

Universidade do Algarve

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrónica e Informática

Formalization of an Enterprise ontology Based on Agents and Contexts

Héctor Rodrigo Rubilar Saavedra

Mestrado em Engenharia Informática

2011

Universidade do Algarve

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrónica e Informática

Formalization of an Enterprise ontology Based on Agents and Contexts

Héctor Rodrigo Rubilar Saavedra

Mestrado em Engenharia Informática

Dissertação orientada por: Prof^a. Doutora Marielba Zacarias.

2011

Abstract

Communication it is a fundamental organizational aspect but not always easy to achieve due to differences between the parts involved (distinct beliefs, backgrounds and needs). Poor communication can lead the organization to several problems like poor resource re-utilization, deficient inter-operability, underutilization of information, among others. These communication problems are translated into information flux problems (we do not use efficiently the information, we do not understand each other and we do not define problems or propose solutions adequately). Addressing these problems its crucial in every organization that understands the value that information has.

Ontologies help us solve the previous situations. They provide a common understanding of a conceptualization of the study object so they allow us to model the study object and to describe it using the same language. Enterprise ontologies allow us model an organization like the enterprise using a unambiguous and commonly accepted terminology who define enterprise terms like activity, resource or process with which we can represent enterprise operation and manage information efficiently. Commonly, enterprise ontologies model formal elements of an organization, but since enterprises are composed by parts of different nature, informal elements emerge and create relations and different perspectives with respect to the rest of the enterprise components.

This document study an agent based and context aware framework who encompasses both, formal and informal element of the enterprise. This work results on a formalized OWL ontology that will allow us to answer question with information inferred from the model.

KEYWORDS: ontology, enterprise model, context, agent oriented, rules, queries.

Resumo

A comunicação é um aspecto fundamental da organização, mas nem sempre fácil de alcançar, devido às diferenças entre as partes envolvidas (distintas crenças e necessidades). A comunicação deficiente pode levar à organização a vários problemas como má utilização dos recursos, inter-operacionalidade deficiente, a subutilização da informação, entre outros. Esses problemas de comunicação são traduzidos em problemas de fluxo de informação (não usamos de forma eficiente a informação e não entendemos um ao outro). A resolução destes problemas é crucial em todas as organizações que entendem o valor que a informação tem.

As Ontologias ajudam-nos a resolver as situações anteriores, fornecem um entendimento comum da conceituação do objecto de estudo permitindo-nos modelá-lo e descrevê-lo usando a mesma linguagem. As ontologias empresariais permitem-nos modelar uma organização como a empresa através do uso de uma inequívoca e comumente aceite terminologia para definir termos de empresa, como recursos, actividades ou processos com o qual podemos representar a operação da empresa e gerir a informação de forma eficiente. Tipicamente as ontologias empresariais modelam os elementos formais de uma organização, mas uma vez que as empresas são compostas por peças de natureza diferente, elementos informais emergem e relacionam-se com o resto dos componentes da empresa. Este documento estuda um framework baseado em agentes e contextos que abrange os elementos formais e informais da empresa. Este trabalho tem como resultado uma ontologia formalizada e escrita na linguagem OWL que é capaz de representar uma empresa nos seus diferentes níveis englobando ambos elementos formais e informais e a qual responde a questões (através de regras e pesquisas) obtendo informação que não está explicitamente presente no modelo.

PALAVRAS CHAVE: ontologia, modelo empresarial, contexto, modelo orientado a agente, regras, metodologias.

Resumo Alargado

A comunicação é uma questão fundamental em todos os aspectos das nossas vidas. Pessoas, organizações, sistemas de software, etcetera, precisam se comunicar entre si. Esta comunicação nem sempre é fácil (às vezes nem mesmo possível) devido às diferenças entre as partes envolvidas. Essas diferenças podem ser devido a muitas razões, tais como: distintas crenças, origens, necessidades, e linguagens de comunicação, entre outras. Todas essas diferenças contribuem para uma “má comunicação”, que por sua vez pode conduzir-nos para os seguintes problemas:

- Pobre re-utilização dos recursos. Neste caso podemos encontrar trabalhos ou investigação realizada numa dada área, materiais que não são utilizado/as por outros investigadores, dado que os resultados embora que relevantes, estão expressos em termos que não são consistentes ou que até são desconhecidos para os investigadores.
- Deficiente inter-operacionalidade. Pessoas, organizações e ferramentas de software precisam se comunicar entre si, especialmente quando se trata de software em grandes organizações ou na integração de sistemas. Neste último caso, as saídas de uma ferramenta podem não estar legíveis ou compreensíveis para outras ferramentas, ou ainda apenas uma parte das informações de saída de uma ferramenta poderá servir como entrada na outra ferramenta, em ambos os casos estamos a perder dados importantes.
- Identificação do problema/solução não eficientes. Em projectos grandes a má comunicação entre as pessoas pode transformar muito difícil a tarefa de identificar problemas. Pessoas com diferentes perspectivas podem ver os mesmos problemas de forma diferente. Assim enfrentamos dificuldades na identificação de problemas e, conseqüentemente, para encontrar soluções para os mesmos.

- Desperdício de esforços. Embora possamos dispor de bons resultados nas pesquisas, de ferramentas de software adequadas e equipas de trabalho competentes, se não utilizarmos correctamente estes mesmos recursos, serão necessários esforços adicionais para atingir os nossos objectivos, isto quando os resultados já poderiam estar nas nossas mãos.
- Utilização de informação deficiente. Os problemas de comunicação reduzem o poder da nossa informação, isto é, impedindo a utilização dos nossos recursos de informação de maneira correcta e direita e impedindo a produção de nova informação (as inferências são limitadas, se mesmo possíveis).

Enfrentar esses problemas é crucial em todas as organizações que entendem o valor que a informação tem. Os problemas citados são questões relacionadas com recursos e não resolvê-los pode tornar a organização menos competitiva ou fazê-la desaparecer.

As ontologias ajudam-nos resolver ou melhorar o desempenho nas situações anteriores. Estas proporcionam entendimento comum sobre a conceituação dum objecto em estudo e permitem-nos modelá-lo e descrevê-lo usando a mesma linguagem.

Ao fazer isto, estamos a melhorar a comunicação entre participantes e partes. Como resultado, as ontologias podem melhorar a inter-operacionalidade permitindo a comunicação entre as diferentes entidades do sistema, aumentar a re-utilização dos recursos tornando as saídas dos processos compreensíveis para todos no sistema, conduzir a uma melhor identificação do problema/solução evitando a ambiguidade na definição dos termos chave, evitar o desperdício de esforços (tentando obter os resultados que já temos) através da padronização das entradas e saídas de processos usando uma única linguagem e também incrementar o valor das nossas informações por permitir inferências.

As ontologias empresariais aplicam todas estas vantagens no mundo empresarial. Modelam a empresa com uma terminologia comum e sem ambiguidades. Costumam definir termos como actividade, recursos e processos, entre outros, que representam a operação da empresa e gerem informações de forma eficiente.

Algumas das ontologias empresariais mais conhecidas são: The Enterprise Ontology (EO), o Toronto Virtual Enterprise (TOVE) e a ontologia empresarial de Dietz. Estas desenvolvem termos essenciais da empresa e as relações que representam as actividades, recursos e processos.

Apesar de que as mais conhecidas ontologias empresariais serem muito úteis para modelar este tipo de organizações, abordam apenas os seus elementos formais, mas as empresas são compostas por peças de natureza diferente, e por tanto, elementos informais emergem e criam relações com o resto dos componentes da organização. Portanto é necessário abordar os elementos formais e informais da organização para o modelo ser o mais fiel possível.

O presente trabalho estuda um framework conceitual baseado em agentes e contextos (desenvolvido no âmbito do trabalho realizado para uma tese de doutoramento), que leva em consideração os elementos formais e informais da empresa. Em particular, esta abordagem tem como objectivo principal tratar a necessidade de uma perspectiva centrada nos agentes e ciente dos contextos, para alinhar as pessoas e organizações, considerando os comportamentos de agentes e empregando a noção de contexto como a chave de ligação entre conceitos centrados no agente com outras perspectivas da empresa.

Para lidar com a complexidade organizacional, juntamente com a complexidade do agente individual, o framework define pontos de vista para captar os diferentes

comportamentos dos agentes, fornecer meios de traçado do comportamento de um único agente e permitir ligações entre o comportamento do agente com outras perspectivas de arquitectura. A arquitectura de agente apresentada aqui é composta por três camadas interdependentes: a deliberação, acção e mudança/aprendizagem.

Para formalizar a ontologia três metodologias foram revisadas, o objectivo deste trabalho foi desenvolver nossa própria metodologia de formalização tendo em conta experiências anteriores. Como resultado, foi desenvolvida uma metodologia iterativa baseada em reuniões regulares e análise de documentos. Esta metodologia é baseada em uma abordagem cliente-fornecedor, onde o cliente é o autor do framework e o provedor é o autor do presente trabalho.

Também foi realizada uma pesquisa comparando quatro ferramentas de desenvolvimento. Após isto, decidimos trabalhar com Protégé como um editor de ontologia principal e com COE como ferramenta de mapas conceituais. A formalização foi concluída com a população da ontologia através da introdução de casos exemplo, os quais foram extraídos da tese do autor do framework.

Após a conclusão da formalização, um conjunto de regras foram adicionadas à ontologia com o objectivo de obter relações que não estivessem explicitamente definidas e obter nova informação. Junto com as regras, queries foram desenvolvidas de modo que seja possível extrair informações específicas da ontologia. No fim do presente trabalho, os resultados de regras e queries foram avaliados comparando-os com a informação presente na tese do autor do framework.

Os resultados da avaliação foram em sua maioria favoráveis e ajudaram-nos a mostrar que as ontologias que consideram elementos formais e informais da empresa são uma forma adequada para modelar-la.

As ontologias são altamente escaláveis, nos permitem integrá-las com outras ontologias ou adicionar-lhes elementos de maneira relativamente fácil. A natureza dinâmica das empresas exige modelos dinâmicos para representá-los e as ontologias estão à altura desse desafio.

O resultado final deste trabalho é a formalização do framework em uma ontologia formal, escrita em linguagem OWL e capaz de responder a queries com informações inferidas.

PALAVRAS CHAVE: ontologia, modelo empresarial, contexto, modelo orientado a agente, regras, metodologias.

Contents

Abstract	ii
Resumo	iii
Resumo Alargado	iv
1 Introduction	1
1.1 Motivation	2
1.2 Objectives	3
1.2.1 General Objectives	3
1.2.2 Specific Objectives	3
1.3 Work Limitations	4
1.4 Work Outcome	5
1.5 Chapters Summary	5
2 Background	6
2.1 Ontologies	6
2.1.1 Formal Definition	6
2.1.2 Ontology in Related Fields	8
2.1.3 Formalization Degree	9
2.1.4 Application Areas	11
2.1.5 Specific Application	12

2.2	Enterprise Ontologies	17
2.2.1	The Enterprise Ontology (EO)	19
2.2.2	Toronto Virtual Enterprise (TOVE)	24
2.2.3	Enterprise Ontology by Dietz	27
2.3	Agent Based Model	32
2.3.1	Framework based on Agents and Contexts for the Alignment between Individuals and Organizations	33
2.3.1.1	Fundamental Concepts	34
2.3.1.2	Additional remarks	38
2.3.1.3	Addressing Complexity	40
2.4	Methodologies Review	51
2.4.1	Uschold and King Methodology	52
2.4.2	Grüninger and Fox Methodology	53
2.4.3	Methontology	55
2.4.4	Methodology Comparison	57
2.5	Tool Review	60
2.5.1	Choosing a Formalization Tool	68
3	Formalization Methodology	70
3.1	Brief Description	70
3.2	Methodology	71
3.2.1	Documentation Review	71
3.2.2	Information Gathering	73
3.2.3	Informal Conceptualization	74
3.2.4	Conceptual Map Formalization	74
3.2.5	Conceptual Map - Protégé Transition	75
3.2.6	Ontology Refinement	76
3.2.7	Ontology Population	76
3.2.8	Rule addition and Query Testing	77

4 Formalization Results	80
4.1 Methodology Results	80
4.1.1 Documentation Review Outcome	80
4.1.2 Information Gathering Outcome	81
4.1.3 Informal Conceptualization Outcome	83
4.1.4 Conceptual Map Formalization Outcome	85
4.1.5 Conceptual Map - Protégé Transition Outcome	85
4.1.6 Ontology Refinement Outcome	90
4.1.7 Ontology Population Outcome	97
4.1.8 Rule addition and Query Testing Outcome	101
4.2 Results Evaluation	116
5 Conclusion and Future Work	126
5.1 Future Work	127
Bibliography	129
APPENDIX	133
A OWL Ontology extract	133
B Informal conceptualization sketches	138
C Prof. Smith's payment context	141

Chapter 1

Introduction

Communication is a fundamental issue in every aspect of our lives. People, organizations, software systems, etcetera, need to communicate each other. This communication is not always easy (sometimes not even possible) due to differences between parts involved. These differences could be due to many reasons, like: distinct beliefs, backgrounds, needs, contexts, even completely different communication languages, among others. All these differences produce “poor communication” which in turn can lead us to several problems like poor resource re-utilization, deficient inter-operability, deficient problem and solution identification, waste of efforts, underutilization of information, among others. Addressing these problems it is crucial in every organization that understands the value that information has. The mentioned problems are resources related issues and do not address them can make the organization less competitive or even disappear.

