, as high level and preferable stable concepts are copied into the Semantic Transit Model, whereas the flexible direct linkage is provided for lower and probably more agile concepts. Combining the selection of the Semantic Transit and supporting the flexible use of the Semantic Tunnel enables to combine both advantages of the two approaches.

Additionally to the aforementioned changes of the meta model, mechanisms need to be in place to handle the redundancy – like notification mechanisms - with the ontology system.

Hence, the Semantic Transit Model Type is defined as:

$$\boldsymbol{MT}_{ST} \coloneqq \{\boldsymbol{O}_{ST}, \boldsymbol{D}_{ST}, \boldsymbol{A}_{ST}\}, \text{ where}$$

$$\boldsymbol{O}_{ST} \coloneqq \{Concept\}, \text{ and}$$

$$\boldsymbol{D}_{ST} \coloneqq \{String\}$$

$$\boldsymbol{A}_{ST} \coloneqq \{\ Name, URI, Access\ External\ Ontology, Referenced\ concept\text{-}from, referenced\ concept\text{-}to\}$$

In order to enable the semantic lifting with a reference from any object in any model type, we use the super class of all objects:

$$\forall\ class\ x\ \in \{\boldsymbol{O}_{BPD}^{T}, \boldsymbol{O}_{CD}^{T}, \boldsymbol{O}_{CM}^{T}, \boldsymbol{O}_{DDM}^{T}, \boldsymbol{O}_{WE}^{T}, \boldsymbol{O}_{DRD}^{T}, \boldsymbol{O}_{STM}^{T}\}$$
$$\exists\ class\ \boldsymbol{super}: \text{super} \geq x$$

and define that link in form of:

domain($Referenced\ concept\text{-}from$)={ $Referenced\ concept$}

range($Referenced\ concept\text{-}from$)= {super}

card($Referenced\ concept\text{-}from = \langle m, \text{n}\rangle$) for m,n$\in N$

The Tunnel is then triggered in the referenced "Concept" –instance within the semantic transit model through the "Access External Ontology" attribute. The attribute attachment according to the FDMM is defined as follows:

$$\text{domain(Access External Ontology)} = (Activity\ Description)$$
$$\text{range(Access External Ontology)} = (String)$$
$$\text{card(Access External Ontology)} = \langle 1{,}1 \rangle$$

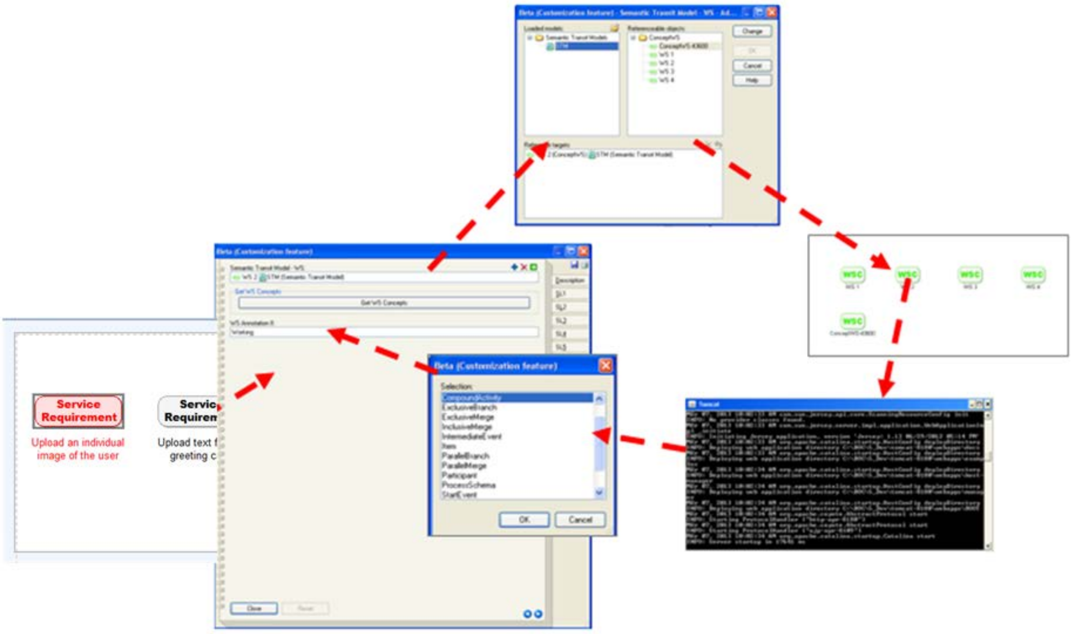

Figure 75: Combination of Semantic Tunnel and Transit Model Example

6.1.6 Loose Coupling: Using Intermediate Reference Ontologies

In this scenario an intermediate ontological layer is introduced that enable the loose – hence intermediate – linkage of concepts in contract to the aforementioned direct linkage. Any aforementioned techniques can be applied, as loose coupling does not introduce a new technical way of introducing semantics but introduces an intermediate ontology acting as reference ontology that both meta models link their concepts towards it.

The following example uses the "Semantic Tunnel for a flexible Enumeration List" scenario to invoke an intermediate ontologies list into the ADOxx environment. The formalism of this case in FDMM language is shown by the example "Access External Input Data Ontology" and is equivalent for all attributes $a_i \in D_{SDM}$, that are triggering a script within ADOxx.

$$\mathrm{domain}(\mathrm{Access\ External\ Input\ Data\ Ontology}) = (Activity\ Description)$$
$$\mathrm{range}(\mathrm{Access\ External\ Input\ Data\ Ontology}) = (String)$$
$$\mathrm{card}(\mathrm{Access\ External\ Input\ Data\ Ontology}) = \langle 1,1 \rangle$$

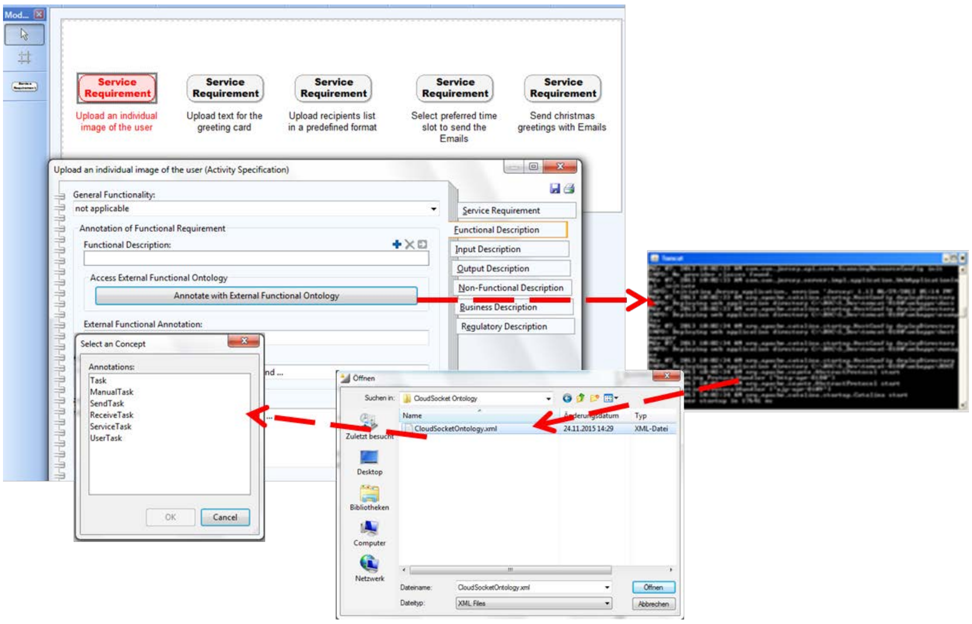

Figure 76: Using Intermediate Reference Ontologies with Semantic Tunnel

6.1.7 Graphical Annotation: The semantic Whiteboard

This scenario uses the graphical position of objects for its annotation. It is hence the realization of a semantic whiteboard, where the background image is the model which is to be annotated. Semantic tags – similar to post-its – are placed above the model objects and hence annotate it. This semantic whiteboard scenario is implemented with an positioning algorithm, that annotates the content of the "Note" instances to an attribute of the "Activity description" instance if they are have overlapping areas.

For this purpose the class "Note" including the attribute "Text" is added into the "Service Description Model" model type in ADOxx as follows:

$$\boldsymbol{MT}_{SDM} = \langle \boldsymbol{O}_{SDM}^{T}, \boldsymbol{D}_{SDM}^{T}, \boldsymbol{A}_{SDM} \rangle \text{ where,}$$

$$\boldsymbol{O}_{SDM}^{T} = \{Activity\ Description, Note\},$$

$$\boldsymbol{D}_{SDM}^{T} = \{String, Longstring, \boldsymbol{Enum}_{GF}, \boldsymbol{Enum}_{GID}, \boldsymbol{Enum}_{GOD}, \boldsymbol{Enum}_{GPS}, \boldsymbol{Enum}_{VC}, \boldsymbol{Enum}_{PAY}, \boldsymbol{Enum}_{DLO}, \boldsymbol{Enum}_{DSC}\}$$

$$\boldsymbol{A}_{SDM} = \{A_{SDM}, Text\}$$

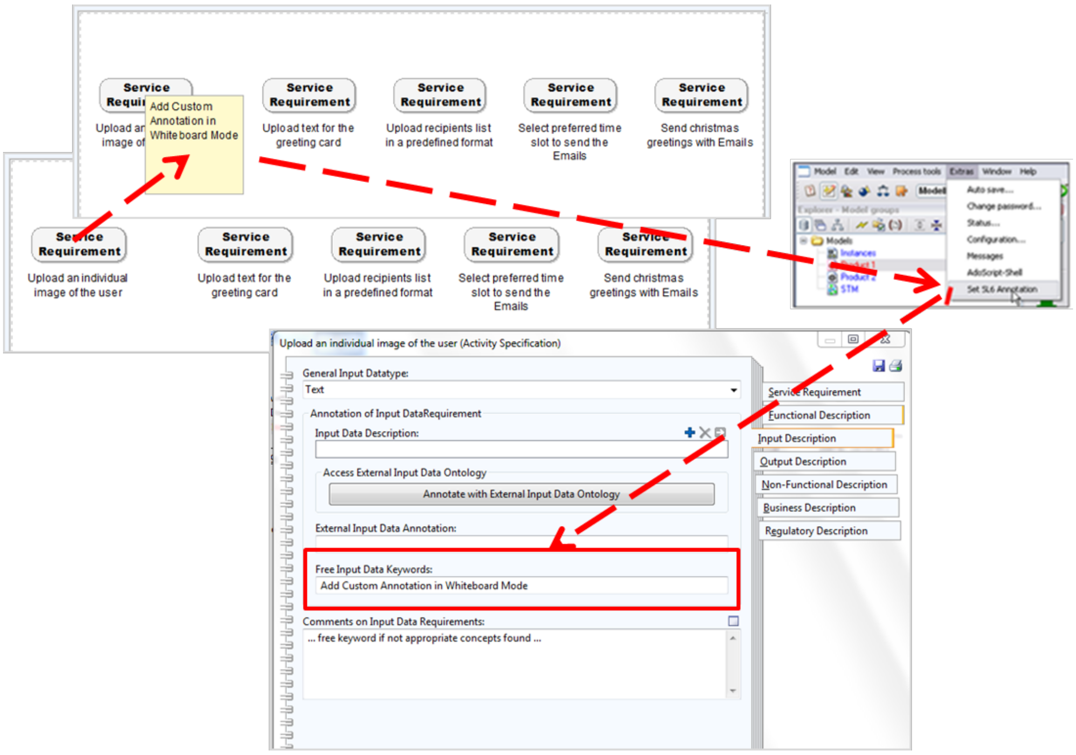

Figure 77: Semantic Whiteboard Example

6.2 Model Transformation and Mapping Rules

In order to enable smart business and IT-Cloud alignment, a transformation is implemented, which creates a formal representation of the graphical and script-based models. The basic mechanism, which is based on the transformation approach of the LearnPAd project (Emmenegger et al. 2016) is shown in Figure 61figure .

