

Università degli studi di Camerino

Scuola di Scienze e Tecnologie

Laurea Magistrale in Computer Science LM-18

Double Degree with

**University of Applied Sciences and Arts Northwestern
Switzerland (FHNW)**

Master of Science in Business Information Systems



Towards a Semantic Enrichment of Business-IT Alignment in the Cloud

Candidate:

Fabio Marcotulli

Supervisor (FHNW):

Knut Hinkelmann

Co-Supervisor (FHNW):

Emanuele Laurenzi

Supervisor (UNICAM):

Barbara Re

Abstract of the thesis

Nowadays Information Technology and Business Process are related to each other and spread in all fields and all possible models. A business process automated with the IT increases the speed and maximize value, monitoring the execution of processes and allowing the company to make analysis and changes according to the current condition of the environment.

Cloud computing has overcome the phase of being a hype and has started establishing itself in business. It plays a valuable role in business processes influencing structures and decisions and causing a paradigm shift on how IT is used, delivered and taken advantage of (Chandrasekaran, 2015).

The problem for companies to identify the appropriate cloud services that can be used to automate and support business processes block the growth and the evolution of these companies. This is due to lack of expertise and poor technical knowledge on business side.

This is the issue that affects especially small and medium enterprises and this is the problem dealt with by this research.

There is a need for a business and IT alignment in the cloud. To fulfil this need, this thesis proposes an approach that maps and comes up with suggestions and specifications of Business-IT alignment model of cloud services.

This thesis provides a matching approach between business process requirement (i.e. Business layer) and cloud service specification (i.e. IT layer). To this end, annotations of functional and non-functional aspects from both Business and IT layers are provided appropriately.

Statement of Authenticity

I confirm that this master thesis research was performed autonomously by myself using only the sources, aids and assistance stated in the report, and that quotes are readily identifiable as such.

Felxo Marcantelli

Table of Contents

Abstract of the thesis	I
Statement of Authenticity	II
Table of Contents.....	III
1. Introduction.....	1
1.1 Research Question	1
1.2 Research Approach.....	2
1.3 Scope and Limitations	3
1.4 Contribution of the Thesis	3
2. Literature Review	5
2.1 Cloud Computing.....	5
2.1.1 Characteristics of Cloud Computing	6
2.1.2 Models of Cloud Computing	6
2.1.3 Types of Cloud Services	7
2.2 Requirement Engineering	11
2.2.1 Requirements	12
2.2.2 Types of Requirements	13
2.2.3 Functional Requirements	14
2.2.4 Non-Functional requirement.....	14
2.3 Business Annotation	16
2.3.1 APQC.....	16
2.3.2 Object / Action.....	18
2.4 IT annotation.....	19
2.4.1 Web Services	19
2.5 Business-IT alignment.....	24
2.6 Ontology	26

2.7	Conclusion and Research Gap	29
3.	Research Design and Methodology	30
3.1	Research Philosophy	31
3.2	Research Approaches	33
3.3	Research Strategy	34
3.4	Research Choice	36
3.5	Time Horizons	37
3.6	Research Method applied in this Thesis	38
3.6.1	Awareness of problem	38
3.6.2	Suggestion	39
3.6.3	Development	39
3.6.4	Evaluation	39
3.6.5	Conclusion	39
4.	Awareness of the problem	40
4.1	Introduction	40
4.2	Application Scenario	41
4.2.1	What is an invoice?	42
4.2.2	Evolving market models	43
4.3	Case Study	45
4.4	Conclusion	52
5.	Suggestion phase	53
5.1	Introduction	53
5.2	Annotation for the mapping	53
5.2.1	Business process functional requirements	54
5.2.2	Cloud services functional specifications	58
5.2.3	Business and IT non-functional specification	60
5.3	Conclusion	68

6.	Development.....	69
6.1	Ontologies.....	69
6.2	Ontology for functional requirements	70
6.2.1	Business Process Functional Requirement	71
6.2.2	Cloud service Specification	77
6.3	Ontology for non-functional requirements	78
6.3.1	Business Process Non-Functional Requirement	78
6.3.2	Cloud Service Non-Functional Specification	80
6.4	Cloud services.....	82
6.5	Conclusion	85
7.	Evaluation.....	87
7.1	Business – IT Functional Layer.....	87
7.1.1	Matching	88
7.2	Business – IT Non-Functional Layer.....	88
7.2.1	Transformation from Business to IT.....	88
7.3	Conclusion	92
8.	Conclusion & Outlook.....	95
	Bibliography	96
	Abbreviations.....	103
	List of Figures / Tables	105

1. Introduction

Almost all business processes are more performant, competitive and effective when based on information systems and IT (Jeanne W. Ross & Peter Weill, 2004). This automatization makes processes more flexible and dynamic. The Business-IT alignment is a paradigm that strives to bring the two worlds together in a shared understanding. There is room for improvement on the alignment; as Sabherwal & Kirs, (1994) showed, that can give a better result and reduce efforts: automatization gives a wide range of benefits that starts from the customer cares to the back office internal work.

Many small and medium enterprises (SMEs) have difficulty to deeply integrate business and IT, showing a gap that involves on useless IT investments (Oliver Akpan, 2007).

There are still cases where business people remain too far away from the IT reality.

The purpose of this thesis is to investigate in specifications that prevent IT from integrating with business level in cloud computing environment, identifying the missing points and finding out an appropriate method to close this gap.

The alignment of business and IT has to create an environment in which installing technology not merely support business processes but technology also is used to shape business strategy.

The thesis statement for the research is the following:

It is possible to represent business process requirements and cloud service specifications in a machine interpretable format such that the given requirements can match with the related specifications.

1.1 Research Question

This research intends to answer the following question:

- 1) How to represent functional and non-functional requirements for business processes?
- 2) How to annotate functional and non-functional specifications for cloud services?
- 3) How to match business process requirements with cloud service specifications in order to discover cloud services?

The first and second questions will be answered in the “Suggestion Phase”, the third answer in the “Development Phase”.

1.2 Research Approach

This thesis follows a constructive approach using design science research. “It involves a rigorous process to make research contributions, to evaluate the designs, and to communicate the results to appropriate audiences. May include constructs, models, methods, and instantiations. They might also include social innovations or new properties of technical, social, and/or informational resources with an embedded solution to an understood research problem.” (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007)

Design research consists of following five phases (Vaishnavi & Kuechler, 2008) :

➤ **Awareness of the problem.**

In the first phase, a specific business process of CloudSocket it is analysed. The analysis will be focused on the identification of the specifications (Functional and Non-Functional), in the respective layers (Business or IT) and how it is possible to align.

➤ **Suggestion Phase.**

The information collected through the awareness phase helps to create a model for the interconnection between concepts.

After choosing the right field and the most appropriate category for gathering the tasks from the business process it is possible the creation of a UML diagram that shows the connection of the requirements from the business level to the IT level and vice versa and the development of rules in natural language that allow these matches.

➤ **Development phase.**

In the development phase, all the interviews and concepts found in the previously phase are re-elaborated in the ontology.

➤ **Evaluation phase.**

Evaluation of the results from the analyses. Using a scenario of use the work of the development phase will be evaluated.

➤ **Conclusion.**

In this phase, all research contributions and outcomes are reported.

1.3 Scope and Limitations

The main scope of this research is to relate the business process with the relative IT infrastructure concerning cloud computing.

For this reason, the following fields will be analysed:

- **Cloud Computing**

Seeing the different service types and describing the most suitable for that case.

- **Cloud Socket Invoice Service**

The thesis is focused on the specific process of sending invoice to customer, presenting the structure of the process, the function, the objects and the actions. This structure will be analysed from business and IT view point, seeing the differences and similarities between the two layers.

- **Modelling requirements and specifications of processes**

Finding out the requirements inside the case study for analysing it and trying to combine the two levels.

- **Modelling and matching algorithm**

Creation of rules that will be applied on the ontologies (spin rules) for the alignment between business process requirements and cloud service specifications.

- **Interview with Mathema Group**

Mathema is a company that operates in the Information Technology and gave the possibility to evaluate the tables for the thesis prototype.

1.4 Contribution of the Thesis

Starting with the assumption that in the era of digitalization IT being driven factor of nowadays companies' success, the need of integrating the business and the IT becomes even more important, especially in SMEs, where IT expertise might be missing. Additionally, there is still the evergreen issue of business people asking for something that is not very in line with the IT language.

For this reason, there is a need for matching the two layers in order to identify the most suitable workflows for the given business process specification, by proposing a hybrid modelling approach that enables both business users and IT experts to a smart alignment interpretation for machine and human.

2. Literature Review

This chapter provides an overview on concepts related to this thesis, which are cloud computing, services discovery and matching methods (for business and IT layer). First a brief definition of the term: Cloud Computing - introducing the main concepts, the services offer and the various types of existing models, then the specific analysis of the current service adopted. In the second part, it will be analysed the research requirements of the services, with a description of what a requirement is, the distinction between functional and non-functional and the different service discovery types. In the last part, the modelling method will be defined. This chapter will end with the identification of the research gap derived from the literature review performed.

2.1 Cloud Computing

The definition of cloud computing according to the NIST (National Institute of Standards and Technology) is: *“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”* (Mell & Grance, 2011).

Technological advantages in information technology seem to have no limits. Even the world of business has been called to keep up with these advantages of technology, which have affected the IT industry in recent years. At the moment, companies are having to deal with another challenge: cloud computing.

The innovation brought by this technology is the ability to access resources easily configured to suit needs and accessible directly from the Internet.

2.1.1 Characteristics of Cloud Computing

NIST also defines 5 essentials characteristics for *Cloud*:

- *On-demand self-service*. The user can choose cloud computing services, such as server time or network storage as needed automatically without requiring human interaction.
- *Broad network access*. Access to these services is available through the network, using standard and common mechanisms like thin or thick client platforms (mobile hand-held, laptops or workstation).
- *Resource pooling*. The resources used by the provider will follow a multi-tenant model, this allows the reallocation of the various resources (physical or virtual) depending on the customer's request.

Customers have little or no control over the exact location of the resources made available, but are able to specify the country, state or data centers where the resource is located. Resources are meant storage, processing, memory, network and virtual machines bandwidth.

- *Rapid elasticity*. Capabilities must be rapidly and automatically adaptable to quickly scale out, giving the possibility to the customer to have the resources available also during peak times in any quantity at any time.
- *Measured service*. Cloud system can automatically control and monitor the service and it is able to optimize the services depending on the use and necessity. All resources used can be measured and controlled for better transparency between the provider and the user.

2.1.2 Models of Cloud Computing

The distribution patterns indicate where a cloud computing model has to be organized. In this case, it is possible to find four alternatives.

Private Cloud: The Cloud computing services are provided by the company, or by an external provider, only for the company itself and its various units. The main purpose is the fruition of the services: the infrastructure can be managed or operated by the company itself or by an external provider, the property can be internal or external, but the services must be the prerogative of a single organization.

Community Cloud: The Cloud computing services are provided by a company or a service provider to a select group of organizations that share some features like, for example, security levels, legal requirements, objectives, policy and so on. Also in this case, the infrastructure can be handled or operated by one of the companies of the group or by an external provider.

Public Cloud: Cloud computing services are provided by a service provider through the Internet to different customers. The infrastructure, platform, applications are owned by the service provider, all the functions are managed by the service providers and are shared with more customers (public or industry group).

Hybrid Cloud: The services are built on hybrid infrastructure that uses two or more clouds (private, community, public) the private mode, for some aspects (for example the retention of data) and the public mode, for other (for example the access interfaces). The different infrastructures are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds). (Mell & Grance, 2011)

2.1.3 Types of Cloud Services

IaaS is a type of cloud computing in which a third-party provider hosts resources of virtualized computing on the Internet. It provides hardware, software, servers, storage and so on, IaaS providers are also home user applications and help to manage the system maintenance, including backups and resilience planning. The resources offered are highly scalable and can be adjusted on demand.

PaaS is the cloud computing model that gives users the applications that are in excess of an infrastructure. In particular, a PaaS provider gives the tools dedicated to application development. The PaaS services in principle are hosted on hardware and software belonging to the infrastructure provider, freeing end users from server maintenance operations. Generally, companies use PaaS not to replace the entire corporate infrastructure, but only for the services that let applications develop without having to purchase and install additional hardware locally.

SaaS is a cloud computing model in which software are hosted by third-party vendors and are available through the Internet. It is a type of cloud that is becoming increasingly

popular, as it allows customers not to worry about any updates, compatibility, accessibility software, and so on.

The figure below shows an overview of the cloud computing models and characteristics

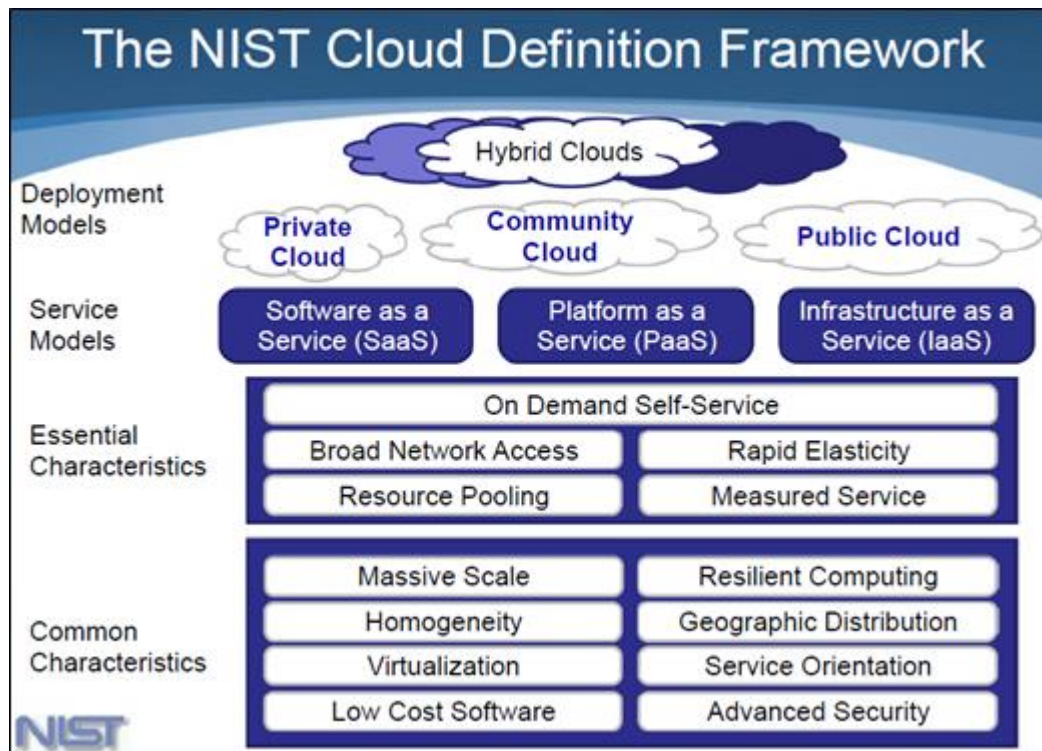


Figure 1 The NIST cloud definition framework. (NIST, 2009)

Recently a new model has emerged, the next level of abstraction above SaaS, PaaS and IaaS, it is Business Process as a Service (BPaaS).

BPaaS: In this fourth service the software is not only provided to the user as a service but also delivers the logic and the control of the flow of the business process that the final user wants to execute as a service. IBM defines BPaaS as “*any business process (horizontal or vertical) delivered through the Cloud service model (Multi-tenant, self-service provisioning, elastic scaling and usage metering or pricing) via the Internet with access via Web-centric interfaces and exploiting Web-oriented cloud architecture*” (Behrendt & Arsanjani, 2011).

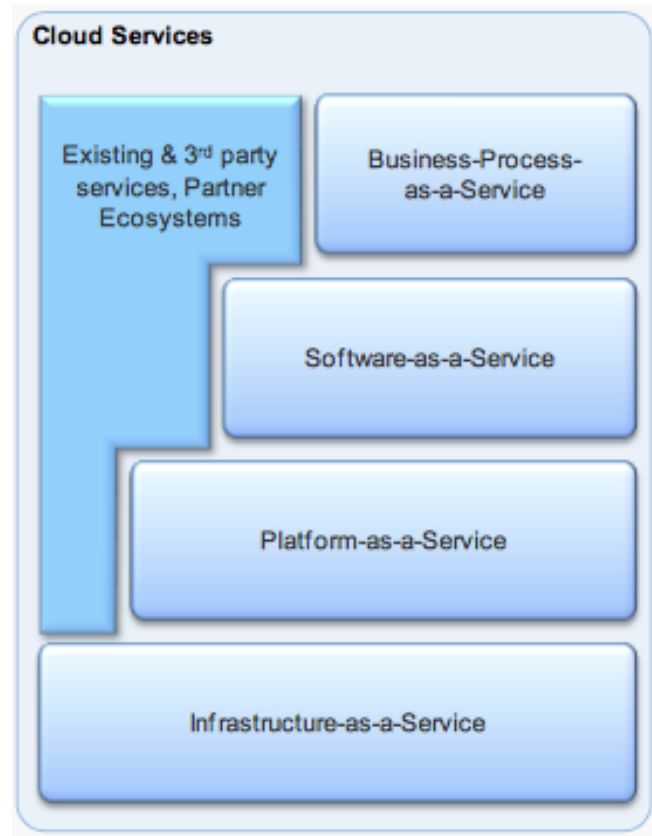


Figure 2 Classification of Cloud Services

BPaaS has gained importance in the scientific literature in 2013. It is compared to SaaS marketplace but the differences for BPaaS are in the possibility to have a selected specific domain and technical criteria; in fact as Woitsch & Utz, (2016) said it is a two-stage marketplace. The purpose of the BPaaS is to decrease costs through automation of business processes.

The main prerogative of these services, in addition to their flexibility, is that in many cases it is possible for the purchaser to make available their own custom services again to final customers. BPaaS is considerate a service-oriented application.

In these services are included those of Business Process Outsourcing (BPO) delivered in Cloud mode, that services built for multi-tenancy use. These types of services are not specifically dedicated to a single customer; in fact, it is standardized service that can be used by many different organizations. These services are optimized to provide consistently automated, standardized and repeatable services.

The main features of a BPaaS are:

- BPaaS sits on top of the other cloud services (SaaS, PaaS, IaaS).
- Having a well-defined APIs to be easily connected to services.
- It supports multiple languages and deployment environments for not limiting it in only a specific development environment.
- Flexibility and resilience to ensure the management of users and services from a few to thousands, simultaneously.

Accorsi, (2011) defines BPaaS as “a special SaaS provision model in which enterprise cloud offerors provide methods for the modelling, utilization, customization, and (distributed) execution of business processes”.

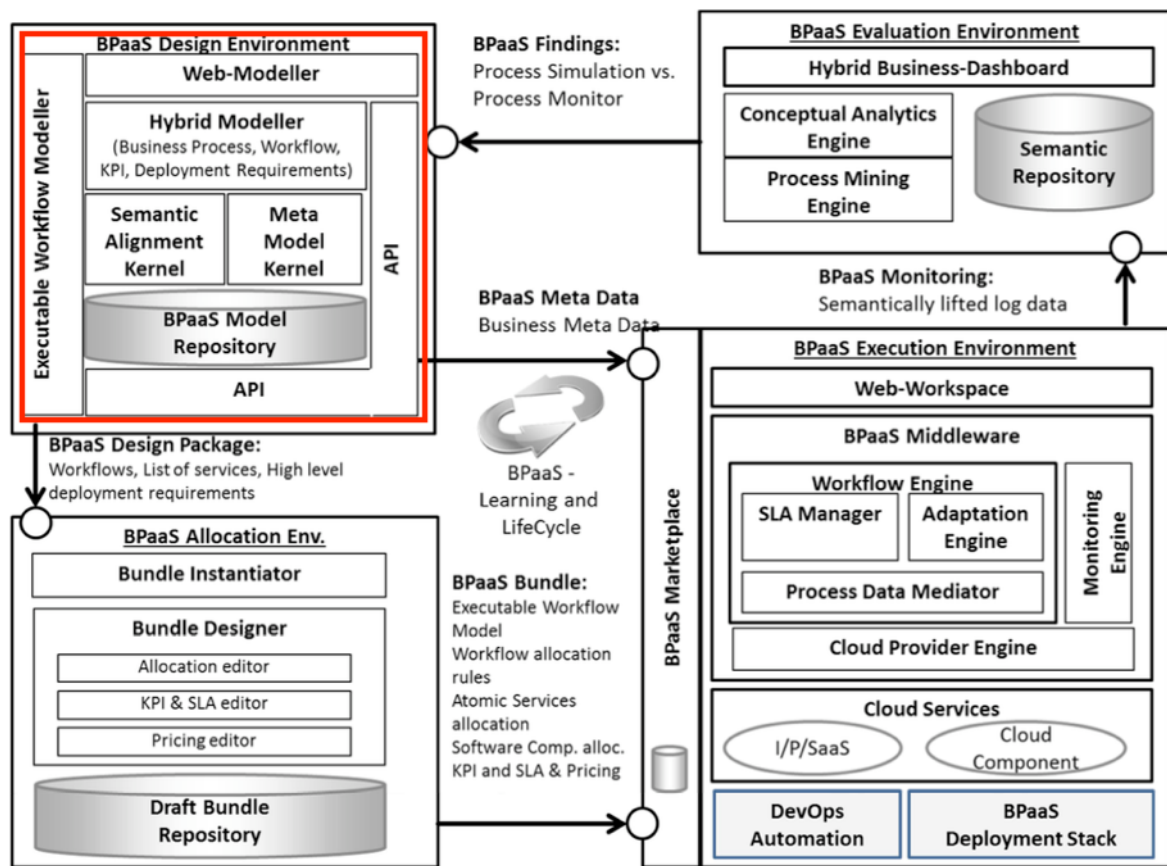


Figure 3 Initial High-level Architecture of CloudSocket (CloudSocket 2015a)

Through the literature analysis, BPaaS is considered the most suitable and appropriate service for the management of cloud services. The use of BPaaS is not a new concept in fact, because also other programs and other business logic software (ERP, CRM) use it.

The advantage of this service model is to choose a business process that is linked with various services including SaaS, IaaS and PaaS. The same service is also used in the European project CloudSocket.

The Figure 3 highlights the BPaaS Design Environment: the thesis will focus precisely on the environment design in which the Business requirement and IT specifications are modelled with the most appropriate language on each side. These two levels are like two parallel worlds that "speak" a different language, so there is a need to find a way to put them in communication. The next chapter will present the requirements and specifications and how they can be annotated.

2.2 Requirement Engineering

Requirement Engineering (RE) is the process by which system requirements are identified, analysed, validated and documented.

During the last decade the field of Requirement Engineering has gained an important and serious role for the development of a system as Besrour et al., (2016) say: "requirement still one of the most critical process in software development." In fact, very often system defects are coming from wrong or missing requirements.

Álvaro Rocha, Ana Maria Correia, Felix . B Tan, & Karl . A Stroetmann (2014) define RE as "*the process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed*".

The purpose of the Requirement Engineering is the production of a document (the *system requirement document*) that defines the functionalities and services offered by the system to achieve.

This document has to define what the system should do. Before starting the development, it is necessary to identify the requirements of the customers. According to Sommerville, (2010) it is possible to divide this process in three main activities that are:

- Requirement Elicitation
- Requirement Specification
- Requirement Validation

In the present study, all three activities have been dealt with. The **Elicitation** through literature review and the present work for the CloudSocket project, the **Specification** through the creation of tables that were then included in the ontology and **Validation** that will be shown in chapter 7. Following subsection, there is a description of what the requirements are and their classifications.

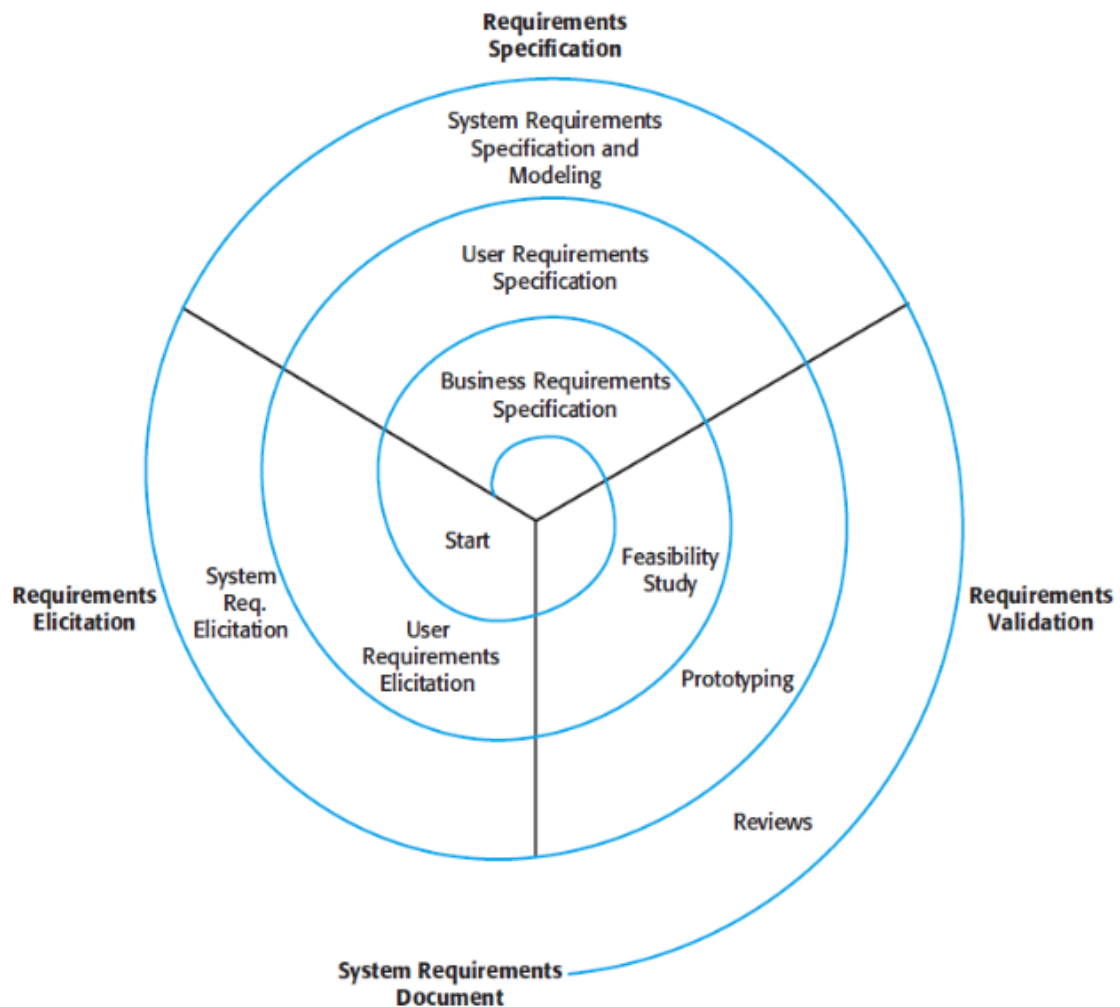


Figure 4 Spiral view of the requirements engineering process

2.2.1 Requirements

The requirements describe the properties of the software system in relation to certain services or functions and can also be related to the process: efficiency, reliability, safety and so on.

IEEE, (1990) defines requirement as:

“(1) A condition or capability needed by a user to solve a problem or achieve an objective.

(2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.

(3) A documented representation of a condition or capability as in (1) or (2).”