The problem of “under-utilization of information” (not using the information efficiently or its sub-utilization) it is an issue that must be specially addressed by organizations. Logically, if we spend our time and money trying to obtain information resources, we should try also to get the most out of them, otherwise we are wasting time and money (another resources).

Ontologies help us solve or improve the performance on the previous situations. Providing a common understanding of a conceptualization of the study object they allow us to model the study object and to describe it using the same language. By doing this we are improving communication among the participants and parts. As a result, ontologies can improve inter-operability by allowing communication among different system entities, increase resource re-utilization by making processes outputs understandable for everyone on the system, lead to a better problem/solution identification avoiding ambiguity in term definition, avoid waste of efforts (trying to obtain results that we already have) by standardize processes outputs and processes inputs using the same language, and increment the value of our information by allowing inferences.

1.1 Motivation

Ontologies are a domain formal specification that allow us to answer specific questions and infer new information starting from domain data.

Enterprise ontologies allow us to describe how enterprise operate and communicate. Enterprises are complex systems, so to describe them, different (heterogeneous) ontologies might be used (to be able to describe activities, resources and represent time). Commonly used (and accepted) ontologies allow us to capture and model formal element of the organization, but organizations are not only about formal elements. Since organizations are also composed by human beings informal elements appears on it. It is desirable and necessary represent both, formal and informal aspects of the enterprise.

The enterprise framework presented in Chapter 2 corresponds to a conceptual

framework for an agent-context based ontology which take into consideration formal and informal elements of organization.

The motivation of this work is to have a functional and formalized agent-context enterprise ontology that covers formal and informal aspects of the organization and which is able to retrieve information which is not explicitly present on the ontology.

1.2 Objectives

The objectives are presented in two groups: General objectives and Specific objectives.

1.2.1 General Objectives

The general objective of this work is to do a formal specification of a enterprise ontology based on enterprise agents and contexts. This ontology was developed within the work done for the Ph.D. Thesis presented in [29] and complements existing ontologies by allowing not only capturing and modelling informal aspects of the organization but also the formal ones.

1.2.2 Specific Objectives

This work has the following specific objectives:

1. Analyse related work concerning ontologies, enterprise ontologies and ontology development methodologies.
2. Analyse the agent-context based ontology described on this work and originally proposed in [29].

3. Relate key concepts of existent ontologies with concepts of the agent-context based ontology.
4. Analyse some of the available ontology development tools and select one to formalize the ontology.
5. Definition of our ontology formalization methodology.
6. Semi-formal specification of the ontology using conceptual maps.
7. Formalization of the ontology in OWL language using the selected tool.
8. Using rules or DL queries¹ , the ontology must be able of retrieve the following information:
 - (a) Which actions are associated to a given context.
 - (b) Commitment of a given agent.
 - (c) Completed commitment of a given agent.
 - (d) Execution context of a given agent.
 - (e) Coordination context of a given agent.
 - (f) Inter personal context.
 - (g) Inter-personal interactions.
9. Validate the ontology.

1.3 Work Limitations

The present formalization only addresses personal and interpersonal relations (individual actions and interactions), not reaching groups and organizational levels.

¹Query language based on the Manchester OWL syntax

Time (temporal aspects of events) is not addressed on the present work due mainly to a reasoner restriction (whit the actual version of Pellet is not possible to work with time). Previous version of this reasoner allowed working with time, but is always preferable work with last versions of software tools.

1.4 Work Outcome

The final result of this work is the formalized ontology, written on OWL language which is able to answer the questions described on the previous section.

1.5 Chapters Summary

This document is organized as follows. Chapter 1 contains introduction to the document, motivation and objectives. Chapter 2 contains necessary background about ontologies, enterprise ontologies, conceptual framework based on agent and contexts, ontology development methodologies and ontology editors (software tools). Chapter 3 contains the methodology used to formalize the ontology and the results of every stage of it. Chapter 4 contains the result of the work done and its correspondent evaluation. Finally, in chapter 5 we present the conclusion and a future work discussion.

Chapter 2

Background

On this chapter we revise background concepts and the related work necessary to start the formalization process. We will describe here ontologies, its types, formalization degree and some application areas. Also we define enterprise ontologies, we show some popular enterprise ontologies and we present the approach of agent and context based enterprise ontology. We also describe some methodologies for develop enterprise ontologies and we compare and select a formalization software tool to work with it.

2.1 Ontologies

On this section we define the term ontology, we show the utilization of ontologies in related fields, we also give a notion of different formalization degrees of ontologies and we describe some application areas and specific applications of ontologies.

2.1.1 Formal Definition

On the literature we can find many definitions of the ontology concept, some of them (the most common and applicable on this study field) are the following:

- Ontology is a formal explicit description of concepts in a domain of discourse (classes, sometimes called concepts), properties of each concept describing various features and attributes of the concept (slots, sometimes called roles or properties), and restrictions on slots (facets, sometimes called role restrictions) [18].
- Ontology is an explicit specification of a conceptualization [10].
- Ontology is the shared understanding of some domain of interest which may be used as a unifying framework to solve problems [26].

These 3 definitions have some common elements which we had combined to have a single sentence defining the term ontology with which we will work. This combined sentence is: “a formal, explicit and shared description of concepts, properties and restrictions of a domain of interest (conceptualization) which may be used as a unifying framework to solve problems”.

Three key concepts compose this definition, these are:

Formal. We need our ontology to be machine readable.

Explicit. No ambiguity is allowed on term and concept definition.

Shared. Definitions must be agreed between all the people involved on the ontology (people may be from different fields and have different backgrounds and needs).

This definition has some degree of generality. More specific definitions can be found for more specific utilization of ontologies and these depends on the field in which we want to use ontologies and on its purposes.

2.1.2 Ontology in Related Fields

Depending on its purposes, in [25] we found three main uses in different fields for different ontology types:

Communication between people. In this case, informal ontologies are allowed (but they still have to be unambiguous).

Inter-Operability among systems. In this case the ontology is used as a interchange method, which can make this translation between different paradigms, languages and software tools.

Systems Engineering Benefits. Which are:

- *Re-Usability.* Formal representation on an ontology may be used as a shared component in a software system.
- *Knowledge Acquisition.* If ontologies are used to gather knowledge on the development of some software system, some extra degree of reliability can be achieved. This also can be a positive factor in increasing early development speed.
- *Reliability.* Ontologies give us the possibility of automatic consistency checking resulting on a more reliable final product.
- *Specification.* Defining specifications and identifying requirements are tasks supported by the ontology.

As different utilizations and purposes, ontologies also have different degrees of formalization depending on what we want to represent and the level of detail we want to achieve.

2.1.3 Formalization Degree

Uschold, in [25] define four degrees of formalization to classify ontologies:

1. *Highly Informal*. Which corresponds to ontologies expressed in natural language.
2. *Structured Informal*. Which corresponds to ontologies expressed in a restricted and structured form of natural language. Unlike the previous formalization degree, here the clarity is increased by reducing ambiguity.
3. *Semi-Formal*. Which corresponds to ontologies expressed in an artificial and formally defined language.
4. *Rigorously Formal*. Which corresponds to ontologies with terms accurately defined, formal semantics, theorems and proofs of such properties, as soundness and completeness.

With similar sense, McGuinness in [15] define the following ontology spectrum, which contain a linear organization of possible ontology schemes ordered ascendantly by expressiveness. This spectrum is represented in Figure 2.1 and is described as follows:

- *Controlled Vocabulary*. A finite list of terms (e.g. a catalog). This is a simple approach of a possible ontology.
- *Glossary*. Corresponds to lists of terms with its meanings. Semantics are given here by the meanings which are expressed in natural language which can be read and interpreted by humans. As these interpretations are not unambiguous, they are not machine processable.

- *Thesauri*. In this case, more semantic information is given (such as synonym relationships). Agents can interpret unambiguously this relationships. Hierarchy can be deduced (typically is not explicit).
- *Informal “is a”*. Here, the relationship is not a strict ”is a”. This means that may occur the case that an instance of a more specific class is also an instance of the more general class but this is not always enforced. Without this enforcement deduction in ontologies might be sometimes complicated (or even not possible).
- *Formal “is a”*. Corresponds to the strict subclass hierarchies. This condition is necessary for exploitation of inheritance.
- *Formal Instances*. In this case class names, ground individual content and instances are included on the scheme.
- *Frames*. Here, classes also include property information (which are inherited by subclasses).
- *Value Restrictions*. Restrictions are placed on what can fill a property.

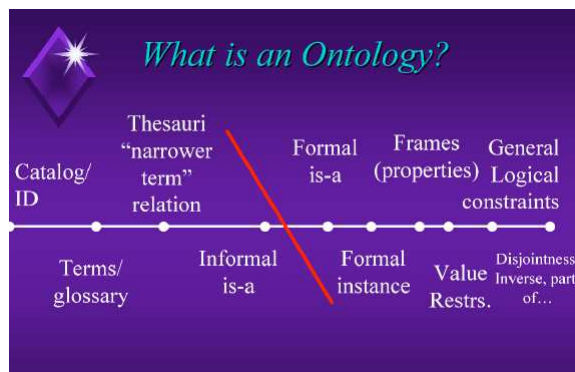


Figure 2.1: An ontology spectrum [15]

As we could see on the ontology spectrum, the expressiveness of the ontology should grow if we need to express more information.

Ontologies have a wide range of applications, in different areas and to accomplish different objectives.

2.1.4 Application Areas

Some of the most important application areas for ontologies that we can find on the literature are:

Communication: Ontologies are a shared understanding of some conceptualization, based on this they avoid ambiguity, which facilitates the communication since all (communication) parts are “speaking” the same language.

Inter-operability: Here we have users with the need of exchange data between different software tools. Ontologies allows communicating between the different applications.

Re-utilization of domain knowledge: In this case we can have an ontology developed by some group of researchers. This ontology can be used for others researchers in case they need it, avoiding the task of developing a new ontology from scratch. Also this ontology can be combined with other ontology on larger projects.

Analyse domain knowledge: In cases when one want to extend a ontology or simply work with it, if a declarative specification is available for the terms, its possible to analyse these terms checking if the ontology is suitable for work with it.

2.1.5 Specific Application

We can find several specific applications involving ontologies in different fields, some of them are described below.

Requirement Engineering

In this area, ontologies are used to specify requirements and domain knowledge. Its degree of expressiveness can be adapted to specific needs so ontologies can cover semi-formal and structured as well as formal representations [23].

Besides the specification of requirements, ontologies has the capability of automate validation and consistency checking.

Component Reuse

The re-utilization of components is a useful way to efficiently utilize resources which helps to save money and efforts (associated to re-work). To search for components for re-utilization its necessary to look into repositories, and since most re-utilization repositories are limited to a plain, syntactical key-word based search, they are suffering from low precision (due to homonyms) and low recall (due to synonyms) [16].

To help in the task of component searching, ontologies can join information about several component descriptions and also describe the functionality of components so more accurate answers are obtained.

Business Rules

The business logics corresponds to the software systems mechanisms to satisfy the business policies of an organization. Business logics are usually coded in programming languages which leads to modify the source code when business logics change.

Using business rules is possible to separate business logic from processing logic. Here, business logic is modelled declaratively with logical statements and processed by a rule engine. While business rule engines are available for quite some time, they can be regarded as “ontology-based” approaches towards software engineering since they run declarative knowledge on a special middleware [23].

Since they are no implicitly embedded on the source code, business rules can be changed more easily.

Semantic Middleware

In three level architectures, middleware layer can have lots of advantages for the application developer in terms of complexity reduction, but to deal with interdependences between modules or legal constrains make the management of middleware based systems a more difficult task [23].