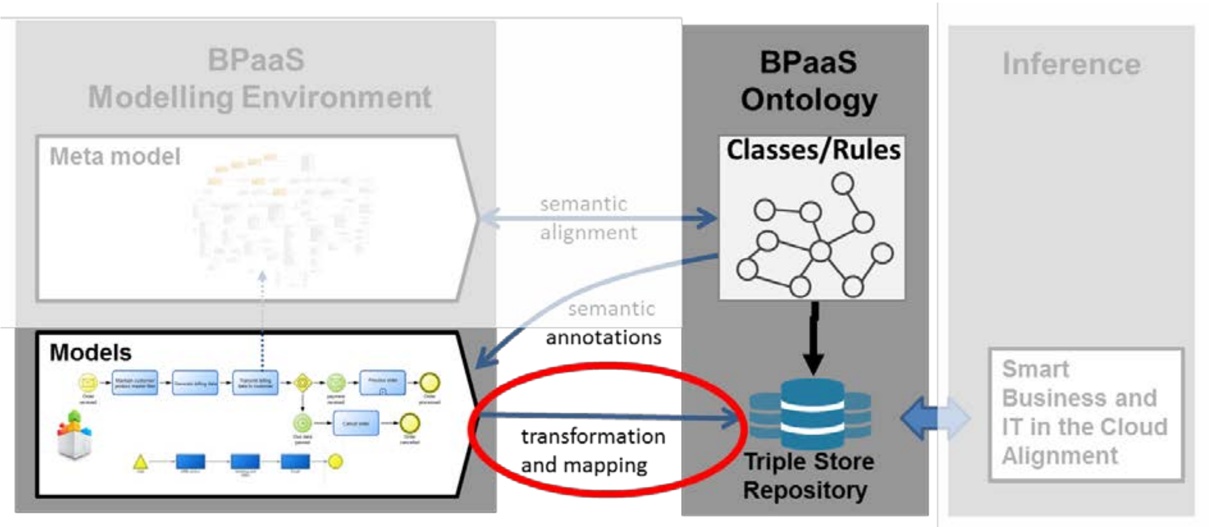

Figure 59: Transformation Mechanism

ADOxx.org exports the annotated models into an Extensible Markup Language (XML) formatted description. The EXtensible Stylesheet Language (XSLT) engine transforms the XML and created instances of classes 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.

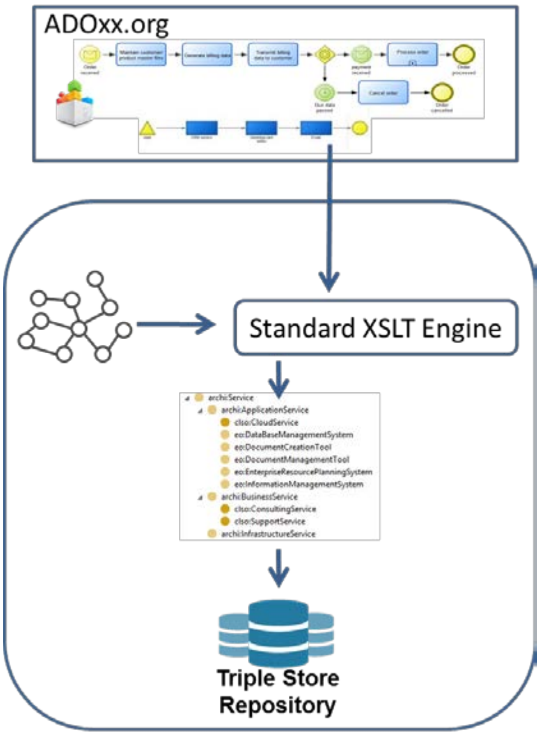

Figure 78: Transformation into Ontology

6.3 Semantic Alignment of Meta Models

It is also possible to semantically enrich the graphical models on the meta-model layer. The meta model definition is exported into an ontology representation in Turtle format. The turtle format can be added to the BPaaS Ontology. In the opposite direction, class definitions of the BPaaS Ontology can be translated into XML, imported by the BPaaS modelling environment and extended syntax definitions.

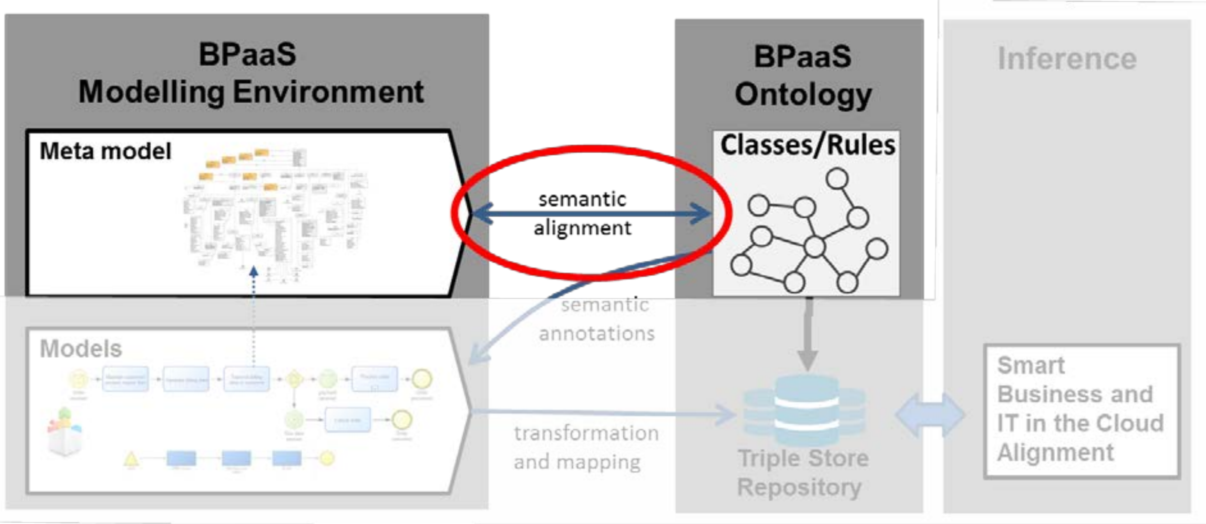

Figure 79: Semantic alignment on the meta-model layer

This type of semantic lifting is a step towards realization of the semantic metamodeling as described in our recent article about enterprise modelling (Hinkelmann et al. 2015). It helps in the semantic alignment of meta models and BPaaS ontology. This allows to ensure consistency between meta models and BPaaS Ontology.

A first version of this semantic alignment has been implemented. The meta models are exported in XML format and converted using standard XSLT Engine into turtle format. The turtle format can be edited using a script editor or extended with class definitions of the BPaaS Ontology. Then the adapted meta model can be fed back to the BPaaS modelling environment.

7 PROTOTYPES AND RESULTS

Prototypes are interpreted as proof of concepts and hence provided and explained in order to trigger awareness with the aim that those prototypes will be integrated – in any appropriate ways – into productive software.

Hence the concept of innovation items is used, where each prototype is an autonomous innovation item that consists of relevant explanation, training, documentation and tool access in order to distribute the underlying idea and enable the evolution of appropriate parts of those ideas towards production.

In the following the two innovation items (a) the BPaaS Modelling Environment including the semantic lifting as well as (b) the BPaaS Ontology including discovery, analysis and composition are introduced. Current existing solutions as well as intended development are explained to indicate the targeted prototype that is developed in the next phase of the research.

7.1 Semantically Enriched BPaaS Design Environment

The semantic enriched BPaaS Design Environment consists of two parts: (a) the BPaaS Design Environment, as well as (b) the semantic enrichment.

Figure 80 depicts the BPaaS Design Environment in form of a public available and for research and academic purpose openly available proof of concept on the ADOxx.org platform[1]. The upper half of the screen shows the sample business process "Sending Christmas Cards", whereas the model in the lower half of the screen represents the service requirement description. The model tree on the left side indicates that there are also other models realised, but in order to focus on the business process and workflow alignment, these two models have been selected for providing an impression.

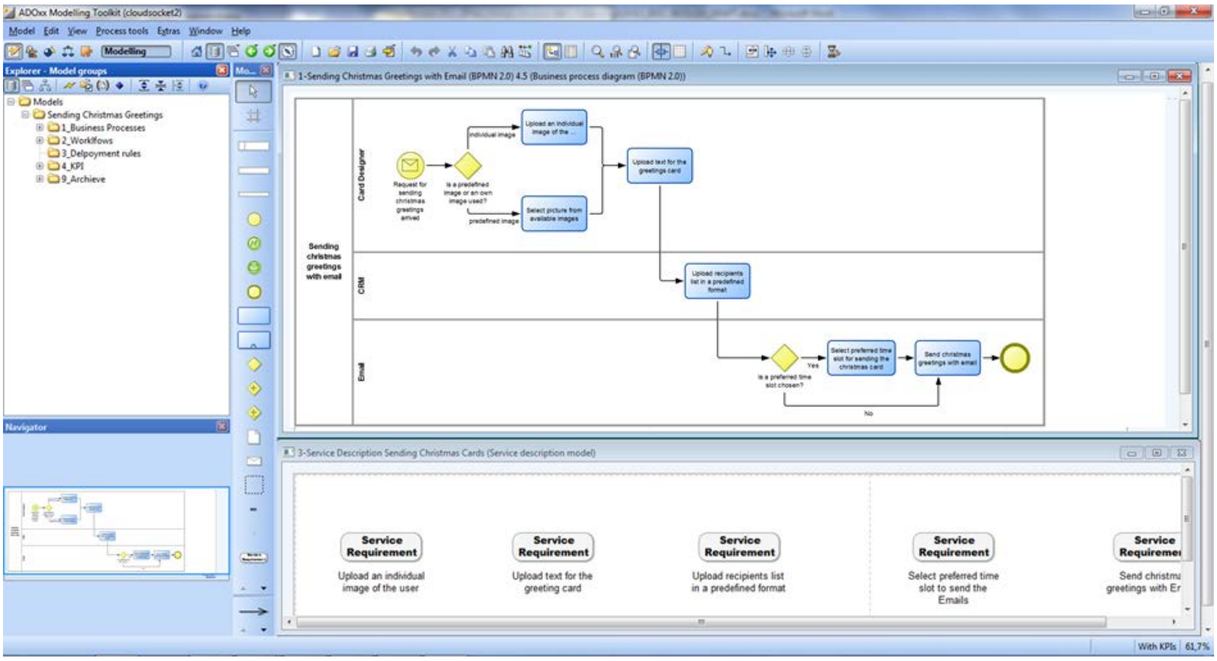

Figure 80: BPaaS Design Environment – Research Prototype

[1] https://www.adoxx.org/live/web/cloudsocket-developer-space/space

The second part of the prototype consists of the semantic enrichment, which is realised via the aforementioned semantic lifting. Currently the semantic lifting is realised on attribute level for the service requirement description. This means that each task of a business process can reference to a service requirement description, which consists of several attributes where each of those attributes can be semantically lifted. Complex attributes are lifted in multiple steps, whereas simple attributes are lifted in a single step.

7.1.1 Semantic Enriched Service Requirement

The **Service Requirement** is a generic description and consists of:

- Name, [String]: indicating an unique id of the requirement
- Description, [String] {Multi-Line}: enabling generic and easy readable text, in case this is not already done in the task description of the business process.
- Responsible Author, [INTERREF] {Person, Role, Organisation}: enabling to indicate the person, the role or the organisation that raises this requirement, which should be contacted, in case anything is unclear.