It means that a requirement is something necessary for the person or the organization, it is asked for satisfy regulation or norm and it can be represented also in a documented version.

2.2.2 Types of Requirements

There are two main types of requirements:

- Functional Requirement Type
- Non-Functional Requirement Type

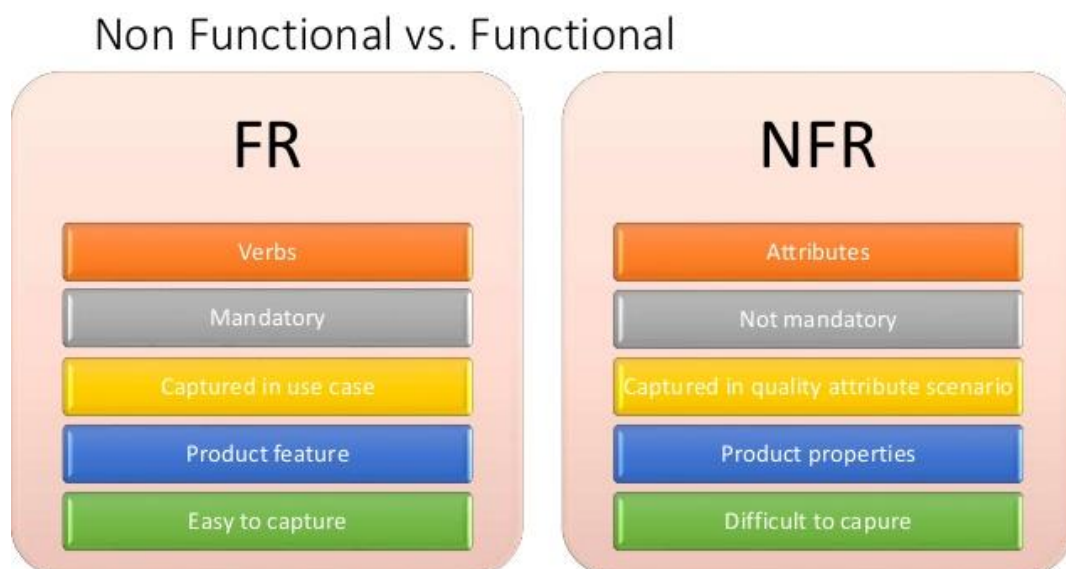


Figure 5 Main differences between Functional and Non-Functional Requirements

Here is specific analysis for each one.

2.2.3 Functional Requirements

The functional requirements specify what the system has to perform. They concern actions that the system must do to satisfy its purpose; it describes the features of the system software in terms of services that should provide, how the software system reacts to specific types of input and how it behaves in a particular situation.

They are lists of services that the system should provide for, describing the functions in detail, its input and its output, with any exceptions and so on.

The specifications of a functional requirement must be:

- **Complete:** because it represents all the services required by users;
- **Consistent:** It indicates that the requirements should not have contradictory definitions.

2.2.4 Non-Functional requirement

The non-functional requirements, as the word suggest, describe something that is not correlated with the specific service that is developed, but still represent a part of basic structure that makes system suitable.

The Non-Functional Requirements define the attributes of the user and the system environment, and give the standards that system must conform to. It contains attributes that define system's functionality regarding use and describes the properties of the software system in relation to certain services or functions, giving constraints on the development of the system.

A non-functional requirement has a critical level, greater than an individual functional requirement; in fact, non-functional requirement may affect the entire system structure / architecture. Each non-functional requirement may generate other requirements (functional) that can improve or provide better development of the service. As it is possible to see in the Figure 6, non-functional requirements can be characterized and divided into subgroups that give a better or more detailed classification. As said before, the implementation of the non-functional requirements is related to the entire system

and this may lead needs to create new functional requirements that can define new system services or even to restrict existing ones.

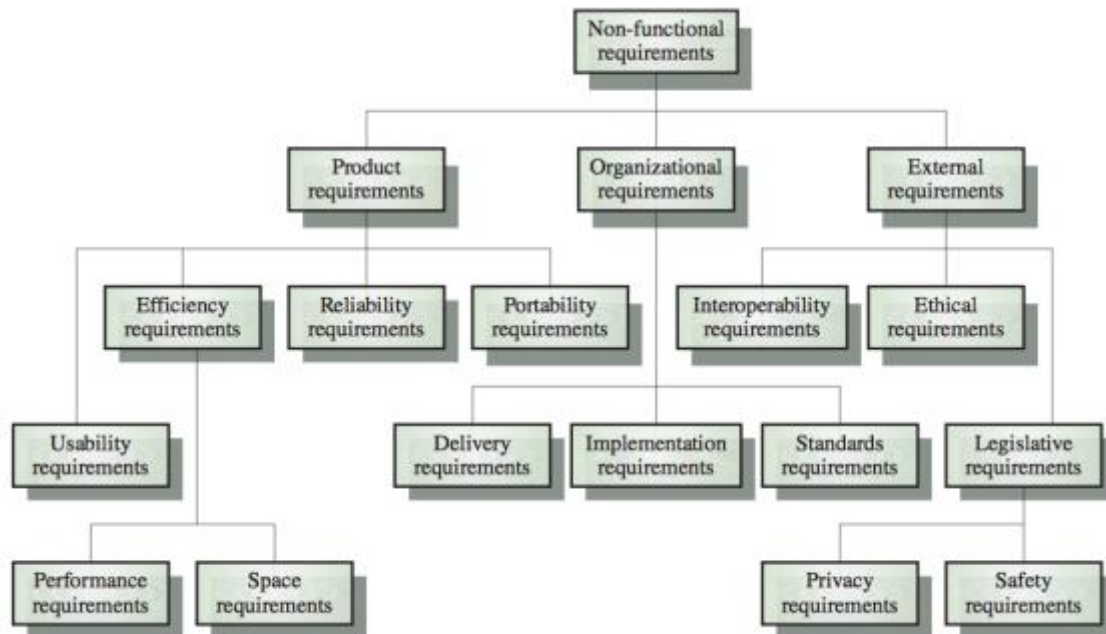


Figure 6 Different types of Non-Functional Requirements

Functional and non-functional requirements are the conjunction between the user and the provider of the service. They represent what the user wants and what the provider has to provide for the execution of the service for fulfil the expectation of the client.

Non-functional requirements of business process and non-functional specification were determined from the Cloud Service Level Agreement Standardisation Guidelines (Van der Wees Arthur et al., 2014)

2.2.4.1 Service Level Agreement for Cloud Service

Service level agreement is a document that defines the level and the minimum standard that the service can provide and that the supplier is obliged to ensure.

Service Level Agreement (SLA) describes a set of non-functional requirements of the service the customer is buying. In the SLA, the mode of delivery or provision of the service are not define (Kolisko, 2009).

The definitions of the service level should be specific and measurable in every area. In this way, it is possible to assess the quality of the service and, if required by the contract, reward or penalize accordingly.

2.2.4.2 Service Level Objective for Cloud Services

Service level objective is the metric to define SLA, a way of defining the service's performance. The difference with SLA is that this latter represents the entire agreement of the service such as, costs, location, performance and everything that characterizes the operation of the service requested by the customer. On the contrary, a SLO describes the same service but in a measurable way. In every single SLA, it is possible to find many SLO. It is the qualitative way to offer a customer service.

In this master thesis, many issues are taken into consideration: performance, security, data management, personal data protection and payments. All these categories have the subset that describes the features and performance in details. The next chapters will show a detailed and practical aspect of the business process taken into consideration.

The next chapter in fact, shows how to proceed with the annotation of business functional and non-functional requirements and cloud service functional and non-functional specifications.

2.3 Business Annotation

Both the APQC framework and the Object / Action characterization for the tasks that are used in the business process to annotate the functional requirement. The following chapter will specifically analyse these annotation models.

2.3.1 APQC

APQC (American Productivity & Quality Center) is an organization founded in 1977, headquartered in Houston. In 1992, they created a process classification model called PCF; it contains 5 levels: - Category, - Process Group, - Process, - Activity, - Task. The goal is to enable comparisons (benchmarking) based on categories and homogeneous processes (standard). The PCF structure operational processes and management are divided into 13 categories, which are structured into process groups and subsequently in over 1000 processes.

The APQC model is an internationally recognized resource for process improvement and performance, creating a common language for companies to communicate and define work processes to track them and compare them according to aspects performance. (APQC, 2017)

APQC standard is used in this study for the classification of the business process in the business layer, as it is possible to see has shown in the next paragraph: for the workflow layer it is used another standard that fits the level.

In this thesis, the business process in question (**Send Invoice**) refers to the **9.2.2 standard, but it is possible to find other standards from other sub-categories**. In the following chapters, there will be the presentation of the differentiation and specification of each standard.

The figure below shows the main 13 categories and the distinction between “operating process” and “management and support services”.



Figure 7 Categories of APQC process classification framework

2.3.2 Object / Action

According to Silver, (2011) it is possible to derive a specification from the verb-noun form to action-object form, where verb corresponds to the action and the noun to the object. That specification with the APQC classification provides the meaning of business process functionalities.

As Roth, Diamantopoulos, Klein, & Symeonidis, (2014) explained, the requirements are often written in natural language, and can run into ambiguity. For this reason, it was conducted a study on how to map them as requirements from formal representation to semantic parsing task.

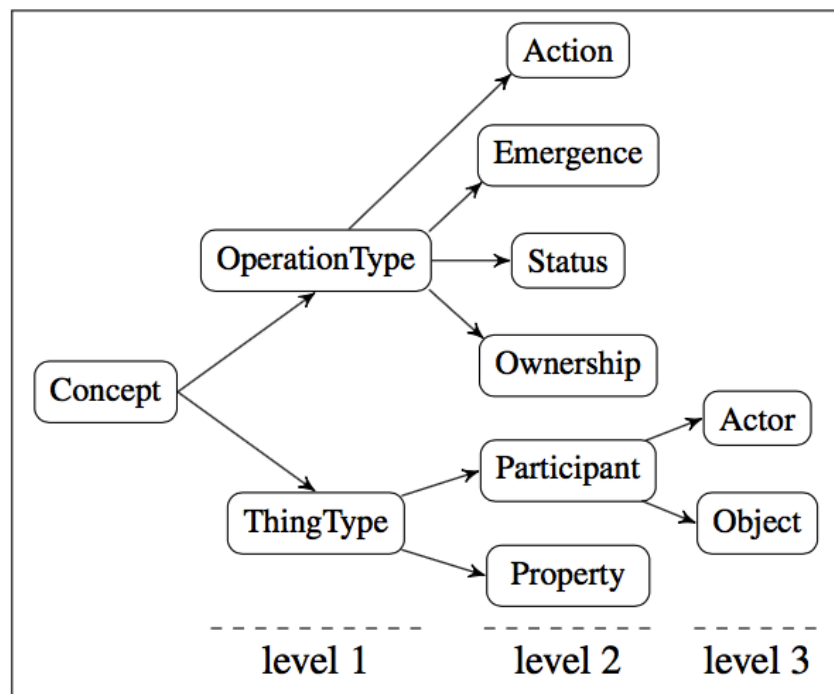


Figure 8 Class hierarchy for modelling requirements

Starting from the highest level of classification it is possible to distinguish "operations" and "things" types, which contain, respectively, for "OperationType" action and object for "ThingType".

Another approach is presented by van Welie, van der Veer, & Eliëns, (1998) which suggest developing an ontology that explicates the relationship of each task. In this ontology, the object is not important whether it is real or non-physical entity. It is

composed of agents, roles, object, task and event. Actions or verbs are used to connect the various parts of the ontology, as it is possible to see in the image below.

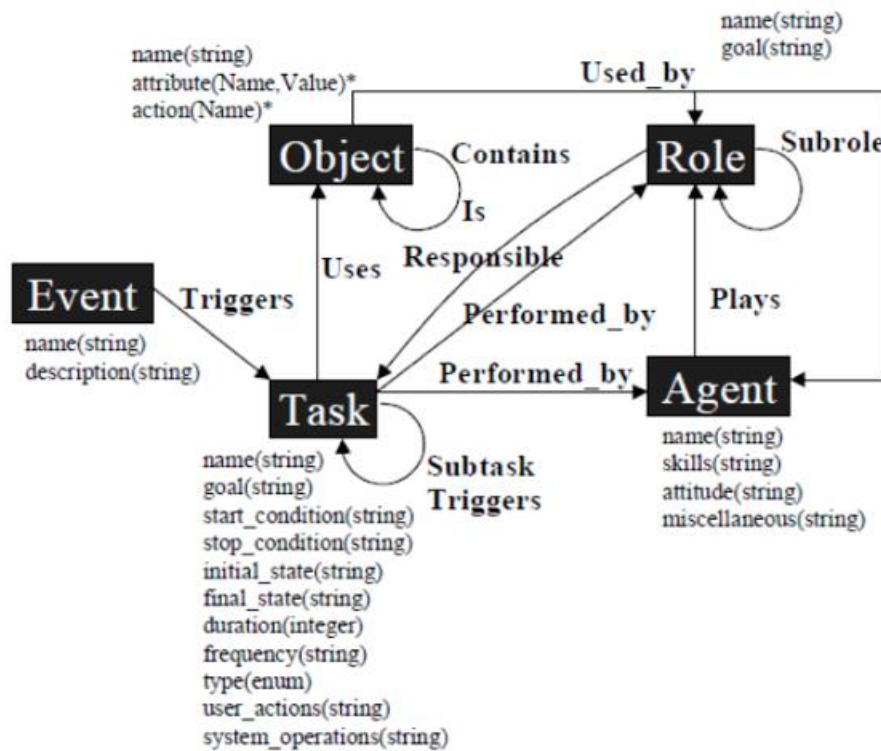


Figure 9 Welie's approach to the ontology

2.4 IT annotation

As for the business, the IT needs to annotate the functional specifications of cloud service through the use of a suitable language for web services. In the following chapters, it will be shown the presentation of the web service and the way for the annotations by this method.

2.4.1 Web Services

The definition of web service for Haas & Brown, (2004) is: “A *Web service* is a software system designed to support interoperable machine-to-machine interaction over a network.”

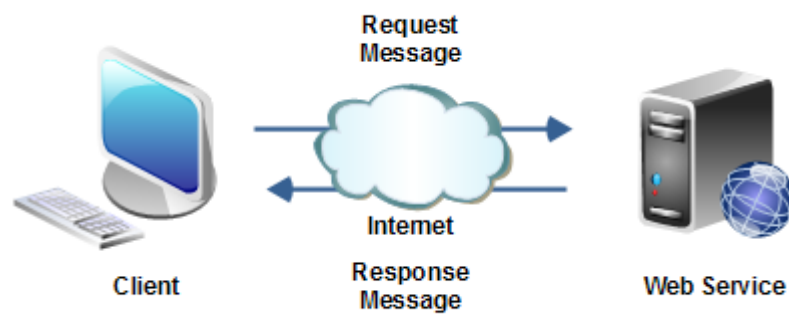


Figure 10 Communication through a Web Service

As Potti (2011) said “web services can be described simply as any service offered over the web.”

The function of the web service is the connection of software programs to each other on the internet; this give the opportunity to share big quantity of data without losing efficiency and with low-cost in term of use. It is a category of web solution which is used in every industry field.

A web service is made up of a *client*, that can have just minimum or even no coding, that is the real service and it is *software* inside the platform (computer, smartphone, etc.), with network access, and a *server* that contains data types, operations and all the information for the localization of the network.

“Web services work at a level of abstraction similar to the Internet and are capable of bridging any operating system, hardware platform, or programming language, just as the Web is”(Aldea, Sangeorzan, & Aldea, 2009).

The Service Oriented Architecture (SOA) is the bases for the creation of the web services, giving the possibility to modify, in very simple way, the methods of interaction between the services, or the combination in which the services are used in the process. It is also easier to add new or modify services depending on the business needs. It can be considered as a component of a broader process that can be reused or changed.

“The advent of web services and SOA offers potential for lower integration costs and greater flexibility.” (Qusay H. Mahmoud, 2005)

The basic protocol for the web services is the HTTP: it is responsible for the connection with the application that intends to make use of its function. In addition to HTTP, web services use many other web standards, based on XML (Extensible Mark-up Language). The XML message is received through the web service that can make it understandable for the interfacing systems.

In order to analyse the most common ways to develop a web service, there are two main methods: the Simple Object Access Protocol (SOAP) and the Representational State Transfer (REST). Both are methods for building a web service with the difference that SOAP is a protocol, whereas REST an architecture: so the analyses over the developing of web services will focus on the performance of the two technologies rather than on comparing there structure.

2.4.1.1 Simple Object Access Protocol (SOAP)

It was designed in 1998 for Microsoft so it has a long history and well defined strong structure. The word Object indicates that the use of the protocol must be done according to the paradigm of object-oriented programming. The protocol defines a set of rules that the client must adhere to for the Web service. Through the exchange of SOAP messages (Remote Procedure Call), a local software component performs an operation through a remote system (the web service).

A SOAP message is structured by a header and a body.

- The header segment is optional and contains meta-information such as routing, security, transactions and parameters required by a procedure.
- The segment body delivers the information content (payload). This must follow a defined XML schema diagram.

2.4.1.2 Representational State Transfer (REST)

REST: it is not a standard but uses HTTP, URI, XML, HTML, GIF, JPEG as standards.

REST is based on the identification of resources explicating the use of HTTP methods with self-descriptive linked resources. Each service is represented in terms of resources, that has a URI (Unique Resource Identifier)

The URI must be self-explanatory, to make clear to understand what the resources do. After finding a way to identify the resources, it is necessary to define what operations can be performed on them.

REST uses the typical CRUD operations (create, read, update, delete) and HTTP methods:

- POST -> Create (Creates a new resource)
- GET -> Read (Gets an existing resource)
- PUT -> Update (Updates a resource or changes its status)
- DELETE -> Delete (Deletes a resource)

REST does not explicit a way to describe how to interact with a resource. The operations are implicit in the HTTP protocol. The REST approach tends to preserve and enhance the characteristics of the Web, highlighting the predisposition to be a platform for distributed processing. So, it is not necessary to add anything to what already exists on the Web, in order to allow remote applications to interact. The table below shows shortly in bullet point, the main characteristics, pros and cons of these two methods.

	REST	SOAP
Characteristics	<ul style="list-style-type: none"> • Operation are defined in the messages • Exposes resources which represent data • Unique address for every process instance • Each object supports the defined (standard) operations • Loose coupling of components 	<ul style="list-style-type: none"> • Operations are defined as WSDL ports • Exposes operations which represent logic • Unique address for every operation • Multiple process instances share the same operation • Tight coupling of components
Self-declared advantages	<ul style="list-style-type: none"> • Late binding is possible 	<ul style="list-style-type: none"> • Debugging is possible

	<ul style="list-style-type: none"> • Multiple data formats (XML, JSON, ...) • Process instances are created explicitly • Client needs no routing information beyond the initial process factory URI • Client can have one generic listener interface for notifications • Flexible (Architecture not require processing) 	<ul style="list-style-type: none"> • Increased privacy • Complex operations can be hidden behind facade • Wrapping existing APIs is straightforward • Distributed Enterprise Environments • Long term use benefits
Possible Disadvantages	<ul style="list-style-type: none"> • Large number of objects • Managing the URI namespace can become cumbersome • Point-to-point communication model over HTTP • Difficult to build the client side • Lack of standards support for security 	<ul style="list-style-type: none"> • Old • Client needs dedicated ports for different types of notification • Process instances are created implicitly • Strong typing • Client needs to know operations and their semantics beforehand

Table 1 Main characteristics and differences between SOAP and REST methodology

As showed in the description above, the two methods are quite similar but REST results most appropriate for this use.

This choice is also supported by the utilization of the BPaaS model. In fact, RESTful services could serve as enabler to connect software components with each other as shown in Figure 11 (Barton & Seel, 2013).

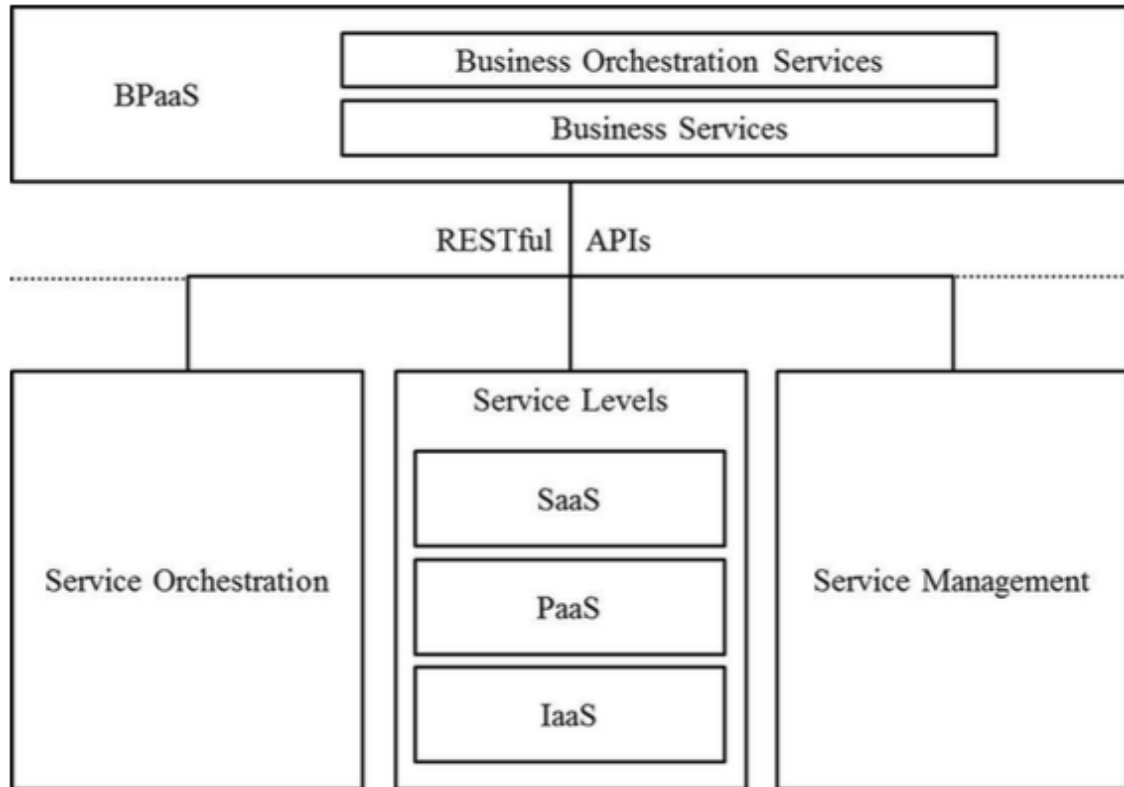


Figure 11 RESTful APIs in BPaaS architecture

2.5 Business-IT alignment

In this context, most of the authors refer only to the service discovery, where a service is found on the basis of the specifications.

As already said in chapter 2.2.1, requirements elicitation is a main activity in any requirements engineering process (Sommerville & Sawyer, 1997). Many techniques and methods were developed and used in real environment for the identification of the requirements (Zowghi & Coulin, 2005).

The importance of the functional and non-functional requirement description is clearly indicated in the findings of the literature analysis. There are plenty of ways for the elicitation of the requirements; for example Nepal, Zhang, Ranjan, Haller, & Georgakopoulos (2012) develop a discovery services system based on requirements.

The Cloud Computing Ontology (CoCoOn) that defines functional and non-functional concepts, attributes and relations of service infrastructure. The researchers proposed an ontology classification, bases on IaaS layer, divided into three categories: computing, network, storage; and each one has a subclass and proprieties that are defined on the cloud services IaaS's layer. Ruiz-Alvarez & Humphrey (2011) proposed a different way for choosing the cloud services, basing on using semi-formal language to express a description of the capabilities and requirements, with the use of XML schema that evaluates cloud services and chooses the one that meets the user's requirements in the best way. Another approach, proposed by Kang & Sim (2010), is the use of a cloud search engine that is specialized to discover cloud service. "Cloudle", the name of that engine, supports matching algorithm for functional, technical and budgetary requirements. Customers insert queries for cloud service using the web interface and receive results with similarities between customer's requirements and providers' functional and technical specification of services.

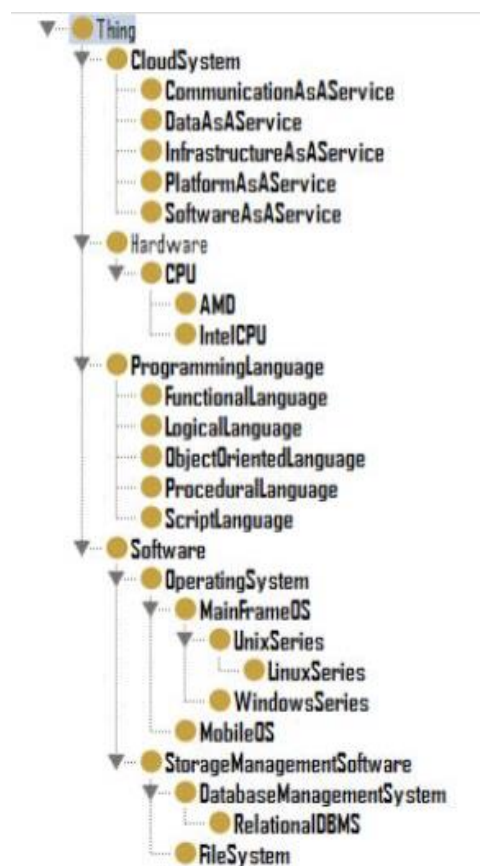


Figure 12 Example of cloud ontology derived from Cloudle

The researches for the matching algorithms conducted in the present study cover both technical aspects and functional requirements based on the need of the customers come out after interviewing SMEs. The knowledge and the information taken in business model help to find and to identify the user requirements for the process in order to execute the business process (Aysolmaz & Demirors, 2014).

The presented approaches are restricted to the IT perspective, which requires a high understanding of IT in order to retrieve service that matches the final user's need. The approach with the ontology is limited to the Business side, and moreover it covers all the requirement level specifications for the classification of the services.

The machine interpretable semantics that the ontology provides to a conceptualization, makes it a highly valuable approach for the present work. An ontology that covers both business process requirements and cloud service specifications can be used for the matching of the two, and therefore it contributes to close the gap between Business and IT in the cloud.

The following section, explains what an ontology is and also the benefits from the use of it.

2.6 Ontology

Analysing literatures, what emerges is how ontologies can be used for the representation of almost every possible scenario, from finite elements (Freitas, Asooja, Jares, Decker, & Sahay, 2014) to the integration of data for the semantic web (Schenk & Staab, 2008). Ontology could be defined as a representation to structure knowledge.

An ontology can be seen as a data model that represents knowledge as a set of concepts within a domain and the relationship between these concepts (Boyce & Pahl, 2007). An ontology is a form of knowledge management that captures the knowledge within an organization as a model; this model can be queried by users to answer complex questions and display relationships across an enterprise. Studer, Benjamins, & Fensel (1998) explained the ontology as: *“conceptualization refers to an abstract model of some phenomena in the world by having identified the relevant concepts of that phenomenon. Explicit means for the type of concepts used and the constraints on their use are explicitly defined. Formal refers to the fact that ontologies should be machine-*

readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual but accepted by a group”.

Ontologies are made up of:

- **Classes:** Collection of instances with similar properties, it defines entity relevant in the context.
- **Properties:** Relations between classes. Both classes and properties can be hierarchically organized in taxonomies. It also gives data values such as integer numbers, strings, Boolean, dates and other types.
- **Instances:** Concrete object generated from the shape of classes. They represent the data of the model.
- **Rules:** Allow to infer implicit information combined with reasoning.
- **Axioms:** Allow to capture additional information about classes and properties.