Ontologies can capture properties, relationships and behaviors of components of the system, which can provide a precise and formal definition of some ambiguous terms from software engineering (like “component” or “service”) as well as structures supporting the formalization of middleware knowledge [23]. In this case, the characteristic of capture knowledge of ontologies is used to deal with interdependence issues.

One example of this are the application servers, which are component-based middleware platforms offering an environment in which users can deploy self-developed or third-party components [1].

Figure 2.2 shows the architecture of the utilization of ontologies in application servers. We can find in the left side potential input sources for the framework. This information is parsed and converted into semantic meta-data (data structured to fit

into the conceptual model). Semantic meta-data and the ontology are fed into the inference engine which in turn is embedded in the application server [19].

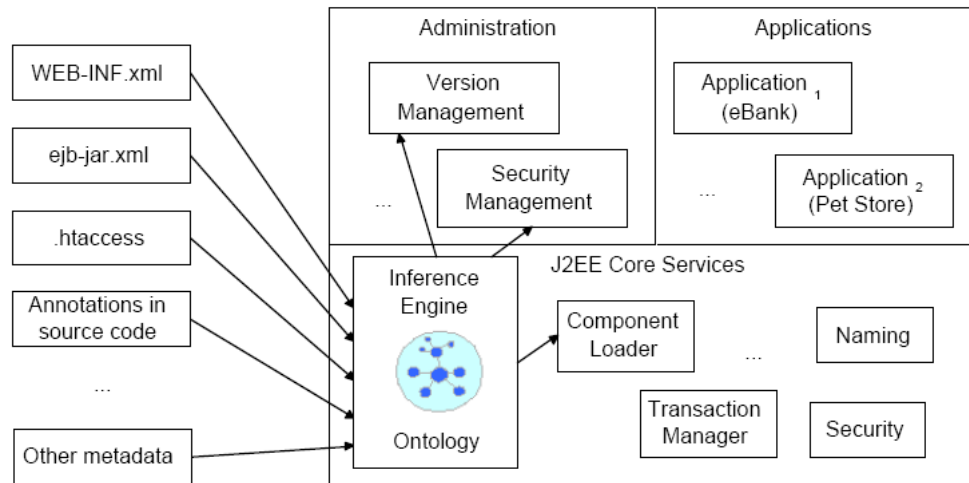


Figure 2.2: Utilization of ontologies in application servers

Semantic WEB

In the beginnings of the WEB, its content sharing characteristic was a key factor for its fast growth. Here, people and organizations were able to write pages by hand. Also there were some software tools that allowed non experienced people to create Websites without any special qualification. These pages were aimed to be read by humans rather than by machines.

Along with the years the tendency continues, most of the pages were developed to be understandable for humans and not for machines. This is a considerable obstacle in the common task of search for content, because although we have machines to search and classify, we still need humans to review the results in order to look for the

answers they are looking for.

Even with the most popular Web search engines, as Google, Yahoo or Bing, we obtain a large amount of pages that may contain (on some part of them) something that the search engine “thought” its the answer to our query. This is due to the ambiguity on the definition of terms and contents and to the fact that most of the contents on the WEB are just documents, not (structured) data [20].

If that large portion of pages were developed to be machine readable (marked up with information concerning what services could be obtained and how that services or information could be obtained [15]) we could use software tools to obtain more accurate and precise answers to our queries.

One of the most important applications of ontologies is the semantic WEB. According to the World Wide WEB Consortium (W3C) Semantic WEB is a WEB of data, rather than a WEB of just documents, and its vision of the Semantic WEB is to extend WEB principles from documents to data [20].

The ultimate goal of the WEB of data is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network. The term “Semantic WEB refers to W3Cs vision of the WEB of linked data [21]. All this work is supported by technologies like RDF, SPARQL, OWL, and SKOS ¹.

Behind the Semantic WEB we can find key elements on which its constructed, these are [21]:

Linked Data. Huge amount of data and relationships among it, structured in a

¹RDF: Resource Description Framework; SPARQL: SPARQL Protocol and RDF Query Language; OWL: Web Ontology Language; SKOS: Simple Knowledge Organization System

standard format that is reachable and manageable by Semantic WEB tools. Here, some technologies like RDF or SPARQL are used to structure and access data, respectively.

Vocabularies. Define, classify and characterize concepts, relationships and constraints of the domain of interest. The word ontology is used to refer to complex and possibly formal vocabularies. Vocabularies are fundamental to apply inference techniques on the Semantic WEB.

Query. Technologies that allow us to obtain information from the WEB of Data. Here, huge amount of data can be accessed and useful (specific) information retrieved.

Inference. Capability of generate new information based on existing data and additional information (set of rules).

Vertical Applications. Is the term used at W3C to denote particular, generic application areas, specific communities, etcetera. That explore how W3C technologies can help their operations, improve their efficiencies and provide better user experiences.

Figure 2.3 shows the Semantic Web layers. The definition of the layers is as follows [12]:

- *Unicode and URI layers.* Make sure that we use international characters sets and provide means for identifying the objects in Semantic Web.
- *XML layer + namespace + schema definitions.* Make sure we can integrate the Semantic Web definitions with the other XML based standards.
- *RDF (RDF) and RDFSchema (RDFS).* Allows to make statements about objects with URI's and define vocabularies that can be referred to by URI's. This is the layer where we can give types to resources and links.

- *Ontology layer.* Supports the evolution of vocabularies as it can define relations between the different concepts.
- *Digital Signature layer for detecting alterations to documents.* Layers that are currently being standardized in W3C working groups.

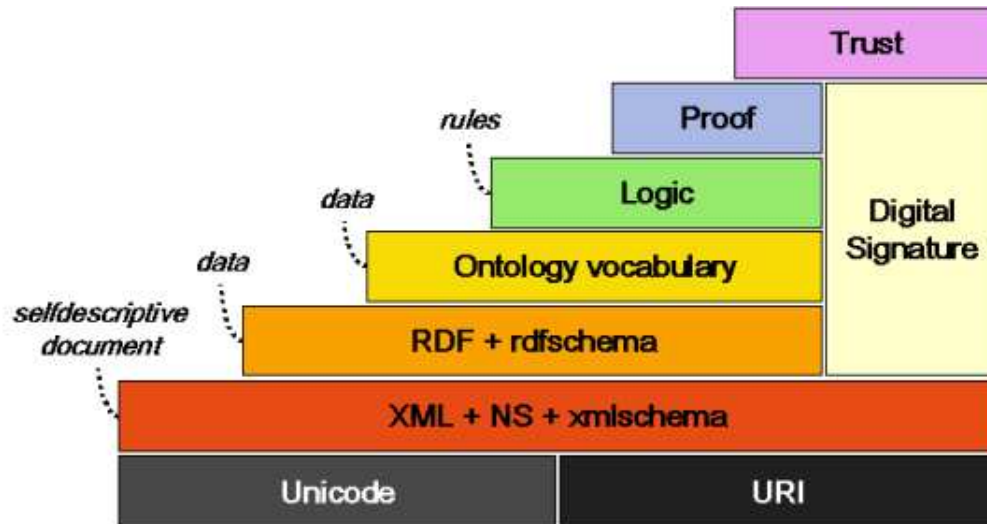


Figure 2.3: The Semantic Web layers

On this section we have described and mention some general aspects of ontologies along with certain applications. The next section introduce the utilization of ontologies for enterprise modelling describing some popular enterprise ontologies.

2.2 Enterprise Ontologies

In this section we review three of the most relevant enterprise ontologies which were a start point for the formalization of ours. Before speaking about a enterprise ontology itself it is important to revise the concept of enterprise modelling and its importance.

According to Mark S. Fox and Michael Gruninger ² in [8] an Enterprise Model is a computational representation of the structure, activities, processes, information, resources, people, behavior, goals and constraints of a business, a government, or other enterprise.

Two perspectives are given with respect to what the model should provide. The first one is the *design* perspective, in which the model should provide the language used to explicitly define an enterprise and a way to determine changes in all parts of the enterprise. In this perspective, also, it is necessary to explore alternative models in the design of the enterprise structure.

The second is the *operations* perspective. The aim here is to represent what has happened in the organization, the plans and what might happen. Also, it must support the operations of the enterprise and must be able to answer questions related with performance tasks.

According to Mike Uschold, Martin King, Stuart Moralee and Yannis Zorgios ³ in [28], the overall goal of enterprise modelling is to take an enterprise-wide view of an organization which can then be used as a basis for taking decisions. This is a view of the subject area, or domain, in which an organization operates. The authors define four key aspects which must be encouraged in order to achieve this view.

Integration of relating information and relating tasks.

Communication between people, tasks and tools.

Flexibility to allow an organization to adapt its business processes to meet changing goals and changes in its environment.

²Authors of Toronto Virtual Enterprise (TOVE)

³Authors of the Enterprise Ontology (EO)

Support to the user by making clear why and what is going on.

Both definitions of enterprise modelling consider the representation of all organizational elements. The difference between both is that Fox and Gruninger separate the organization into two perspectives and Uschold just have one enterprise-wide view of it.

Here, we present some of the most recognized enterprise ontologies used to model enterprises.

2.2.1 The Enterprise Ontology (EO)

The Enterprise Ontology was developed within the Enterprise Project. This project aims to provide a framework for enterprise modelling by giving a collection of terms and definitions relevant to business enterprises.

In [28], the authors link the concept of enterprise modelling with enterprise ontologies stating that an important way to achieve both, effective integration and effective business planning is ensuring that all parties involved (from business managers to software engineers) have a shared understanding of the relevant aspects of a business enterprise. The utilization of *shared understanding* is in the same sense in which we defined it before in Chapter 1.

The authors define an enterprise ontology including a wide variety of carefully defined terms which are widely used for describing enterprises in general. The idea is to provide one set of terms and definitions which adequately and accurately covers the relevant concepts in the enterprise modelling domain [28].

This ontology was aimed mainly to support the communication between human beings, and between systems, to help in the elaboration of software specifications and to help in the task of system interoperability.

The Enterprise Ontology has various forms of manifestation. The first form of EO was informal, describing core terms and definitions in natural languages. Later, core EO was extended by individual applications to specify the ontology according with their objectives. Another form of EO corresponds to a formal one, which is expressed using Ontolingua⁴. This section will describe the informal form of EO since its extensions are at this level [28].

The EO terms and definitions are organized around five categories [28]:

Meta-Ontology and Time. Here are described concepts used to define the terms of the Ontology. The basic concept of the Meta-Ontology is *Entity*. Terms that do not depend from others to mean something (like *Person*) are directly classed as entities. Other concepts can be considered relationships between two or more other entities (like *Sale*) so they are described as relationships.

An *Entity* may have a *Role* within a *Relationship*. An *Entity* playing a *Role* is regarded as *Actor* and a *Role* played by an *Entity* is regarded as *Role Entity*. Entities that can play a *Role* are called *Potential Actors*.

State of Affairs corresponds to the situation in which one or more *Entities* participate in one or more *Relationships* with one or more other *Entities*.

⁴Formal language grounded in first order logic, with built-in facilities used to represent knowledge in an object-oriented style [28]

Time Interval refers to when *Activities* are performed and is defined in terms of *Time Points*, which in turn make up a *Time Line*.

Activity, Plan, Capability and Resource. Here are described terms related to processes and planning. The central term is *Activity* which intends to capture the notion of anything that involves actual doing, in particular including action.

Activity Specification refers to specifications or plans for activities. A *Doer* (a *Person*, *Organizational Unit* or *Machine*, all of them potential actors) is the one that executes the *Activity Specification* by performing the specified activities. To be a *Doer*, a *Potential Actor* must have some *Capabilities* or *Skills* (if the *Doer* is a *Person*).

Resource is something that can be used or consumed in an *Activity*. An *Activity* also have outputs or *Effects* and may be simple or complex (in this case can be de-composed into many *Sub-Activities*).

Activities with intended *Purposes* are called *Plans*. A *Process Specification* refers to being able to repeatedly execute the same *Plan*.

Authority is the right of an *Actor* to perform one or more activities.

Organization. Here are described terms related to how organizations are structured. Central terms of this category are the concepts of *Legal Entity* and *Organizational Unit* (*OU* for short). *Legal Entity* have rights and responsibilities in the world and *Organizational Entities* have full recognition within an organization.

A *Machine* is a non-human, non-legal *Entity* that may play certain *Roles* otherwise played by a *Person* or *OU*.

Ownership (of rights and responsibilities) covers *Legal* and *Non-legal* aspects to encompass *Legal Entities* and *OUs*.