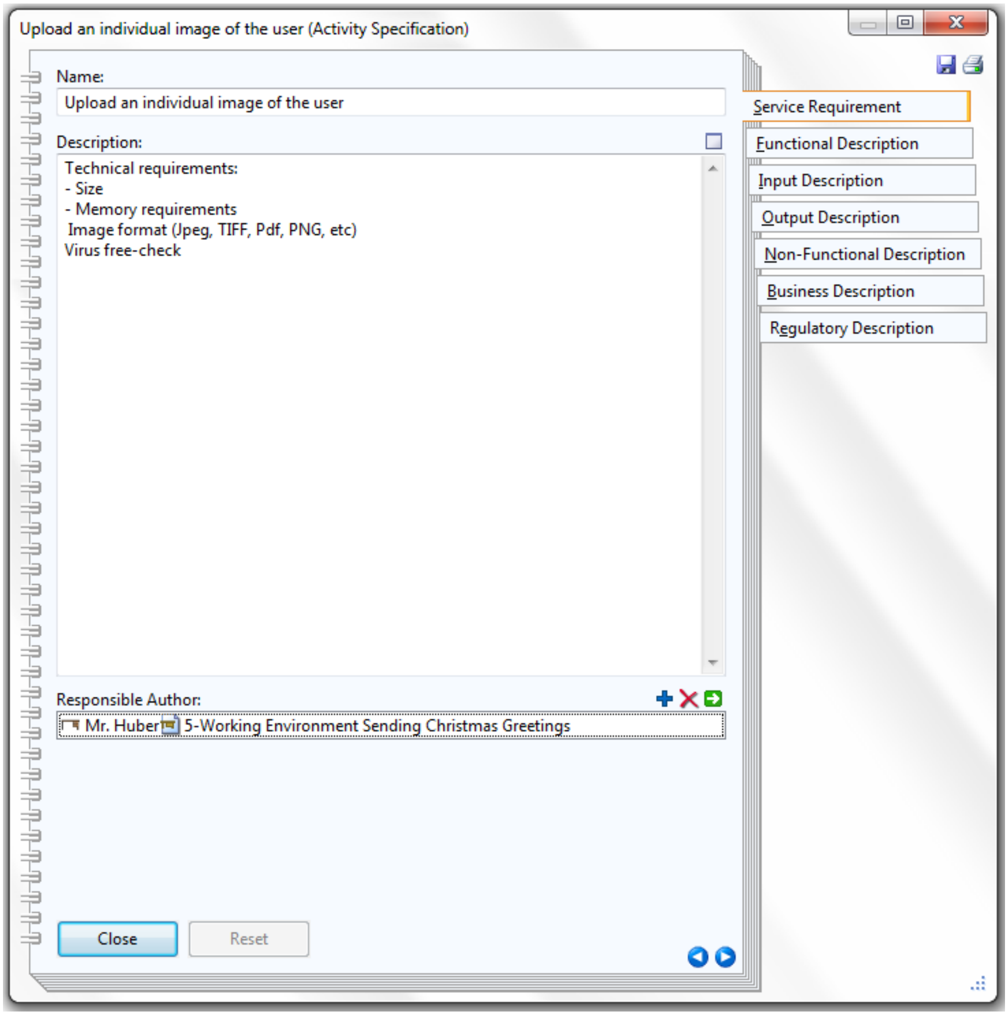

Figure 81: Service Requirement Description

Figure 81 shows the three attributes, without semantic enrichment.

7.1.2 Semantic Enriched Functional Description

The Functional Description indicates what the service is supposed to do and realises a semantic lifting in three steps:

- General Functionality, [Enumeration], { not applicable; Apply Rule; Manual; Receive; Transform; Send; Store; Wait; Update; Create; Assign; Others}: defines the expected service behaviour on very high level. This semantic lifting is a pre-defined flat list in form of a pull down menu.
- Annotation with External Ontology, [INTERREF, PROGRAMMCALL, STRING]: enables the selection of an ontology that describes the functional description in the appropriate details. The button invokes an ontology access, whereas the selected concepts are stored in the External Functional Annotation. This semantic lifting is using a Semantic Transit as well as a Semantic Tunnel to access an ontology.
- Free Functional Keywords, [STRING]: In case aforementioned semantic expressions are insufficient or need further elaboration, the author can describe its concerns here.
- Comments on Functional Requirements [STRING]: This feedback enables an evolution based on user feedback.

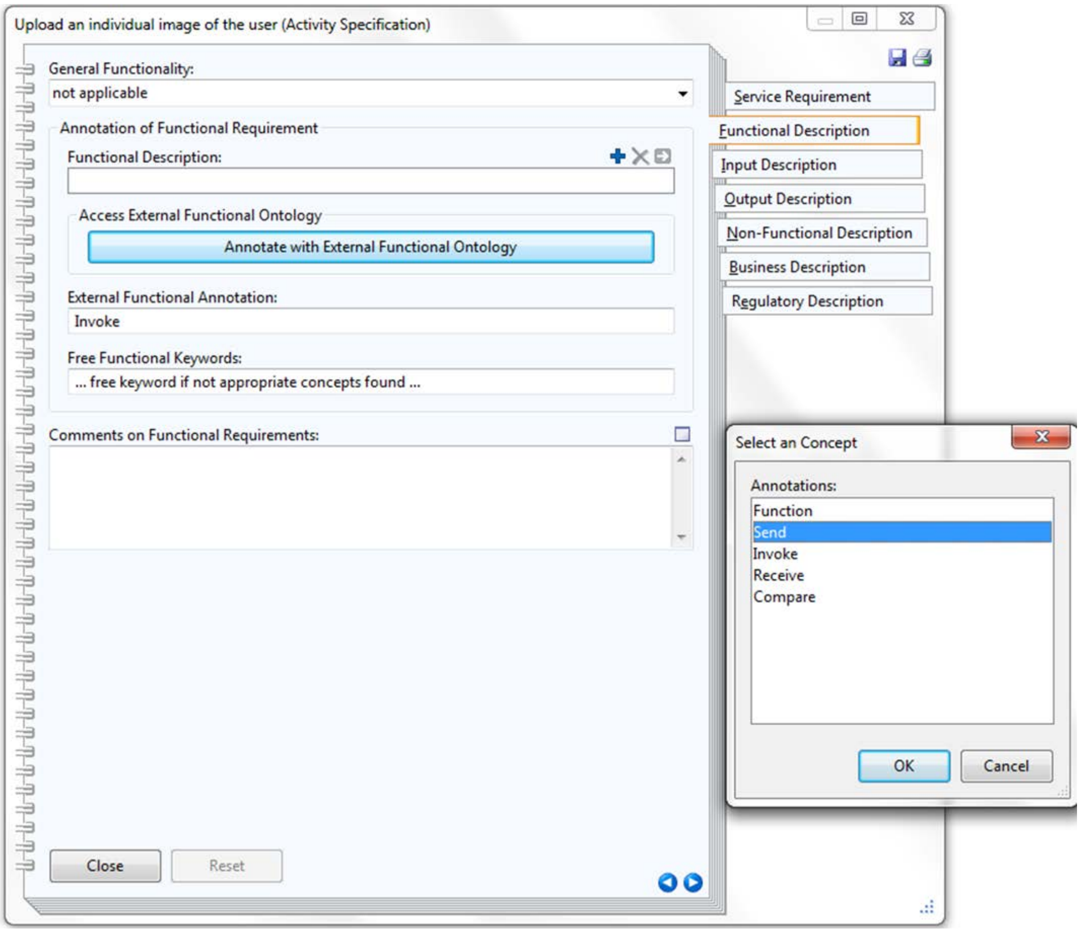

Figure 82: Functional Description

Figure 82 shows the implemented semantic lifting features for functional service descriptions and demonstrates the access of an external ontology – hence flexible evolution of concepts – in the pop-up window. The button "Annotate with External Functional Ontology" accesses an ontology with the concepts listed in the pop-up window. The selected concepts are stored.

7.1.3 Semantic Enriched Input and Output Description

Input and Output Description have the identical structure and similar to the functional description provide a three step semantic lifting.

- General Input/Output Datatype, [Enumeration] {Text; Number; Date; Picture; File; Geo Data; User Data; Weblink; Configuration Format; Others}: defines the expected data format on high level. A pre-defined flat list for semantic annotation is used.
- Annotation with External Ontology, [INTERREF, PROGRAMMCALL, STRING]: enables the selection of an ontology that describes the input / output format in the appropriate details. The button invokes an ontology access, whereas the selected concepts are stored in the External Functional Annotation. This semantic lifting is using a Semantic Transit as well as a Semantic Tunnel to access an ontology.
- Free Input/Output Data Keywords, [STRING]: In case aforementioned semantic expressions are insufficient or need further elaboration, the author can describe its concerns here.
- Comments on Input/Output Data Requirements, [STRING]: This feedback enables an evolution based on user feedback.

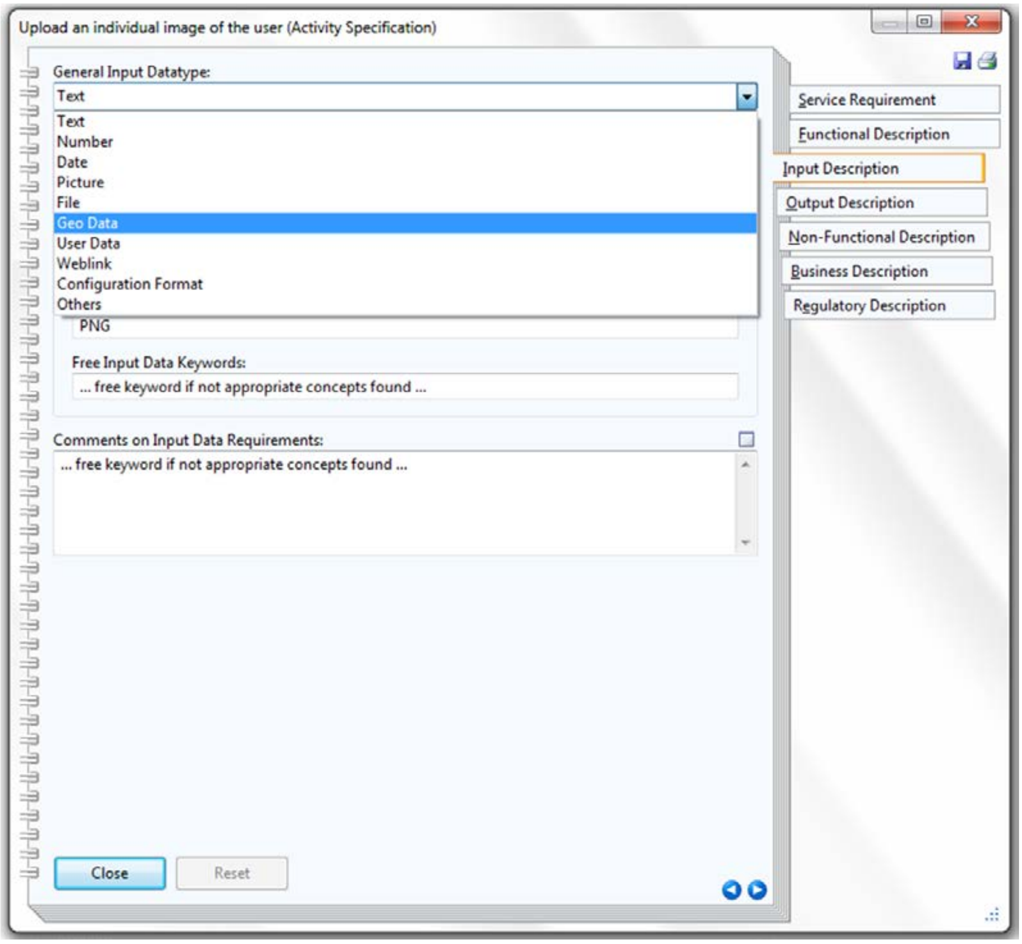

Figure 83: Input and Output Description

Figure 83 depicts the three step semantic lifting for input and output data format, and shows the drop down list that can be used for pre-defined flat list semantic lifting.

7.1.4 Non-Functional Requirements

The non-functional requirements are a collection of important attributes, with different complex semantic lifting approaches.

- Reliability Requirements
  - Targeted Error Rate, [DOUBLE]: A double value in percent without semantic lifting.
  - Thresholds, [DOUBLE]: Two double values in percent without semantic lifting.
  -
- Availability Requirements
  - Targeted Availability Time, [DOUBLE]: A double value in percent without semantic lifting.
  - Thresholds, [DOUBLE]: Two double values in percent without semantic lifting.

- Sample
  - Sample Service, [PROGRAMMCALL]: A reference to a service that describes the intended behaviour, without semantic lifting.
  - Sample Service Comments [STRING]: Additional comments on the sample service providing explanation why this sample had been chosen and what needs to be different in the requested services.

- Planning and Schedule is realised in a three step semantic lifting
  - General Planning Schedule, [ENUMERATION],{ on-demand, no-specific time slot, Calendar specific time slots, Season specific time slot, Day specific time slot, Other time slot}:
  - Annotation with external Ontology, [INTERREF, PROGRAMMCALL, STRING]: enables the selection of an ontology that describes the planning options in the appropriate details. The button invokes an ontology access, whereas the selected concepts are stored in the External Functional Annotation. This semantic lifting is using a Semantic Transit as well as a Semantic Tunnel to access an ontology.
  - Free Planning and Scheduling Keywords, [STRING]: In case aforementioned semantic expressions are insufficient or need further elaboration, the author can describe its concerns here.
  - Comments on Planning and Scheduling, [STRING]: This feedback enables an evolution based on user feedback.