Ontology is an important and useful tool, it reduces or eliminates the conceptual and terminological mistakes and help to come into a shared understanding (Uschold, Gruninger, Uschold, & Gruninger, 1996), providing:

- **Better communication:** Users and software have a unified understanding, ontologies enable communication between people with different needs and viewpoints.
- **High degree of Inter-operability:** Ontology is an extremely compatible system, any information technology environment should integrate activities, resources, organization, goals and so on.

- **Support for system engineering:** Ontology could support design and development of software systems. It facilitates the understanding of the relationship among the components of the system, especially for designers working in different domains and fields.
- **Reusability:** Ontology's libraries and components are easy to reuse in a different domain, so that it is possible to import and export modules among different software systems.
- **Flexibility:** Ontologies can be extendible, allowing the incorporation of new classes and a deep specialisation of concepts.

As mentioned, ontologies "work" through rules: the use of semantic rules it is possible to create new knowledge. The use of rules and the application on the ontologies are useful to see how ontologies work, which will be particularly seen next chapters, where rules will be re-used for creating the mapping between the business and IT level using the ontologies.

The ontologies, to express a concrete notation, need a language that must extend existing web standards to simplify their use, easy to understand and use, specified in a formal way and must have an appropriate expressive power to describe the domain. There are different languages, proprietary or standards-based, for the definition of ontologies, the ones used in this case are:

- **RDF (Resource Description Framework)** is a framework for the description of knowledge on the web. This framework is the basis of the semantic web, and allows the sharing of knowledge on the web. The RDF data model is formed by resources, properties and values. The properties are the relationships that bind together resources and values, and are also identified by URI (Universal Resource Identifier). The basic unit for representing RDF information is the statement. A statement is a collection of Subject - Predicate - Object, where the subject is a resource, the predicate is a property, and the object has a value. The RDF data model is used to define a simple model to describe the relationships between resources, in terms of properties identified by a name and its values, but does not provide any mechanism for declaring these properties, nor to define the relationships between these properties and other resources (W3C, 2014).

- OWL is developed as a next step of RDF; it is a language for defining structured, web-based ontologies which enable greater integration and interoperability of data between applications. OWL allows to perform a range of descriptive applications such as managing web portals, collections management, content-based searches, enabling intelligent agents and web services. OWL uses URI to identify a resource, and the linking provided by RDF adding redistributive capacity between multiple systems, compatibility and scalability for web standards, regarding accessibility and internationalization, making the language more extensible, adding a larger vocabulary for describing properties and classes (W3C, 2012).

2.7 Conclusion and Research Gap

The literature review tries to give a first approach to the covered topics, giving a definition and a preliminary analysis of these. It has revealed the different types of currently existing models and structures, which can be used to close the gap between the business and IT layer. Through this analysis problems for classification and for the proper language and standards to define some topics have arisen.

Through this literature, it has been possible to find all the components that are part of aligning between business and IT.

This thesis tries to close the gap between the business and IT introducing a new conceptualization for the alignment of the requirement, trying to give the correct standards for the classification of these. This analysis and alignment work can be useful for companies, especially in small and medium enterprises (SMEs), to have a benchmark with the companies present in the market and be able to keep up with new technologies, improving profitability and performance.

3. Research Design and Methodology

This chapter describes the approaches and strategies used for the research; it shows the research method applied and the philosophies in this thesis.

The research design is the plan and the procedures followed for the search, a kind of map to define the structure and comprehensibility of the thesis.

It helps to create a solid basis for all research giving reliability regards to the results. Planning helps the organization of ideas and content, avoiding mistakes and improving the end result. The methodology is the approach adopted to achieve the objective pursued and answers the research questions.

Research design means the plan or the technique of modelling research. The success of the research study is correlated with the correct and defined structure of research, finding the right goals and justifications. According to Creswell (2013), the design research is defined as "*the plans and the procedures for research that span the decision from broad assumption to detailed methods of data collection and analysis*". This quote, provides for many issues that are possible also to be found in the research onion.

The research onion is used to describe the steps of the present research. As the name suggests, it is a “multi-layer diagram” that helps to develop the research. In order to analyse the research onion starting from the external layer, it is possible to find the Philosophical Prospective, after the Approach, Strategies, Choices, Time Horizon and then reaches the most internal layer, the one of the Techniques and Procedures for the Data Collection and Data Analysis. Figure 13 shows the structure and all the layers with the possible methods for the analysis.

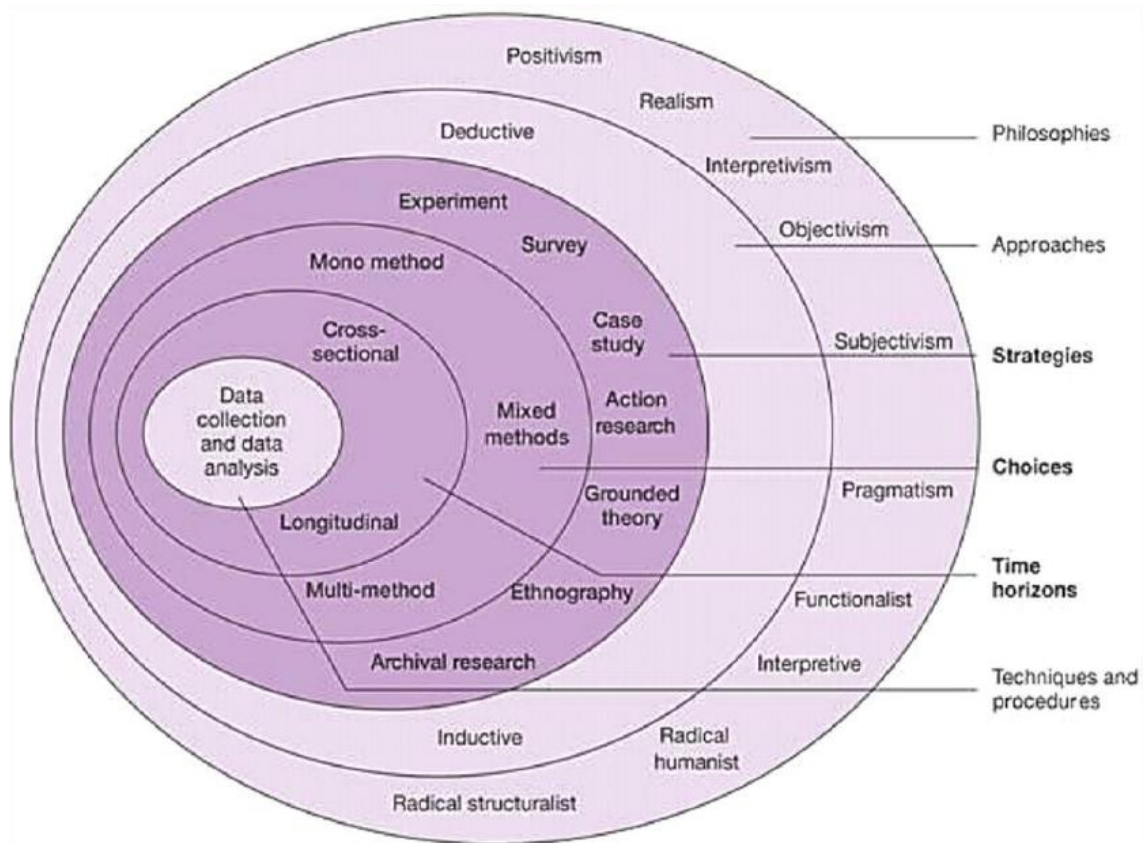


Figure 13 Research onion according to Creswell

3.1 Research Philosophy

In this section, it is possible to see the different research philosophy, with the description of selected philosophy related to this thesis.

According to Vaishnavi & Kuechler (2008) research philosophy is “*an activity that contributes to the understanding of a phenomenon*”. It gives a view regarding how the phenomenon should be collected, analysed and used. According to M. Saunders, Lewis, & Thornhill (2009) there are three main ways of thinking about research philosophy:

- Ontological assumptions: It concerns with the nature reality or being. It is related on how the events are related in the world and what they do.
- Epistemological assumptions: It concerns in the exploration of the nature of the knowledge (Vaishnavi & Kuechler, 2004); it is the researcher’s point of view of and highlights what can be considered acceptable in the study.
- Axiological assumptions: It concerns in the study and the role of values, giving the added value of truth or understanding.

However Vaishnavi & Kuechler (2008) include also another paradigm:

- **Methodological assumptions:** It means the work with methods for the research in order to obtain knowledge with solid basis and relevant subject.

Basic Belief	Design Research
Ontology	Multiple, contextually situated alternative world-states. Socio-technologically enabled
Epistemology	Knowing through making: objectively constrained construction within a context. Iterative circumscription reveals meaning.
Axiology	Developmental. Measure artefactual impacts on the composite system.
Methodology	Control; creation; progress (i.e. improvement); understanding

Table 2 Philosophical Assumption of the Three Research Perspectives according to (Vaishnavi & Kuechler, 2004)

Analysing layer-by-layer the structure of the onion starting with the external one, here are main philosophical assumptions that characterize the research:

- *Positivism:* Positivism researches use only phenomena that exist, some that are possible to measure, to test or that derive from existing theory to create credible data. In this approach the main assumption will be, depending on the result, confirmed (whole or part) or refuted for leading to further development (M. Saunders et al., 2009).
- *Realism:* It is similar to positivism; it requires a scientific approach to the development of knowledge. Realism assumes that reality exists independent of our mind and it is perceived by the senses. There are two types of realism:
 - *Direct:* everything perceived with senses accurately describes the world.
 - *Critic:* All that is perceived by the senses is reworked in the mind which gives

a certain “vision” of the world, sensations that reflect reality but are not the real world (M. Saunders et al., 2009).

- *Interpretivism*: Research is with empathetic stance not law-like as the positivism generalization. The point of view is from the research subjects, and it is necessary to understand the differences between humans and their role as social actors (M. Saunders et al., 2009).
- *Pragmatism*: The adoption of epistemology, ontology and axiology, are important and determinant to answer the research question. One approach can be “better” than another, but if the research question results ambiguous between positivism or interpretivist philosophy, it is possible from the point of view of the pragmatism to work with both (M. Saunders et al., 2009).
- *Design Science Research*: It is similar in some aspect to the pragmatism. According to Hevner & Chatterjee (2010) it is defined as: “*a research paradigm in which a designer answers questions relevant to human problems via the creation of innovative artifacts, thereby contributing new knowledge to the body of scientific evidence. The designed artifacts are both useful and fundamental in understanding that problem.*”

The strategy selected for this study is Design Science Research. Figure 16 will show the research cycle that begins with all the steps needed for the research and the outputs that come out from each level. This research is based on Design Science Research Framework adopted from the General Design Cycle by Vaishnavi & Kuechler (2008).

3.2 Research Approaches

The second layer of the research regards the approach to the research. There are two possibilities:

- *Deductive*: The approach goes from general to specific. First of all, there is the development of a theory, based on hypothesis derived by previous findings or literature, after that the observation and the test give confirmation or rejection of the hypothesis.

- *Inductive*: On the contrary, the inductive approach starts from the observation of the phenomena and after finding patterns and creating hypothesis out of it, the realization of a theory occurs.

Figure 14 shows the two approaches and their similitudes in the workflows.

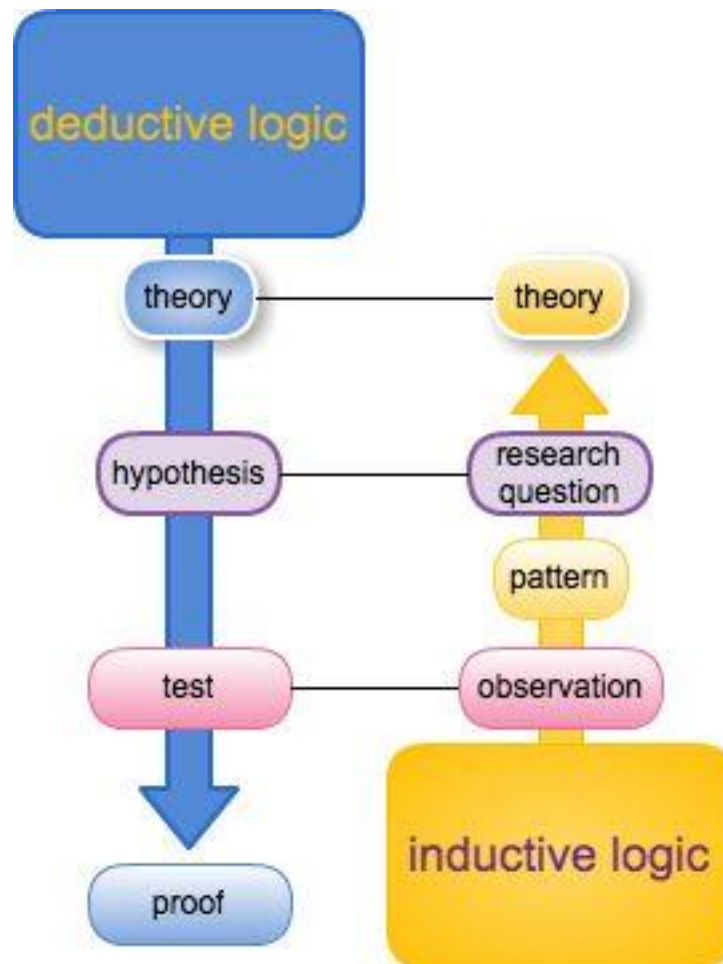


Figure 14 Deductive and Inductive approach

The best suitable research approach for that master thesis is inductive. This approach fits best because the creation of the artefact comes out from the observation of a real problem.

3.3 Research Strategy

Moving through the research onion it is possible to find the research strategy layer. There are different strategies available for the research:

- **Design Science Research:** Development of a new artefact address to a specific problem with the aim to produce new knowledge on that specific field. It follows a specific research framework that is composed with the following step:
 - **Awareness of problem**
 - **Suggestion**
 - **Development**
 - **Evaluation**
 - **Conclusion**

The analysis of the use and of the performance gives a better understanding and improves the artefact itself.

- **Action Research:** There are some similitudes between Design Research and Action research as Hevner & Chatterjee (2010) showed. In this typology of research the participant has to solve a problem or to achieve an objective.
- **Case Studies:** Starts from a single specific case in a strong structure way and draws generalizations. These methods test hypothesis and re-use them when exhaustive knowledge from a particular case is needed. There is a data collection that can involve interviews, observation, documentation and questionnaires.
- **Survey:** The studies use deductive approach and the gathered data can be analysed empirically. It is used in particular for the quantitative research project also because used to examine causative variables between different types of data.
- **Experiments:** It can be used in all area of research; it is analysed the relation between two or more variables (limited number), answering the “how” and “why” questions. The factors are analysed and judged between the expectation and the real outcomes.
- **Ethnography:** it is closely related to the inductive approach. Involves the close observation of a group of people in social world to understand the functions in its natural setting.
- **Archival research:** The researcher uses existing materials from other researches to sum up all the findings to obtain the sum of knowledge for a specific case study.

- **Grounded theory:** Through an inductive approach patterns are derived from the data as a precondition for the study (May, 2011).

3.4 Research Choice

The fourth layer of the research onion refers to the choices. According to M. Saunders et al. (2009) there are two ways to combine those data:

- **Quantitative:** questionnaires are used for data collection and graphs or statistics are used for the procedure analysis, the output being numerical data.
- **Qualitative:** interviews are used for data collection and the categorising of data is used for the analysis procedure that at the end generates non-numerical data.

There are also three further different research choice methods:

- **Mono-method**
- **Multi-method**
- **Mixed-method**

Starting from these differentiations, combining methods together it is possible to obtain:

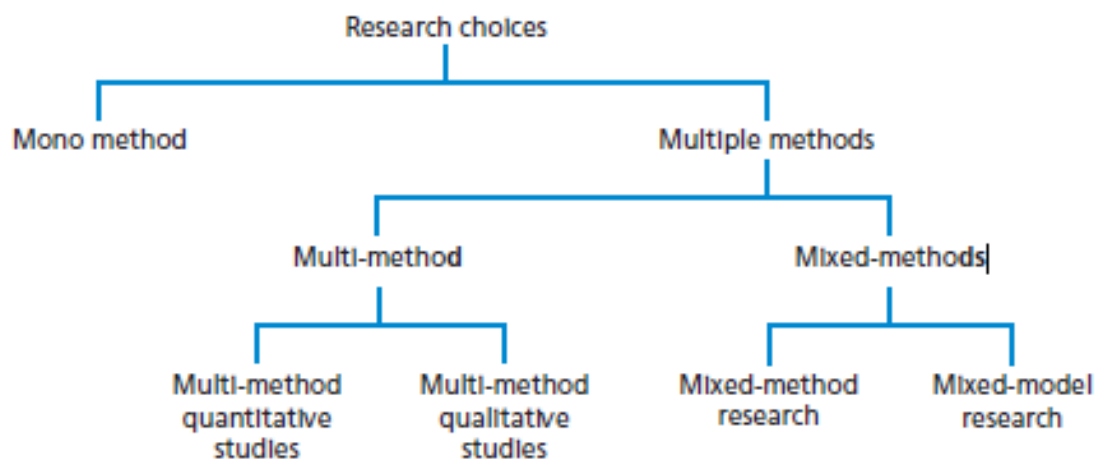


Figure 15 Research choice methods

Mono-method: Single data collection technique with qualitative data collection analysis or quantitative mono-method with related quantitative analysis.

The *multi-method* means the combinations of several data collection technique in order to get analysis data from different sources. It is not possible to mix qualitative and quantitative technique and procedures and so they are divided in:

- *Multi-methods quantitative study*: in which the research use more than one quantitative data collection (questionnaires and structured observation) using statistical procedures.
- *Multi-methods qualitative study*: with the use of qualitative collection data (interviews or diary accounts for example) and the analysis using non-numerical procedures.

For the *mixed-methods*, as the word suggest, it is possible, to mix quantitative and qualitative data collection and analysis procedure at the same time (parallel) or one after the other (sequential), but not combining together. There are two mixed methods that are:

- *Mixed-method research*: where quantitative data are analysed quantitatively and qualitative data are analysed qualitatively.
- *Mixed-model research*: where quantitative and qualitative data collection technique and analysis procedures are combined together.

The multi-methods qualitative is chosen because different methods can be used for different purposes in the study and because the approach is not only a collection of data from theory but also from user part. In this case thought interviews to SMEs, cloud service's provider and workshop with Mathema, a company that operates in information technology.

3.5 Time Horizons

The final layer of the research onion, before reaching the core, is the time horizon where the researcher defines the time of the research. There are two possible horizons:

- *Cross-sectional studies*: research is undertaken to answer a question or address problem at particular time. (B. M. Saunders & Tosey, 2013)

- *Longitudinal studies*: data being collected for an extended period of time. Experiment or data collection need to be re-analysed over time to achieve a final output from the research.

Cross-sectional study is the time horizon for this thesis because the research is trying to answer a specific problem in a specific time.

3.6 Research Method applied in this Thesis

The aim of this thesis is to develop a method for the alignment between the business and the IT layer in cloud computing. For this reason, the focus lays on looking at an existing problem and trying to close the gap between these levels in an innovative way providing a helpful contribution for the future business and performance results.

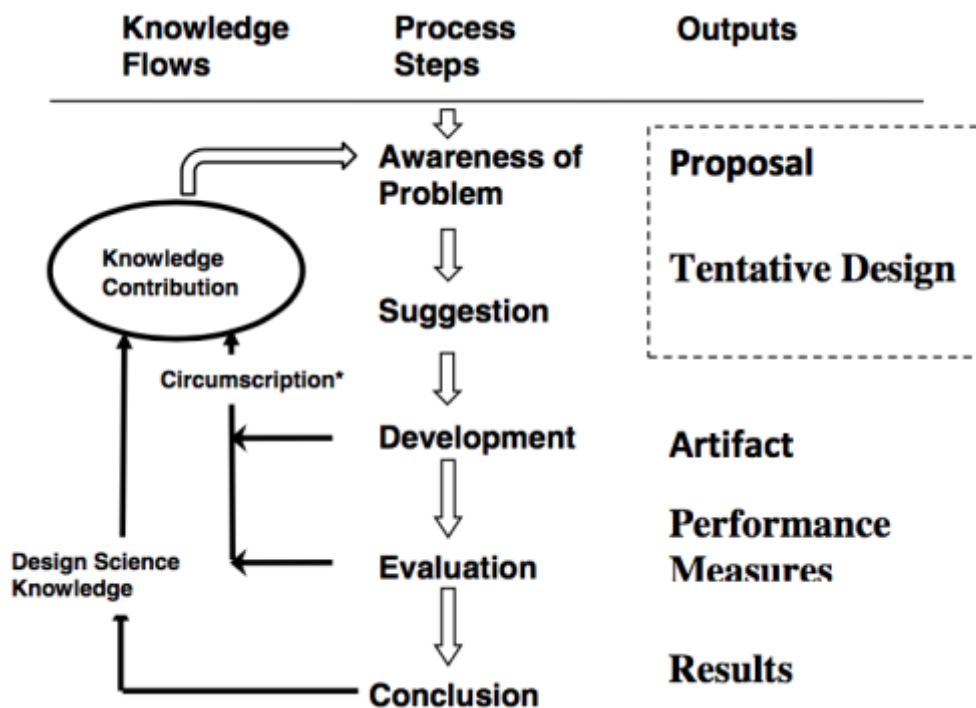


Figure 16 Design Science Research Framework (Vaishnavi & Kuechler, 2008)

Analysing step by step every single component of the framework here are the procedures:

3.6.1 Awareness of problem

This step is triggered by the need of a new approach between the business and the IT layer, in order to close the gap between these two “worlds”, especially for the SMEs

that suffers from this problem, with the resulting loss in terms of performance, time and development compared to large companies. The research will be focusing on the identification of the requirements in both layers and how to align. Inspired by CloudSocket project and additionally making interviews with CloudSocket use case partner and a CRM provider.

3.6.2 Suggestion

The suggestion is derived from the information collected from the awareness phase. The suggestion phase is the “creative” phase where the model for the interconnection between the concepts is created. The specific process, which shall be supported by cloud service, is modelled in Business Process Model and Notation BPMN (Object Management Group (OMG), 2011) using the AdoXX tool (<http://adoxx.org>), a meta modelling development and configuration platform for implementing modelling tools. After choosing the right field and the most appropriate category to gather the tasks from the business process it is possible to create a UML diagram, that shows the connection of the requirements from the business level to the IT level and vice versa and development of rules in natural language that permit to do these matches.

3.6.3 Development

All the requirements are distinguished from business and IT point of view and associated with the “appropriate language”. After that there is the matching part that associates the two layers together for the creation of an artefact that is understandable for business and IT people. In this part, it takes place the development of ontology.

3.6.4 Evaluation

In this part, the created artefact is evaluated; in this case the outputs from the previously steps are analysed. This phase permits to understand the artefact in a real business world from the point of view of business companies and IT experts.

3.6.5 Conclusion

This is the final phase of the research; the results of the evaluation are reported together with starting points for future research.

4. Awareness of the problem

This chapter presents the problem taken into consideration in this research. It is the first phase of the design research process. The following case study will help for a rapid understanding of the problem and to obtain a clear point of view of the current situation. The other subchapters give an analysis and findings from the literature review and an example from an existing real business process taken into analysis. At the end, some consideration and conclusion with respect to the goal of the master thesis are given.

4.1 Introduction

As seen in chapter 1.1 the gap between business and IT especially for small and medium enterprises is a big problem regarding profit, sustainability and development for company. As Mcdavid (2007) said, this integration is necessary because *“modern business could not exist without technology, and complex technology could not exist without modern business”* this means that they depend on each other for a good evolution and benefit.

Factors that increase or create this gap are many, starting from the lack of a way to model the requirements in the most appropriate manner according to the level of association, to the lack of expertise able to model these processes in the correct way, especially in SMEs. The statistics of industry analysis report a higher level of 50% of failure rates for IT initiative; this percentage is unacceptable for many companies, because it produces a huge drain of capital without benefits.

As mentioned also in the previous chapters, the main causes of this gap is the difference between the two levels, mainly due to the lack of a “language” understandable by both levels as Mcdavid (2007) stated. According to him: “business communication tends to be informal, stochastic, and conducted in natural language with a common context assumed by human beings, while IT aims toward the formal, deterministic artificial language communication Within an explicit context of machine processing”. Another cause is the lack of background information: in this specific case, business people come out from business world and IT people from another one, so there is a dearth of knowledge. Big companies or enterprises cover that lack with the adoption of an IT

department, but for the SMEs that have different budgets it is more difficult to bridge this gap and often what they do is not enough to cover that aspect.

Another motivation for this lack of alignment between business and IT levels is the different "speed" of development. In other words, one does not follow the change of the other (for example, technology works, but organizational factors are not aligned with the change or vice-versa), thus failing to take advantage from the change and curbing development. This may happen also because, although dealing with the same subject, there may be different points of view, business take care more about strategy, problems, and external demands, while IT works on platforms, networks, applications, and systems development.

These are just some examples where the gap between business and IT prevents proper development for the SMEs and can fail to deliver expected returns (Hinkelmann & Pasquini, 2014).

This chapter analyses the characteristics, requirements and structure of the process. In the next chapters, one can see how to match business and IT levels is possible. The future implementations that emerge from interviews with cloud brokers, specialists and companies (SMEs) who are facing this digitization process.

4.2 Application Scenario

During a period of study abroad, through an internship for the European project CloudSocket a work is being taken up with the aim, to find functional and non-functional requirements and together with a way to align them in both two layers of business and IT in order to communicate more easily.

This project includes many services; the one analysed in this thesis is the **invoicing service**. The choice fell on this service because it is relevant not only for this case study, but also for many other enterprises, as it is a crucial and necessary step in any profit-making company and in some cases even in non-profit ones.

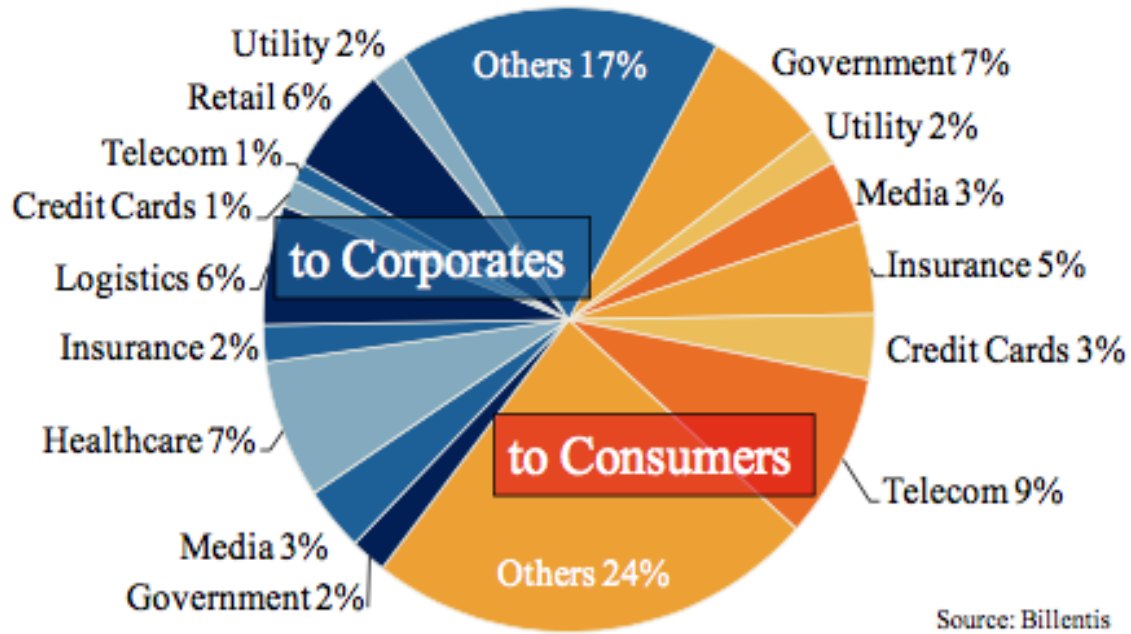


Figure 17 Invoice/Bill volume breakdown by industry

4.2.1 What is an invoice?

An invoice is a list of products or services, given or used by a client and their relative cost. In other words, an invoice is a bill. It is a list of services offered and their price for that performance. Before the advent of computers an invoice was a simple handwritten sheet that had the task of certifying the sale or purchase of a service or product. Thanks to the continuous technological development, also the invoice was updated and has become e-invoice. Gradually there is an approach to replacing these paper-based systems with digital substitutes. Obviously, there are many pros of this new electronic billing method, such as the reduction of process time, costs, and mistakes.