Strategy. Here are described terms related to high level planning for an enterprise. The central concept of this category is *Purpose*, which captures the notion of the intended reason for *Executing an Activity Specification* and the notion of something that an *OU* can be responsible for. A *Purpose* can be composed or decomposed.

Strategy is defined as a *Plan* to achieve a *Strategy Purpose*. Key concepts related to *Strategic Planning* are the terms *Decision*, *Assumption*, *Risk*, and various kinds of *Factor*.

Marketing Here are described terms related to marketing and sale (goods and services). The central concept of this category is *Sale* which is an agreement between two *Legal Entities* for the exchange of a *Product* (normally a good or a service) for a *Sale Price*. Here, the *Legal Entities* play the *Role* of *Vendor* and *Customer*.

A *Market* is all *Sales* and *Potential Sales* within a scope of interest and may be decomposed into *Market Segments* in many ways and levels of detail. Other terms related with *Market* are *Image*, *Brand*, *Need of Customers* and *Promotions*.

<i>ACTIVITY etc.</i>	<i>ORGANISATION</i>	<i>STRATEGY</i>	<i>MARKETING</i>	TIME
Activity	Person	Purpose	Sale	Time Line
Activity Specification	Machine	Hold Purpose	Potential Sale	Time Interval
Execute	Corporation	Intended Purpose	For Sale	Time Point
Executed Activity Specification	Partnership	Purpose-Holder	Sale Offer	
T-Begin	Partner	Strategic Purpose	Vendor	
T-End	Legal Entity	Objective	Actual Customer	
Pre-Condition	Organisational Unit	Vision	Potential Customer	
Effect	Manage	Mission	Customer	
Doer	Delegate	Goal	Reseller	
Sub-Activity	Management Link	Help Achieve	Product	
Authority	Legal Ownership	Strategy	Asking Price	
Activity Owner	Non-Legal Ownership	Strategic Planning	Sale Price	
Event	Ownership	Strategic Action	Market	
Plan	Owner	Decision	Segmentation Variable	
Sub-Plan	Asset	Assumption	Market Segment	
Planning	Stakeholder	Critical Assumption	Market Research	
Process Specification	Employment Contract	Non-Critical Assumption	Brand	
Capability	Share	Influence Factor	Image	
Skill	Shareholder	Critical Influence Factor	Feature	
Resource		Non-Critical Influence Factor	Need	
Resource Allocation		Critical Success Factor	Market Need	
Resource Substitute		Risk	Promotion	
			Competitor	

Figure 2.4: Terms formally defined in the Enterprise Ontology [28].

The previous description corresponds to a overview of the main categories in EO with its correspondent central terms and some of its related terms. Figure 2.4 shows all terms formally defined in the Enterprise Ontology.

A more detailed description of the EO ontology can be found at [28].

2.2.2 Toronto Virtual Enterprise (TOVE)

TOVE project aims to create a data model that [7]:

1. Provide a shared terminology for the enterprise with which each agent can jointly understand and use,
2. Define the meaning of each term as precise and unambiguous as possible,
3. Implement the semantics in a set of axioms that will enable TOVE to automatically deduce the answer to many “common sense” questions about the enterprise, and
4. Define a symbolism for depicting a term or a concept in a graphical context.

TOVE defines a generic level representation in which generic concepts (including Time, Causality, Activity and Constraints) are represented. Application level representations will be defined in terms of the generic level.

A description of part of the terminology defined on this generic level is described as follows [7]:

- **Activity:** Activities are one of the basic elements in a organization. With the definition of activities we can represent from actions to processes and operations. Activities have an attribute called status. This status can be one of: *dormant*, *enabled*, *suspended*, *re-enabled* and *completed*.

- **State:** In activities we can find two states, *enabling* (which define the condition for the activity to be performed) and *caused* (what will be true after the activity is completed). In TOVE we can find terminal states and non-terminal states. Also, there is a set of status for activities, which are: *possible*, *not possible*, *committed*, and *completed*.
- **Activity abstraction:** Activities and its states can be aggregated to compose levels of abstractions. In turn, these aggregated activities can be divided in sub-activities.
- **Time:** Time is used on this ontology to situate the occurrence of events. Time points, periods and relations are used to represent time.
- **Resources:** Entities can play the role of resources when they are related or associated to activities. Machines, human skills, tools, electricity are examples of resources.

Figure 2.5 shows a representation of TOVE terms.

TOVE also define a set of axioms (or rules) that define common-sense (the more obvious definitions/deductions about the entities and attributes) meanings for the terminology. Obvious deductions should be determined by a subset of questions used also to determine the competence of a representation. The competence criterion focuses on how well the model supports problem solving.

The following are a subset of questions the authors have considered in the creation of the TOVE model. The questions are presented grouped by main terms [7].

Activities and Causality. • Conditions: What is the current status of an activity? What alternatives exist?

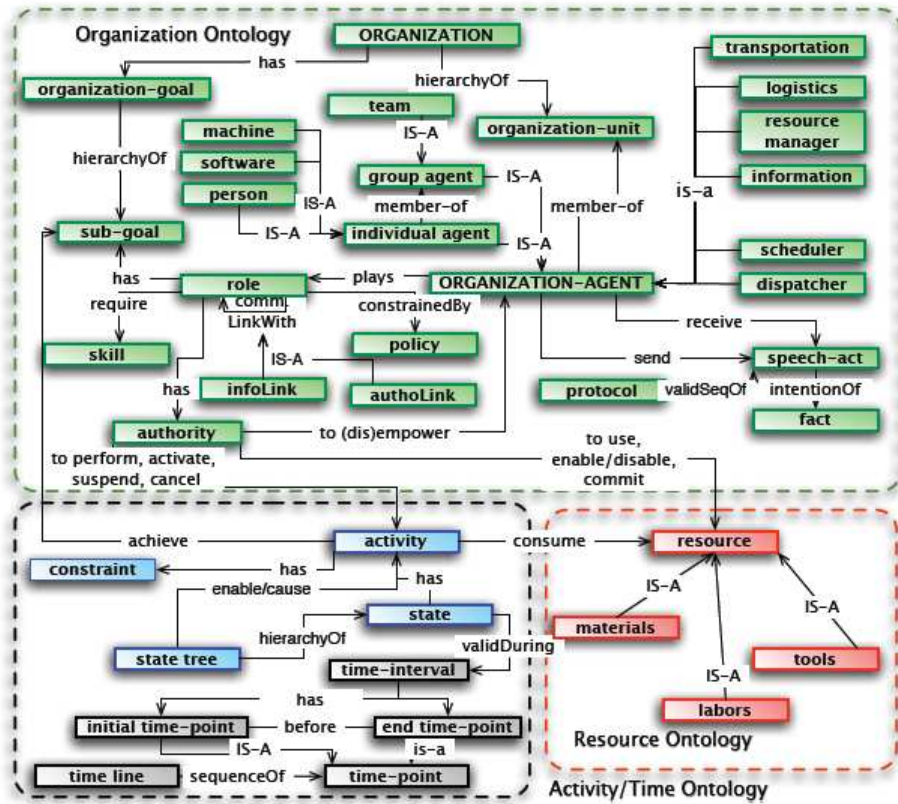


Figure 2.5: TOVE Terms [29].

- Causality: What conditions have to be satisfied to perform activity? What conditions will be satisfied when the activity has been performed?
- Abstraction: What subactivities can the activity be divided into? What superactivities is an activity part of?

- Time**
- Time Point: When does the activity start?
 - Time Period: What is the start time and end time of the activity?
 - Time Window: What is the earliest/latest start/end time?
 - Time Relation: Is activity 1 before/after/during activity 2?

- Resource**
- Existence: How much of the resource exists at time t?
 - Consumption: Is the resource consumed by the activity? If so, how much?
 - Divisibility: Can the resource be divided and still be usable? Can two or more activities use the resource at the same time?
 - Structure: What are the subparts of resource R?
 - Capacity: Can the resource be shared with other activities?
 - Location: Where is resource R?
 - Commitment: What activities is the resource committed to at time t?

A more detailed description of TOVE ontology can be found at [7].

2.2.3 Enterprise Ontology by Dietz

In [4], the author describes the theory and methodologies for enterprise ontologies. This theory is based on a set of axioms (operation, transaction, composition and distinction) that serve to achieve its overall goal of extract the essence of an organization from its actual appearance.

The *operation axiom* states that the implementation independent essence of an organization consists of subjects fulfilling actor roles, these subjects are called actors and represent the operation of an organization by performing two kinds of acts: production and coordination acts.

In the case of production acts (P-act) the actor helps achieving the goods or services that are delivered to the environment of the organization. The outcome of this acts can be material (something manufactured, stored or transported) or immaterial (decisions or judgements) facts (P-fact). In the case of coordination acts (C-act) subjects commits each other with respect to the performance of production acts. These acts are composed by a intention (like request, promise or assertion) and a proposition (intention's fact and associated time) and result in coordination facts (C-fact) [4].

Figure 2.6 shows a graphical representation of the operation axiom.

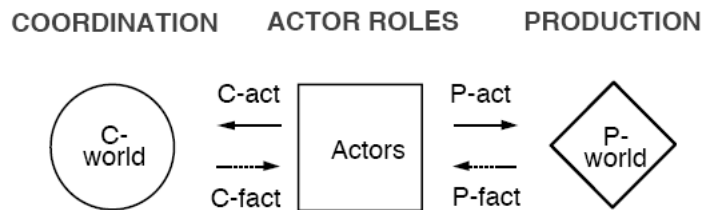


Figure 2.6: Graphical representation of the operation axiom [4].

The *Composition axiom* state that transactions can be:

- enclosed in some other transaction.
- a customer transaction of the organization under consideration.
- a self-activation transaction.

The author states that this axiom provides the basis for a well-founded definition of the notion of business process, which says that a business process is a collection of causally related transaction types, such that the starting step is either a request performed by an actor role in the environment (external activation) or a request by an internal actor role to itself (self-activation) [4].

The *distinction axiom* state that subjects have or play three basic human abilities: *performa*, *informa*, and *forma*. The *forma* (Latin for “form”) ability concerns the form aspects of communication and information. The *informa* (in Latin, what is in the form) ability concerns the content aspects of communication and information (abstracting from the form aspects). The *performa* ability (in Latin, through the form) brings new things, directly or indirectly by communication [4].

Figure 2.7 shows a summary of distinction axiom.

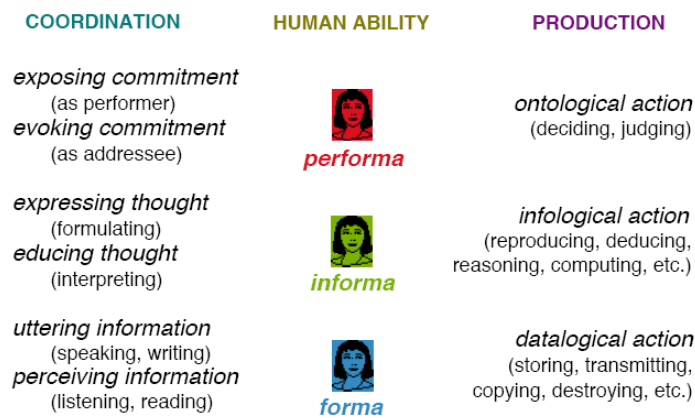


Figure 2.7: Summary of distinction axiom.

The *transaction axiom* states that coordination acts are performed as steps in universal patterns (transactions) who involve two actor roles and are aimed at achieving

a particular result.

A transaction consists of two conversations: order conversation and result conversation. A conversation is defined as a sequence of coordination acts between two actor roles (one called initiator, and the other executor of the transaction). The three phases of a transaction are the order phase (O-phase), the execution phase (E-phase), and the result phase (R-phase).

On the order phase, the initiator and the executor work to reach an agreement about the intended result of the transaction, in the execution phase this result is actually brought about by the executor and in the result phase the initiator and the executor work to reach an agreement about the production fact that is actually produced, as well as the actual time of creation [4].

Figure 2.8 shows the basic pattern of a transaction.