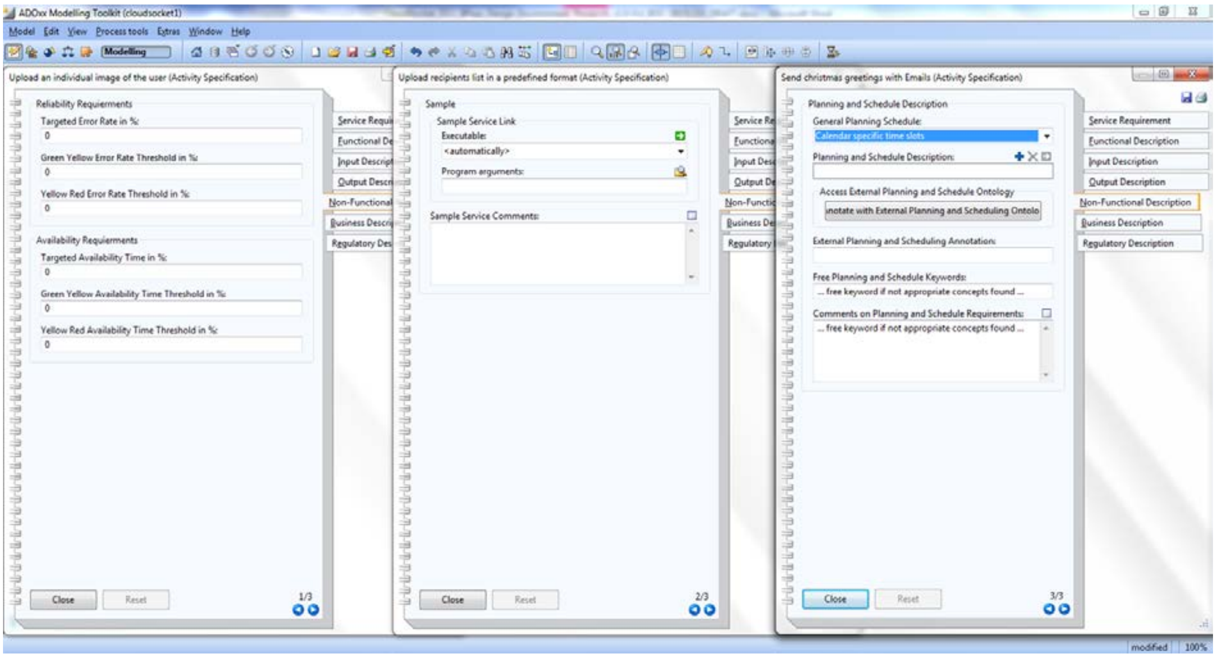

Figure 84: Non-Functional Description

Figure 84 depicts the three pages for describing non-functional requirements, by first starting on the left side with simple Reliability and Availability requirements, introducing a sample link in the middle of the figure as well as introducing the three step semantic lifting approach for planning and scheduling on the right side.

7.1.5 Business Description

The business description is separated in the following attributes, enabling to enforce business rules:

- Vendor Criteria [ENUMERATION] {Avoid Vendor Lock, Trust, Security, Helpdesk available, Onsite Visits possible, Maintenance available, Training offered}: Generic vendor selection criteria are proposed in a pre-defined flat list. The difference to previous drop-down lists, is that here multiple selection is enabled.

- Trust with single step semantic lifting
  - Trust Description, [INTERREF, PROGRAMMCALL, STRING]: enables the selection of an ontology that describes the trust options in the appropriate details. The button invokes an ontology access, whereas the selected concepts are stored in the External Functional Annotation. This semantic lifting is using a Semantic Transit as well as a Semantic Tunnel to access an ontology

- Security with single step semantic lifting
  - Security Description, [INTERREF, PROGRAMMCALL, STRING]: enables the selection of an ontology that describes the security options in the appropriate details. The button invokes an ontology access, whereas the selected concepts are stored in the External Functional Annotation. This semantic lifting is using a Semantic Transit as well as a Semantic Tunnel to access an ontology

- Payment [ENUMERATION], {Not-Decided-Yet, Pay-Per-Process, Pay-On-Demand, Pay-Monthly, Pay-Three-Monthly, Pay-Bi-Yearly, Pay-Yearly, Other }: enables the selection of high level payment models in a pre-defines flat list. A single step semantic lifting is used.

- Costs [STRING]: Comments about costs are possible without semantic lifting.

- Comments on Business related Requirements [STRING]: Comments on business related requirements are possible without semantic lifting.

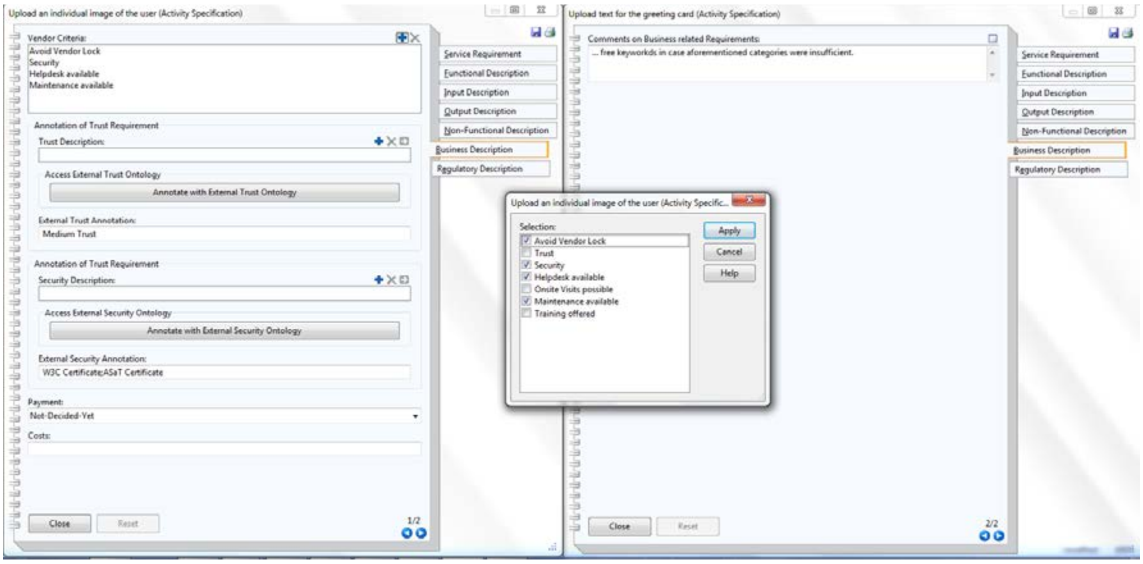

Figure 85 Business Description

Figure 85 depicts the business related description, showing the multiple selection list of vendor criteria. Those requirements are derived from the business philosophy of the broker, hence do not address technical issues.

7.1.6 Regulatory Description

Regulatory description is intended to describe domain specific regulations, which must be kept.

- Data Location
  - Data Location Overview, [ENUMERATION], {World-Wide, Europe, National, Regional, Local, Other}: A pre-defined flat list to indicate generic location requests
  - Data Location Description, [INTERREF]: The Semantic Transit model enables the selection of semantic concepts in more detail. As it is not expected that location requirements are evolving rapidly, this two-step semantic lifting approach is considered as appropriate.
  - Comments on Data Location [STRING]: Comments for evolving the alignment and semantic lifting configuration.
- Annotation of Certificate
  - Domain Specific Certification [ENUMERATION]: A pre-defined flat list of major certifications.
  - Keywords [STRING]: Additional free keywords in case special certification is requested.

- Data Protection
  - Domain Specific Data Protection [ENUMERATION]: A pre-defined flat list of major data protections issues
  - Keywords [STRING]: Additional free keywords in case special data protection is requested

- Maturity model
  - Domain Specific Maturity Model [ENUMERATION]: A pre-defined flat list of maturity models
  - Keywords [STRING]: Additional free keywords in case special maturing models are requested
  - Relevant Regulation List [STRING]: A free comment field to point to relevant laws and compliance issues.

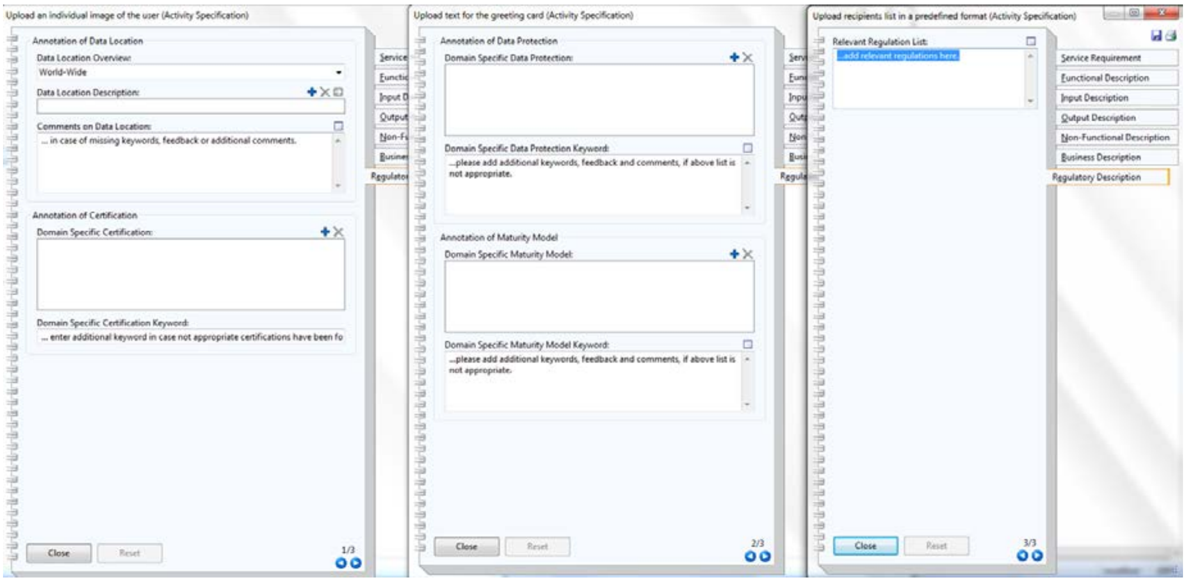

Figure 86: Regulatory Description

Figure 86 indicates the Regulatory Description, where some semantic lifting is applied.

7.2 Prototype Environment Setup

In order to setup the environment, the ADOxx platform can be downloaded from https://www.adoxx.org/live/download. Request a free installation code[2] and follow the instructions on ADOxx "New installation Guide"[3].

7.2.1 Import CloudSocket Library

Once the platform has been installed, two icons will appear on the desktop, as shown in Figure 87.

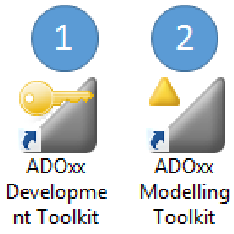

Figure 87: ADOxx Desktop Shortcuts

(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 88 - (1) and open the settings dialog (2).

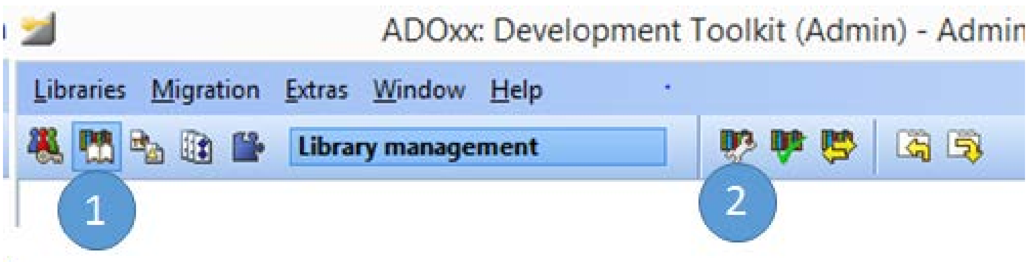

Figure 88: ADOxx Development Toolkit Library Management

The library management dialog will appear as shown in Figure 89. Click on management (1) and select import (2).