In fact, both parties (supplier and customer) are able to move the invoice directly from one AR or ERP system to another and the information into a standardized form from their portal to buyer's system. Reducing the processing costs of 54%, the possibility of loss or missing invoices of 31%, speeding up the approval cycles of 30%, giving the possibility to focus on higher-value activities, improve cash management, reduce fraud, and improve opportunity for process analytics (PayStream Advisors, 2015).

4.2.2 Evolving market models

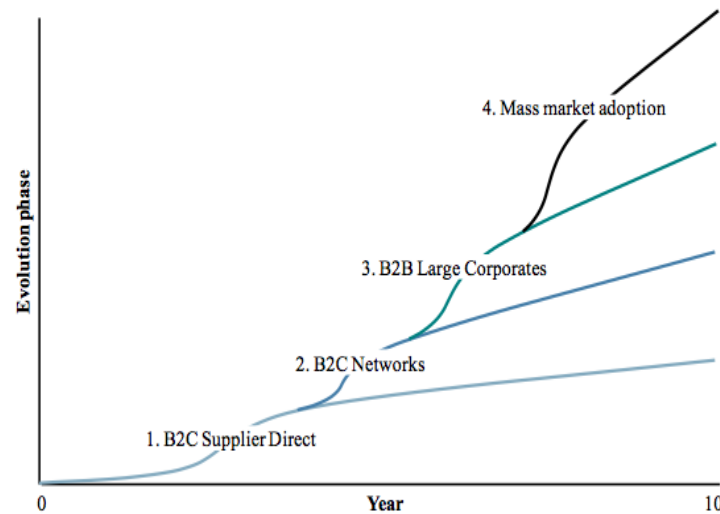


Figure 18 Evolution phases of the market's experiences with e-billing system

In the graph, it is possible to see the evolution of the market and the various types of experiences with e-billing system.

- The first to use this type of service have been the large Organisations (telecoms, sector utility, card issuers, etc.) also offering incentives for the use of this method.
- In the second phase, there was a preference for a single point of contact (aggregating website, online banking) for bills of all their suppliers.
- In the third phase, also enterprises begin to use this service and to benefit of automated processes in their role as issuer as well as recipient. Due to the high benefits for issuers and recipients, e-invoicing in the B2B and B2G segment is typically more successful than in the B2C. However, it is still a challenge for large organisations to push a high number of mid-sized and small trading partners to exchange invoices electronically.
- In the last stage, public sector initiates the breakthrough in the mass market, pushing suppliers to send invoices electronically and modifying the legislation in a user-friendly way, if necessary. In this way, a large number of providers offer solutions for e-invoicing services for the mass market adoption.

The maturity of the market varies according to continents and countries on each continent, as shown in Figure 19.

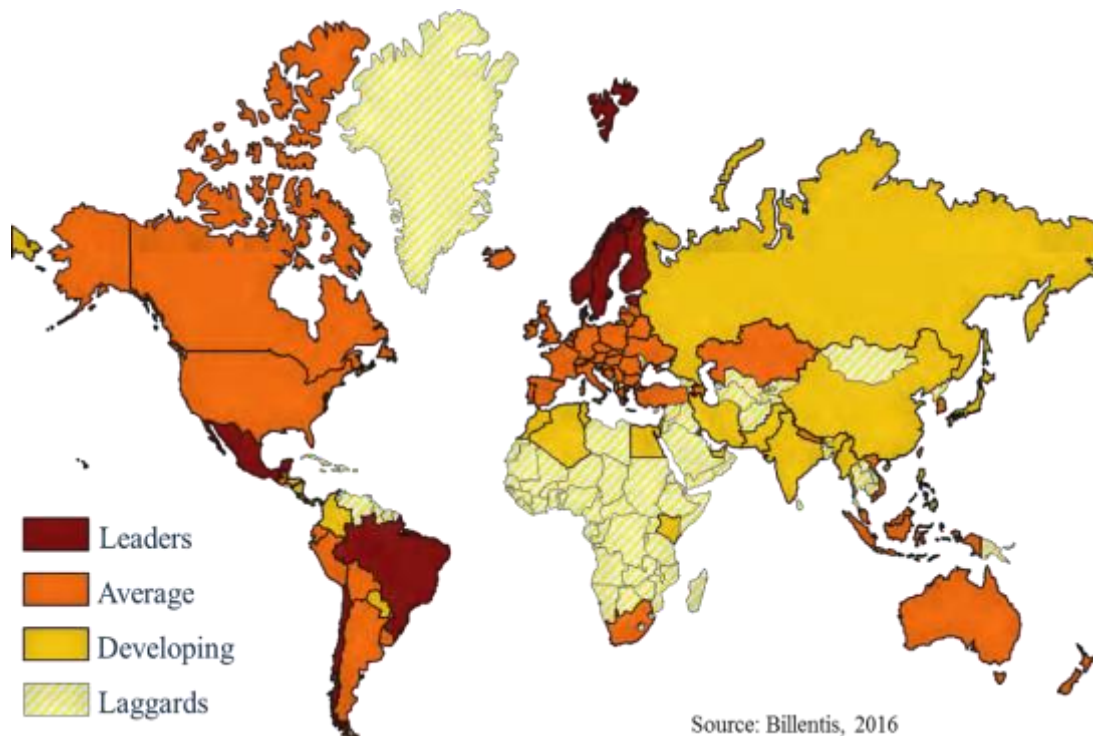


Figure 19 Maturity level of the market in the world

In the 2016 volume for e-bills / e-invoices achieved around 30 billion worldwide with annual growth rates of 10-20%. (Koch & Billentis, 2016)

Below, some examples made by Billentis that show the saving potential with e-invoice system compared with classical paper way.

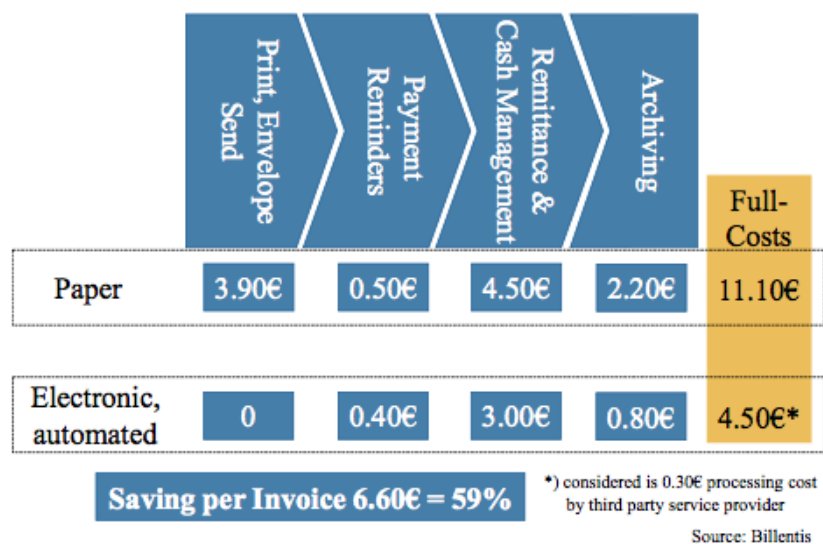


Figure 20 Saving potential for invoice/bill issuers

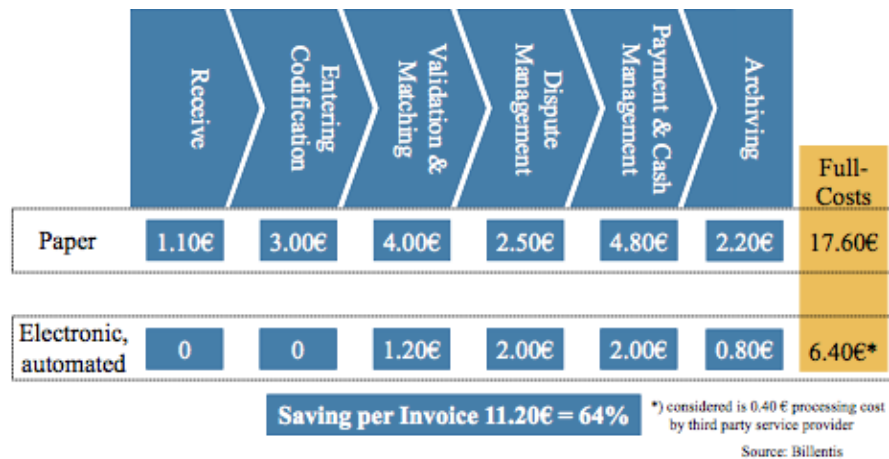


Figure 21 Saving potential for invoice recipients

Having explained what an invoice is and its application in the current business context, further analysis focusses on how the e-invoicing system works with the CloudSocket project.

4.3 Case Study

Literature research and interviews have highlighted that SMEs, in particular, are still lagging in the process of moving from simple invoice to e-invoice. At present, especially small enterprises, use billing spreadsheet application because they do not have a specialized department dedicated to IT or because they have little knowledge on this field that does not allow them to be able to evolve digitally as happens to most large companies.

From interviews with business people from SMEs, emerged the need for a service that is not just a compilation of invoice, in fact to the request to specify what are the key features of e-invoice for this service, the following features have been pointed out:

- Customer management (something that can track the client characteristics and information).
- Ease of use (business people do not have a IT background so there is a need of an easy interface, annotation easy to understand and not in technical language).
- Availability (as a versatility and quickly access in any place without the need to install specific software)
- Customer support
- Security guarantee

- Possibility of personalization (as a possibility to choose customized template, logo, type of payments and so on)
- Integration with other services

These needs have been extrapolated also from phone interview with a CRM consultant for Teamleader, a provider for CRM service, asking what are the main requests from a business user who wants to move from the simple paper or spreadsheet to a more informatics system.

The functionalities derived from the analysis of the interview are the managing of a customer information, the creation or the modification of an invoice and the sending part.

Inspired by a CloudSocket Project workflow and interviews, a business process diagram has been created.

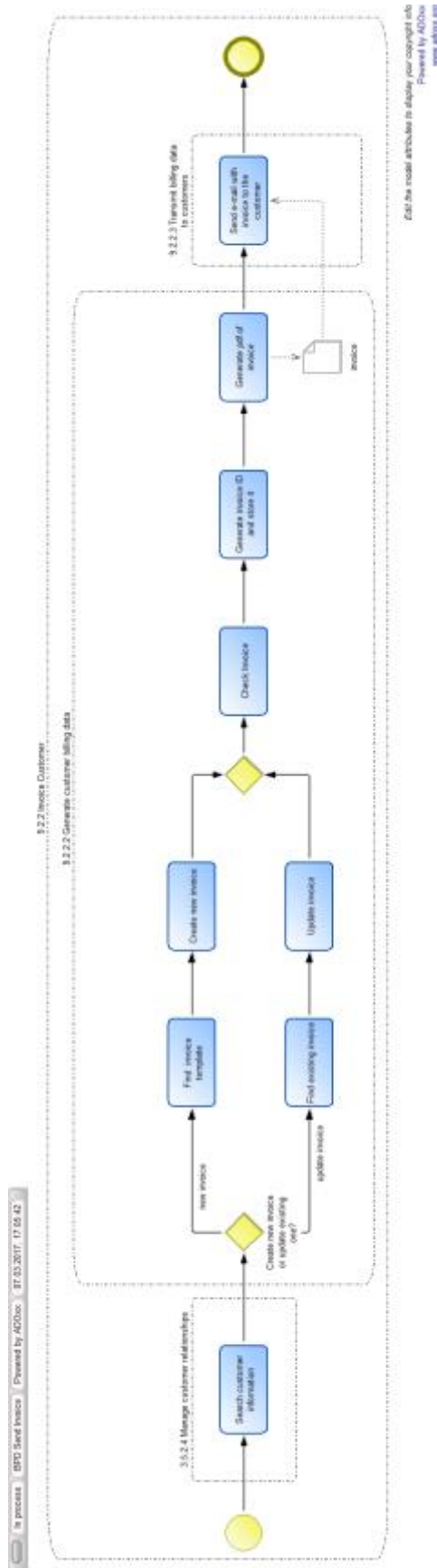


Figure 22 Business Process Diagram Send Invoice

Analysing the business process structure, it is possible to understand the functionality of the service taken under consideration. A business process model can be defined as a structured set of logically related activities and attributes. These activities describe the functionality that identifies the tasks and makes up the model. Once the feature is known, its description can be formulated. The description may be part of the asset or may require other additional information such as inputs / outputs.

Specifically, the process starts with the research of the customer or company information, after that there is a gateway that asks for the creation of a new invoice or the updating of existing one. Depending on the decision chosen there is, in the first case, the selection of template and the creation of the invoice or the selection of existing invoice with the respective update. At the end of this gateway, the task check, as the word suggests, checks the information and after that generates the invoice with a unique ID to store it. The process ends with the generation of an invoice in a PDF file and the sending of this file to the customer.

For the business process modelling it is used the BPMN 2.0 standard from OMG is used. It is possible to see that a group of tasks are grouped in order to specify the requirement. A group can include either a single task or several tasks. For each group there is a business process requirement specification.

In the following workflow model it is possible to see the technical aspect of the process, designed by technical people, in particular, in contrast to business processes, workflow model does not have requirements, but refers to descriptions of service functionalities and non-functional properties (Hinkelmann, Kurjakovic, Lammel, Laurenzi, & Woitsch, 2016).



49

In the workflow, the lanes represent services, because these are the actors in an automated process; they execute the tasks. The modelling language is the same BPMN 2.0 like for the business process, but there are different attributes and references for the workflow elements, which will be analysed in the suggestion phase.

From the workflow diagram, it is possible to see the proper identification of cloud services found for the invoice system after the interviews.

As shown in the workflow in Figure 23 there is not just one cloud service. In this case there are three, because one workflow can contain one or more cloud services, and at the same time cloud services can enclose more workflows.

The cloud services found in the process are:

- Customer Relationship Management (CRM)
- Invoice Service
- Webmail service

The first one is the service for the management of the client, so in the workflow there is the log in into the service task, for looking at the customer's personal information, after that; there is the invoicing system that is more or less similar to the one shown in the business process, where the invoice is filled in with information, and at the end a webmail service is necessary for the sending of the final invoice to the customer.

The interaction between the domain of the business layer (business process) and the workflow that includes the IT aspect is the main problem that emerges when trying to make the two models communicate, the attempt fails due to the different language used. This different "language" spoken by business and IT people is not only found in the functional requirement just shown in business process and workflow, but also in the non-functional requirements.

The Figure 24 will show an example of cloud service's comparison for non-functional specifications. Especially here it is possible to see the difference in the annotation of the attributes that are clear for the IT layer, but sometimes incomprehensible for a business level.

Non-Functional requirements categorie	Annotation Attributes	TMENS CRM	Zoho CRM	Ninja (Invoicing System)	Ninja (e-mail system) In the ontology this	Simple Invoices (Invoicing System)	Mailjet (e-mail system)	Open Source Billing (Invoicing System)	Gmail (e-mail system)
Performance	Availability	100%	100%	100%	100%	99,70%	96%	99%	99,90%
	Response Time in ms	<=1ms	<=5ms	<=5ms	<=5ms	<=1ms	<=1ms	<=2ms	<=1ms
	n. of simultaneous users	500	200	500	500	400	100	200	100
Service Support	data storage	<=10GB	<=5GB	<=5GB	<=5GB	<=10 GB	<=1 GB	<=1GB	<=10
	execution capacity	<=30	<=30	<=30	<=30	<=10	<=20	<=20	<=30
	Support Coverage	twentyfourseven	twentyfourseven	Monday-Friday	Monday-Friday	Monday-Friday	twentyfourseven	mon-fri	twentyfourseven
Data Security	Support Responsiveness	at most 2 hrs	at most 2 hrs	at most 1 working day	at most 1 working day	at most 4 hours	at most 1 working day	at most 1 working day	at most 4 hours
	Backup retention time	up to 1 month	up to 1 month	up to 1 month	up to 1 month	up to 2 months	up to 1 month	up to 2 month	up to half year
	Backup frequency	Daily	Monthly	Monthly	Monthly	Monthly	Daily	Monthly	Daily
Payment	Backup restore time per hour	<=50GB per hour	<=100 GB per hour	<=50GB per hour	<=50GB per hour	<=50GB per hour	<=1 GB per hour	<=5 GB per hour	<=5GB per hour
	Stored Data location	Europe	USA	Italy	Italy	China	Russia	Spain	Switzerland
	Processed Data Location	Europe	America	Europe	Europe	Tailand	Russia	Europe	Switzerland
Logging and Monitoring	Encryption Type	Sha-256	Sha-1	Sha-256	Sha-256	Sha-1	Sha-256	Sha-1	Blowfish
	Payment plan	Prepaid annual plan	Prepaid annual plan	Monthly fee, Free of charge, Prepaid annual plan	Monthly fee, Free of charge, Prepaid annual plan	Monthly fee,	Monthly fee, , Try First	Customizable plan	Monthly fee OR Prepaid annual plan
	Target Market	Businesses AND School and Universities	Businesses AND Individuals	Businesses	Businesses	Businesses	bpaas:Businesses	Universities AND School, Businesses, Individuals, Small Teams	bpaas:Businesses;bpaas:GrowingTeams;bpaas:SchoolsAndUniversities;bpaas:Individuals;bpaas:SmallTeams
Logging and Monitoring	Access log availability	6 months	1 month	2months	2months	1month	1 month	12 months	1 month
	Access Log retention period	6 months	2 month	2months	2months	1 month	2 month	24 year	2 month
	Access Log retention policies	Overwrite events as needed	Overwrite events as needed	Overwrite events as needed	Overwrite events as needed	Archive the log when full	Overwrite events as needed	Archive the log when full AND do not overwrite events	do not overwrite events
	Audit log availability	2months	1 month	2months	2months	1month	1 month	12 months	1 month
	Audit Log retention period	2months	2 month	2months	2months	2 month	2 month	24 months	2 month
Logging and Monitoring	Audit Log retention policies	Overwrite events as needed	Overwrite events as needed	Overwrite events as needed	Overwrite events as needed	Archive the log when full	Overwrite events as needed	Archive the log when full AND do not overwrite events	do not overwrite events

Figure 24 NonFunctionalRequirements cloudservices (CloudSocket Project, 2017)

The analysed process shows that there is a gap between the business and IT level. The discovered cloud services showed how, through the integration of the two levels it is possible to obtain a complete and effective service regarding the business process, especially for finding requirements. However, it has been shown that results that will be achieved only by integrating these two levels, integration that currently does not exist (especially in SMEs).

4.4 Conclusion

This chapter relates to the first awareness phase of the design science research. Here is the collection of the information of the current situation and the gap between business and IT, after the analysis of a real business process. After the reasons for choosing of this subject and description of the case study, it is shown an awareness of the need for a method of linkage between the business and IT layer, especially for the research of functional and non-functional requirements in a business process.

This chapter emphasises the need of a proper way to align the language of business and IT in order to find appropriate cloud services based on the functional and non-functional requirements.

This need has emerged especially after conducting interviews with business people who are facing a process of digitalization, and that would like to close this gap as already happened with big companies, and interviews conducted in IT companies. One of major companies dealing with the closure of the gap is Mathema, an Italian company that has been operating in the information technology field since 1987, expert in supporting small and medium enterprises and that tries to overcome this business - IT communication problem. Thanks to its unique knowledge accumulated in years of work in these fields, this company works hard to find the most appropriate way for put in communication these two different worlds.

The creation of a service that automates these processes, especially for the choice of appropriate cloud service, can be an excellent solution from all points of view, improving communication between the two levels by closing the gap between business and IT, bringing advantages in management, financial and organizational of the entire enterprise.

5. Suggestion phase

Following the approach proposed by Hevner & Chatterjee (2010), the second phase of the design science research is the suggestion as described in chapter 3 of Methodology.

5.1 Introduction

For the choice of the appropriate cloud services a good description of the functional and non-functional specifications for the service is required.

A good style to describe the features in the business process is, according to Silver (2011), the use of verbs and nouns. However, in this way there is not enough information for cloud service matching.

Other methods, as Ukolova (2016) showed in her thesis, are to map the requirements and services with a match one to one; obviously this type of mapping is effective and easy to understand for both of the levels in some services, but the use of different languages in most cases involves misunderstandings.

As seen specifically in the next subchapters, a APQC or Object / Action mappings for a business process cannot be reused in the same way for a workflow. To a different language corresponds a different classification type. In the following chapters, it will be possible to observe the evolution of the mapping process.

5.2 Annotation for the mapping

As Francescomarino, Rospocher, Ghidini, & Valerio (2014) observed, the semantical annotation for the business process can be improved taking elements of the process description and annotation from ontology with the specific process domain.

The use of task ontology can support the discovery of the services. A task ontology contains knowledge to achieve a task, describing specification and relationship where the task is applied and the specific field of environment (Falquet, Métral, Teller, & Tweed, 2011).

The annotation will be divided in:

- Business process functional requirements

- Cloud service functional specifications

5.2.1 Business process functional requirements

As said by Hinkelmann et al. (2016), functional requirements specify what the system has to do, hence the functionality of the task or the group of tasks. The functionalities are specified by attributes, as follows:

- Categorization of the task or group of tasks according to the APQC standards.
- Object and action specification.

Starting with the APQC classification, the example of current task ontologies is the American productivity and quality center process classification framework (APQC). It is made of five levels that start from 13 business process categories, the use of this taxonomy for the business process helps to track and compare benchmarking.

The process classification framework uses structure operational processes and management categories that are in turn structured in process groups of over 1,000 processes and selects the process depending on the granularity of the task or group of tasks.

The following table gives an overview of the main categories of the framework.

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

Table 3 APQC top level categories

Analysing Figure 22 that represents the business process for the invoicing system, it is possible to select the appropriate level on the hierarchy; in the specific case, there is not a single task but groups of tasks. The whole process can be grouped under the hierarchy level 9.0, the “Manage Financial Resources” category, in this case 9.2.2 “Invoice Customer”. Going into details, it is possible to see that inside that category other tasks are grouped. In addition to the 9.2.2.2 “Generate Customer Billing” and 9.2.2.3 “Transmit Billing Data to Customers”, which are subcategories of the main one, there is also 3.5.2.4 “Manage Customer Relationships”, which refers to the task “Search customer information” through the use of CRM service.

In the following table, it is possible to see the specific processes and activities taken by the process group of APQC with its ID, name and description.

Hierarchy ID	Name	Description
3.5.2.4	Manage customer relationships	Managing the organization's relationship with its customers, by systematically coordinating interactions over multiple touch points, on a regular basis. Coordinate the organization's efforts to reach out to its customers. Create and manage effective touch points for interactions from the customers, which could include emails, social-media interactions, newsletters, and direct conversations.
9.2.2	Invoice customer	Preparing detailed reports of customer purchases. Prepare a commercial document between the seller and customer with details

		about transaction. Detail the quantity purchased, price of products/services, date, parties involved, unique invoice number, and tax information.
9.2.2.2	Generate customer billing data	Preparing detailed reports about products purchased by customers. Record and generate a detail account of transactions made by customers fat a particular time and location. Include all details about products such as price, quantity, and name.
9.2.2.3	Transmit billing data to customers	Providing information to customers about purchases made by them. Communicate the details of purchases. Provide customers with a copy of details for their reference.

Table 4 APQC classification relative at the invoice process

The framework creates a common language for organizations in order to communicate and define work processes in an understandable way (APQC, 2017).

As mentioned before, the features are not only specified by APQC categorization, in fact: this method does not provide for an exact and specific definition of its activities, and defines the functionality in specific areas without giving semantic annotation of individual tasks.

Therefore, in additions to APQC the Functional Description Ontology was introduced. Considering what said by Silver (2011), an opportunity has arisen to derive a specification from the verb-noun form to action-object form, where verb correspond to the action and noun to object. That specification with the APQC classification provides the meaning of business process functionalities.

A task ontology hierarchy was created: on the top it is possible to distinguish action (verbs describing the action performed) and objects (nouns of the activities of the business process). Each class has a subclass that always goes into more specific description of the action or process name. Actions or nouns are always part of the same class but more specification and details of the specific topic are given.

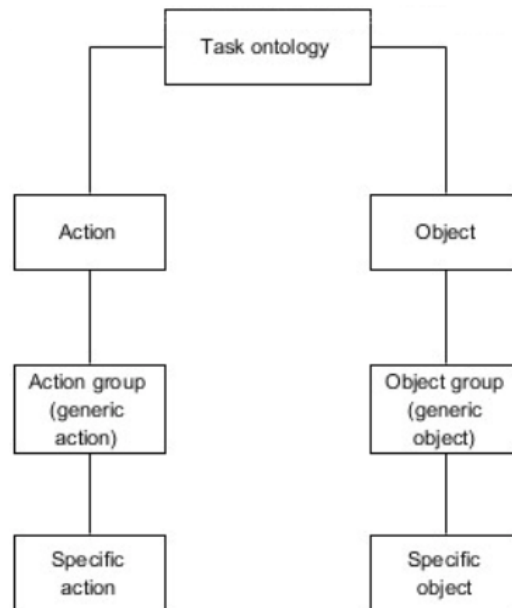


Figure 25 Task ontology for classification of Action and Object

The use of a taxonomy allows to add extra information to the business process or workflow. In this case, the use of this taxonomy can represent a specific domain of APQC ontology.