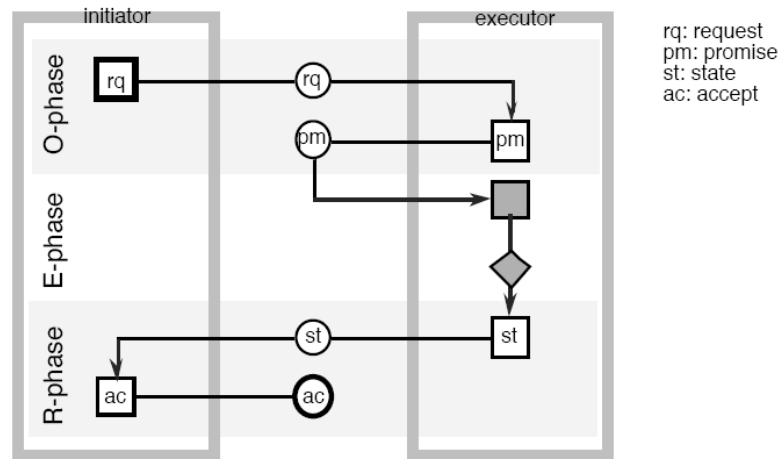


Figure 2.8: Basic pattern of a transaction [4].

All the benefits of the previously explained axioms are combined in a concise, comprehensive, coherent, and consistent notion of enterprise by the organization theorem, which states that the organization of an enterprise is a heterogeneous system that is constituted as the layered integration of three homogeneous systems: the B-organization (from Business), the I-organization (from Intellect), and the D-organization (from Document) [4].

Figure 2.9 shows the organization theorem. Coordination parts of systems: B-organization, I-organization and D-organization are similar but differ in the kind of production: ontological for the B-organization, infological for the I-organization and datalogical for the D-organization.

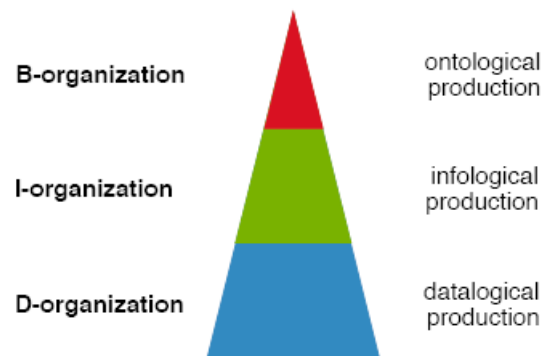


Figure 2.9: Representation of the organization theorem [4].

The following corresponds to a set of the main concepts of this ontology.

- **Time related concepts:** Requested, Promised, Stated, Actual, Creation Time, Time Unit, Time.
- **Process related concepts:** Phase, Transaction, Process, Conversation, Initiator, Fact, Event, state, Transition, World, Act, Executor.

- **Actor related concepts:** Agenda, Actor Role, Subject, Action Rule, Agendum, Intention, Proposition.

Detailed information about this ontology can be found at [4].

The previously explained ontologies are based on models focused mostly on an organizational perspectives (processes and resources). The next section will introduce an approach that includes individual perspectives and recognizes the importance of contexts in modelling the enterprise.

2.3 Agent Based Model

Traditional enterprise modelling is focused mostly on an organizational perspectives (processes and resources) disregarding the importance that individuals have in the organization. Not enough importance has been given to individual or inter-personal perspectives. This section describes a model that addresses the modelling of individual and inter-personal behaviors combining them with classic approaches.

In [29], the author develops a conceptual framework based on agents and contexts for the alignment between individuals and organizations. The aim of this framework is to take into consideration agents' points of view for enterprise modelling and, as a consequence of this improvement, the alignment between individuals and organizations.

Achieving alignment between individuals and organizations is not an easy task. Most models assume that the views of some organization are shared by all its members. The argument presented in [29] states that individual and collective members have different views of each other, and coherency between such views cannot be taken for granted, this is why individual and organizational behaviors should be modelled

in inter-dependent, but separate perspectives that are not necessarily aligned.

In particular this approach mainly intends to bring out the need of an agent-centric and “context-aware” perspective to align individuals and organizations, concerning about agent behaviors and employing the notion of context as the key in linking agent-centric concepts to other enterprise perspectives.

All the previous reasons lead the author of [29] to the specification of an conceptual framework encompassing formal and informal aspects of the enterprise which aims to achieve alignment between different enterprise elements. This framework is described on the next section.

2.3.1 Framework based on Agents and Contexts for the Alignment between Individuals and Organizations

The framework described on this section was developed within the context of a Ph.D. thesis. The reader can found this work in [29]. Here, the author describes a ontological position on organizations. This ontological position defines how organizations are regarded in terms of their nature, essential components and inter-relationships. This position is a base to construct an ontology. The ontology that the author presents is supported on a set of premises. The following summarises that premises.

1. Organizations are complex, adaptive systems created and maintained by interactions among its agents.
2. The operation of organization is described in terms of activities.
3. Agents are complex, adaptive entities that have acting, monitoring, reflexive, and learning capabilities.

4. Associated to activities are resources and agents.
5. Organization and agent complexity can be addressed defining different architectural views.
6. Agents have their own architecture. Typical agent architectures encompass three layers addressing different behavioral concerns; (1) acting, (2) deliberation, and (3) learning.
7. A single agent exhibits different behaviors.
8. Agent behavior is determined by the role played at a given moment, and within a given context. A single agent may play different roles, at different times.
9. Agent interactions are mediated by artifacts. Artifacts can be of physical, psychological or social nature. The specific artifact involved depend on the particular interaction context surrounding agent actions and interactions.
10. A proper modelling of agent behavior cannot be isolated from the contexts associated to such behavior.

The framework also define a set of fundamental concepts needed to model the enterprise.

2.3.1.1 Fundamental Concepts

This ontology is based on the following fundamental concepts:

1. **Resource.** Resources are elements that the organization needs for its operation. They can be:
 - *Physical*: Like machines, persons, reports, buildings or tools.
 - *Abstract*: like concepts, ideas, information, skills or time.
 - *Active*: When they have, or they are capable of having active behavior.

- *Passive*: When they are only activated by its utilization by active resources.

Resources are identified by nouns and may be single and simple items or complex resources composed by several items.

2. **Agent.** Agents are physical and active resources. They can interact with their activities (performing, coordinating and re-designing them) and with other agents activities (monitoring, coordinating and re-designing them). Agents represented on this framework must satisfy the following requirements:

- *Autonomous*: Actions performed by the agent can be performed without direct external intervention. Agents have some control over its action and internal state.
- *Interactive*: Agents can communicate and interact with its environments and with other agents.
- *Adaptive*: The agent can respond to the stimulus of its environment and of other agent and in some cases can modify its own behavior.
- *Sociable*: Agent must be capable of establish social relations.
- *Proactive*: Not only a reacting agent, but a purposeful one that can act on initiative.
- *Rational*: Agents capable of choosing the best action based on its knowledge and goals.
- *Not fully predictable*: Agents are capable of act in a non-deterministic way, even if all initial conditions are known.
- *Coordinative*: Agents can perform activities in heterogeneous environments with other agents and coordinate that activities.
- *Intelligent*: Agent behavior is governed by knowledge, beliefs, goals, plans and assumptions.

- *Mobile*: Agents are capable of transporting themselves between environments.
 - *Credible*: Agents must have a believable personality and emotional state.
 - *Rugged*: Agents must be able to deal with errors, mistakes and problems with data.
 - *Accountable*: Agents must be able to provide their activity logs.
3. **(Agent) Actions.** Are atomic operations (identified with verbs), performed by agents and involving organizational items.
- *Communicative Actions*: Are actions that involve agents as resource, as sender (performer) and as receiver. The intention of communicative actions is to generate or trigger another action (nested or embedded on it) by the receiver(s). If the embedded action is executed by the agent(s), the communicative action succeeds.
 - *Non-communicative Actions*: Actions that have an agent as executioner, does not have embedded actions and do not trigger any other action.
4. **(Agent) Interactions.** Occurs when agents exchange successfully adjacent pairs of communicative actions.
5. **Roles.** Correspond to agent behavior (actions performed by them) expressed in terms of capabilities. In this framework are used generic role definitions, such as performer or supervisor. About roles, in [29] seven properties are given, based on characteristics given in [14]. These properties are the following:
- At all times, agents play at least one role.
 - Agents may acquire additional roles (classify operations).
 - Agents may relinquish roles (declassify operations).
 - Agents may play multiple roles, requiring multiple classification.

- Agents may play the same role, several times i.e. agents activate and suspend classified roles.
- During particular time intervals, agents have only one active role.
- The same role can be played by different agents, simultaneously or at different times.

6. **Context.** Situations in which agents play roles supported or restricted (behavior) by the situation itself. From the previous definition is clear that the concept of context is important due to its natural binding with other important concepts as role and agent behavior. In this framework, the author describe context as abstract entities:

- Created and updated by agent actions and interactions (action streams and their associated resources).
- Composed of a set mediating artifacts used in agent interactions.
- Reflects states of affairs of activity execution.
- With no clear-cut boundaries. These boundaries are influenced by some topic, a particular set of agents, action/interaction types and rules and time-related factors.

7. **Activities.** Activities are abstractions about what the organization do. To define activities, it is mandatory to have a shared understanding about its outcomes, its coordination mechanism and resources (used and produced). Activities can be decomposed in sub-activities and in turn may be part of another activities. Given this, coordination is needed (between them) to achieve what we intend to do with them.

Besides the above fundamental concepts some additional remarks are needed to describe and understand and the framework.

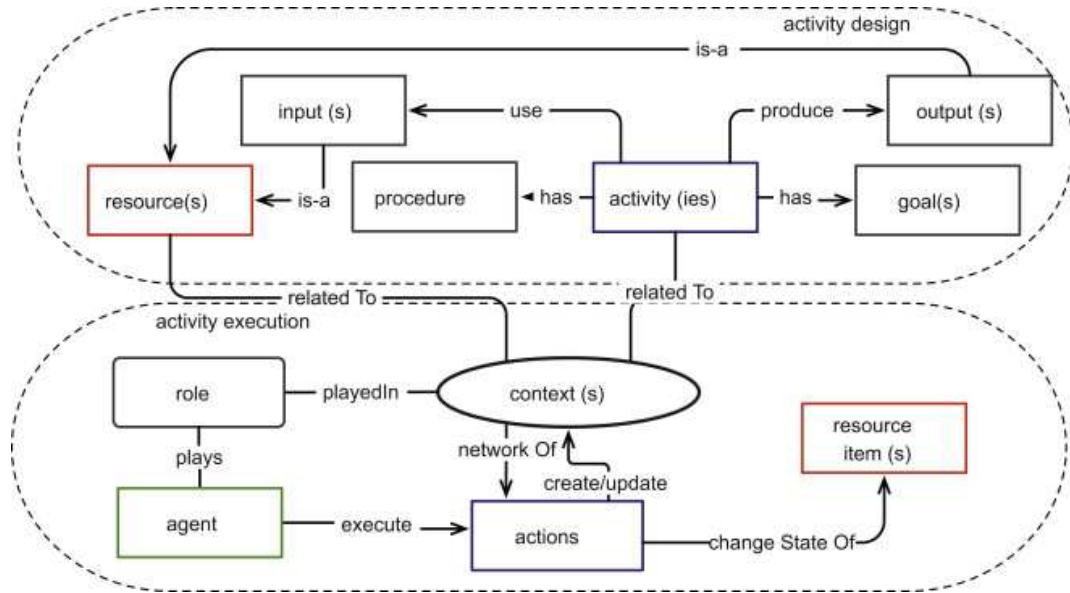


Figure 2.10: Framework fundamental concepts and their inter-relationships [30].

2.3.1.2 Additional remarks

The following is a description of some additional remarks which, along with the description and fundamental concepts, shows us the semantics of the framework and help us to understand it better.

- *Agent sentences.* Agent sentences describe what organizational agents do. This sentences can be formed joining together agents (nouns), actions (verbs), and resources (nouns). Here, agents corresponds to the subject of the sentence, the predicate identifies the action performed by the agent and the verb identifies the type of action and its object, the specific resource(s) used or produced [29].
- *Agent-resource duality.* Being an agent is a capability, a potential behavior. Sometimes entities behave as plain resources (passing from the subject of the sentence to the predicate). Due to agent-resource duality, the specific role (agent or resource) played by people in a given action can only be determined through the position occupied within the corresponding sentence [29].