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

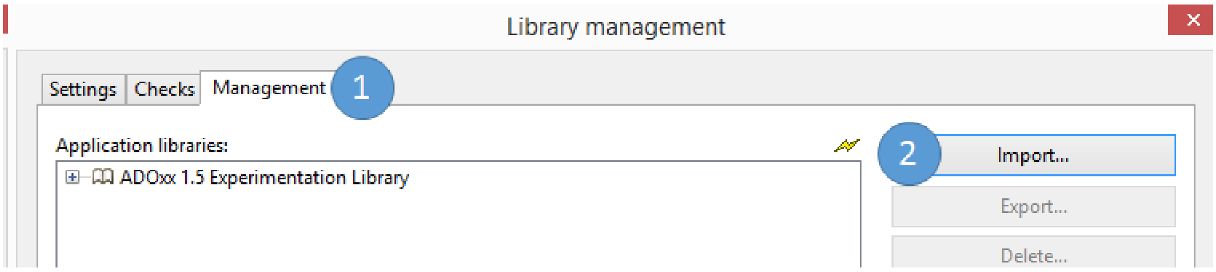

Figure 89: ADOxx Development Toolkit Import

Download the example library[4] and browse to the folder where the CloudSocket Prototype Library is stored. Select the ADL file and click import.

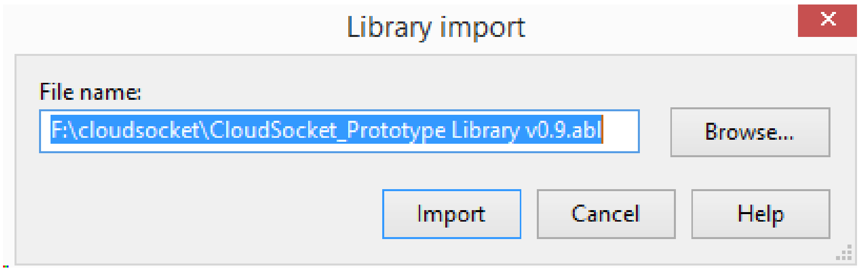

Figure 90: ADOxx Development Toolkit – Browse for Library File for Import

Once the import process finished click on yes for creating a default model group (Figure 91).

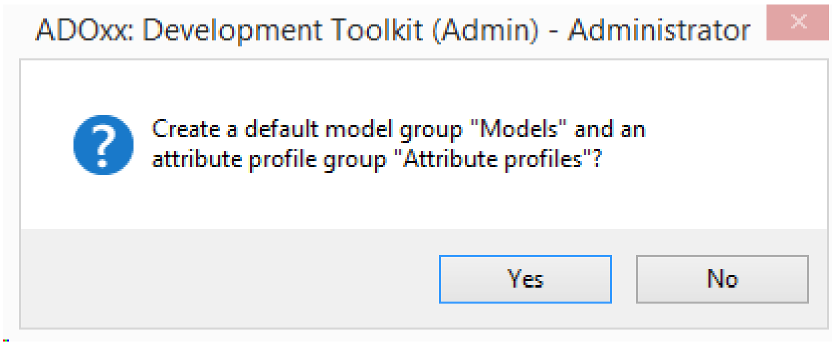

Figure 91: ADOxx Development Toolkit - Create Default Model Group

If the import has been successful, the report will look like Figure 92.

[4] https://www.cloudsocket.eu/resources/CloudSocket_Prototype_Library_v0.8.abl

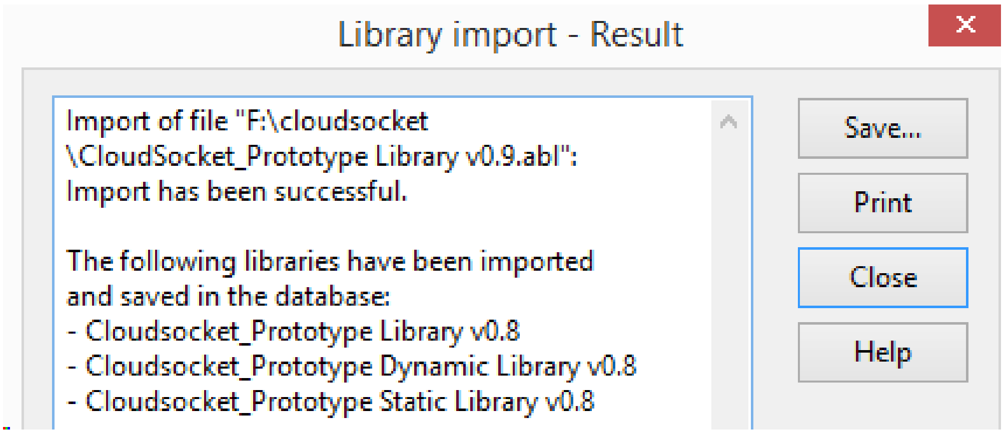

Figure 92: ADOxx Development Toolkit - Library Import Result

Create a new user and make the imported library. Click on User management (1) and select User List (Figure 93).

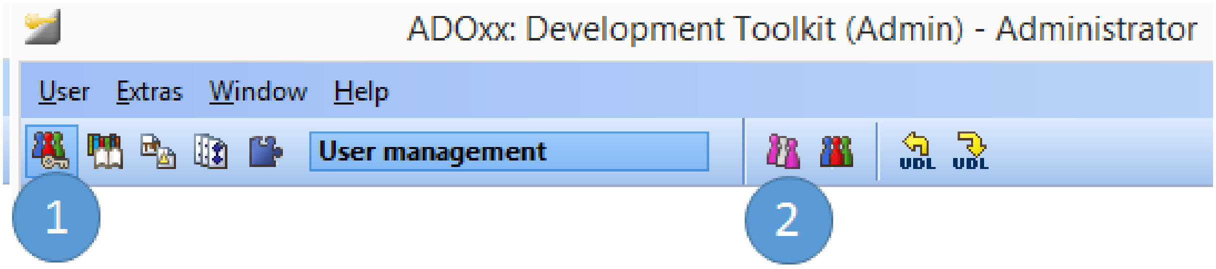

Figure 93: ADOxx Development Toolkit - User Management

The User management – User list dialog will appear as shown in Figure 94.

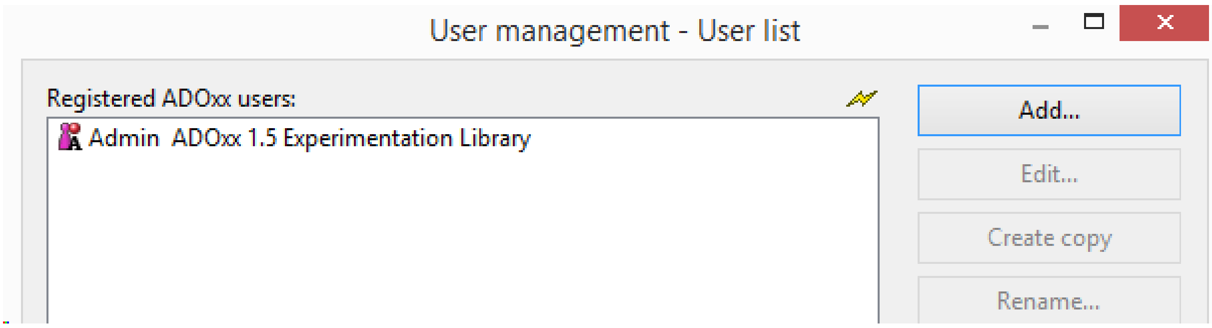

Figure 94: ADOxx Development Toolkit - User List

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

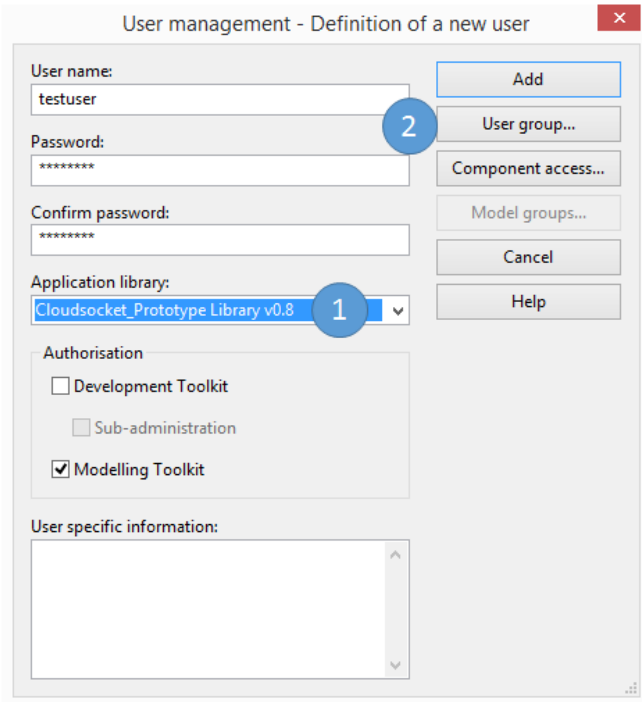

Figure 95: ADOxx Development Toolkit – Create a new User

Assign the user to a user group and click ok.

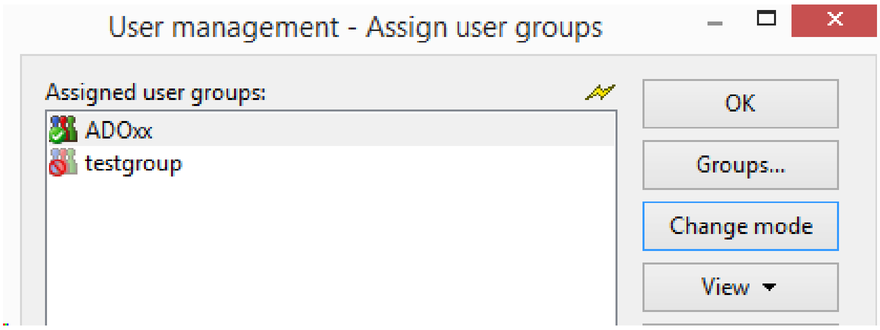

Figure 96: ADOxx Development Toolkit – Assign user groups

The user will be created.

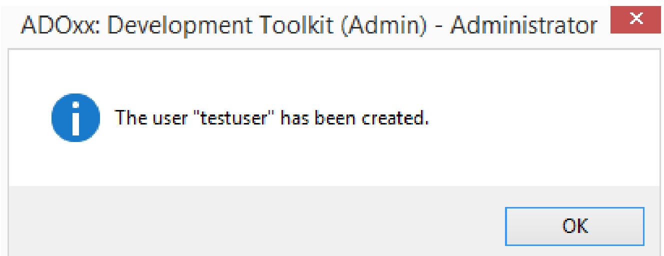

Figure 97: ADOxx Development Toolkit - User Management Dialog

7.2.2 Start the Modelling Environment

The next step will show how to make use of the library in the modelling toolkit. Right click on ADOxx Modelling Toolkit (2) and select run as administrator.

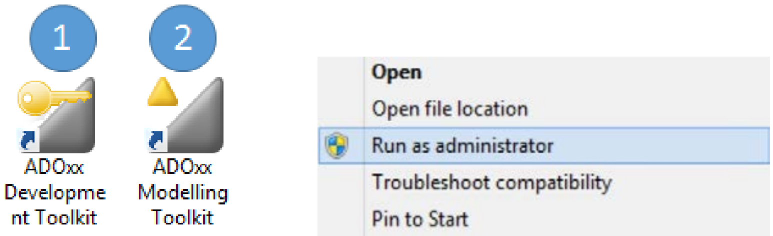

Figure 98: ADOxx Desktop Shortcuts – Run as administrator

Click on yes at the user account control dialog.

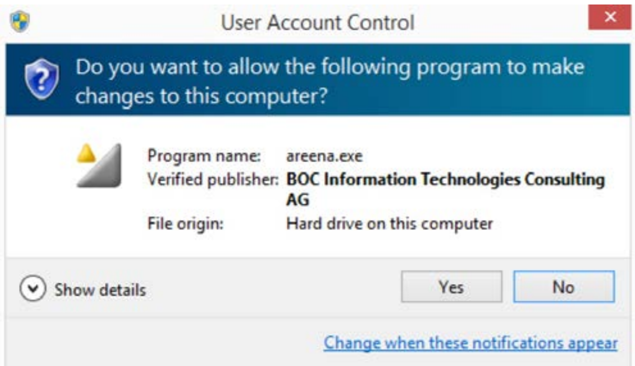

Figure 99: User Account Control Dialog

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).