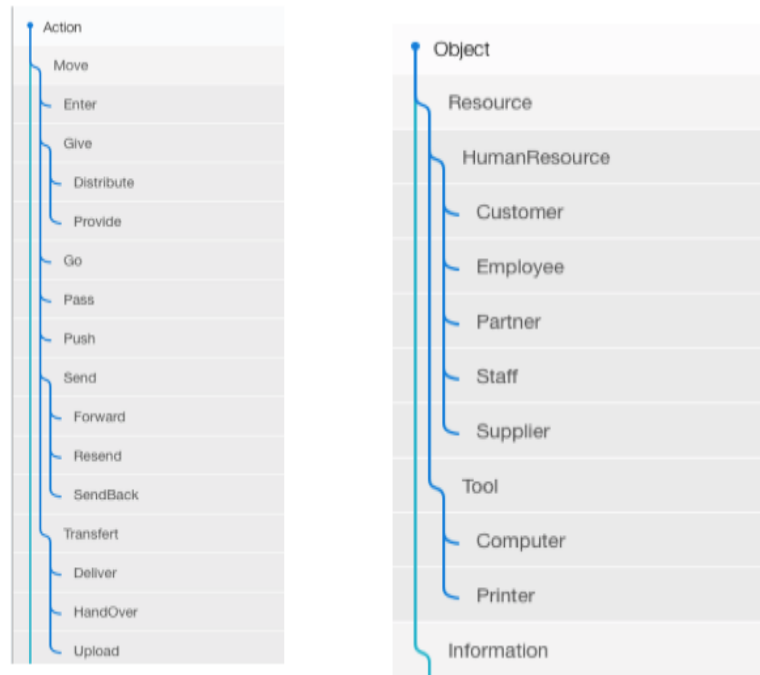


Figure 26 Part of Action and Object taxonomy

As it can be seen from the figure above, in the Action taxonomy, “Move” is one of the Action subclasses; at the same time “Move” has other subclasses such as Enter, Give, Go, Pass, Push, Send, Transfer, and some of these are divided into several subclasses with specific action related to the main one.

5.2.2 Cloud services functional specifications

As mentioned above, for the functional annotation of the cloud service a different approach from that used in the business process has been used. This approach fits in a better way and is more appropriate for annotating the discovered web services.

After an analysis of potential existing methods to annotate the semantic API for web services, as already seen in chapter 2.4, proof has emerged that the SOAP and REST standards are the best for the classification and semantic annotation in the ontology.

As Cretella & Di Martino (2012) suggest, this kind of approach for annotation it will get advantage and will not incur incompatibility as with an OWL mapping. It is also possible to describe the new relationships through the annotations.

As adduced in previous chapters, the method used is the REST for its properties like the uniform interface, addressability, statelessness and connectedness. (Kumari & Rath, 2015)

The figure below shows a summary of the characteristics between the SOAP and REST standards.

SOAP	REST
A XML-based message protocol	An architectural style protocol
Uses WSDL for communication between consumer and provider	Uses XML or JSON to send and receive data
Invokes services by calling RPC method	Simply calls services via URL path
Does not return human readable result	Result is readable which is just plain XML or JSON
Transfer is over HTTP. Also uses other protocols such as SMTP, FTP, etc.	Transfer is over HTTP only
JavaScript can call SOAP, but it is difficult to implement	Easy to call from JavaScript
Performance is not great compared to REST	Performance is much better compared to SOAP - less CPU intensive, leaner code etc.

Figure 27 SOAP / REST characteristics

The various methods considered, according to RESTfulAPI.net (2017) are:

- HTTP GET: to retrieve resource representation / information only. A GET does not change the state of the resource, it is just a request. Even if the request is done multiple times it produces the same result. (Idempotent)
- HTTP POST: to create new resources or subordinated new resource into collection of resources.
- HTTP PUT: to update existing resources. Unlike the POST, its primary purpose is to modify an already existing resource, if it does not exist, it is possible to choose to create a new one.
- HTTP DELETE: to delete resources

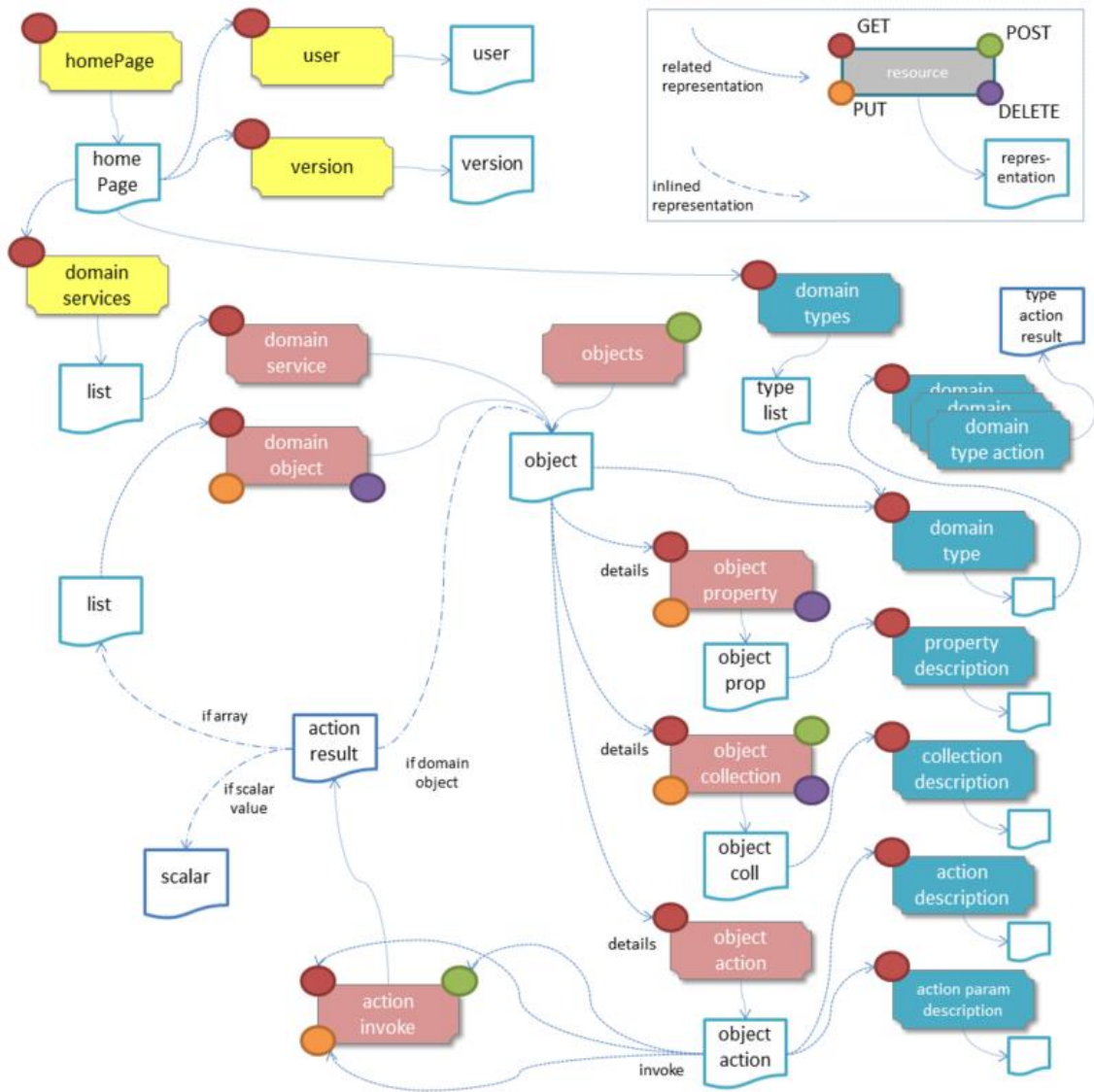


Figure 28 Relationship between resources

The image shows how the resources are defined by the specification. The resources that can be invoked are indicated by the shaded box. The colours indicate the specification where resources are defined (yellow: supporting resources, pink: object resources, blue: type resources). The circles indicate the supported methods (GET, PUT, POST, DELETE) (Haywood, 2012).

5.2.3 Business and IT non-functional specification

The non-functional requirements, as mentioned in previous chapters, unlike the functional expressing the "what", are all the other requirements which may express characteristics, qualities, and specific constraints on functionality. They are still a very important and crucial part that should not be underestimated: in fact, the failure to meet

a non-functional requirement may make the system unusable, even thwarting the correct interpretation of the functional requirements.

The non-functional requirements and specifications were analysed and determined by the Cloud Service Level Agreement Standardisation Guideline (C-SIG SLA, 2014). The guideline is an outcome of the European 2020 initiative “Digital Agenda for Europe” and has the purpose to standardize and streamline the terminologies and understanding of Cloud Level Agreements.

The service level agreements are contractual instruments through which to define service metrics to be respected. The definition of the SLA is based on the determination of the level of service that the customer wants for their needs. Following the European document, the attributes that are present in the invoice service were grouped into five categories that enclose the SLO (description, in measurable terms, benchmarks or goals of the service). Each section covers a group of non-functional requirements that refer to the main category.

- **Performance:** This section deals with services related to the performance of the cloud service.

Availability	Define the time that the service was available, expressed as a percentage.
Average response time	Statistical mean of response time observation for a particular form of request.
Number of simultaneous cloud service users	Maximum number of customer users that can use the service at the same time.
Maximum resource capacity	Maximum amount of given resource available for the customer. In this case refers to data storage.
Support hours	The time during the cloud service provider provides support service to the customer.
Data retention period	Length of time which the cloud provider will retain backup copies of the customer data.

Table 5 Performance Service Level Objectives Overview

- **Security:** The security level offered by the cloud service and the level requested by the customer.

Authentication	Describes the authentication methods offered by the cloud service.
Cryptographic brute force resistance	The strength of cryptographic protection.

Table 6 Security Service Level Objectives Overview

- Data Management: Quantitative and qualitative indicators related the data life cycle management.

Data backup method	Methods used to backup customer data.
Data backup frequency	Period of time between a complete backup of customer data.
Backup retention time	Period of time a given backup is available for the data restoration.
Maximum data restoration time	Time taken to restore data from backup.

Table 7 Data Management Service Level Objectives Overview

- Personal Data Protection: The data processed in the cloud may be transferred to third countries, whit different laws that do not guarantee an adequate level of data security.

Data geolocation list	Geographical location where the customer data may be stored and processed.
-----------------------	--

Table 8 Personal Data Protection Service Level Objectives Overview

- Payment:

Payment plan	Type of payments available for the use of the cloud service.
--------------	--

Table 9 Payment Service Level Objectives Overview

The following table shows the difference language for the description of each requirement.

NON-FUNCTIONAL REQUIREMENTS	BUSINESS LAYER	IT LAYER
Performance		
Availability	Low / Medium / High	Percentage
Average Response Time	Low / Medium / High	Milliseconds
Number of Simultaneous Cloud Service Users	Numerical	Numerical
Maximum Resource Capacity	Numerical	Gb
Support Hours	Personalised, Low / Medium / High	Time
Security		
Authentication	Low / Medium / High	SFA, CERT, TFA
Cryptographic Brute Force Resistance	Low / Medium / High	DES, 3DES, AES
Data Management		
Data Backup Method	None / Low / Medium / High / Very High	Raid Type
Data Backup Frequency	Time (daily, week, months, years)	Time (daily, week, months, years)
Backup Retention Time	Time (week, months, years)	Time (week, months, years)
Maximum Data Restoration Time	None / Low / Medium / High	Gb x Time
Personal Data Protection		
Data Protection	Location, Low / Medium / High	Location / Level
Payment		
Payment Plan	Time	Time

Table 10 Non-functional Business - IT layer matching

As a result, for the specification of non-functional requirements a more technical language is used. In fact, there are descriptions or characteristics that the service must have to meet certain technical minimum standards from the machine view point, user-defined.

On this basis, researches were done on how to make these records more understandable for the business-speaking world. Through interviews and comparisons with other cloud services in the marketplace and consulting with Mathema, evidence emerged as a classification in low / medium / high is the best choice to understand the requirements also for a business user that is not very well versed in the IT field, assuming a basic IT background for business people, that allows to understand the characteristics the user may need. Obviously, there are also requirements that have a matching 1:1, then understandable for both levels.

Analysing in details each non-functional requirement under the IT and business point of view, it is possible to understand the mapping between non-functional business process requirements and non-functional workflow description. In the next chapter, matching rules will be presented. The following tables shows how the comparison works with the two levels.

AVAILABILITY				
Uptime perMonth	Less-equal 96,7%	99%	99,8%	100%
Downtime perMonth	Less-equal 1440 min	420 min	60 min	0 min
Performance	Low	Medium	High	Very High

Table 11 Availability description

- Time in a defined period the service was available, over the total possible available time, expressed as a percentage. The formula to calculate this rate is:

$$f(x) = \frac{\frac{365 \times 24 \times 60}{12} - x}{\frac{365 \times 24 \times 60}{12}} \times 100$$

- Time in a defined period the service is not available. Downtime perMonth

AVERAGE RESPONSE TIME			
Time in milliseconds	More than 30ms	Between 29ms and 10ms	Less than 9ms
Performance	Low	Medium	High

Table 12 Average Response Time description

- Time interval between a cloud service customer initiated event (stimulus) and a cloud service provider initiated event in response to that stimulus.
- Time that is required between the requests from the stakeholder to the system and the receipt, elaboration, execution and response of the request. Expressed in milliseconds.

MAXIMUM RESOURCES CAPACITY				
Data storage perMonth	5 Gb	15 Gb	45 Gb	135 Gb
N. invoice perMonth	1 - 99	100 - 299	300 - 999	1000 - 3000

Table 13 Maximum resources capacity description

- To calculate the monthly data storage, 3 factors are taken into account, the number of monthly invoice, the invoice weight and time. In order to calculate the necessary data storage, it is just necessary to ask the user for an average of invoice sending monthly. The formula for the calculation is:

Number of invoices per month X size (1.5 Mb average size of an invoice) X time (in this case a 30-day month).

SUPPORT HOURS				
Availability perWeek	None	Monday to Friday	Monday to Friday	7 days
Availability perHours	None	9:00 – 20:00	24 hours	24 hours
Performance	None	Low	Medium	High

Table 14 Support hours description

- The time period an interface is made available by the cloud service provider to handle issues and queries raised by the cloud service customer.

AUTHENTICATION			
Types	SFA	TFA (2FA)	MFA
Description	Alphanumeric Password with special characters (min 8 to 14 characters)	USB OTP (One-Time password) 44 characters (12 unique ID, 32 encrypted information given by OTP); SMS OTP; n. max of attempt	Password, Security Token, Biometric Verification.
Level of Assurance	2	3	4
Data Security Level	Low	Medium	High

Table 15 Authentication description

- Specifies the available authentication mechanisms supported and the relative grade of security. The information is derived from The Electronic Authentication Guideline of NIST (Burr et al., 2013)

CRYPTOGRAPHIC BRUTE FORCE RESISTANCE				
Type	DES	3DES	AES	BLOWFISH
Key Length	56 bits	112(k1, k2), 168(k1, k2, k3) bits	128, 192, 256 bits	32, 448 bits
Cipher Type	Symmetric block cipher	Symmetric block cipher	Symmetric block cipher	Symmetric cipher algorithm
Keys	Single	Single (3 parts)	Single	Public
Block Size	64 bits	64 bits	128, 192, 256 bits	64 bits
Possible Keys	2 ⁵⁶	2 ¹¹² , 2 ¹⁶⁸	2 ¹²⁸ , 2 ¹⁹² , 2 ²⁵⁶	2 ³² , 2 ⁴⁴⁸

Possible ASCII character keys	95^7	$95^{14}, 95^{21}$	$95^{16}, 95^{24}, 95^{32}$	$95^4, 95^{56}$
Data Security Level	Low	Low	Medium	High

Table 16 Cryptographic Brute Force Resistance description

- Strength of a cryptographic protection applied to a resource based on its specific characteristic and services discovered for the security of the data (Mathur & Kesarwani, 2013), (Singh, 2013).

DATA BACKUP METHOD					
Type	Single Disk	RAID 0	RAID 3	RAID 5	RAID 6
Level	None	0	3	5	6
Description		Block Interleave Data Striping w/o Parity	Bit Interleave Data Striping with Parity Disk	Block Interleave Data Striping with Skewed Parity	Block-level striping with double distributed parity
N. min of disk	1 max	2	3	3	4
N. max faulty disk	0	0	1	1	2
Capacity	C	C x N	C x (N - 1)	C x (N - 1)	C x (N - 2)
High I/O Rate	Good	Very Good	Poor	Good	Good
Data Management Level	None	Low	Medium	High	Very High

Table 17 Data backup method description

- Types of services available for the data saving and characterization based on available data level in case of damaged or broken disks. (CISCO, 2016)

MAXIMUM DATA RESTORATION TIME				
Speed	None	50 GB perHour	100 GB perHour	200 GB perHour
Data Management Level	None	Low	Medium	High

Table 18 Maximum data restoration time description

- Refers to the committed time taken to restore cloud service customer data from a backup.

DATA GEOLOCATION LIST					
Level	F	D	C	B	A
Description	No rules, no data protection enforcement.	Negligible rules for intelligence and law enforcement activity, non-existing data protection enforcement.	Minimal rules for intelligence and law enforcement activity, limited data protection enforcement.	Laws for intelligence and law enforcement activity, some data protection enforcement.	Strong laws for intelligence and law enforcement activity, strong data protection enforcement.
Location	North Korea.	Russia, China, Pakistan, Saudi Arabia, Egypt, Libya, Hong Kong.	India, Thailand, Sumatra, South Africa.	United States, European Union (EU), Australia, Mexico.	Canada, Switzerland, Spain, Brazil, Japan.
Data Protection	None	Low	Medium	High	Very High

Table 19 Data geolocation list description

- Specifies the geographical location(s) where the cloud service customer data may be stored and processed by the cloud service provider. The location can be characterized also for categories (Continent/Nation/State/City). Based on mutual legal assistance treaty (MLAT), it is possible to characterize five different groups of nations that have similar and specific laws and rules for the storage of the data.

In the comparison, there is attention to the detailed description of the services that is essential for the customer, especially for SMEs, but often the IT side considers these aspects secondary to others.

Any reference concerning IT specifications refers to existing and solid sources with technical details that can help not only the customers to better understand what they choose, but also a specialized broker (a kind of IT / Business intermediary) to have excellent vision of the whole.

5.3 Conclusion

This chapter defines the suggestion phase of the science research design.

The case study was divided into four parts, respectively: business process and cloud service functional specifications and business process and cloud service non-functional specifications. The initial suggestion is found on the appropriate way to classify the various requirements.

For this reason, in order to describe functional requirements on business process, it was considered appropriate a task ontology that consists of Object / Action and APQC classification, while for the workflow, appropriate was the classification of the services and Object / Property for tasks.

As for the non-functional requirements, in addition to the match 1:1, used in some cases, after opportune interviews and collaborations with expert companies in this industry field, it was created a low / medium / high classification method for the business people that refers to specific IT layers in the technical language. The next chapter will show how these activities are noted in the ontology and how they interact with one another.

6. Development

This chapter describes the development phase, in this case the development of an ontology, which refers to the third stage of the design science research. After a brief introduction of the structure, the development of ontology for functional requirement, with APQC, Object / Action for the business layer and Object / Property for cloud service is explained. For the non-functional requirement, the comparison tables presented in chapter 6 are inserted in the ontology to describe cloud specifications in order to achieve a smart alignment of business and IT.

6.1 Ontologies

Gruber (1993) defines ontology as "a specification of a conceptualization". An ontology can be defined as a formal representation of a set of knowledges in a conceptualization, in a particular area of interest. The conceptualization means an abstract and simplified representation of the particular field of knowledge that it is possible to describe.

An ontology is an attempt to formulate an exhaustive and rigorous conceptual scheme within a given domain. Any information will be mapped by its own ontology and inserted in a context of relationship with other ontologies, in order to create the logical relationships that allow any semantic program to be comprehensible. This type of structuring can group the information in a certain context or logical space, according to the binding mechanism of the human mind.

Ontology provides for an explicit conceptualisation that describes the semantics of the data, with a syntactic and semantic language using a common terminology to make the ontology reusable in similar contexts.

As said in chapter 2.6, the components of the ontology are:

- Classes
- Property
- Instance
- Rules
- Axioms

Ontologies can be very useful to structure and define the meaning of the terms used. Could be critical when applications try or mix information from different environments. The same terms can be used with different meanings in different contexts, whereas different terms may be used for documents that have the same meaning. The mapping process allows to make compatible different content, resolving the mixing information problem. The mapping can have different levels of complexity and can involve other ontologies. The figure below presents the structure of the BPaaS ontology, that includes REST, APQC, Functional Description and the Service Level Agreement and extends the ArchiMEO ontology (an enterprise ontology based on ArchiMate).

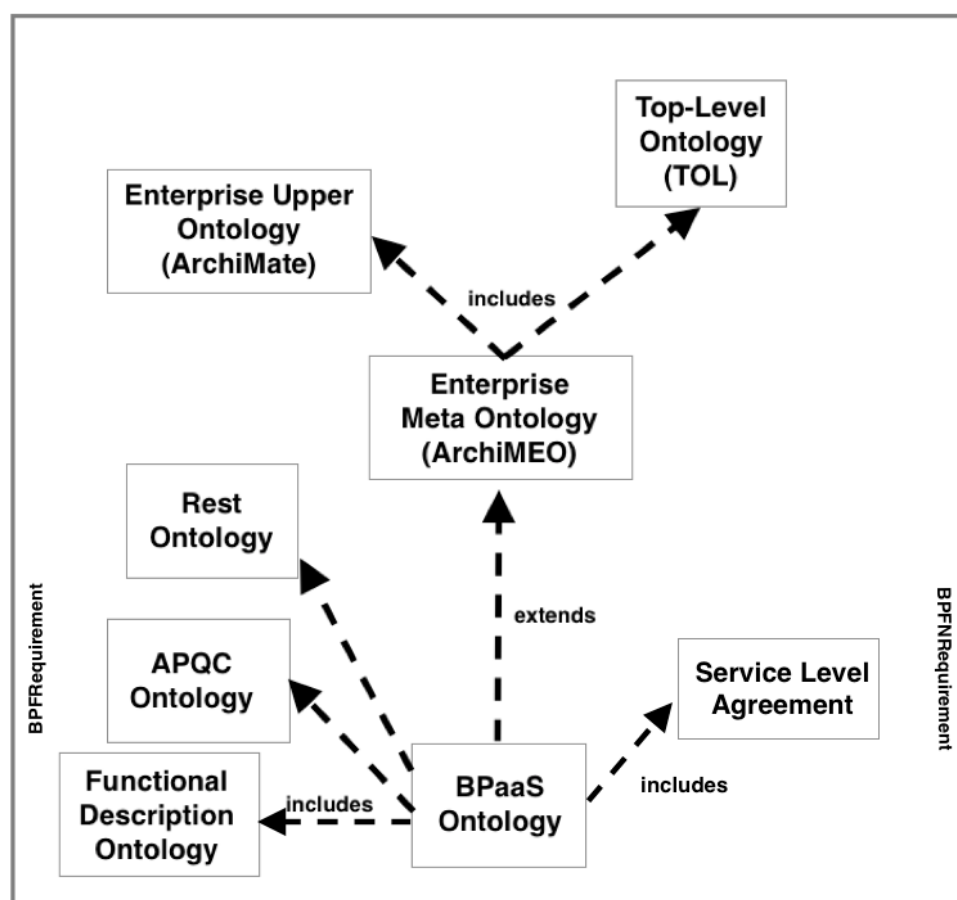


Figure 29 Adapted from ArchiMEO ontology.

6.2 Ontology for functional requirements

The semantics of functional requirement is represented depending on the level of belonging, either business or IT. For each level, the ontologies, thus created, are used to record the single activities or group of activities. In the next chapter, more details of this distinction occur, including how they were recorded in the ontology.

6.2.1 Business Process Functional Requirement

For the business layers, were represented two ontologies that complement each other: APQC ontology and the Functional Business Process Description ontology for the Object / Action. Both ontologies are used for the annotation of the business process (Figure 22). In particular, APQC is used for the context of the process and Object / Action taxonomy for the annotation of each single activities in order to specify detail not clarified by APQC.

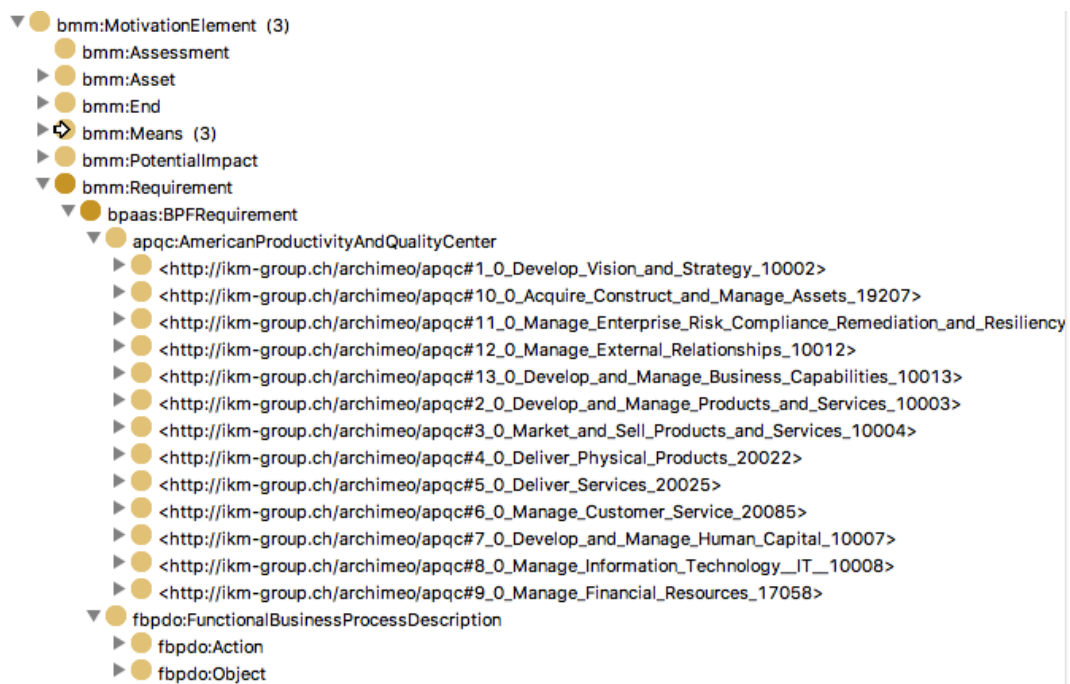


Figure 30 Business process functional requirement ontology

The figure shows the implementation of these requirements in the ontology. The APQC ontology and Functional Business Process Description (Object / Action) are subclasses of Business Process Functional Requirement (BPFRequirement), which is a subclass of Requirement, which in turn, is Motivational Element subclass. This characterization is consistent with the model of Motivational Extension Metamodel of ArchiMate (The Open Group, 2013). These motivations influence, guide, and constrain the design.

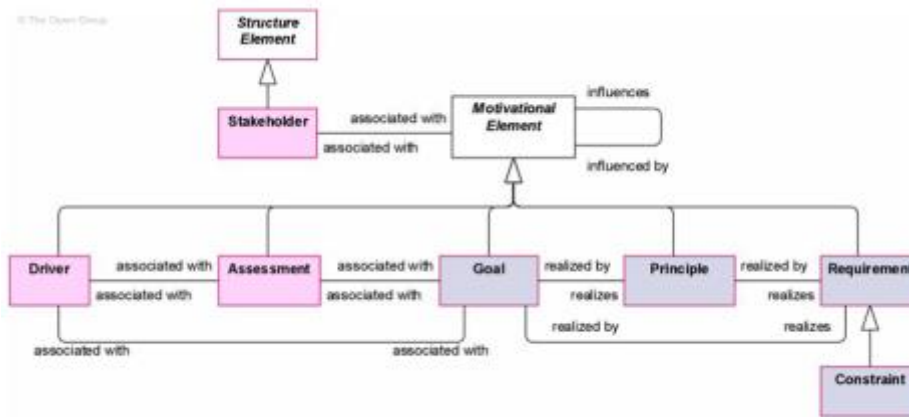


Figure 31 Motivation extension metamodel (The Open Group, 2013)

6.2.1.1 APQC

American productivity and quality center process classification framework creates a common language for organizations to communicate and define work processes comprehensively and without redundancies (APQC, 2017).