- When the action in the sentence is a communicative action, the agent in the predicate of the sentence is also an active resource that eventually will execute the nested action.
- In this framework, actions executed within a context are characterized by its subject, agent that performs the action, action type and associated resource items, and its execution date. This is why defining when a context ends and another begins is not trivial. Identifying contexts from higher numbers of actions requires support of automated techniques [29].
- The relevance of actions is given by its presence on contexts, i.e. its relevance will decrease until the action is no longer considered as part of the context. Particularly, entities like subjects, agents, action types and resources will have each, a different relevance degree for a particular topic.
- Contexts have some emergent properties, like a state (initiated/terminated or active/suspended) and importance or priority attributed to the context as a whole.
- Mediating artifacts refer to physical or symbolic, external or internal instruments, including both tools and signs, which are used to transform objects [29]. Contexts provide mediating artifacts between agents (such as tools used, documents exchanged, information, language used and commitments derived from communicative actions).
- Action sequence can show us history of actions and interactions, which corresponds to context history. The state of actions and interactions is also defined by context history and time.
- Interactions are governed by rules that condition or restrict the occurrence of certain events. These rules can be formal and explicit (how the events should happen) or informal (alternative events or exceptions to events).

- Contexts are a key element to understand agents behavior. Agents may behave differently according to the role that they play in different contexts. A single agent may play several roles in several contexts. While the same role may be played in several contexts, within a context, each agent play a unique role [29].
- The execution of action streams by agents creates and updates context. Mapping actions streams into activities is not a trivial task because of the nature of activities. Since activities are abstractions they don't have clear boundaries, therefore they may overlap with each other. Moreover, activity and context boundaries are not necessarily the same, this means that the execution of an activity may involve more than one context. On the other hand a context can incorporate more than one activity.
- Considering the time factor, we have that action streams performed at different times may be associated to the same activity but to a different context.
- Resources need to be carefully considered. There are some type of resources called transient resources that are only important within a particular context. This type of resources are not necessary activity inputs or outputs but are manipulated through actions.
- In this framework, contexts have a many-to-many relationship with activities [29].

The framework address complexity of agent behavior defining different views to capture different kinds of behavior.

2.3.1.3 Addressing Complexity

To address organizational complexity together with individual agent complexity the framework define views to capture the different concerns of agent behavior providing

means of tracing all the behavior of a single agent and enabling linking agent behavioral concerns with other architectural perspectives [29].

On the present framework is presented an agent architecture composed by three inter-dependent layers: (1) action, (2) deliberation and (3) change/learn. Figure 2.11 shows this agent architecture.



Figure 2.11: Agent Architecture [29].

Action Layer. In this layer agents are considered actors possessing sets of pre-defined action strategies (reactive, pre-defined behaviors). Action strategies show how agents perform their work and are related to topics or subjects, specific sets of resources and are situated in time. Task execution may involve several action strategies and a strategy may be employed in several tasks. In this layer agent behavior is mostly deterministic. Action layer strategies only apply in normal situations, to handle exceptions upper layers are needed.

Deliberation Layer. In this layer agent deliberative behavior, like planning, decision making, or scheduling, is captured, but only scheduling is addressed. Deterministic representations are not suitable for this type of behavior because this one has a dynamic nature given by the decision making process which is conditioned by incoming events and the current state of specific activities, resources and other agents. This is why in this layer is used a state-based representation regarding agents as non-deterministic finite state machine (FSM). In this type of representations, transitions from one state to another are made when a certain condition (that defines the transition) is true. Deliberation behavior covers partially pre-defined behavior in a way that for each current state and inputs, there may be a set of possible states.

Agents are considered as Abstract State Machines (ASM) which are defined as single-agent machine that may dispose potentially unrestricted non-determinism and parallelism [3].

Basic ASM are finite sets of transition rules where each rule transform abstract states. Activation is the processes affecting agent timing and activity sequencing [2]. Strategies are defined as activation strategies and are represented using state-transition rules, where the state of an agent is defined in terms of its active context [29].

Change/Learn Layer. This layer encompasses agent reflective behavior. How action and activation strategies evolve in time, based on performance attributed by agents to each strategy [29]. Agents in this layer can create, change or eliminate strategies.

In this layer, changes occurs when a new strategy arises and learning occurs when a new strategy or a re-designed one improve the previous performance.

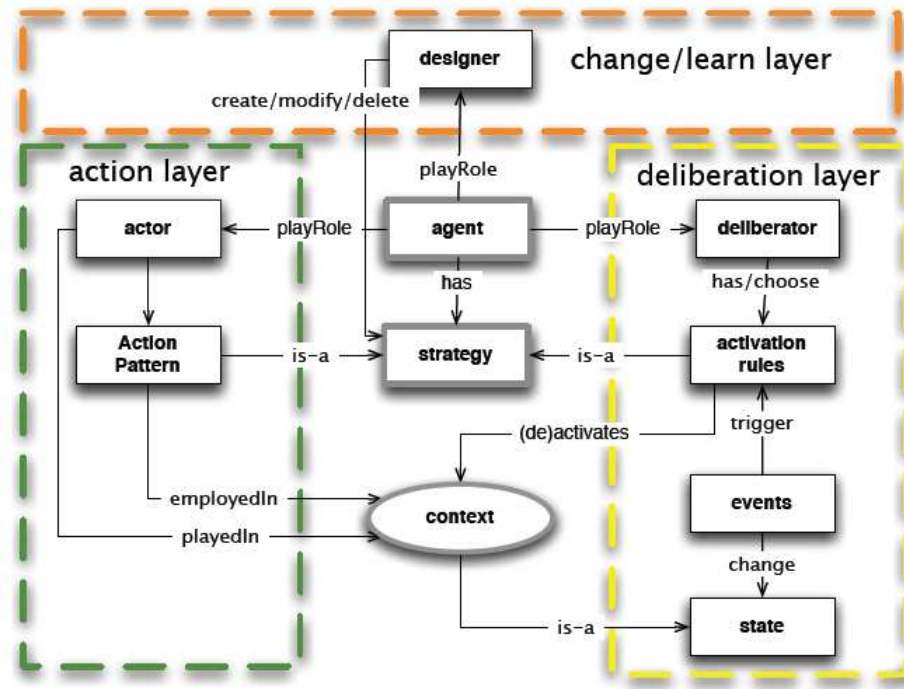


Figure 2.12: Organizational agent architecture summary [29].

In Figure 2.12 we can observe that at the action layer agents are considered as actors and their patterns of action are captured within contexts. At the deliberation layer agents are regarded as decision-makers, focusing in agent states and state-transition rules. Finally, at the change/learn layer is shown that strategies are not static and evolve in time.

In this framework the concept of actor is placed inside of the concept of agent. Moreover, in the action layer, agents are regarded as actors and in upper layers agent are regarded as deliberators and designers (with learning/changing capabilities).

Strategies can move between layers. Agents also switch among layers and using its deliberation capabilities they can decide what to do activating a specific context.

Integrating agent and enterprise architectures

The present framework integrates agent and organization's complexity by taking into consideration both, agent and enterprise architectures. Enterprise architecture is modelled with two dimensions: activities and resources and organizational agents are modelled according the three previously discussed layers. Hence, the organization is modelled as a dynamic network of activity and resource-related agents [29].

Artifacts (objects made, used or modified by people) provided by contexts regulates and facilitates agent interaction at each architectural layer. In this framework mediating artifacts are regarded as abstract resources created, modified, updated or used by agents to support and constrain organizational actions [29]. The nature of artifact depends on the layer, for example:

- Action layer artifacts are pre-defined interaction patterns composed of typical interaction which is related to a specific set of resource types that enable and constrain, agent interactions.
- Deliberation layer mediating artifacts are the state of agents (defined by its active execution context) and their interactions (given by agent commitments). Commitments are typically related to the notions of pledge, promise, agreement, or contract, so, mediating artifacts of deliberation layer are the agreed pledges, promises, agreements or contracts between activity and resource-related agents [29]. At this point it is possible to infer agent commitments from its actions because communicative action are related with commitments and obligations which in turn involve the execution of specific actions.

- Change/learn layer artifacts corresponds to interaction rules (higher-order rules governing design-layer behaviors). Between resource and activity (re)design and interaction rules exists a two way relation because the second constrain the first and the first, in turn, can trigger changes on the second.

A system can be composed or decomposed in several systems. These sub-systems can also be agents, for example: an organizations, and can be modelled using the architecture that we have been discussing on this section. Concepts, elements and relations can be applied, in a recursive way, to all organizational levels. Human agents are typically studied at an individual, inter-personal, group and organizational levels and business activities are commonly analysed and designed at process, activity and task levels [29]. This allow us to use the same ontology in all levels but with different universe of discourse (set of nouns, verbs, attributes, state variables and rules) and different kind of agent (at the individual and inter-personal layers subjects are persons, at the group layer subjects are formal or informal teams and at the organizational level, subjects may be organizational units or whole organizations).

The following corresponds to a description of the usage of the concepts in the different levels of the architecture.

Agents. Behavior of agents refers mostly to the behavior of individuals. Behavior of collective agents composed of two individuals (dyads) is also represented. The modelling of groups is not not addressed on this research.

Actions and Interactions. Actions are “atomic” acts performed by individuals and interactions are adjacent pairs of communicative actions exchanged between two or more individuals [29].

Activities. Activities are associated to a group level. Another similar terms, but with different levels of complexity are tasks (associated to personal and inter-

personal levels) and processes (associated to a whole organizations or organizational unit levels).

Resources. Three different types of resources are distinguished:

- **Tools.** Artifacts used by individuals in performing actions. Can be physical, material or psychological.
- **Information items.** Are information elements used or exchanged by individuals in performing actions. Can be formal (items acknowledged as task inputs or outputs) or informal (partial inputs or outputs, or transient items used in producing outputs). Formal items are typically embedded in documents and files of different kinds, informal items are typically embedded in messages or action descriptions [29].
- **People.** Are considered human resources when they play a resource-related roles.

Contexts. The two concepts of context corresponding to the levels addressed are (personal) *action contexts* and (inter-personal) *interaction contexts*. The present research addresses and develops both.

Personal Action Contexts. (Personal) action contexts are created and updated by streams of related actions performed by a single individual during a given time interval [29]. A task may be related to one or several contexts and an action context may be associated to one or several tasks (many-to-many relationship). In different layers personal context reflect different things. At the action layer this context reflect the personal view of the individual with respect to a given interaction context. Here, each individual is aware only of the interactions he sends and receives. The individual regards himself as a task performer and the other individuals as resource producers or consumers. At the deliberation layer, a personal context reflects personal commitments. Here, obligations are

generated from interactions and their associated commitments. Finally, at the change/learn layer, a personal context reflects personal rules.

Inter-Personal Interaction Contexts. Corresponds to interactions among two individuals. In different layers inter-personal interaction contexts reflect different things. At action layer interaction between two individual is captured. Individuals interact from specific personal contexts so the inter-personal network is allocated on the personal context of the individuals that are interacting. Inter-personal interaction contexts are considered multiplex networks because individuals may share several inter-personal contexts. At the deliberation layer inter-personal contexts reflect the agreed commitments produced by interactions between specific personal action contexts of two given individuals, at given time points [29]. Finally, at the change/learn layer, inter-personal contexts represent interaction rules governing interaction patterns among individuals and shared by individuals (two).

Strategies. Can be personal and inter-personal.

- *Personal strategies.* Represent individual action patterns at work and are distinct in different layers.
 - At deliberation layer. Represent the rules used by individuals in activating their personal contexts.
 - At change layer. Represent hidden personal rules which govern the individual strategies of the previous layers.
- *Inter-personal strategies.* Corresponds to structures and dynamics of networks created by interactions between set of individuals and also are different in distinct layers.
 - At action layer. Represent typical interaction patterns between any two individuals.

- At deliberation layer. Represent the dynamics of commitment creation, update and accomplishment that is produced by interactions among individuals.
- At change/learn layer. Represent hidden inter-personal rules that governs inter-personal strategies of the previous layers.

To summarize, we present a comparison about the 4 enterprise ontologies presented above. This aims to recognize the most important aspects that led us to select one of these ontologies and formalize it.

Ontology Comparison

Figure 2.13 shows a comparative table summarizing the principal aspect of the four ontologies presented on this chapter. The idea is to present the main characteristics of them to justify the selection that we have made for our formalization process. This figure shows the following aspects:

- Purpose. Principal objective for using these ontologies or “why” were they developed.
- Focused In. Main aspect in which the ontologies are based on.
- Main Terminology/Concepts. Principal terms or concepts.
- Presentation. How the ontology is presented to the readers/users/developers.
- Agent Perspective. How the ontologies encompasses the presence of agents.
- Context Presence. If/How the ontologies recognizes the presence of contexts.
- Complexity. If complexity is addressed by the ontologies.
- Aspects. Which aspects of the enterprise are addressed by the ontologies.