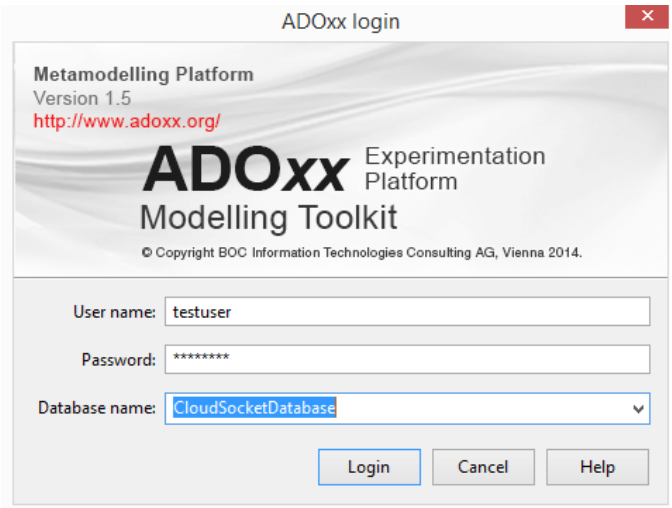

Figure 100: ADOxx Login

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

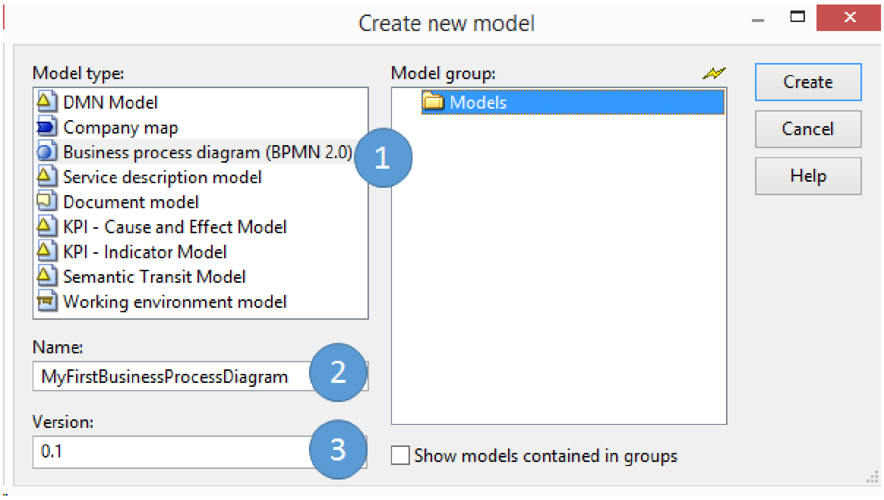

Figure 101: ADOxx Modelling Toolkit: Create Model

In order to make use of the annotation, select the Activity Specification element and drag it to the working space.

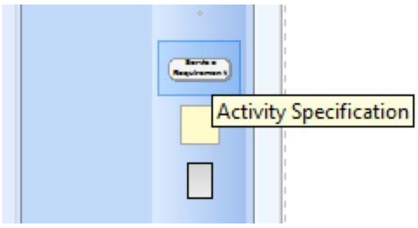

Figure 102: ADOxx Modelling Toolkit - Activity Specification Element

Double click the Activity Specification element to open the notebook to make use of the annotation.

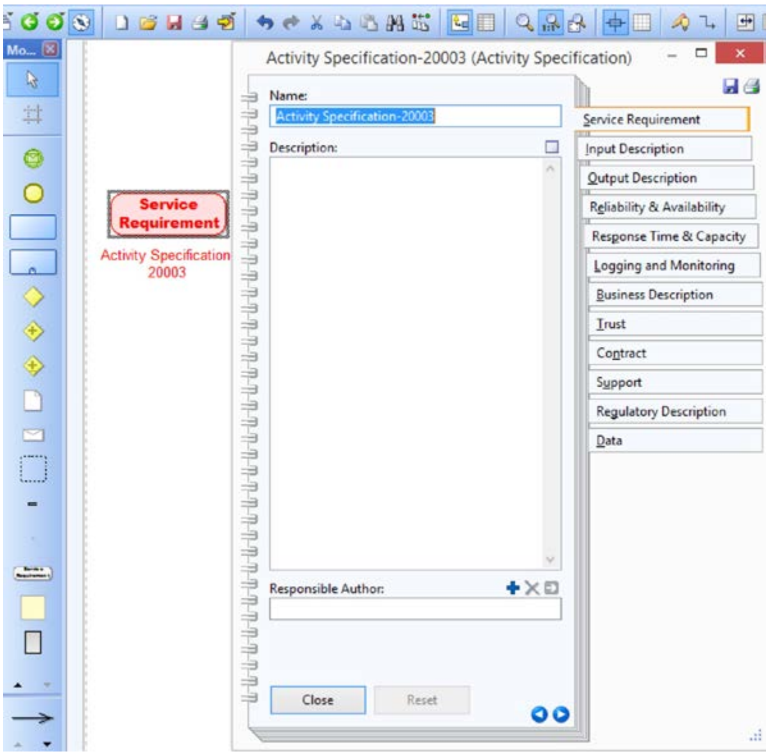

Figure 103: ADOxx Modelling Toolkit - Annotation Notebook

7.2.3 Ontology Download and Editing

The prototype of the current BPaaS Ontology is available from the CloudSocket website using the link https://www.cloudsocket.eu/resources/20151225_CLSO.ttl.

In order to access the prototype of the current ontology, it is necessary to have an ontology environment. Protégé is free and well-known ontology editor that can be downloaded at http://protege.stanford.edu/products.php#desktop-protégé.

After installing Protégé, download the prototype ontology open it directly from the URL.

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

The design environment of the smart Business and IT-Cloud Alignment uses informal (text), semi-formal (graphic) and formal (ontology, rules) knowledge representation languages. The meta model development as suggested by the OMILAB has been adapted to developing both formal ontologies as well as graphical modelling methods in ADOxx.org. In the first phases, the development is synchronized in order to make sure that ontology and meta models are consistent.

Business scenarios were analysed and competency questions were derived in order to determine the scope of the modelling framework. The BPaaS modelling method is implemented in ADOxx meta-modelling platform. The modelling stack shows several model types that are relevant to describe the processes with their contexts and requirements, the workflows and the services. Its purpose is to enable the cloud broker to describe the business process and to specify the requirements for the services to be selected. The model types are extended with algorithms and mechanisms of the semantic lifting to connect the graphical models with the BPaaS ontology The BPaaS Ontology contains the relevant classes for the smart Business and IT-Cloud alignment. The BPaaS Ontology extends the already existing enterprise ontology ArchiMEO with Cloud-specific classes and relations. Inference rule can support reasoning for selecting appropriate workflow, cloud providers and services.

Thus, the research objectives as specified in section 1.2 have been achieved:

- A BPaaS Ontology and meta models for business-IT in the cloud alignment have been developed. The modelling extends standard modelling approaches for business processes and enterprise architecture with cloud-specific modelling elements.
- The BPaaS Design Environment combines human-interpretable graphical modelling with a machine-interpretable formal representation.
- Three approaches for semantic lifting have been developed which (1) align metamodels with semantic definitions, (2) annotate graphical models and (3) transform graphical models into a semantic representation.

8.2 Future Work

A first preliminary version of the prototype is available free for download from the CloudSocket webpage.

The prototype has a focus on the modelling parts: The business process and workflow models, the respective semantic annotations and the rules for the business-IT alignment. The creation of the BPaaS Design Package still requires manual work by the Cloud Broker. To increase automation, the protoype has to be extended with alignment rules and support for semantic annotations.

To identify the appropriate rules and necessary extensions of the BPaaS Ontology and metamodel, the prototype will be extended and adapted in order to satisfy requirements from the business scenarios of the use case partner. The prototype will be applied for smart business IT-Cloud alignment of the Christmas card process. For this we will extend the BPaaS Ontology with rules for smart ontology-based service discovery and business-IT in the cloud alignment. Ontology-based service discovery approaches will be adapted for BPaaS Design Environment.

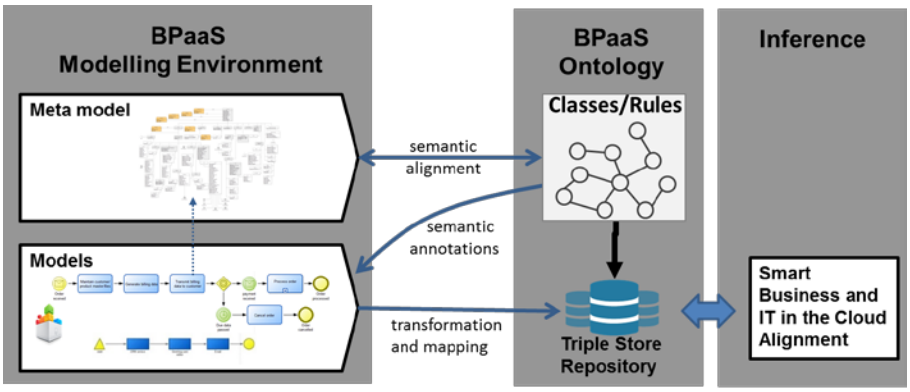

Figure 104: Smart business-IT Alignment for Service Discovery

The application of the modelling and inference approaches to service discovery for some of the business scenarios will show strengths and limitations of the modelling framework. The BPaaS Ontology, the Modelling methods and the semantic lifting will be adapted based on these experiences.

The prototype is due in June 2016 as Deliverable D3.2. The use of the hybrid modelling and semantic lifting for BPaaS allocation and execution will be described in Deliverable D3.3.

9 REFERENCES

Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., & Sintek, M. (1998). Toward a Technology for Organizational Memories. IEEE Intelligent Systems and Their Applications, 13 (3), 40–48.

ADOxx (2015). ADOxx Website. http://www.adoxx.org [Access December 30, 2015].

Akkermans J, Baida Z, Gordijn J (2004) Value Webs: ontology-based bundling of real-world services. IEEE Intelligent Systems 19 (44), 23–32

Albrecht, J. (1996). Universal GIS operations - a task-oriented systematization of data-structure independent GIS functionality leading towards a geographic modelling language. ISPA. Vechta, University of Vechta.

Al-Roomi, M., Al-Ebrahim, S., Buqrais, S., & Ahmad, I. (2013). Cloud Computing Pricing Models: A Survey. *International Journal of Grid and Distributed Computing*, *6*(5), 93–106.

Allemang, D., & Hendler, J. (2011). Semantic Web for the Working Ontologist Effective Modeling in RDFS and OWL Second Edition (Second Edi.). Morgan Kaufmann Publishers.

Alrifai, M., Risse, T. (2009). Combining Global Optimization with Local Selection for Efficient QoS-Aware Service Composition. In WWW, 881–890.

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

APQC (2015). APQC's Process Classification Framework®(PCF) http://www.apqc.org/pcf [Access December 23, 2015].

Ardagna, D., Pernici, B. (2007). Adaptive Service Composition in Flexible Processes. IEEE Transactions on Software Engineering, 3(6), 369–384.

Baida Z, Gordijn J, Omelayenko B (2004) A shared service terminology for online service provisioning. In: ICEC

Barr, J. (1998) Integrating the Management System and Balanced Scorecard. Annual Quality Congress.

Battisto, D., Pak, R., Vander Wood, M. A. and Pilcher, J. J. (2009) Using a Task Analysis to Describe Nursing Work in Acute Care Patient Environments. The Journal of Nursing Administration, 39 (12), 537-547.

Beco, S., et al. (2005). OWL-WS: A Workflow Ontology for Dynamic Grid Service Composition. e-Science, IEEE.

Belhajjame, K., & Brambilla, M. (2009). Ontology-based description and discovery of business processes. In *Enterprise, Business-Process and Information Systems Modeling* (pp. 85-98). Springer Berlin Heidelberg.

Bertolazzi, P., Krusich, C., Missikoff, M., & Manzoni, V. (2001). An Approach to the Definition of a Core Enterprise Ontology : CEO. In International Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations - OES-SEO 2001 (pp. 104–115). Rome.

Bertoli P, Pistore M, Traverso P (2010) Automated composition of Web services via planning in asynchronous domains. Artif Intell 174(3–4):316–361 Brandtzæg E, Parastoo M., Mosser S. (2012). Towards a Domain-Specific Language to Deploy Applications in the Clouds. In Cloud computing 2012: 3rd international conference on cloud computing, grids, and virtualization, pages 213-218. IARIA

Brogi A, Corfini S (2008) Ontology- and behavior-aware discovery of Web service compositions. Int. J. Co-op Inf. Syst. 17(3):319–347

Budinsky F., Steinberg. D., Merks E., Ellersick R. and Grose T.J. (2004). Eclipse modelling Framework. The Eclipse Series. Addison Wesley.

Byrne, J. A. (1993). The Virtual Corporation. Available at http://www.bloomberg.com/bw/stories/1993-02-07/the-virtual-corporation

Buyya, R., Ranjan, R., Calheiros, R. N. (2010). InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services. In ICA3PP, 13–31.