It is used as a benchmark and standard by organizations for performance management activities. The framework, which is available from the official website APQC, is in comma-separated value (CSV) Converting this later into turtle file format (that expresses data in the RDF data model with a syntax similar to SPARQL) it is possible to import all classifications APQC directly into the ontology.

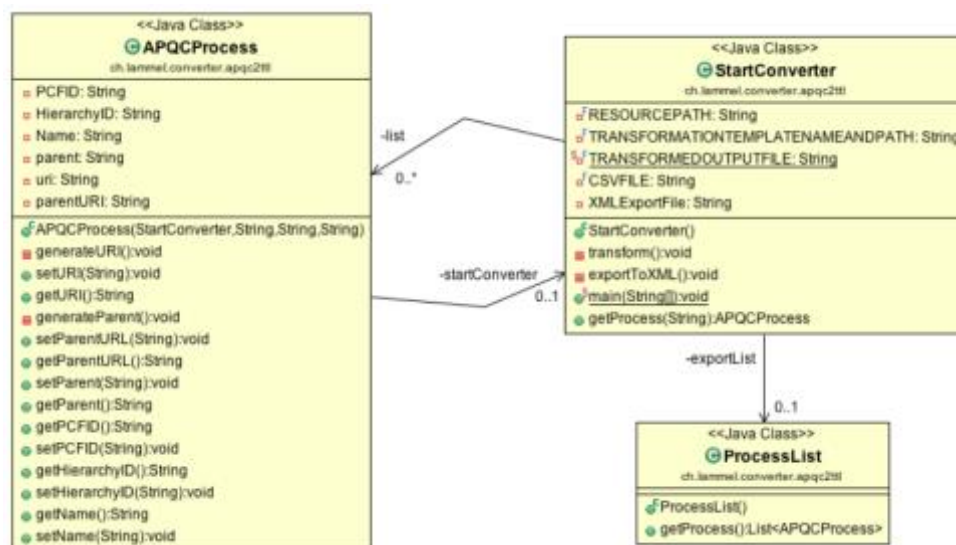


Figure 32 APQC conversion process (CloudSocket Project, 2017)

In Figure 33, it is possible to see how the transformation happens, from CSV file to ontology. On the left side, there are all the information from the CSV file (name, hierarchy, PCF ID and subclasses). On the right, it is shown the conversion, that start with the analysis of the CSV and afterwards the creation of a list that contains all of APQC processes in the form of object occurs. This list is converted into XML format; once this is done the last step is to convert the just created XML file in the turtle format and import this creation into the main ontology, in this case in the subclass under "BPFRequirement".

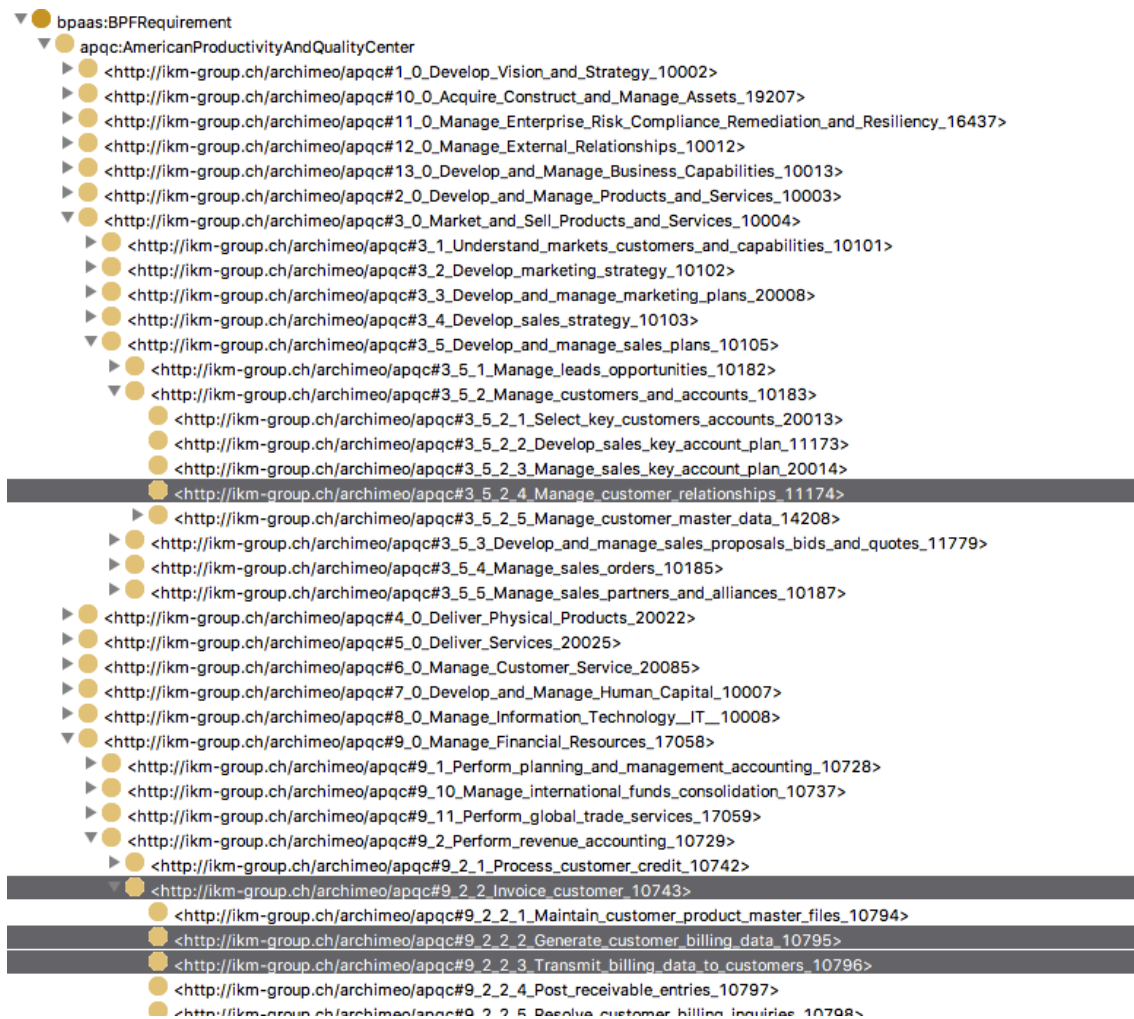


Figure 33 Part of the APQC ontology

The result of the conversion is displayed above: it is possible to see how the sub-categorizations have been maintained after the transformation into the structure (specifically: hierarchy, name and PCF ID). In Figure 33, categories are highlighted that will define the activity or group of activities of the business process taken into account.

6.2.1.2 Object / Action

As already mentioned, classification with APQC is only a part of the record of the activities for the business process. The second part is based on the categorization of each task with Object / Action taxonomy.

The creation of the taxonomy occurred during the development of CloudSocket project, after interviews and business process modelling. These processes have given evidence of the names and the most suitable verbs to categorize each task. As mentioned earlier, some nouns or verbs can be misunderstood or interpreted in a different way, for this reason, they have been characterized using WordNet, a lexical database of English used to give hierarchy to words (Fellbaum, 2005).

This analysis, has produced a list of nouns and verbs that offers taxonomy from the words that were extracted, based on their meaning and compared with other existing ontologies, in order to obtain a clear structure. Subclasses help to group objects or actions in the field of the most appropriate meaning.

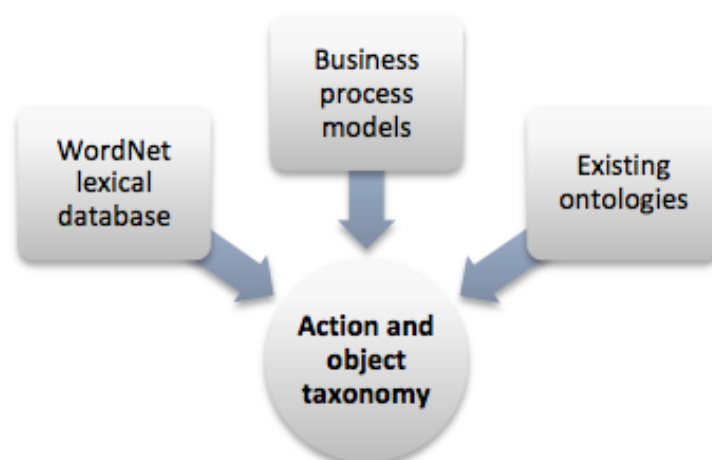


Figure 34 Action and object taxonomy

Observing the ontology under the “Action” point of view, Figure 35, shows how the whole part of “Action” is structured with all its classes and subclasses. In the example, the left side lists all the action classes, while the right-side zooms in into the specific action "Move" which groups all the verbs of motion (i.e.: "Give", "Send", "Resend", ...).

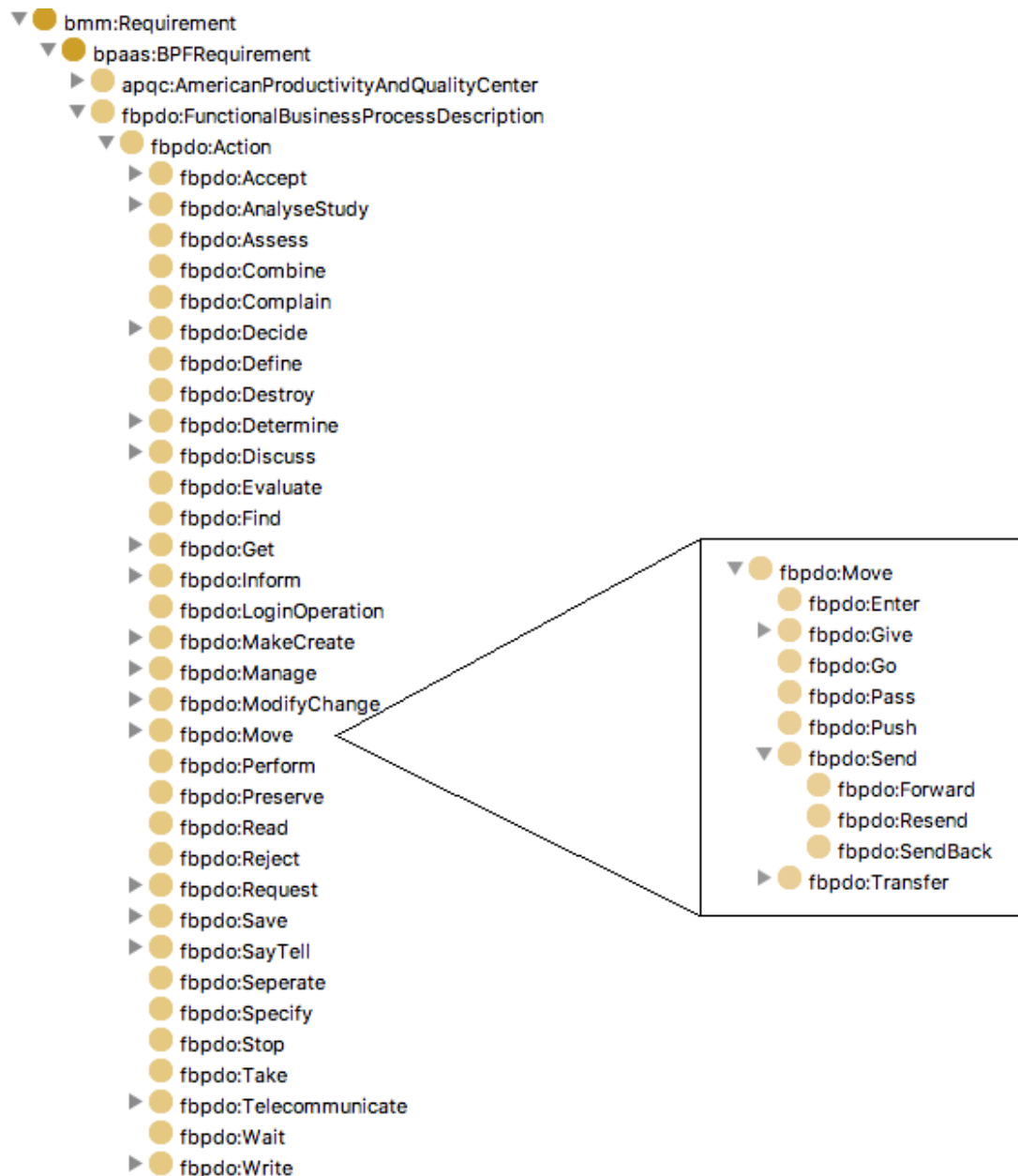


Figure 35 Action part of the ontology with "Move" example

The same type of approach has been followed for the “Object” part: after analysing the business process from CloudSocket project, a list of names corresponding to the process activities was compiled. That lead to took place the grouping of activities that had the same class of belonging and reported in the ontology.

Figure 36 shows, on the left side, the complete list of “Objects”, and on the right side a zoom in into the case "Invoice", subclass of “Information” and differentiated depending on the type.

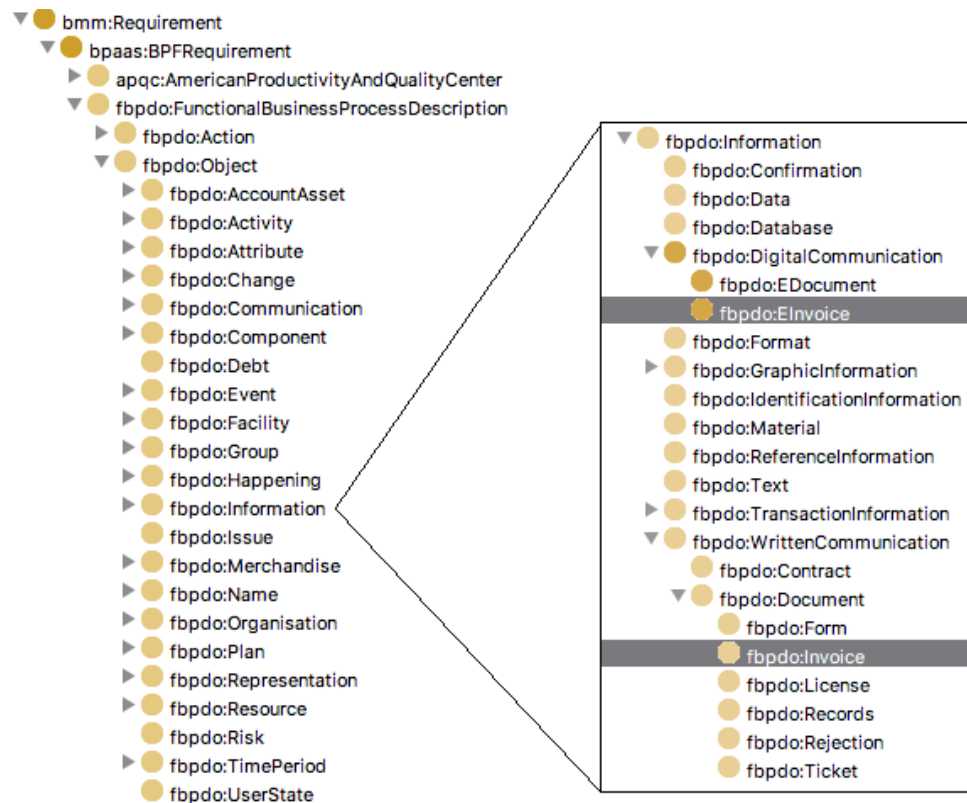


Figure 36 Object part of the ontology with "Invoice" example

The use of APQC framework ontology together with Functional Business Process Description ontology describes the functional part of the business process.

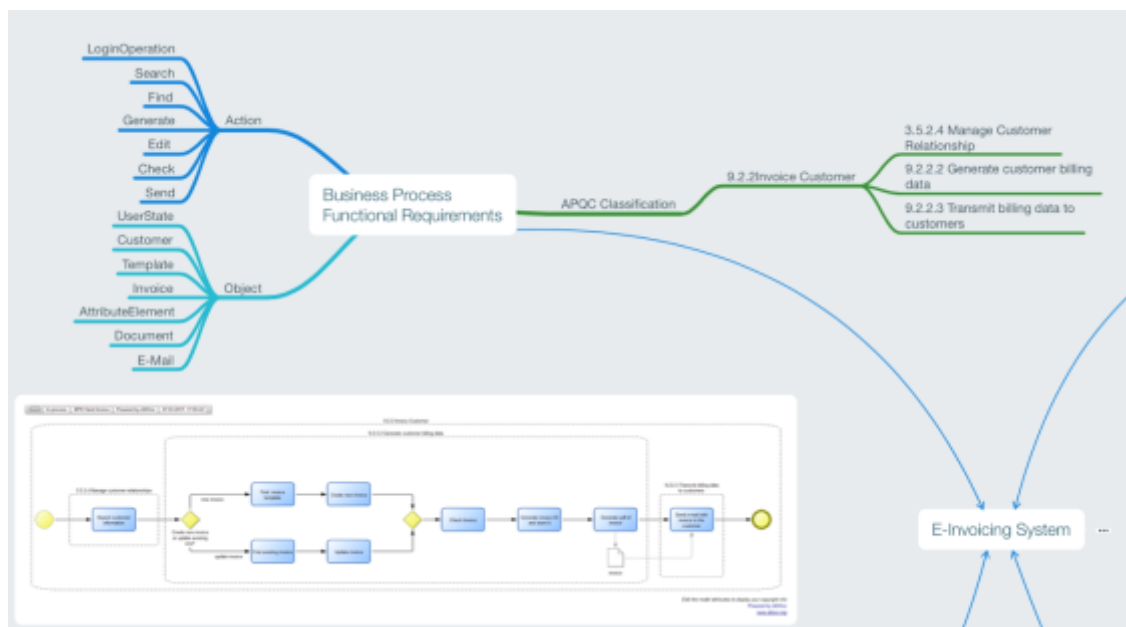


Figure 37 Business process functional requirement annotation

6.2.2 Cloud service Specification

Completed the description of the functional part of the business process, analysis of the cloud service specification is required. For categorization of this part, the REST protocol, subclass of “SpecificationStandard”, was used.

With the constant evolution of the web, the REST service is becoming a good alternative to the traditional SOAP. The technical characteristics and comparison with the SOAP service has already been covered in chapter 2.4.1.2. This service was chosen for its ability to combine data from multiple resources and to create more meaningful datasets (Y.-J. Lee & Kim, 2010). The same concept of “mashup” is also taken by Jung, Kim, & Kim (2013), saying that the combination of two or more services can create a new one, and thanks to modelling in REST, it is easy to combine several web services and semantic queries (Y. J. Lee & Kim, 2011). The ease of use and manipulation of resources through representations by this method makes working with the REST standards, easier, especially in the learning part, more secure, and lightweight, in terms of use and consumption of bandwidth, in contrast to SOAP (Lanthaler, Granitzer, & Gütl, 2010).

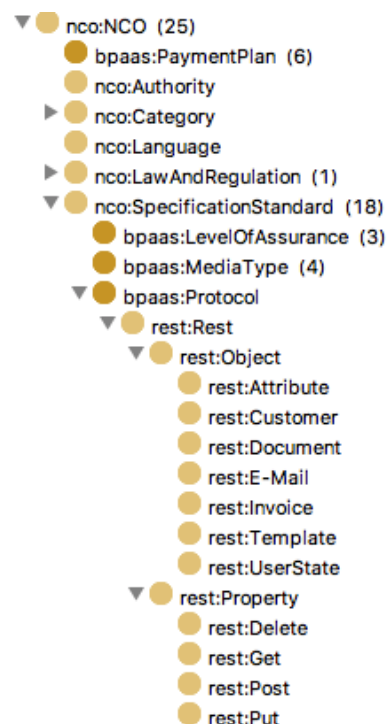


Figure 38 Rest ontology

6.2.2.1 Object / Property

As for the creation of Object / Action ontology in the business layer, elements of the workflow have been taken into account. After an analysis of the most appropriate names to rank each task, there is the creation of “Object” class into the ontology and the classification of properties, which uses a “CRUD” mapping, four subclasses being created with their meanings in HTTP (Post, Get, Put, Delete).

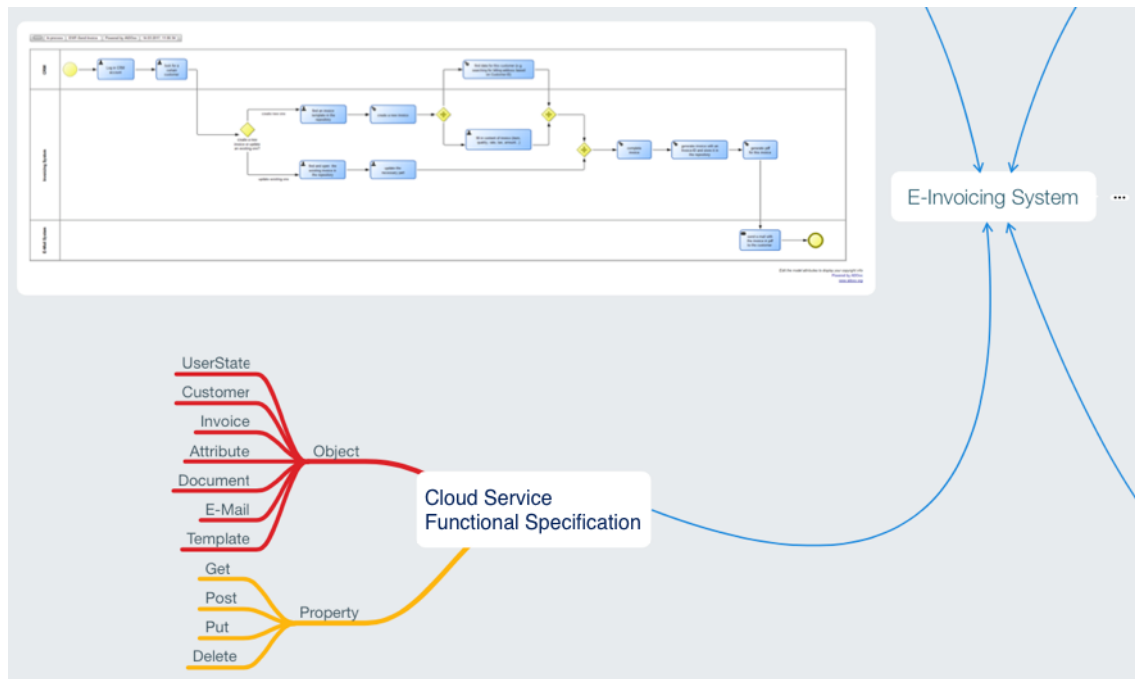


Figure 39 Cloud service functional specification

6.3 Ontology for non-functional requirements

As mentioned, non-functional requirements are based on the Cloud Service Level Agreement Guideline. The next chapter, will show more in detail the way to express non-functional preferences for a business user.

6.3.1 Business Process Non-Functional Requirement

As also happened with the functional part, the non-functional requirement is a subclass of "Requirement" and in turn, of the "MotivationalElement". The class "BPNFRequirement" collects all the non-functional choices that the user prefers, all the features that the service will have plus the main process. Figure 40 shows the structure of this class and its subclasses, that represent the service level objectives with instances

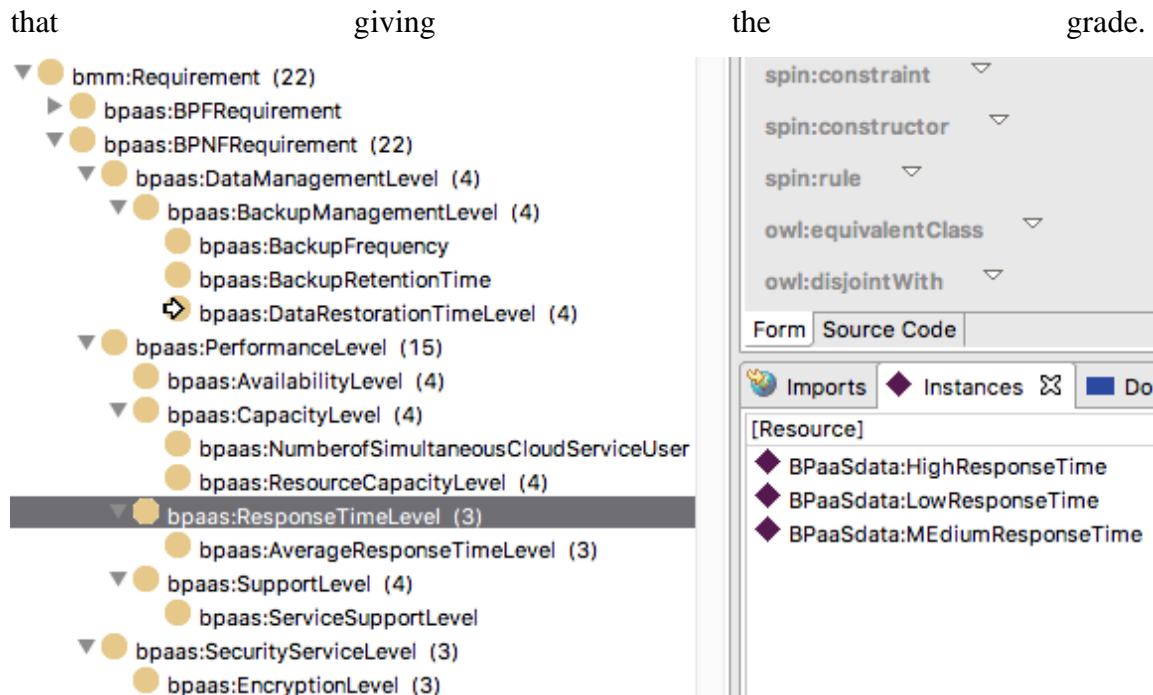


Figure 40 Business Process Non-Functional requirement

Analysing in detail the figure it is possible to distinguish for each classes the level (High / Medium / Low) referred to the specific non-functional class it belongs to.

In chapter 7, using matching rules, that classification level will match with the equivalent of the IT layer.

Within the use of this standard, None, Low, Medium, High, Very High, that defines the level of service, the process of alignment of the Business-IT layers starts. In this way, the business user only has to choose the level of service and enters specific values they prefer, that are automatically are matched with the IT specifications.

Figure 41 shows the graphically structure of business process non-functional requirements and their grades.