Enterprise Ontology	Purpose	Focused in	Main Terminology/Concepts	Presentation	Agent perspective	Context presence	Complexity	Aspects
EO	Communication among people, systems and helps in system specification.	Activities	Activity, Person, Activity Specification, Sale, Time Line, Time Interval, Machine, Purpose.	Five Main categories	Organisation perspective			Formal
TOVE	Provide a enterprise shared understanding, implement axioms to answer questions.	Activities/ Processes	Activity, State, Activity Abstraction, Time, Resources	An generic level representation (generic concepts) and a set of axioms	Organization perspective			Formal
Dietz Ontology	Describe a well-founded method to model the essence of an organisation	Processes/ Actors	Time, Time Unit, Process, Transaction, Subject, Agenda, Event, Actor Role	Four axioms and an organization theorem	Actor perspective (roles)			Formal
Conceptual Framework	Describe an organization taking into consideration its formal and informal aspects	Agents/ Contexts	Resource, Agent, Action, Role Interaction, Context, Activity	Set of fundamental concepts and 3 layers to address complexity	Agent centric perspective	Aware of enterprise contexts	Three layers to address complexity	Formal and informal

Figure 2.13: Enterprise Ontologies Comparison

We have chosen to formalize the conceptual framework presented on 2.3.1. The reasons are explained in terms of the criteria presented on Figure 2.13.

- Purpose. This ontology encompasses formal and informal aspects of the organization. With this takes into consideration the presence of the individual and different points of view generated from its presence. We have selected the conceptual framework since we want to formalize an ontology that take cares of individuals and by that incorporates informal aspects of enterprises.
- Focused In. This ontology is focused in Agents and contexts rather that just processes. This is a fundamental aspect in our choice because these two element together help us modelling more accurate individual behavior.
- Main Terminology/Concepts. Principal terms of this ontology help us to describe individuals and its contexts, not just processes as the other described ontologies.
- Presentation. This is not a determinant aspect in our choice.
- Agent Perspective. This is an agent-centric ontology, which give us more tools to represent and address agent behavior.
- Context Presence. From the four revised, this was the only ontology which is aware of contexts on the enterprise. This is a complement to the agent centric view and an upgrade to the three other ontologies.
- Complexity. This ontology specify three layers to address different behavioral concerns. Again, addressing behavior in such detailed and systematic way help us to model better the informal elements of the enterprise.
- Aspects. Incorporate informal aspect to its representation. This is the main reason of choosing this ontology because is in correspondence with one or our main objectives.

As we have chosen the conceptual framework presented in 2.3.1, we have described its essential details (more detailed than the other ontologies) with which we can understand it better. The idea is to have a deep view of this work because the results presented on Chapter 4 corresponds to the results obtained from the formalization of this framework. A more detailed description of the framework can be found in [29].

The process of formalization can not be taken or conducted in a lightly way. This is why we need a set of structured steps to do a more robust work. Given this we analysed a set of ontology development methodologies with the purpose of learn about the process and adapt some development methodology to the formalization process. It is important to stress that the following are development methodologies (not formalization methodologies) so it is not possible just selected and apply one.

2.4 Methodologies Review

In this section we review three of the most important methodologies used to develop enterprise ontologies that are available on the literature. These methodologies are a important part of the background of ontology development and are broadly discussed on specialized parers, magazines and articles.

The studied methodologies describe a set of steps and activities needed to build ontologies. This activities and steps range from very early development stages (like motivational aspects and capture) until post-specification stages (like documentation and evaluation).

2.4.1 Uschold and King Methodology

In [27], Uschold and King address a methodology to building ontologies which includes the following steps.

1. **Identify Purpose.** Identifying purpose its very important for the development of every project, including ontologies. In this stage we need to be clear about why are we building the ontology and what we aim with it (utilization and uses).
2. **Building the ontology.**
 - *Ontology Capture.* The aim of this phase is to reach an agreement of the terms used to describe the organization. To do this first we need to define the scope of our modelling identifying key concepts and relationships to be modelled, then we need to define unambiguous text definitions for those concepts and relationships. The process and results of this stage are independent form any coding languages.
 - *Ontology Coding.* In this stage, the conceptualization of the previous stage is represented explicitly in some formal language. To do this we have to choose a representation language and create the code. In some methodologies, capturing phase and coding phase are mixed in just one stage. The authors recommend, by experience, handle this stages separately.
 - *Integrating Existing Ontologies.* On the previous stages the developer face the decision of whether to use (integrate) or not existing ontologies on the one that he is developing. Much care must be taken to reuse ontologies that are shared by multiple or large communities because of the need to reach agreement, although this is not a trivial task.
3. **Evaluation.** To explain this stage, in [27] the authors make reference to [9] which states that to evaluate technically the ontology and its components, it is

necessary to do it in term of requirement specification, competency questions and/or comparisons with the real world.

4. **Documentation.** This stage is very important to have a formal and/or informal documentation of the ontologies. To make the ontology more explicit, is important to document all the assumptions regarding to the ontology concepts and the meta-ontology.

This methodology provides guidelines to develop ontologies and is based on the work done to construct The Enterprise Ontology, described in 2.2.1.

2.4.2 Grüninger and Fox Methodology

This methodology is composed by the following steps [11]:

1. **Motivating scenarios.** The first step on this methodology is the definition or identification of scenarios. Scenarios are problems found on enterprises that are not well addressed for the current ontology.

This scenario also provides an intuitive solution to the problem which later can be included on the resulting ontology. The scenarios give us the motivation for the ontology that is being build.

2. **Informal Competency Questions.** Requirements to the ontology are obtained from scenarios of the previous stage. These requirements takes form of questions that the ontology must be able to answer.

This questions are called informal (not expressed on the formal language of the ontology) competency questions and can be used to partially evaluate the new ontology (or the new upgrade to the existing ontology) by means of determine if the new ontology can answer the questions or if the questions can be answered by the existing ontology.

it is desirable that competency questions were stratified so that the response to one question can be used to answer more general questions from the ontology. Finally this questions are used to evaluate ontological commitments that have being made (does not generate new ones) which allows to see if the requirements are meet.

3. **Specification in First-Order Logic - Terminology.** In this step, the terminology of the ontology is specified using first order logic or an equivalent. This specification correspond to a language used to express definitions and constrains in the axiom.

To complete this stage, first, objects of the domain of discourse need to be identified (by constant and variables), then, it is necessary to define attributes and relations among objects with unary and n-ary predicates, respectively.

4. **Formal Competency questions.** Once we have the informal competency questions and also the terminology of the ontology, we can formally define competency questions. The terminology of the ontology must include all the terms in the statements of formal competency questions.
5. **Specification in First-Order Logic - Axioms.** Axioms in the ontology specify definitions of terms in the ontology and constraints on their interpretation. They are defined as first-order sentences using axioms to define the terms and constraints for objects in the ontology [13].

To define the meaning of the ontology it is necessary to provide axioms. it is not enough with objects and terms. Development of the axioms is an iterative process because if the axioms are not enough it is necessary to add more until the formal competency questions are represented and the solutions characterized.

6. **Completeness Theorems.** On this stage the conditions under which the solutions to the questions are complete are defined. Completeness make explicit the role that each axiom plays in providing the ontology, with these feature we can also determine its extendibility. Future extensions to the ontology must the completeness theorem.

This methodology provides guidelines to develop ontologies and is based on the work done to construct the TOronto Virtual Enterprise (TOVE) Ontology, described in 2.2.2.

2.4.3 Methontology

This is a framework developed within the Laboratory of Artificial Intelligence at the Polytechnic University of Madrid. This framework is used to construct ontologies and encompasses the identification of the ontology development process and a life cycle based on evolving prototypes [13].

Ontology development process. Corresponds to the activities that are performed to build ontologies. Three categories of activities are presented:

- *Project Management Activities.* This category includes:
 - Planing. Is focused on tasks, which of them are needed, how are going to be performed and the resources used on when performing them.
 - Control. Is a manner to check if there is alignment between the actual way in which tasks were completed and the way that was planed for them.
 - Quality Assurance. Assures the quality of output products.
- *Development-Oriented Activities.* This category includes:
 - Specification. States purpose, intended users and end users for the ontology.

- Conceptualization. Is a structuring of the domain knowledge into models at the knowledge level.
- Formalization. Codification of the conceptual model into a formal or semi-computable model.
- Implementation. Build the previous model into a computational language.
- Maintenance. Updates, upgrades and corrections to the ontology.
- *Support Activities*. This category includes:
 - Knowledge Acquisition. Gathers knowledge from the domain.
 - Evaluation. Technical assessment of the ontologies, associated software environments and documentation. This is made during and between phases of its life cycle.
 - Integration. In the case of building an ontology by re-using another one.
 - Documentation. Detailed explanation of ontology building process and the resulting product.
 - Configuration Management. Version control of the ontology, documentation and software.

Ontology Life Cycle. In the previous stage were given some activities for ontology building, but the execution order of each activity was not addressed. Ontology life cycle identifies the set of stages through which the ontology moves during its life, when you should perform the activities to move from a given state to the next and in how much depth.

Building ontologies from scratch. In [5] are given several steps to build ontologies from scratch. These steps are the following:

- *Specification.* The objective is to produce a document written in natural language containing an informal, semi-formal or formal ontology specification. The document must include information about purpose, level of formality and scope of the ontology.
- *Knowledge Acquisition.* Gather knowledge from multiple sources (even other ontologies) using multiple gathering techniques, such as brainstorming, interviews or analysis of texts.
- *Conceptualization.* Obtaining a conceptual model that describes the problem and its solution which is expressed in terms of domain vocabulary.
- *Integration.* This is the case in which definitions of other ontologies are reused to construct the current one. If this is the case, the authors propose to inspect meta-ontologies to select the right one and check which ontology libraries have definitions of terms that are coherent with the terms of our conceptualization.
- *Implementation.* Codify the ontology in a formal language.
- *Evaluation.* As we stated before, this corresponds to technical assessment of the ontologies, associated software environments and documentation and must be made during and between phases of its life cycle.
- *Documentation.* Detailed explanation of the ontology building process and the resulting product. In Methontology the authors state that documentation is an activity that must be done during the whole ontology development process.

2.4.4 Methodology Comparison

The previously explained methodologies describe a series of steps to develop ontologies. Figure 2.14 shows a comparison between these three methodologies based on a set of steps or activities identified from them. The aim of this comparison is to

identify which activities are recurrent on these methodologies so we can take them in to consideration for our formalization methodology (presented in Chapter 3).

Activity	Uschold and King	Gruninger and Fox	Methontology
Purpose	X		
Problem analysis		X	X
Requirements analysis		X	
Integration	X	X	X
Informal specification		X	X
Formal specification	X	X	X
Documentation	X		X
Evaluation		X	X
Maintenance			X

Figure 2.14: Methodology Comparison.

The comparison is based on the presence or absence of the following activities:

- Purpose. Activity aimed to define why are we building the ontology and for what purposes.
- Problem analysis. Activity to analyse the enterprise looking for problems or situations not well addressed for the current ontology.
- Requirements analysis. Description of requirements needed to solve enterprise problems.
- Integration. Activity which considers the integration/revision of other ontologies to the development process.
- Informal specification. Create an informal specification of the conceptualization.
- Formal specification. Create the formal specification of the conceptualization.

- Documentation. Activity which makes the ontology more explicit by documenting assumptions regarding its elements.
- Evaluation. Activity necessary to check if the requirements are met.
- Maintenance. Activity to update and upgrade the ontology.

From Figure 2.14 we can see that *Integration* and *Formal Specification* activities are present on the three methodologies. On our methodology we will consider the *Formal Specification* activity to formalize the ontology in OWL language. *Integration* activity will be considered but for future work (as we already have a solid set of terms on the framework).

Problem analysis, *Informal specification* and *Evaluation* activities are present on two of the methodologies so they also will be considered on ours. *Problem analysis* activity will be taken more like framework analysis since the analysis of the problem was already done and detailed in [29].

The previously presented methodologies corresponds to *development methodologies* (covering all steps of the development process). In the case of this work, our goal is to formalize an ontology (conceptual framework), so we are not using any of these methodologies as they were conceived. In turn, we have analysed them and highlighted their principal activities/steps that could help us in the formalization process. With this, we take into consideration the experience showed on the development methodologies to count with a formalization process (which would be less complex and extensive than the development process). So, based on the previous *development* methodologies we constructed our own *formalization* methodology which will be explained in Chapter 3.

Next section shows a review of 4 ontology edition tools. These are 4 of the most

used tools available. The aim of this section is to, at the end, make a comparison of these tools and select one for our formalization process.