Camarinha-Matos, L. M., & Afsarmanesch, H. (1999). Infrastructure for Virtual Enterprises.

Cloud Security Alliance (2015). STAR Self Assessment. Available at https://cloudsecurityalliance.org/start/self-assessment [Access: December 2, 2015].

CloudSocket (2014). Business and IT-Cloud Alignment using a Smart Socket, Grant Agreement Annex 1 - Description of Action, Part B.

CloudSocket (2015a). Use case analysis and evaluation criteria specification D2.1, April 2015.

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

CloudSocket (2015c). Initial CloudSocket Setup Report D5.1, December 2015

CloudSocket (2015d). Cloud Transformation Framework D2.3, December 2015

Cook S., Jones G., Kent S., und Wills A.C. (2007). Domain Specific Development with Visual Studio DSL Tools (Microsoft .Net Development). Addison-Wesley Longman.

Confora, G., Di Penta, M., Esposito, R., and Villani, M.L. (2008). A framework for QoS-aware binding and re-binding of composite web services. Journal of Systems and Software 81, 10, 1754-1769.

Cortés, A. R., Martín-Díaz, O., Toro, A. D., Toro, M. (2005). Improving the Automatic Procurement of Web Services Using Constraint Programming. Int. J. Cooperative Inf. Syst., 14(4), 439–468.

C-SIG SLA (2014) Claud 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).

D'Andria, F., Gorronogoitia Cruz, J., Ahtes, J., Bocconi, S., Zeginis, D. (2012). Cloud4SOA: Multi-Cloud Application Management Across PaaS Offerings. In MICAS.

Darby, M. R., & Karni, E. (1973). The University of Chicago The Booth School of Business of the University of Chicago The University of Chicago Law School FREE COMPETITION AND THE OPTIMAL AMOUNT OF FRAUD *, *16*(1), 67–88.

da Silva Santos, L. O. B, Ferreira Pirres, L., van Sinderen, M. (2008). A Goal-Based Framework for Dynamic Service Discovery and Composition. In ACT4SOC.

Dastjerdi, A.V, S.G.H. Tabatabaei and R. Buyya (2010). An Effective Architecture for Automated Appliance Management System Applying Ontology-Based Cloud Discovery. IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. Melbourne, Australia pp. 104 – 112.

De Nicola, A., Di Mascio, T., Lezoche, M., Taglino, F. (2008) Semantic Lifting of Business Process Models, Proceeding EDOCW '08 Proceedings of the 2008 12th Enterprise Distributed Object Computing Conference Workshops Pages 120-126

De Nicola, A., Missikoff, M., Proietti, M., & Smith, F. (2010). An open platform for business process modeling and verification. In *Database and Expert Systems Applications,* pp. 76-90. Springer Berlin Heidelberg.

Den Haan, J. (2009, October 15). An Enterprise Ontology based approach to Model-Driven Engineering. TU Delft, Delft University of Technology. Available at http://repository.tudelft.nl/view/ir/uuid:e2093132-9db7-4cba-bc68-9355f93cb9e3/

Diskin, Z. and Dingel, J. (2006). Mappings, Maps and Tables: Towards Formal Semantics for Associations in UML2, in O. Nierstrasz et al., (Eds.): MoDELS 2006, LNCS 4199, pp. 230–244, Springer-Verlag Berlin Heidelberg

Dong, X., Haley, A., Madhavan, J., Nemes, E., and Zhang, J. (2004). Similarity search for web services. In VLDB: Proceedings of the Thirtieth International Conference on Very Large Data Bases. VLDB Endowment, 372-383.

Eclipse (2015). Xtext. http://projects.eclipse.org/projects/modeling.tmf.xtext [Access: December 30., 2015)-

EuroCloud (2011). Leitfaden Cloud Computing. Recht, Datenschutz & Compliance.

EuroCloud (2012). Leitfaden Cloud Computing. Risk & Compliance.

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.

Fensel D., Bussler C. (2002). The Web Service Modeling Framework WSMF, in Electronic Commerce Research and Applications, Vol. 1, No. 2,p 113-137, http://www.wsmo.org/papers/publications/wsmf.paper.pdf, access: 20.06.06

Ferreira, A. M., Kritikos, K., and Pernici, B. (2009). Energy-aware design of service-based applications. In ICSOC. Springer, Stockholm, Sweden.

Ferry, N., Song, H., Rossini, A., Chauvel, F., Solberg, A. (2014). CloudMF: Applying MDE to Tame the Complexity of Managing Multi-Cloud Applications. In: Bilof, R., editor. UCC 2014: 7th IEEE / ACM International Conference on Utility and Cloud Computing. IEEE Computer Society, p. 269–277. doi:10.1109/UCC.2014.36.

Fill, H.-G., Redmond, T., Karagiannis, D. (2012). FDMM: A Formalism for Describing ADOxx Meta Models and Models. In: Maciaszek, L., et al. (eds.): ICEIS 2012, Wroclaw, Poland, Vol. 3, pp. 133-144.

Fox, M. S., Barbuceanu, M., & Grüninger, M. (1996). An organisation ontology for enterprise modelling: Preliminary concepts for linking structure and behaviour. Computers in Industry, 29(1-2), 123–134. doi:10.1016/0166-3615(95)00079-8

García-Gómez, S., Jiménez-Gañán, M., Taher, Y., Momm, C., Junker, F., Bíró, J., Menychtas, A., Andrikopoulos, V., Strauch, S. (2012). Challenges for the comprehensive management of Cloud Services in a PaaS framework. Scalable Computing: Practice and Experience, 13.

Garg, S. K., Versteeg, S., Buyya, R. (2011). SMICloud: A Framework for Comparing and Ranking Cloud Services. In UCC.

Gomez-Perez, A., Fernandez-Lopez, M. & Corcho, O. (2004). *Ontological Engineering* 4th ed., Springer-Verlag London, UK.

Gordijn, J., Akkermans, H., & van Vliet, H. (2000). What's in an Electronic Business Model? Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management SE - EKAW '00, 257–273.

Gordijn, J., & Akkermans, H. (2001). Designing and Evaluating E-business Models. *Ieee*, (July/August), 11–17. http://doi.org/10.1109/5254.941353

Grüninger, M. & Fox, M.S. (1995). Methodology for the Design and Evaluation of Ontologies. *Industrial Engineering (1995)*, 95, pp.1–10. Available at: http://ibict.phlnet.com.br/anexos/ grninger95methodology.pdf.

Gutierrez-Garcia, J. O. Sim, K. (2013). Agent-based cloud service composition. Applied Intelligence, 38, 436–464.

Haller, A., Gontarczyk, J. and P. Kotinurmi. (2008). Towards a Complete SCM Ontology - The Case of Ontologising RosettaNet. In *Proceedings of the 2008 ACM Symposium on Applied computing*, pp. 1467-1473.

He, Q., Han, J., Yang, Y., Grundy, J., Jin, H. (2012). QoS-Driven Service Selection for Multi-Tenant SaaS. In Cloud, 566–573.

Hepp, M. (2005). A Methodology for Deriving OWL Ontologies from Products and Service Categorization Standards. In ECIS, 2005.

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

Hinkelmann, K., Martin, A., Nikles, S., & Thönssen, B. (n.d.). ArchiMEO : Formal Semantics for ArchiMate.

Hinkelmann, K., Merelli, E., & Thönssen, B. (2010). The Role of Content and Context in Enterprise Repositories. In 2nd International Workshop on Advanced Enterprise Architecture and Repositories - AER 2010.

Hlaoui, Y. B. Benayed, L.J. (2011). Model Transformation Approach Based on Homomorphic Mappings between UML Activity Diagrams and BPEL4WS Specifications of Grid Service Workflows, 35th IEEE Annual Computer Software and Applications Conference Workshops.

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

Marinchev, Ivo (2012). Semantic Lifting of Unstructured Data Based on NLP Inference of Annotations, International Conference on Computer Systems and Technologies - CompSysTech'12

IKS (2012). *AcademicTraining/Semantic Lifting*. The Semantic CMS Community IKS. http://wiki.iks-project.eu/index.php/AcademicTraining/Semantic_Lifting [Access December 30, 2015].

ISIS (2008) Generic Model Environment GME, Institute for Software Integrated Systems, Available at www.isis.vanderbilt.edu/Projects/gme [Access: December 30, 2015].

Jaeyong Kang, Kwang Mong Sim (2010). "Cloudle: A Multi-criteria Cloud Service Search Engine", IEEE Asia-Pacific Service Computing Conference (APSCC), Hangzhou, China.

Jinho On, Sujeong Woo, Mooknun Lee (2012). A Graphical Language to Integrate Process Algebra and State Machine Views for Specification and Verification of Distributed Real-Time Systems, IEEE 36th International Conference on Computer Software and Applications Workshops

Kang, D., Lee, J., Choi, S., & Kim, K. (2010). An ontology-based Enterprise Architecture. Expert Systems With Applications, 37(2), 1456–1464. doi:10.1016/j.eswa.2009.06.073

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 Metamodels 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. (2015). Agile Modeling Method Engineering. *Proceedings of the 19th Panhellenic Conference on Informatics*, 5–10.

Karagiannis, D., Höfferer, P. (2006): Metamodels in Action: An overview, In: J. Filipe, B. Shishkov, M. Helfert, ICSOFT 2006 - First Int. Conf. on Software and Data Technologies:IS27-36. Setúbal: Insticc Press.

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

Karim, R., Ding, C., Miri, A. (2013). An end-to-end qos mapping approach for cloud service selection. In SERVICES, 341–348.

Karagiannis, D and Höfferer, P. (2006): Metamodels in action: An overview. ICSOFT

Karagiannis, D. and Kühn, H. (2002). Metamodelling Platforms. Invited Paper. In: Bauknecht, K.; Min Tjoa, A.; Quirchmayer, G. (Eds.): Proceedings of the Third International Conference EC-Web 2002 – Dexa 2002, Aix-en-Provence, France, September 2-6, LNCS 2455, Springer-Verlag, Berlin, Heidelberg, p. 182

KBSI (1995) IDEF3 Process Description Capture Method Report, Knowledge Based Systems, Inc KBSI-IICE-90-STR-01-0592-02 and AL-TR-1995-XXXX. Available at http://www.idef.com/pdf/Idef3_fn.pdf

Kehrer, T., Kelter, U. and Taentzer, G. (2011). A Rule-Based Approach to the Semantic Lifting of Model Differences in the Context of Model Versioning, ASE 2011, Lawrence, KS, USA

Kehrer, T., Kelter, U., Ohrndorf, M. and Sollbach, T. (2012). Understanding Model Evolution through Semantically Lifting Model Differences with SiLift, 28th IEEE International Conference on Software Maintenance (ICSM)

Kell, U., Lara, R., Polleres, A., Toma, I., Kifer, M., and Fensel, D. (2004). WSMO Web service discovery. http://www.wsmo.org/2004/d5/d5.1/v0.1/20041112.

Kelly S., Tolvanen J.-P. (2008) Domain-Specific Modelling: Enabling Full Code Generation, John Wiley & Son, Inc.

Kern, H., Hummel, A., Kühne, S. (2011). Towards a Comparative Analysis of Meta-Metamodels, In The 11th Workshop on Domain-Specific Modeling, Portland, ISA. www.dsmforum.org/events/DSM11/Papers/kern.pdf (last access 05.01.2012)

Klein, M. (2002). DAML+OIL and RDF Schema representation of UNSPSC. Available at: http://www.cs.vu.nl/~mcaklein/unspsc/ (Retrieved 23.04.2004).

Klein, A., Ishikawa, F., Honiden, S. (2012). Towards network-aware service composition in the cloud. In Proceedings of the 21st International Conference on World Wide Web (WWW 2012).

Klusch, M., Fries, B., and Sycara, K. (2009). OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services. Web Semantics: Science, Services and Agents on the World Wide Web 7, 2, 121-133.

Knublauch,H., Hendler, J.A. and Idehen, K. (2011). SPIN - Overview and Motivation. W3C Member Submission. Available at http://www.w3.org/Submission/spin-overview/ [Access December 23, 2015].

Kofler, K., Haq, I. U., Schikuta, E. (2010). User-centric, heuristic optimization of service composition in clouds. In EuroPar, 405–417.

Kolovos, D.S., Paige, R.F. and Polack, F.A.C. (2006). Merging Models with the Epsilon Merging Language (EML), in O. Nierstrasz et al., (Eds.): MoDELS 2006, LNCS 4199, pp. 215–229. Springer-Verlag Berlin Heidelberg.