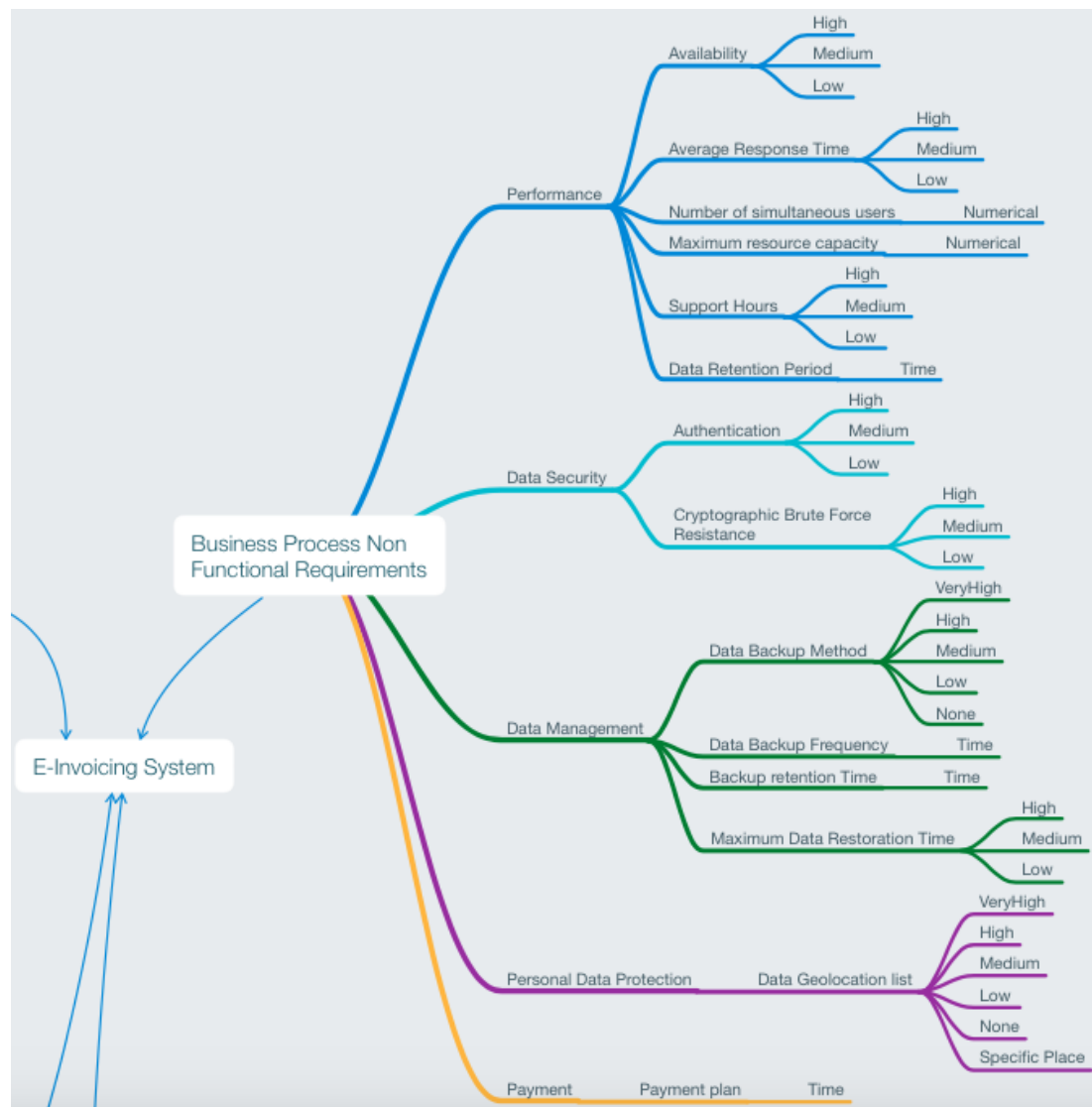


Figure 41 Business process non-functional requirements

6.3.2 Cloud Service Non-Functional Specification

For the non-functional specifications of the cloud service, it has been reported in the ontology all the objectives of service level agreement guidelines that are part of the business process “Send Invoice”. Figure 42 shows the various categorizations expressed in classes and subclasses of the SLA standards.

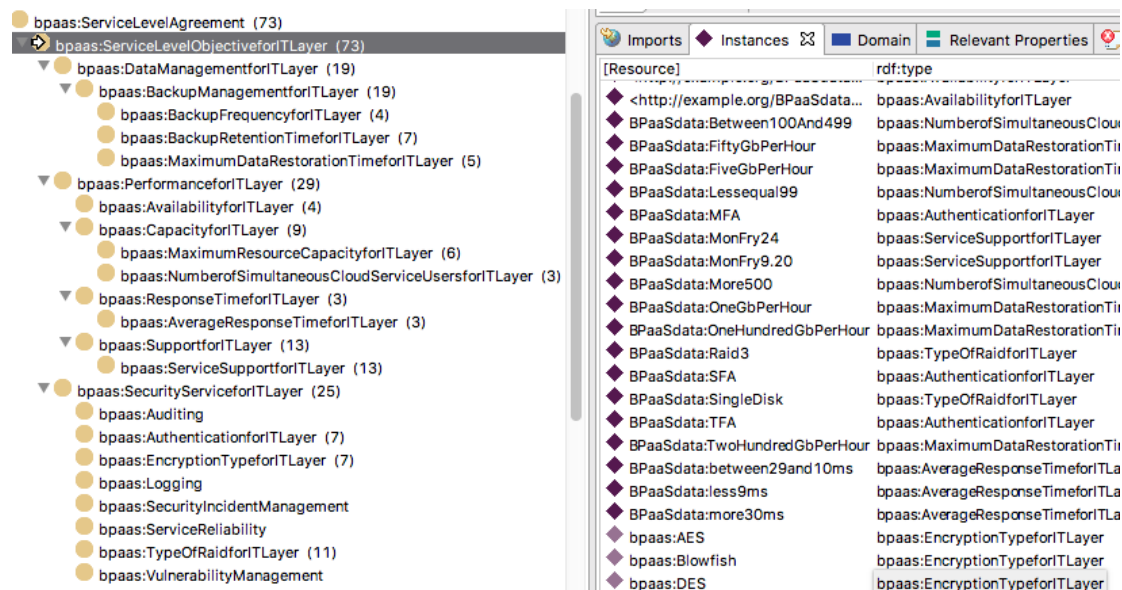


Figure 42 Service level agreement ontology representation

Each class will be put in relation with “BPNFRequirement” in order to be able to associate to a high / medium / low a IT specification; as said before, these matching rules will be shown in the chapter 7.

A detailed analysis presents, on the right, all the instances that represents all the possible technical specifications for the services.

Figure 43 shows the graphic structure of cloud service non-functional specifications and their specifications.

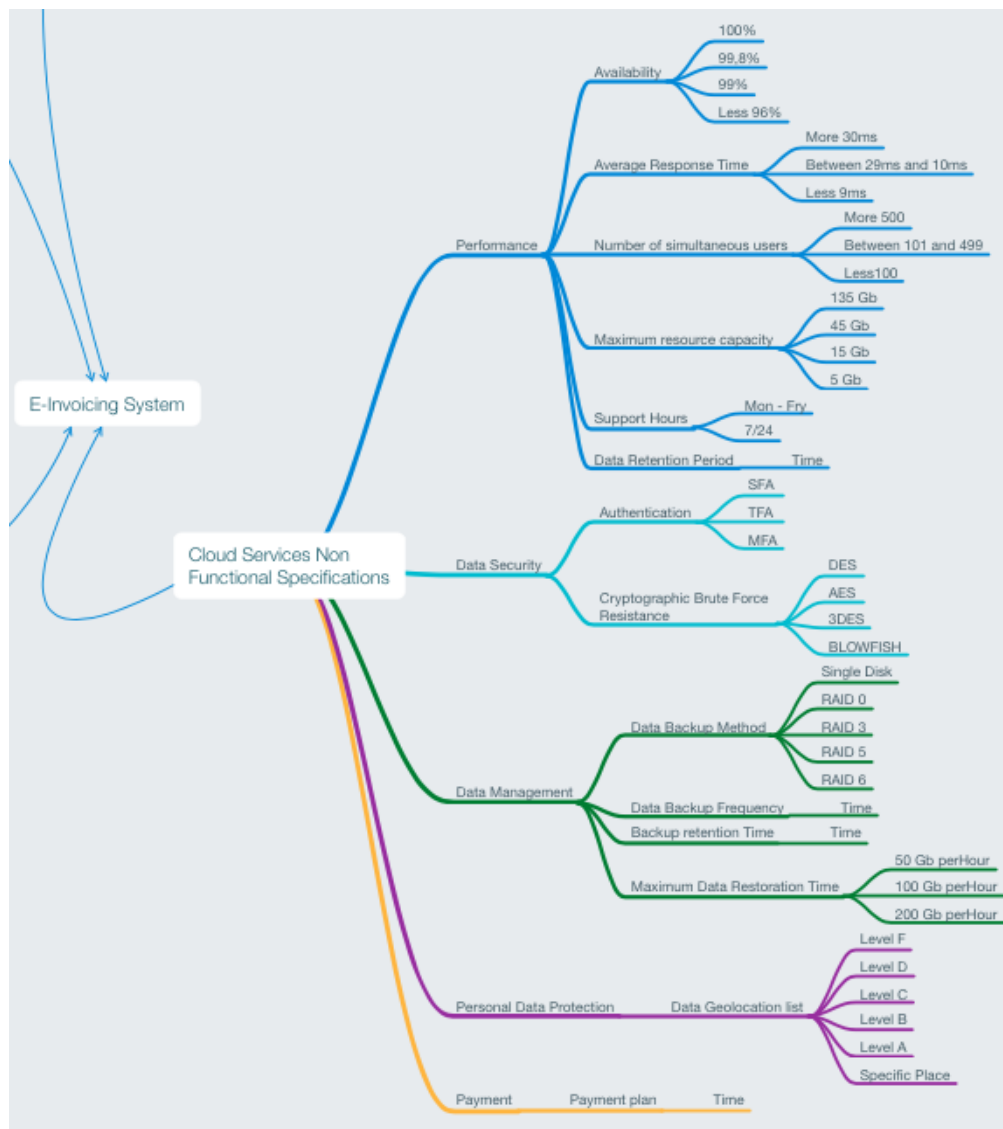


Figure 43 Cloud service non-functional specifications

6.4 Cloud services

In the ontology, class “Service” is over “CloudService” class, that collects existing cloud services that can be used for the Send Invoice service and the class “CloudServiceSpecification” that collects all the exact technical specifications for each service (the technical specifications of the services are taken from (CloudSocket Project, 2017), Figure 24). The creation of these two classes reflects the specifications that come from the model developed for the European Project CloudSocket, in the specific example of Workflow and Workflow Description.

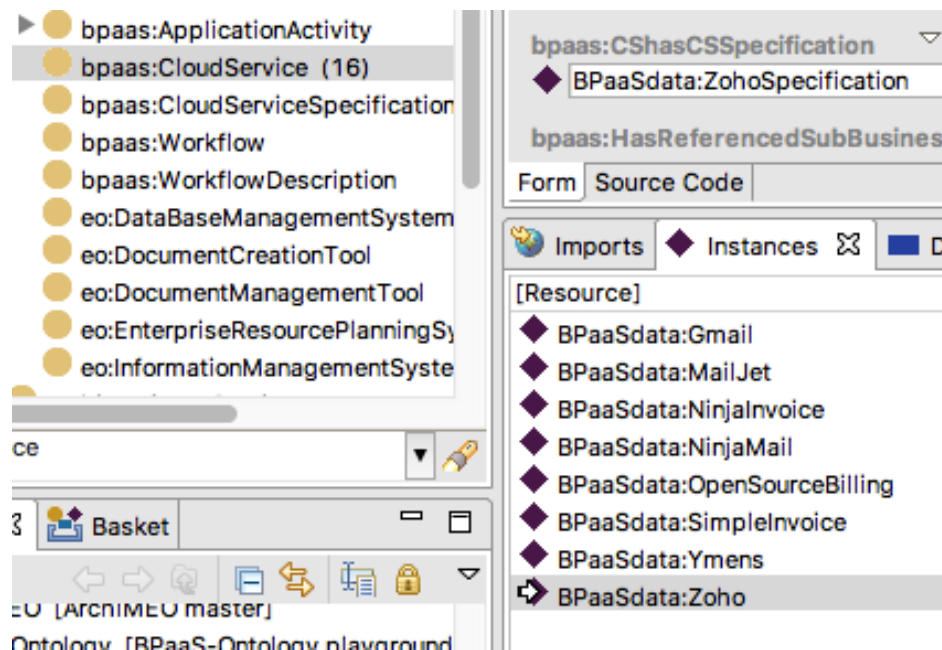


Figure 44 Cloud services invoices for Send Invoice process

Each instance is a cloud service, that is specified with object properties to the “CloudServiceSpecification” and to the REST annotation.

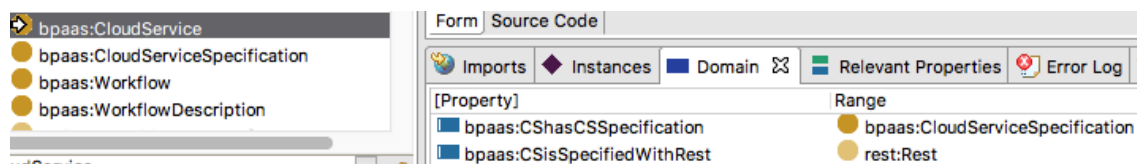


Figure 45 Cloud service properties

Considering the “CloudServiceSpecification” class, as it is possible to see in the figure below, there are instances that refer (are specification of) to the cloud services showed before.

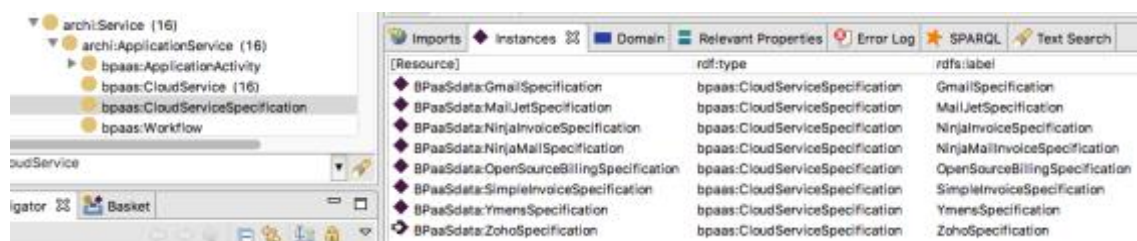


Figure 46 CloudServiceSpecification instances

In Domain, there are all properties (objective properties) that make possible to assign to each individual cloud service the respective technical specification related to service level objective. In particular, each attribute refers to a specific value from SLO IT.

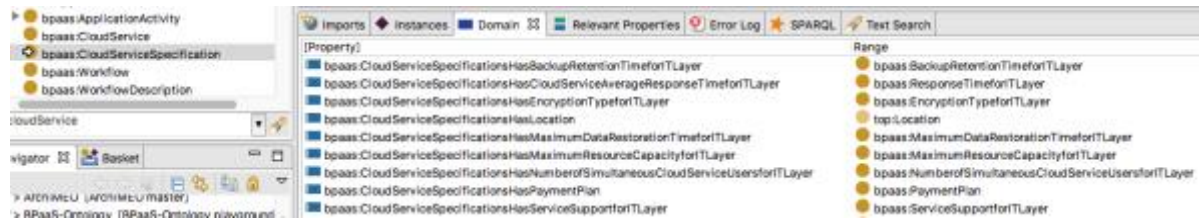


Figure 47 CloudServiceSpecification properties

The example in Figure 48 shows the technical IT specifications of the instance “NinjaInvoice”.

Resource Form

Name: BPaaSdata:NinjaInvoice

Annotations

rdfs:label
NinjaInvoice

Other Properties

bpaas:CloudServiceHasAvailabilityAvailabilityforITLayer
⬢ <http://example.org/BPaaSdata#100>

bpaas:CloudServiceHasBackupFrequencyforITLayer
⬢ bpaas:monthly

bpaas:CloudServiceHasBackupRetentionTimeforITLayer
⬢ bpaas:up_to_1_month

bpaas:CloudServiceHasCloudServiceAverageResponseTimeforITLayer
⬢ BPaaSdata:less9ms

bpaas:CloudServiceHasEncryptionTypeforITLayer
⬢ bpaas:Blowfish

bpaas:CloudServiceHasLocation
⬢ eo:Italy

bpaas:CloudServiceHasMaximumDataRestorationTimeforITLayer
⬢ BPaaSdata:FiftyGbPerHour

bpaas:CloudServiceHasMaximumResourceCapacityforITLayer
⬢ <http://example.org/BPaaSdata#5Gb>

bpaas:CloudServiceHasNumberOfSimultaneousCloudServiceUsersforITLayer
⬢ BPaaSdata:More500

bpaas:CloudServiceHasPaymentPlan
⬢ bpaas:FreeofCharge
⬢ bpaas:MonthlyFee
⬢ bpaas:PrepaidAnnualPlan

bpaas:CloudServiceHasServiceSupportforITLayer
⬢ bpaas:Mon-Fri

Figure 48 NinjaInvoice Instance properties

6.5 Conclusion

This chapter shows the approach used for the creation and extension of the BPaaS ontology already used in the CloudSocket Project (2017). There is a detailed description of the ontology concepts, starting from the annotation of function business process by APQC and Object / Action taxonomy to workflow annotation using the REST standard. For the non-functional aspect, using properties, it is possible to associate classes and instances to a standard, in this case, high / medium / low, allowing to match specification of IT with business requirements and vice versa. The addition of the cloud services specifications allows the choice of the most appropriate service depending on the requirement chosen by the customer.

The following chapter, will show this selection process in a scenario of use.

The following figure will give an example of how the ontology works and how concepts are related.

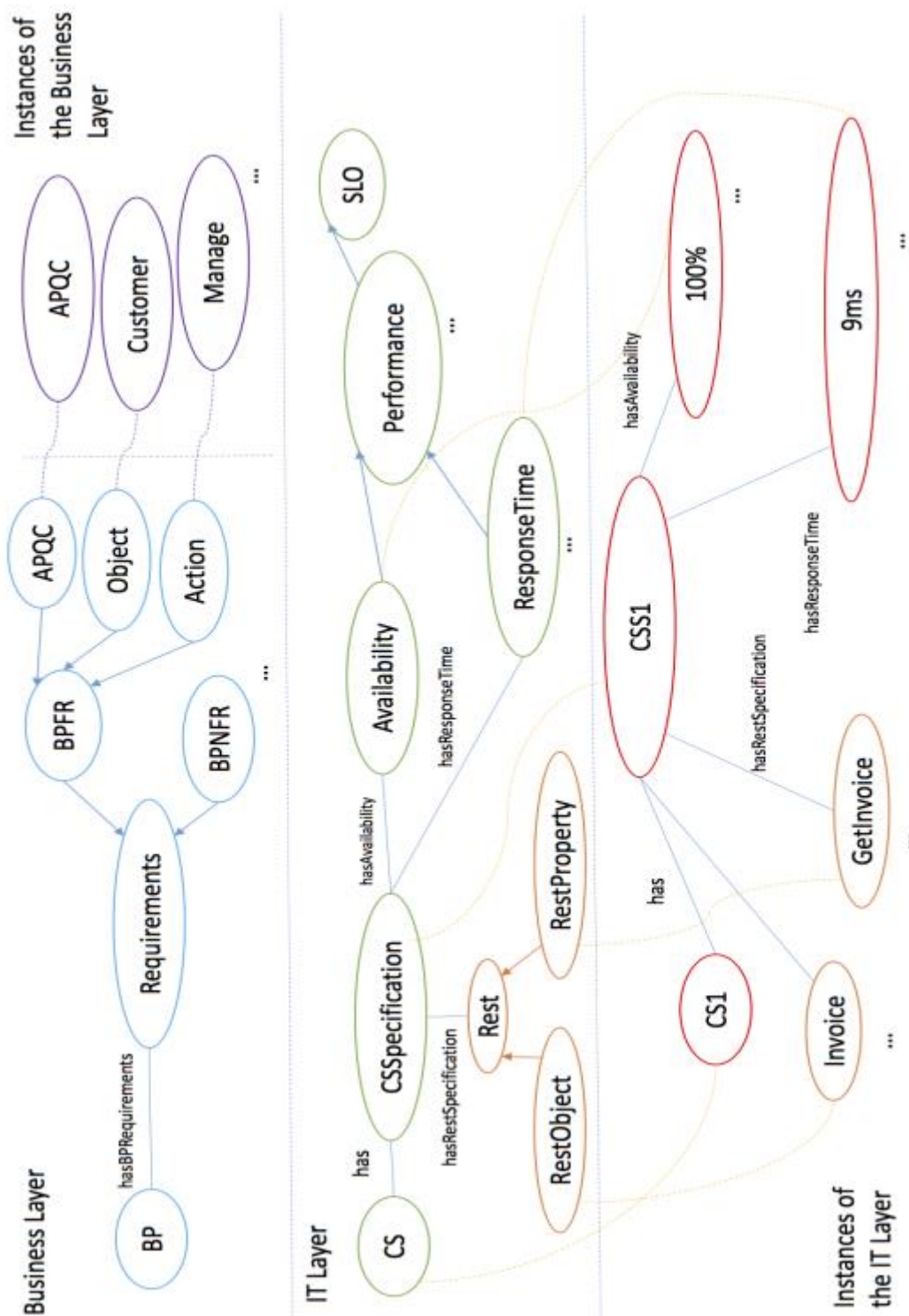


Figure 49 BPaaS Ontology example

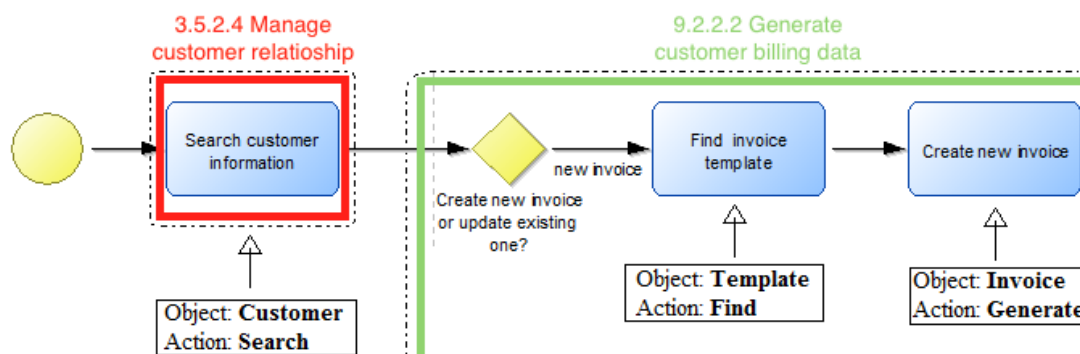
7. Evaluation

This chapter describes the evaluation of the results from the analysis. Through a scenario of use the work of the development phase will be evaluated.

7.1 Business – IT Functional Layer

For the business – IT alignment in the functional layer, as presented in the suggestion phase, it is used an annotation APQC and Object / Action, the example below shows a part of the annotation that refers to the business process in Figure 22.

The classification by the use of APQC (in red and green) and task ontology with the respective Object / Action taxonomy was added during the development phase to the ontology.



50 Example of business annotation using APQC and Object/Action taxonomy

In the following figure, there is an example for the ontology classification based on identification of Object / Property with REST architecture.

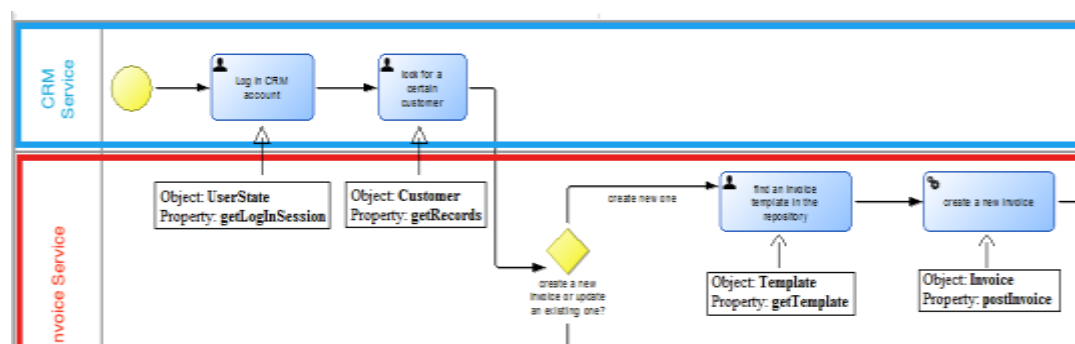


Figure 51 Example of workflow annotation using object / property classification

7.1.1 Matching

The picture below shows the alignment of annotation of each task between business and IT. On the left are all the tasks that came out from the process and on the right the equivalent in business and IT language. For the annotation in the business level, the study used Object and Action and the APQC framework, that is shown with squares, respectively, yellow for 9.2.2, red for 3.5.2.5, blue for 9.2.2.2 and green for 9.2.2.3 classification. For the IT side, using REST, the annotations are divided by Object and Property, while the squares refer to the various cloud services used in the process where the red indicates the CRM, the blue invoice and the green the mail.

Task	Business Layer		IT Layer	
	Object	Action	Object	Property
Log In	UserState	Login	UserState	GetLoginSession
Identify Customer Data	Customer	Search	Customer	GetRecods
Find Template	Template	Find	Invoice	GetTemplate
Find Invoice	Invoice	Find	Invoice	GetInvoice
Create New Invoice	Invoice	Generate	Invoice	PostInvoice
Update Invoice	Invoice	Edit	Invoice	PutInvoice
Check Invoice	Invoice	Check	Invoice	GetInvoice
Generate Invoice ID	AttributeElement	Generate	Attribute	PostAttribute
Generate Invoice PDF	Document	Generate	Document	PostDocument
Send Invoice	E-Mail	Send	E-Mail	PostMail

Figure 52 Business - IT functional alignment

7.2 Business – IT Non-Functional Layer

For the alignment in the non-functional level, an interview to a company that would like to move to an invoice cloud service has been reported. The interview and the non-functional specification are expressed in business language. The following sub-chapters will present how these specifications can be translated into IT language and how they can be matched to the best service.

7.2.1 Transformation from Business to IT

Using the following scenario non-functional business requirements can be adducted and then transformed into IT language, using the development approach shown in chapter 6.

“A medium enterprise needs a new billing system that includes a customer management and mail delivery system. An interview with the customer focuses on the requirements that the system should have. The customer needs a service that is always available, which does not have long waiting times between request and reception of data, that can be used by multiple users simultaneously, for example 100. Monthly he sends roughly no more than 80 bills, so the data capability must be wide enough to contain this number of invoices. These data are important as evidence of fiscal payment, so it should be possible to perform data backup monthly and this backup must be available up to one month, if there would be the need of a recovery. It can take place slowly because of the not so much big number of data. For the security and protection of the data, the customer wants a high standard. Another requirement is the support service in case of problems: the client requires availability from Monday to Friday according to the company's working days. For payment, the client prefers prepaid annual plan.”

	Non-Functional Requirement	BUSINESS LAYER	IT LAYER
1	Availability	Very High	100%
2	Response Time	High	Less 9ms
3	Number of Simultaneous Users	100	100
4	N. Invoice Per Month for data capacity calculation	80	3.6 Gb
5	Data Backup Frequency	Monthly	Monthly
6	Backup Retention Time	1 Month	1 Month
7	Restoration Time	Low	50 Gb Per Hour
8	Data Security Encryption	High	Blowfish
9	Data Protection	High	Level B (United States, Europe, Australia, Mexico).
10	Service Support	Mon - Fry	Mon - Fry
11	Payment Plan	Prepaid annual Plan	Prepaid annual Plan

Table 20 Transformation from business to IT

For each line there are analysis, description and rule, written in pseudo-code, humans understandable but with a well-defined syntax that refers to the developed ontology. Here is the analysis of each line:

1. The customer asks for a service that is always available, so at a very high level. According to Table 11, for the IT level it means a service that has an uptime of 100% that means 0 min of downtime. “IF standard for availability from business area is VERY HIGH, THEN the uptime per month for IT layer is 100%”

```

CONSTRUCT ?AVAILABILITYLEVEL hasAvailabilityFromBusinessToIT ?ITVALUE
WHERE
?AVAILABILITYLEVEL is bpaas:AvailabilityLevel
?ITVALUE is bpaas:AvailabilityForITLevel
IF (?AVAILABILITYLEVEL = "VERY HIGH") THEN ("100") AS ?ITVALUE
IF (?AVAILABILITYLEVEL = "HIGH") THEN ("99.8") AS ?ITVALUE
IF (?AVAILABILITYLEVEL = "MEDIUM") THEN ("99") AS ?ITVALUE
IF (?AVAILABILITYLEVEL = "LOW") THEN ("96.7") AS ?ITVALUE

```

2. For the response time, the customer asks for a high standard so according to Table 12 it means less than 9ms. “IF standard for response time from business area is HIGH, THEN the response time in the IT layer is less 9ms”.

```

CONSTRUCT ?RESPONSETIMELEVEL hasResponseTimeFromBusinessToIT ?ITVALUE
WHERE
?RESPONSETIMELEVEL is bpaas:AverageResponseTimeLevel
?ITVALUE is bpaas: AverageResponseTimeForITLevel
IF (?RESPONSETIMELEVEL = "HIGH") THEN ("Less 9ms") AS ?ITVALUE
IF (?RESPONSETIMELEVEL = "MEDIUM") THEN ("Between29and10ms") AS ?ITVALUE
IF (?RESPONSETIMELEVEL = "LOW") THEN ("More30ms") AS ?ITVALUE

```

3. The user asks for 100 simultaneous users, the value is the same also for the IT layer because the match is 1:1. “IF value for simultaneous users from business area is 100, THEN the simultaneous user value for IT layer is 100”.
4. The customer sends no more than 80 invoice per month, for the calculation of the data capacity is applied the formula: N. invoice x 1.5mb x time (30 days). In this case, the result is 3.6 Gb per month. “IF invoices per month from the business area are 80, THEN the formula is applied and the data capacity for IT layer is equal to 3.6 Gb”.
5. For the frequency of the backup the user asks for a monthly option. The same choice will be in the IT Layer. “IF backup frequency from business area is monthly, THEN the frequency of backup in the IT layer is monthly”.
6. For the retention time of the backup the user asks for a monthly option as for the frequency. The same time choice will be in the IT Layer. “IF retention time from

business area is monthly, THEN the backup retention time for IT layer is monthly”.

7. The number of invoices, as reported by the customer, are not so high, so for the recovery service a low standard is chosen, according to Table 18; for IT layer low means 50 Gb per hour. “IF standard for restore time from business area is LOW, THEN the restore time for IT layer is equal to 50 Gb per hour”.

```
CONSTRUCT ?RESTORATIONTIMELEVEL hasRestorationTimeFromBusinessToIT ?ITVALUE
WHERE
?RESTORATIONTIMELEVEL is bpaas:DataRestorationTimeLevel
?ITVALUE is bpaas: MximumDataRestorationTimeForITLevel
IF (?RESTORATIONTIMELEVEL = "HIGH") THEN ("TwoHundredGbPerHour") AS
?ITVALUE
IF (?RESTORATIONTIMELEVEL = "MEDIUM") THEN ("OneHundredGbPerHour") AS
?ITVALUE
IF (?RESTORATIONTIMELEVEL = "LOW") THEN ("FiftyGbPerHour") AS ?ITVALUE
```

8. The high standard for the security is translated in the IT language with the use of Blowfish encryption according to Table 16. “IF standard for security from business area is HIGH, THEN the encryption type for the IT layer is equal to blowfish encryption”.

```
CONSTRUCT ?ENCRYPTIONLEVEL hasEncryptionFromBusinessToIT ?ITVALUE
WHERE
? ENCRYPTIONLEVEL is bpaas:EncryptionLevel
?ITVALUE is bpaas: EncryptionTypeForITLevel
IF (?ENCRYPTIONLEVEL = "HIGH") THEN ("Blowfish") AS ?ITVALUE
IF (?ENCRYPTIONLEVEL = "MEDIUM") THEN ("AES") AS ?ITVALUE
IF (?ENCRYPTIONLEVEL = "LOW") THEN ("DES") AS ?ITVALUE
```

9. High data protection is translated, according to Table 19, to a set of places that have laws and regulations that reflect this standard, in this case level B (United States, European Union (EU), Australia, Mexico). “IF standard for data protection from business area is HIGH, THEN the data protection for IT layer is equal to B level”.

```

CONSTRUCT ?PROTECTIONLEVEL hasProtectionFromBusinessToIT ?ITVALUE
WHERE
? PROTECTIONLEVEL is bpaas:ProtectionLevel
?ITVALUE is bpaas: AverageResponseTimeForITLevel
IF (?PROTECTIONLEVEL = "VERYHIGH") THEN ("LevelA") AS ?ITVALUE
IF (?PROTECTIONLEVEL = "HIGH ") THEN ("LevelB") AS ?ITVALUE
IF (?PROTECTIONLEVEL = "MEDIUM") THEN ("LevelC") AS ?ITVALUE
IF (?PROTECTIONLEVEL = "LOW") THEN ("LevelD") AS ?ITVALUE
IF (?PROTECTIONLEVEL = "NONE") THEN ("LevelF") AS ?ITVALUE

```

10. For the service support the customer asks for a service that reflects the working days of the company, so the value for business and IT is the same. “IF type for service support from business area is Mon-Fry, THEN the support service type for IT is Mon-Fry”.
11. For the payment, there is no difference between business and IT. “IF type of payment from business area is prepaid annual plan, THEN the payment plan for IT is prepaid annual plan”.

7.3 Conclusion

In the previous chapter, it was possible to see how to extrapolate, from a given scenario, non-functional requirements in business language and how to match them with the equivalent in IT language. In order to see how these IT specifications are compared with the specifications of some CRM, invoice and email cloud services, and find the best one that reflects the requirements of the customer, the following rule has been created:

```

SELECT ?CS
WHERE
?CS CShasCSSpecification ?CSS
?CSS hasRestObject = "UserState OR Customer OR Invoice OR Attribute OR
Document OR E-Mail OR Template"
?CSS hasRestProperty = "GetLoginSession OR GetRecords OR GetTemplate OR
GetInvoice OR PostInvoice OR PutInvoice OR PosteAttribute OR PostMail"
?CSS hasAvailabilityforITLayer = "100%" AND
?CSS hasResponseTimeforITLayer = "<=9ms" AND
?CSS hasSimultaneousUsersforITLayer = ">=100" AND
?CSS hasCapacityforITLayer = ">=3.6gb" AND
?CSS hasSupportServiceforITLayer = ">= Mon-Fry" AND
?CSS hasBackupRetentionTimeforITLayer = ">=1Month" AND
?CSS hasBackupFrequencyforITLayer = ">=1Month" AND
?CSS hasRestorationTimeforITLayer = ">=50gbperHour" AND
?CSS hasDataProtection = ">=LevelB" AND
?CSS hasDataSecurityforITLayer = "Blowfish" AND
?CSS hasPaymentPlan = "PrepaidAnnualPlan"

```

At this stage the rule conveys the services that reflects the specific of the customer, the results, as showed in the Figure 53, identify “Ninja Invoice and Mailing” and “Ymens CRM” as the best cloud services for the specifications of the client, that also follows the REST standard.

The table, from the left, shows the SLO standards, the requirements of the customer and the specification of the various services. In green, all the specifications that match with those of the user are highlighted: all the values that are equal or higher to the customer’s requests are taken into consideration. The same is applied to data protection the user has requested a high standard; according to Table 19, it corresponding in IT layer to a B level which includes United States, Europe, Australia, Mexico, so all our countries and continents that are part of these macro categories were considered.

Non-Functional Service Level Objectives	Cloud Non-Functional Specification after the transformation	YMENS CRM	Zoho CRM	Ninja (Invoicing System and Mailing)	Simple Invoices (Invoicing System)	Open Source Billing (Invoicing System)	Mailjet (e-mail system)	Gmail (e-mail system)
Availability	100%	100%	100%	100%	99,70%	99%	96%	99,90%
Response Time	Less 9ms	<=1ms	<=5ms	<=5ms	<=1ms	<=2ms	<=1ms	<=1ms
Number of Simultaneous Users	100	500	200	500	400	200	100	100
N. Invoice Per Month for capacity calculation	3.6 Gb	10Gb	5Gb	5Gb	10 Gb	1Gb	1 Gb	10 Gb
Service Support	Mon - Fry	24/7	24/7	Mon - Fry	Mon - Fry	Mon - Fry	24/7	24/7
Backup Retention Time	1 Month	Up to 1 month	Up to 1 month	Up to 1 month	Up to 2 months	Up to 2 month	Up to 1 month	Up to half year
Data Backup Frequency	Monthly	Daily	Monthly	Monthly	Monthly	Monthly	Daily	Daily
Maximum Data Restoration Time	50 Gb Per Hour	50GB per hour	100 GB per hour	50GB per hour	50GB per hour	5 GB per hour	1 GB per hour	5GB per hour
Data Protection	Level B (United States, Europe, Australia, Mexico).	Europe	United States	Italy	China	Spain	Russia	Switzerland
Data Security	Blowfish	Blowfish	AES	Blowfish	AES	AES	Blowfish	Blowfish
Payment Plan	Prepaid annual Plan	Prepaid annual plan	Prepaid annual plan	Monthly fee, Free of charge, Prepaid annual plan	Monthly fee,	Customizable plan	Monthly fee, Try First	Monthly fee, Prepaid annual plan

Figure 53 Matching specification with cloud services

8. Conclusion & Outlook

This chapter describes the last part of the design science research paper as illustrated in the methodology.

Based on the analysis of the problem, namely the communication gap between business and IT, especial regard to SMEs, the study has dealt with research and interviews with people involved in this issue to find a way to align IT and business level. The purpose of this thesis is to go towards a semantic enrichment of business - IT alignment in the cloud.

The approach used to cover this gap was to divide the various levels to analyse and find the best way to annotate them for finding the best way to link layers also if expressed in different languages.

Chapter 5 shows the various methods used for annotation and classification of the levels in order to assign a business value to the equivalent for IT and vice versa. For the functional business side APQC Framework and Object / Action annotations have been used and the same has been used for the IT part with REST architecture associated to Object / Properties. For the non-functional part, tables have been created to help the transformation from an informatics language to a standard one, that may be understandable also by business people, who have no or just little knowledge on this field. These translations have been tested thanks to the help of Mathema that validated the results.

All this knowledge has been implemented in the ontology in chapter 6, where the annotation and concepts were converted and extended into the BPaaS ontology from CloudSocket Project (2017).

The evaluation proposed rules in pseudo-code in order to test the ontologies in a scenario that shows the most suitable cloud services from customer's requirements.

In the last chapter, all the rules used for the conversion and the match with the most appropriate cloud service can be reused and translated into SPARQL, another future implementation that can be the automate translation of the Object / Action annotation with the REST architecture.

Bibliography

- Accorsi, R. (2011). Business process as a service: Chances for remote auditing. *Proceedings - International Computer Software and Applications Conference*, 398–403.
- Aldea, C., Sangeorzan, L., & Aldea, A. (2009). Web Services and Enterprise Games. *Proceedings of the 9th WSEAS International Conference on Simulation, Modeling, and Optimization*, 298–303. Retrieved from <http://www.wseas.us/e-library/conferences/2009/budapest/SMO/SMO53.pdf>
- Álvaro Rocha, Ana Maria Correia, Felix . B Tan, & Karl . A Stroetmann. (2014). *New Perspectives in Information Systems and Technologies*. Springer Science & Business Media.
- APQC. (2017). Productivity and Quality with Performance Measures & Metrics - APQC. Retrieved January 16, 2017, from <https://www.apqc.org/>
- Aysolmaz, B., & Demirors, O. (2014). Deriving user requirements from business process models for automation: A case study. *2014 IEEE 1st International Workshop on the Interrelations Between Requirements Engineering and Business Process Management, REBPM 2014 - Proceedings*, 19–28.
- Barton, T., & Seel, C. (2013). Business Process as a Service – Status and Architecture, 145–158.
- Behrendt, M., & Arsanjani, A. (2011). Introduction and Architecture Overview IBM Cloud Computing Reference Architecture 2.0. *Governance An International Journal Of Policy And Administration*, 1, 1–36.
- Besrou, S., Bin, L., Rahim, A. B., Dominic, P. D. D., Rahim, A. B., & Dominic, P. D. D. (2016). Investigating Requirement Engineering Techniques in The Context of Small and Medium Software Enterprises Investigating Requirement Engineering Techniques in The Context of Small and Medium Software Enterprises, 519–523.
- Boyce, S., & Pahl, C. (2007). *Developing domain ontologies for course content. Educational Technology and Society* (Vol. 10).

- Burr, W. E., Dodson, D. F., Newton, E. M., Perlner, R. A., Polk, W. T., Gupta, S., & Nabbus, E. A. (2013). Electronic Authentication Guideline. *NIST Special Publication, 800*, 112. Retrieved from <http://dx.doi.org/10.6028/NIST.SP.800-63-2>
- C-SIG SLA. (2014). Cloud Service Level Agreement Standardisation Guidelines, 1–41.
- Chandrasekaran, K. (2015). *Essentials of cloud computing*. Boca Raton, FL [u.a.] : CRC Press,.
- CISCO. (2016). RAID Overview, *Cisco UCS*.
- CloudSocket Project. (2017). CLOUDSOCKET. Retrieved March 1, 2017, from <https://www.cloudsocket.eu/>
- Creswell, J. W. (2013). *Research Design Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications.
- Cretella, G., & Di Martino, B. (2012). Semantic web annotation and representation of cloud APIs. *Proceedings - 3rd International Conference on Emerging Intelligent Data and Web Technologies, EIDWT 2012*, 31–37.
- Falquet, G., Métral, C., Teller, J., & Tweed, C. (2011). *Ontologies in Urban Development Projects* (Vol. 1). Retrieved from <http://link.springer.com/10.1007/978-0-85729-724-2>
- Fellbaum, C. (2005). WordNet and wordnets. In *Encyclopedia of Language & Linguistics* (Vol. 13, pp. 665–670).
- Francescomarino, C. Di, Rospocher, M., Ghidini, C., & Valerio, A. (2014). The Role of Semantic Annotations in Business Process Modelling. *2014 IEEE 18th International Enterprise Distributed Object Computing Conference*, 181–189. Retrieved from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6972066>
- Freitas, A., Asooja, K., Jares, J., Decker, S., & Sahay, R. (2014). *A Semantic Web Platform for Automating the Interpretation of Finite Element Bio-simulations*.
- Gruber, T. R. (1993). *A translation approach to portable ontology specifications*.

- Knowledge Acquisition* (Vol. 5). Retrieved from <http://www.sciencedirect.com/science/article/pii/S1042814383710083>
- Haas, H., & Brown, A. (2004). W3C. Web Services Glossary. Definitions. Web service. Retrieved November 27, 2016, from <https://www.w3.org/TR/ws-gloss/>
- Haywood, D. (2012). *Restful Objects*.
- Hevner, A. R., & Chatterjee, S. (2010). Design Research in Information Systems. *Design Research in Information Systems*, 28, 75–105. Retrieved from <http://desrist.org/design-research-in-information-systems>
- Hinkelmann, K., Kurjakovic, S., Lammel, B., Laurenzi, E., & Woitsch, R. (2016). Explanatory Notes : Modelling Prototypes for Bpaas D3.2, 1–52.
- Hinkelmann, K., & Pasquini, A. (2014). Supporting business and IT alignment by modeling business and IT strategy and its relations to enterprise architecture. *Proceedings - 2nd International Conference on Enterprise Systems, ES 2014*, 149–154.
- Ieee. (1990). *IEEE Standard Glossary of Software Engineering Terminology*. Office (Vol. 121990). Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342
- Jeanne W. Ross, & Peter Weill. (2004). *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business Review Press.
- Jung, W., Kim, S. Il, & Kim, H. S. (2013). Ontology modeling for REST Open APIs and web service mash-up method. *International Conference on Information Networking*, 523–528.
- Kang, J., & Sim, K. M. (2010). Cloudle: A multi-criteria cloud service search engine. *Proceedings - 2010 IEEE Asia-Pacific Services Computing Conference, APSCC 2010*, 339–346.
- Koch, B., & Billentis. (2016). *E-Invoicing / E-Billing*.
- Kolisko, L. (2009). SLAs in Cloud Computing. Retrieved January 9, 2017, from

https://blogs.oracle.com/lkolisko/entry/slas_in_cloud_computing

- Kumari, S., & Rath, S. K. (2015). Performance comparison of SOAP and REST based Web Services for Enterprise Application Integration. *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1656–1660.
- Lanthaler, M., Granitzer, M., & Gütl, C. (2010). Semantic Web Services: State of the Art. *Proceedings of the IADIS International Conference on Internet Technologies & Society (ITS 2010)*, 107–114. Retrieved from <http://www.iadisportal.org/digital-library/semantic-web-services-state-of-the-art>
- Lee, Y.-J., & Kim, C.-S. (2010). Building semantic otologies for RESTful web services, 383–386.
- Lee, Y. J., & Kim, C. S. (2011). A learning ontology method for RESTful semantic web services. *Proceedings - 2011 IEEE 9th International Conference on Web Services, ICWS 2011*, 251–258.
- Mathur, M., & Kesarwani, A. (2013). *Comparison Between Des , 3Des ,. Proceedings of National Conference on Horizons in IT-NCNHIT.*
- May, T. (2011). *Social research: Issues, methods and research.* (McGraw-Hill International, Ed.). London: Open University Press; 4 edition.
- Mcdavid, D. (2007). The Business-IT Gap : A Key Challenge. *IBM Almaden Service Research.* Retrieved from <http://almaden.ibm.com/coevolution/pdf/mcdavid.pdf>
- Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. *NIST Special Publication, 145*, 7.
- Nepal, S., Zhang, M., Ranjan, R., Haller, A., & Georgakopoulos, D. (2012). An Ontology-based System for Cloud Infrastructure Services' Discovery. *Proceedings of the 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 524–530. Retrieved from <http://ieeexplore.ieee.org.eresources.shef.ac.uk/ielx5/6415460/6450881/06450944.pdf?tp=&arnumber=6450944&isnumber=6450881%5Cnhttp://ieeexplore.ieee.org>.

- eresources.shef.ac.uk/xpl/freeabs_all.jsp?arnumber=6450944%5Cnhttp://eudl.eu/doi/10.4108/icst.collabora
- Object Management Group (OMG). (2011). Business Process Model and Notation (BPMN) Version 2.0. *Business*, 50(January), 170. <https://doi.org/10.1007/s11576-008-0096-z>
- Oliver Akpan. (2007). *Strategic Alignment: The Business Imperative for Leading Organizations*. Tate Publishing & Enterprises.
- PayStream Advisors. (2015). *Invoice Receipt Management Report 2015*.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45–78.
- Potti, P. K. (2011). On the Design of Web Services: SOAP vs. REST. *Master Thesis*, 106. Retrieved from <http://digitalcommons.unf.edu/cgi/viewcontent.cgi?article=1139&context=etd>
- Qusay H. Mahmoud. (2005). Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI). Retrieved from <http://www.oracle.com/technetwork/systems/soa-142870.html>
- RESTfulAPI.net. (2017). HTTP Methods [RESTful APIs Verbs] – RESTful API Tutorial. Retrieved March 14, 2017, from <http://restfulapi.net/http-methods/>
- Roth, M., Diamantopoulos, T., Klein, E., & Symeonidis, A. (2014). Software Requirements: A new Domain for Semantic Parsers. *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, 50–54. Retrieved from <http://www.aclweb.org/anthology/W/W14/W14-2410>
- Ruiz-Alvarez, A., & Humphrey, M. (2011). An Automated Approach to Cloud Storage Service Selection. *Proceedings of the 2nd International Workshop on Scientific Cloud Computing - ScienceCloud '11*, 3, 39. Retrieved from <http://www.cs.virginia.edu/~ar5je/sci705-ruiz-alvarez.pdf%5Cnhttp://portal.acm.org/citation.cfm?doid=1996109.1996117>

- Sabherwal, R., & Kirs, P. (1994). The Alignment between Organizational Critical Success Factors and Information Technology Capability in Academic Institutions. *Decision Sciences*, 25(2), 301–330. Retrieved from <http://doi.wiley.com/10.1111/j.1540-5915.1994.tb00805.x>
- Saunders, B. M., & Tosey, P. (2013). The Layers of Research Design. *Rapport*, 14(4), 58–59. <https://doi.org/08 jun 2015>
- Saunders, M., Lewis, P., & Thornhill, A. (2009). Understanding research philosophies and approaches. *Research Methods for Business Students* 4, 106–136.
- Schenk, S., & Staab, S. (2008). *Networked graphs: a declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web. Proceeding of the 17th international conference on World Wide Web - WWW '08*. New York, New York, USA: ACM Press.
- Silver, B. (2011). *BPMN method and style : with BPMN implementer's guide*. Cody-Cassidy Press.
- Singh, G. (2013). *A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. International Journal of Computer Applications* (Vol. 67).
- Sommerville, I. (2010). *Software Engineering 9. Software Engineering* (Ninth Edit).
- Sommerville, I., & Sawyer, P. (Pete H. . (1997). *Requirements engineering : a good practice guide*. Wiley.
- Studer, R., Benjamins, V. R., & Fensel, D. (1998). *Knowledge engineering: Principles and methods. Data & Knowledge Engineering* (Vol. 25).
- The Open Group. (2013). *ArchiMate 2.1 Specification*.
- Ukolova, O. (2016). *Modelling requirements for Business Process as a Service*.
- Uschold, M., Gruninger, M., Uschold, M., & Gruninger, M. (1996). *Ontologies : Principles , Methods and Applications. Knowledge Engineering Review* (Vol. 11).
- Vaishnavi, V., & Kuechler, B. (2004). *Design Science Research in Information Systems*

- Overview of Design Science Research. *Ais*, 45. Retrieved from <http://www.desrist.org/design-research-in-information-systems/>
- Vaishnavi, V., & Kuechler, W. (2008). *Design science research methods and patterns : innovating information and communication technology*. Auerbach Publications.
- Van der Wees Arthur, Daniele, C., Jesus, L., Edwards Mike, Schifano Nicholas, & Maddalena, S. L. (2014). Cloud Service Level Agreement Standardisation Guidelines. *C-Sig Sla*, 1–41.
- van Welie, M., van der Veer, G. C., & Eliëns, A. (1998). An Ontology for Task World Models (pp. 57–70). Springer, Vienna.
- W3C. (2012). OWL - Semantic Web Standards. Retrieved March 17, 2017, from <https://www.w3.org/OWL/>
- W3C. (2014). RDF - Semantic Web Standards. Retrieved March 17, 2017, from <https://www.w3.org/RDF/>
- Woitsch, R., & Utz, W. (2016). Business Processes as a Service (BPaaS): A model-based approach to align business with cloud offerings. *eChallenges E-2015 Conference Proceedings*, 1–8.
- Zowghi, D., & Coulin, C. (2005). Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In *Engineering and Managing Software Requirements* (pp. 19–46). Berlin/Heidelberg: Springer-Verlag.

Abbreviations

APQC	American Productivity and quality Center
AR	Accounts Receivable
BPaaS	Business Process as a Service
BPMN	Business Process Model and Notation
CRM	Customer Relationship Management
CSV	Comma-separate values
DSR	Design Science Research
ERP	Enterprise Resource Planning
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
JSON	JavaScript Object Notation
NIST	National Institute of Standards and Technology
OMG	Object Management Group
OWL	Web Ontology Language
PaaS	Platform as a Service
PCF	Process classification framework
RDF	Resource Description Framework

RE	Requirement Engineering
REST	Representational state transfer
SaaS	Software as a Service
SLA	Service Level Agreement
SLO	Service Level Objective
SME	Small and Medium Enterprise
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
URI	Uniform Resource Identifier
WSDL	Web Services Description Language
XML	Extensible Mark-up Language

List of Figures / Tables

Figures

Figure 1 The NIST cloud definition framework. (NIST, 2009)	8
Figure 2 Classification of Cloud Services	9
Figure 3 Initial High-level Architecture of CloudSocket (CloudSocket 2015a)	10
Figure 4 Spiral view of the requirements engineering process	12
Figure 5 Main differences between Functional and Non-Functional Requirements	13
Figure 6 Different types of Non-Functional Requirements	15
Figure 7 Categories of APQC process classification framework	17
Figure 8 Class hierarchy for modelling requirements	18
Figure 9 Welie's approach to the ontology	19
Figure 10 Communication through a Web Service	20
Figure 11 RESTful APIs in BPaaS architecture	24
Figure 12 Example of cloud ontology derived from Cloudle.....	25
Figure 13 Research onion according to Creswell	31
Figure 14 Deductive and Inductive approach.....	34
Figure 15 Research choice methods	36
Figure 16 Design Science Research Framework (Vaishnavi & Kuechler, 2008)	38
Figure 17 Invoice/Bill volume breakdown by industry	42
Figure 18 Evolution phases of the market's experiences with e-billing system	43
Figure 19 Maturity level of the market in the world	44

Figure 20 Saving potential for invoice/bill issuers	44
Figure 21 Saving potential for invoice recipients.....	45
Figure 22 Business Process Diagram Send Invoice	47
Figure 23 Workflow Invoice Service	49
Figure 24 NonFunctionalRequirements cloudservices (CloudSocket Project, 2017)	51
Figure 25 Task ontology for classification of Action and Object	57
Figure 26 Part of Action and Object taxonomy.....	58
Figure 27 SOAP / REST characteristics.....	59
Figure 28 Relationship between resources	60
Figure 29 Adapted from ArchiMEO ontology.	70
Figure 30 Business process functional requirement ontology.....	71
Figure 31 Motivation extension metamodel (The Open Group, 2013)	72
Figure 32 APQC conversion process (CloudSocket Project, 2017)	72
Figure 33 Part of the APQC ontology	73
Figure 34 Action and object taxonomy	74
Figure 35 Action part of the ontology with "Move" example	75
Figure 36 Object part of the ontology with "Invoice" example	76
Figure 37 Business process functional requirement annotation	76
Figure 38 Rest ontology	77
Figure 39 Cloud service functional specification	78
Figure 40 Business Process Non-Functional requirement.....	79

Figure 41 Business process non-functional requirements	80
Figure 42 Service level agreement ontology representation	81
Figure 43 Cloud service non-functional specifications	82
Figure 44 Cloud services invoices for Send Invoice process	83
Figure 45 Cloud service properties.....	83
Figure 46 CloudServiceSpecification instances	83
Figure 47 CloudServiceSpecification properties	84
Figure 48 NinjaInvoice Instance properties.....	84
Figure 49 BPaaS Ontology example	86
50 Example of business annotation using APQC and Object/Action taxonomy	87
Figure 51 Example of workflow annotation using object / property classification.....	87
Figure 52 Business - IT functional alignment	88
Figure 53 Matching specification with cloud services	94

Tables

Table 1 Main characteristics and differences between SOAP and REST methodology	23
Table 2 Philosophical Assumption of the Three Research Perspectives according to (Vaishnavi & Kuechler, 2004).....	32
Table 3 APQC top level categories	55
Table 4 APQC classification relative at the invoice process.....	56
Table 5 Performance Service Level Objectives Overview	61
Table 6 Security Service Level Objectives Overview	62

Table 7 Data Management Service Level Objectives Overview	62
Table 8 Personal Data Protection Service Level Objectives Overview	62
Table 9 Payment Service Level Objectives Overview	62
Table 10 Non-functional Business - IT layer matching	63
Table 11 Availability description	64
Table 12 Average Response Time description	64
Table 13 Maximum resources capacity description	64
Table 14 Support hours description.....	65
Table 15 Authentication description.....	65
Table 16 Cryptographic Brute Force Resistance description	66
Table 17 Data backup method description	66
Table 18 Maximum data restoration time description.....	66
Table 19 Data geolocation list description	67
Table 20 Transformation from business to IT	89