Kritikos, K., Kubicki, S., Dubois, E. (2013). Goal-based business service composition. Service Oriented Computing and Applications, 7(4), 231-257.

Kritikos, K. Plexousakis, D. (2014). Novel Optimal and Scalable Nonfunctional Service Matchmaking Techniques. IEEE T. Services Computing, 7(4), 614–627.

Laatikainen, G., Ojala, A., & Mazhelis, O. (2013). Cloud Services Pricing Models. In *4th International Conference, ICSOB 2013* (pp. 117–129).

Lapouchnian A, Yu Y, Mylopoulos J (2007). Requirements-driven design and configuration management of business processes. In: BPM, pp 246–261

Leppänen, M. (2005). A Context-Based Enterprise Ontology. In G. Guizzardi & G. Wagner (Eds.), Proceedings of the EDOC International Workshop on Vocabularies, Ontologies and Rules for the Enterprise (VORTE'05) (pp. 17–24). Enschede, Netherlands: Springer Berlin.

Liao, Y., Roman, D. and Berre, A.J. (2010). Model-driven Rule-based Mediation in XML Data Exchange, MDI2010, October 5, Oslo, Norway.

Lin, Y., (2008). Semantic Annotation for Process Models. PhD thesis, Norwegian University of Science and Technology.

Lo A, Yu E (2007) From business models to service-oriented design: a reference catalog approach. In: ER. Springer, pp 87–101

LSDIS (2005). Semantic annotations for WSDL., http://lsdis.cs.uga.edu/projects/metero-s/SAWSDL/ [December 22, 2015]

Malone, T. W., Crowston, K. and Herman, G. A. (2003). Organizing Business Knowledge: The MIT Process Handbook. Cambridge, MA: MIT Press.

Meixner, F., & Buettner, R. (2012). Trust as an Integral Part for Success of Cloud Computing. *ICIW 2012, The Seventh International Conference on Internet and Web Applications and Services*, 207–214.

Matthes, D. (2011). Enterprise Architecture Frameworks Kompendium. Heidelberg: Springer-Verlag.

MODAClouds (2015) *MOdel-Driven Approach for design and execution of applications on multiple Clouds*. MODACloluds Project Website. http://www.modaclouds.eu/ [Access December 30, 2015)

Mohagheghi P., Berre J-J, Henry A., Barbier F., and Sadovykh A. (2010). REMICS- REuse and Migration of Legacy Applications to Interoperable Cloud Services - REMICS Consortium. In Di Nitto E.,Yahyapour R., ServiceWave, vol. 6481 of Lecture Notes in Computer Science, pages 195–196. Springer-Verlag.

Nelson, P. (1973). Information and Consumer Behavior. *Journal of Political Economy*, *78*(2), 311–329.

Nelson, P. (1974). Advertising as information. *Journal of Political Economy*, *84*(4), 729–754.

NIST (2001). Security Requirements for Cryptographic Modules (FIPS PUB 140-2). National Institute of Standards and Technology (NIST), Available at http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. Palo Alto: Stanford Knowledge Systems Laboratory Technical Report KSL-01-05. Available at http://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Obeo (2015) Obeo Deisgner Key Features. Available at http://www.obeodesigner.com/product/key-features [Access December 30, 2015).

OMG (2009) Unified Modelling Language™ (OMG UML), Infrastructure, Version 2.2, Object Management Group OMG. Available at: http://www.omg.org/docs/formal/09-02-04.pdf

OMG (2010). Business Motivation Model, Version 1.1. Object Management Group OMG. Available at http://www.omg.org/spec/BMM/1.1/PDF/

OMG (2011). Business Process Model and Notation (BPMN) Version 2.0. Needham, MA: Object Management Group OMG. Available at http://www.omg.org/spec/BPMN/2.0/PDF/

OMG (2013). Case Management Model and Notation (CMMN), Version 1.0. Needham, MA: Object Management Group OMG. Available at http://www.omg.org/spec/CMMN/1.0/PDF

OMG (2015a) OMG Meta Object Facility (MOF) Core Specification, Version 2.5, formal/2015-06-05. Available at http://www.omg.org/spec/MOF/2.5

OMG (2015b). *Decision Model and Notation Version 1.0*. Needham, MA: Object Management Group OMG. Available at http://www.omg.org/spec/DMN/1.0/PDF

PaaSage (2015). *PaaSage: Model-based Cloud Platform Upperware*. PaaSage Project website http://www.paasage.eu/ [Access December 30, 2015].

Paolucci, M., Kawamura, T., Payne, T. R., Sycara, K. P. (2002). Semantic Matching of Web Services Capabilities. In ISWC '02: Proceedings of the First International Semantic Web Conference on the Semantic Web, 333–347.

Park, H., Yoon, A. and Kwon, H.-C. (2012). Task Model and Task Ontology for Intelligent Tourist Information Service. International Journal of u- and e-Service, Science and Technology, 5(2), 43-58.

Peffers, K. et al., (2008). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), pp.45–77.

PCI (2015). Datensicherheitsstandard. Anforderungen und Sicherheitsbeurteilungsverfahren, Version 3.0. Payment Card Industry (PCI) Security Standards Council. Available at https://de.pcisecuritystandards.org/minisite/en/ [Retrieved December 4, 2015]

Plebani, P. and Pernici, B. (2009). URBE: Web Service Retrieval Based on Similarity Evaluation. IEEE Transactions on Knowledge and Data Engineering 21, 11, 1629-1642.

plugIT (2011) *plugIT - Business and IT Alignment using a Model-based Plug-in Framework*. plugIT Project Website http://www.plug-it-project.eu [Access December 30, 2015].

Ramel S, Kubicki S, Vagner A, Braye L (2010) Viewpoints reconciliation in services design: a model-driven approach for highly collaborative environments. In: BPMDS

Repschläger, J., Wind, S., Zarnekow, R., & Turowski, K. (2011). Developing a Cloud Provider Selection Model. In EMISA (pp. 163-176).

Rolland C, Kirsch-Pinheiro M, Souveyet C (2010) An intentional approach to service engineering. Serv Comput IEEE Trans 3(4):292–305. doi:10.1109/TSC.2010.26

Ross, R. G., & Lam, G. S. W. (2011). *Building Business Solutions: Business Analysis with Business Rules*. Business Rule Solutions Inc.

Ruiz-Alvarez, A. Humphrey, M. (2011). An Automated Approach to Cloud Storage Service Selection. In ScienceCloud.

Saaty T (1980) The analytic hierarchy process. McGraw-Hill, New York Hamza Saouli, Okba Kazar, and Aïcha Nabila Benharkat. 2015. SaaS-DCS: software-as-a-service discovery and composition system-based existence degree. Int. J. Commun. Netw. Distrib. Syst. 14, 4 (May 2015), 339-378.

Salesforce. (2014). Die passende Sales Cloud- Edition für alle Unternehmen. Retrieved November 2, 2015, from http://www.sfdcstatic.com/de/assets/pdf/datasheets/DS_SalesCloud_EdCompare.pdf

Shahar, Y., Miksch, S. and P. Johnson, P. (1998). The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines. Artif. Intell. Med., 14 (1-2), 29-51, 1998.

Smart, N. (Ed.) (2012). ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012), Revision 1.0. ICT-2007-216676 ECRYPT II, http://www.ecrypt.eu.org/ecrypt2/documents/D.SPA.20.pdf

Sohrabi S, McIlraith SA (2010) Preference-based web service composition: a middle ground between execution and search. In: ISWC

Sowa, J. F., & Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture. IBM Systems Journal, 31(3).

Srinivasan, S. (2014). Building Trust in Cloud Computing: Challenges in the Midst of Outages. In Proceedings of Informing Science & IT Education Conference (pp. 305-312).

Srinivasan, S. S., & Till, B. D. (2002). Evaluation of search, experience and credence attributes: Role of brand name and product trial. *Journal of Product & Brand Management*, *11*(7), 417–431.

Stollberg, M., & Fensel, D. (2010). Semantics for Service-Oriented Architectures. In Agent-Based Service-Oriented Computing (pp. 113-139). Springer London.

Strahringer, S. (1996) Metamodellierung als Instrument des Methodenvergleichs: eine Evaluierung am Beispiel objektorientierter Analysemethoden. Shaker, Aachen

Stuckenschmidt, H. (2011). *Ontologien. Konzepte, Technologien und Anwendungen* 2nd ed. O. P. Günther et al., eds., Springer Berlin Heidelberg.

SUPER (2009). *SUPER - Semantics Utilized for Process management within and between EnteRprises*. http://cordis.europa.eu/ist/kct/super_synopsis.htm [Access December 30, 2015].

Sycara, K., Wido, S., Klusch, Mand Lu, J. (2002. Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. J. Auton. Agents and Multi-Agent Syst. 5, 173–203.

Teuteberg, F., Kluth, M., Smolnik, S. and Ahlemann, F. (2009). Semantic Benchmarking of Process Models - An Ontology-Based Approach. In Proceedings of the International Conference on Information Systems ICIS.

The Free Dictionary (n.d.) *Lifting*. http://www.thefreedictionary.com/lifting [Access December 30, 2015].

The Open Group. (2011). TOGAF® Version 9.1. Van Haren Publishing..

The Open Group. (2012). ArchiMate 2.1 Specification. The Open Group. http://pubs.opengroup.org/architecture/archimate2-doc/ [Access November 3, 2015]

Thönssen, B. (2013). Automatic, Format-independent Generation of Metadata for Documents Based on Semantically Enriched Context Information. Knowledge Creation Diffusion Utilization. University of Camerino.

Thönssen, B., & Lutz, J. (2012). Semantically Enriched Obligation Management. An Approach for Improving the Handling of Obligations Represented in Contracts. In Proceedings of 4th Conference on Knowledge Management and Information Sharing (KMIS2012).

Uschold, M. & Grüninger, M., (1996). Ontologies : Principles, Methods and Applications.Technical Report, University of Edingburgh Artificial Intelligence Institute AIAI, (191).

Uschold, M., King, M., Moralee, S., & Zorgios, Y. (1997). The Enterprise Ontology. The Knowledge Engineering Review, 13.

Verma, H. V. (2012). *Services Marketing. Text and Cases*. Dorling Kindersley (India) Pvt.

von Hunolstein, S. and Zipf, A. (2003). Towards Task-Oriented Map-Based Mobile Guides. In Mobile HCI.

Wagner, C. (2004). Enterprise strategy management systems: Current and next generation. Journal of Strategic Information Systems, 13(2), 105–128.

Walton, J., & Whicker, L. (1996). *Virtual Enterprise: Myth & Reality*.

WfMC (1999). *Workflow Management Coalition: Terminology & Glossary*. wfmc-tc-1011.

Woitsch R., Karagiannis D., Plexousakis D., Hinkelmann K., (2009). Business and IT Alignment: the IT-Socket, Elektrotechnik & Informationstechnik 127/6.

Wittern, E., Kuhlenkamp, J., Menzel, M. (2012). Cloud service selection based on variability modelling. In ICSOC, 127–141.

W3C (2012). *OWL 2 Web Ontology Language Document Overview (Second Edition)*. Available at http://www.w3.org/TR/2012/REC-owl2-overview-20121211/

W3C (2013) SPARQL 1.1 Overview. W3C Recommendation. Available at http://www.w3.org/TR/sparql11-overview/ [Access December 30, 2015).

W3C (2014). *RDF 1.1 Concepts and Abstract Syntax*. Available at http://www.w3.org/TR/rdf11-concepts/

Zachman, J. A. (2008). John Zachman's Concise Definition of The Zachman Framework. Zachman International. Available at http://zachmaninternational.com/

Zdravković, M., Panetto, H., Trajanović, M., and Aubry, A. (2011). An Approach for Formalizing the Supply Chain Operations. Enterprise Information Systems, Taylor & Francis, 5 (4), pp.401-421.

Zeng, C., Guo, X., Ou, W., Han, D. (2009). Cloud computing service composition and search based on semantic. In CloudCom, 290–300.

Zhang, M., Ranjan, R. Haller, A., Georgakopoulos, D., Menzel, M.and Nepal, S. (2012): An ontology-based system for Cloud infrastructure services' discovery. In CollaborateCom 2012: 524-530

Zhou, C., Chia, L.-T., Lee, B.-S. (2004). DAML-QoS Ontology for Web Services. In ICWS, 472– 479.