2.5 Tool Review

In this section we compare four ontology edition tools. The aim of this comparison is to select the most adequate tool to formalize the ontology in the practical part. The four tools to compare are the following:

Neon Toolkit. Neon toolkit is a state-of-the-art, open source multi-platform ontology engineering environment, which provides comprehensive support for ontology engineering life-cycle [17]. Neon Toolkit was developed as part of the Neon Project and supported by the Neon Foundation, is based on the Eclipse platform and supports plug ins for different ontology engineering activities.

Protégé. Protégé is a free, open source ontology editor and knowledge-base framework that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies [22]. Based on Java, is extensible, and provides a plug-and-play environment (plug in capability). It is a resource for biomedical ontologies and knowledge bases supported by the National Library of Medicine.

TopBraid Composer. TopBraid Composer is an enterprise-class modelling environment for developing Semantic Web ontologies and building semantic applications [24]. It is implemented as an Eclipse plug-in and incorporates a flexible and extensible framework with a published API for developing semantic client/server or browser-based solutions.

COE. COE is a project whose goal is to develop an integrated suite of software tools for constructing, sharing and viewing OWL encoded ontologies based on Cmap-

Tools [6].

The comparison is made with respect to the following criteria:

- **Visualisation features.** This feature corresponds to how the tool allow us to graphically visualize the ontology. Also the graphic features or controls that help the user to understand better the ontology.
- **Import/Export capabilities.** Corresponds to the options to import and export ontologies, including file types.
- **Ontology edition.** This characteristic corresponds to how the tool help the user to edit an existing ontology, giving tips, facilitating it edition or helping to reduce errors on the editing task.
- **Ontology evaluation capabilities.** Describe how (if) the tool help to discover and repair inconsistencies on the ontologies.
- **Reasoners.** What reasoners are available in the tool.
- **Ontology capture.** How the software helps to write a new ontology i.e. incorporate new classes, restrictions, individuals, etcetera.
- **Usability.** It is the software tool easy to use?, the interface is clear and somehow maintain the aspect of other similar software tools for ontology development?, how intuitive is the navigation?.
- **Plug Ins.** Does the tool support the addition of plug ins to increase its power?.

The four software tools were evaluated according to the above criteria. The evaluation is the following:

Visualisation

- *Neon Toolkit*: On the visualization environment we can find utilities like Zoom, Rotation, Undo and Redo, Search, History, Legend and Tooltip. Visualisation is very clear, easy to understand and to navigate. Navigation history allows us to look into the ontology elements that we checked before, since this feature is graphically implemented one can jump from one level to another any time because all the visited elements are at sight all the time.

The “Legend information is very useful to identify the elements of the ontology on the graphic representation. Search capability allows us to go and visualize directly the part of the ontology we want to. This visualizer already comes with Neon Toolkit. We have the alternative to download different visualization plugins for Neon Toolkit. One is KC-Viz, this one has useful screen adjustments that facilitates the visualization of the ontology, information about the type, URI, direct super and sub-classes are shown when the mouse pointer is over one element of the diagram. There is no option to export the diagram as an image file.

We have also available a relationship visualization plug in, this one shows the relation between elements (different elements are shown in different colors), this plug in also have screen adjustments controls.

- *Protégé*: To visualize the ontology we need the OwlViz plug-in. This plug in offer us screen adjustment controls, show/hide options (for classes, children and parents). In OwlViz the icons can be too big, sometimes visualizations of large ontologies can lead to large diagrams (some parts of the ontology does not fit on the screen). This plug-in offer us a color differentiation between inferences and asserted classes. Also the inconsistencies are highlighted in red. We can choose to visualize the asserted model or the inferred model.

- *TopBraid*: The free edition of TopBraid does not come with the visualization tool.
- *COE*: A conceptual map way to visualize the ontology. Can be confusing with large ontologies. Zoom controls does not allow us to see a large diagram without reduce too much name labels o tags of ontology elements (if those are reduced too much might be not readable).

Import/Export

- Neon ToolKit:
 - Import: Open ontology from file and load ontology from the web (URI).
 - Export: OWL2 in RDF/XML (.owl, .rdf), OWL/XML syntax (.owlx), OWL2 functional style syntax (.owl2), OWL Manchester syntax (.own) and Turtle syntax (.ttl).
- Protegé
 - Import: Open ontology from file and load ontology from the web (URI).
 - Export: RDF/XML, OWL/XML, OWL functional syntax, Manchester OWL syntax, OBO 1.2 flat file, KRSS2 s syntax, Latex and Turtle
- TopBraid
 - Import: Open ontology from file and load ontology from the web (URI).
 - Export: RDF/XML Abbreviated (.rdf, .rdfs, .owl), RDF/XML (.rdf, .rdfs, .owl), Turtle/N3 (.ttl, .N3) and N-Triple (.nt).
- COE
 - Import: Open ontology from file and load ontology from the web (URI).
 - Export: OWL, N-Triple, N3-PP, N3-Plain, N3-triple and Turtle.

Edition

- Neon Toolkit: Clear and intuitive edition interface that shows suggestions as the user is writing so typing errors are minimized. Based on tab navigation with which we can select what do we want to see or edit. Edition task can be done mainly with two panels: ontology navigation (in which we select the ontology elements) and property window (in which we see/edit the element).
- Protégé: Edition task can be done mainly in two panels: ontology navigation (in which we select the ontology element) and property window (in which we see/edit the element). Through tabs in the top of the window we can change among the different elements of the ontology (entities, classes, object properties, data properties, individuals, etcetera.). To edit some element we can select the new value from a list, with this, we avoid writings errors.
- TopBraid: A double click on a ontology element give us access to its properties. Not so intuitive to add or edit characteristics, does not complete the text when we are editing it, this can lead us to writings errors. As it is everything on the same tab, if the element has large descriptions, its display it is sometimes confuse.
- COE: Edition must be done in a graphical way, directly on the conceptual map.

Ontology evaluation

- Neon Toolkit: To evaluate the ontology we need the RaDON plug in. This plug in allow us to check and repair inconsistencies and incoherences on the ontology. We can visualize unsatisfied classes, see where (justification) the inconsistency or incoherence is and also we have the options of repair the ontology automatically or manually (if it is manually, the plug in show us some suggestions for repairing).

- Protégé: There is no build in or plug in capability for ontology evaluation (the only option is using reasoners, that we are going to review ahead).
- TopBraid: There is no build in or plug in capability for ontology evaluation on the free edition of TopBraid.
- COE: There is no build in capability for ontology evaluation.

Reasoners

- Neon Toolkit: Here we have an infrastructure plug in which provides access to the Pellet2 and HermiT3 reasoners. When we use this functionality, the tool materializes inferences into a new ontology.
- Protégé: In Protégé we have available the following DL reasoners: FaCT++, HermiT, Pellet and Pellet Incremental.
- TopBraid: We do not have reasoners on the free edition of TopBraid composer.
- COE: No reasoners available.

Capture

- Neon Toolkit: Intuitive and easy way to create ontologies. When adding restrictions to the classes and object properties the tool show suggestions while we are writing so we can choose the correct element we want to add and no make typing mistakes.
- Protégé: Intuitive and easy way to create ontologies and add elements to it. When adding restrictions to classes and object properties we have to select the element from lists, avoiding typing errors.
- TopBraid: Intuitive way to add classes and object properties. Can be confusing because its interface shows too much information. More time to learn how to use the tool is needed in comparison with previously tested tools.

- COE: Graphic way to create the conceptual map representing the ontology. The user can select templates to add new classes and relations among them.

Usability

- Neon Toolkit: Easy to use and intuitive, with its interface similar to other ontology editors facilitates its utilization. Clear tab and panel navigation.
- Protégé: Interface similar to other common used editors makes Protégé look familiar. As a consequence of the previous characteristic is intuitive and easy to use and fast to learn.
- TopBraid: As the previous tools has a similar interface. Relatively easy to use and fast to learn. The fact that shows too much information on a single tab make it a little confuse. Relatively fast to learn.
- COE: Different from all the other editors presents on this comparison but at the same time very simple to understand and use. Its simple interface make it less confusing and very fast to learn.

Plug Ins

- Neon Toolkit: Support different plug ins that can be used for multiple purposes.
- Protégé: Support different plug ins that can be used for multiple purposes.
- TopBraid: The free edition supports the addition of plug ins which can be for multiple purposes.
- COE: Does not support plug ins.

Figure 2.15 shows a summary of the tool comparison.

Tool	Visualisation	Edition	Evaluation	Reasoners	Capture	Usability	Plug Ins
TopBraid	Clear visualisation with useful controls. Built in visualiser and plug in visualiser. Also available a relationship visualiser.	Less intuitive than others. Does not complete names (easy to make mistakes).	No build in or plug in capability for ontology evaluation.	No reasoners on the free edition.	Intuitive interface but shows too much information.	Relatively easy to use and fast to learn shows too much information-	Supported
COE	Conceptual map visualisation Can be confusing on large ontologies.	Graphical edition	No build in capability for ontology evaluation.	Not supported	Graphical capture. Templates available.	Simple and easy to use and understand.	Not supported
Protégé	Acceptable plug in visualisation. Color differentiation, useful visualisation controls, possibility of visualise asserted and inferred model.	Clear and easy edition. Tab navigation showing the different elements to edit. Its possible to choose values from lists (avoid errors).	Available via reasoners.	DL reasoners: FaCT++, HerMiT, Pellet and Pellet Incremental.	Intuitive and easy to learn Its possible to select elements from a list.	Simple and easy to use and understand.	Supported
NeOn	Clear visualisation with useful controls. Built in visualiser and plug in visualiser.	Clear and intuitive with suggestions as one is writing Similar to other editors.	We need the RaDON plug in which hepl us to check and repair inconsistencies and incoherences on the ontology.	Pellet2 and HerMiT3	Intuitive and easy to learn Suggestions are shown as the user is writing.	Simple and easy to use and understand.	Supported

Figure 2.15: Tool Comparison Summary.

2.5.1 Choosing a Formalization Tool

Based on the previous tool comparison we have selected as a formalization tool Protégé. On our evaluation COE is the first to be dismissed, because just provide a graphical way of work, does not count with reasoners, does not allow plug ins, among other reasons (although is dismissed to be the primary formalization tool, we are going to use it on a early development stage).

The second tool to be dismissed is TopBraid Composer. Although it is a very powerful tool to our purposes, the free tool available for download it is limited on functionality. For example, we don't have reasoners on the free edition, and an important part of our work is based on reasoners.

We have two options left, Protégé and NeOn Toolkit. Both tools are very similar based on our comparison criteria but we choose Protégé because its graphical capabilities are superior. Also, another key reason is that the support community behind Protégé is very solid and large.

The version of Protégé used to formalize the ontology is the 4.0.0(Build 213). The reasoner used to obtain inferences is Pellet.

As we said before COE is dismissed as a primary formalization tool but we are going to use it to develop conceptual maps of the ontology which will help us to visualise it better. These conceptual maps will be later exported to Protégé. The version of COE used to construct our conceptual maps is 5.0.03.

On this chapter we revised key concepts like ontologies, enterprise ontologies, agent and context based models, ontology development methodologies, and ontology development tools. This corresponds to the necessary background to start the devel-

opment of our work. Next chapter will describe the methodology that we have used to formalize the conceptual framework described earlier on this chapter.

Chapter 3

Formalization Methodology

On this Chapter we describe the methodology used to formalize the ontology presented in section 2.3.1 of Chapter 2.

3.1 Brief Description

Basically, the formalization methodology is based on a iterative process executed along with a client-provider approach. This means that the client (in this case the framework author) request the provider (in this case the author of the present Master thesis) the framework formalization.

The iterative nature of this methodology is given by regular meetings in which goals are agreed between client and provider and results of previous meetings are checked and evaluated. Depending on these results, goals are adjusted. i.e. depending on how the initial framework elements are being modelled its following elements may be re-conceived. Also, the review of the documentation is present throughout the entire process of formalization helping and supporting the iterative approach.

This methodology encompasses all the process of formalization, starting very early

with the revision of associated documentation until the evaluation of the final product. With this we tried to reflect all the necessary stages that we performed to obtain the final result.

The following corresponds to a detailed description of the different stages of the formalization methodology.

3.2 Methodology

The following corresponds to a description of the methodology used to formalize the conceptual framework presented in section 2.3.1 of Chapter 2.

3.2.1 Documentation Review

In this stage a revision is made about relevant information sources needed to formalize the ontology. This revision is made in a Bottom-Up fashion, starting first from the most general information sources to the most specific ones. The order followed was:

1. *Ontologies*. Ontology definition and history material was reviewed. Basically trying to situate the ontology on the information science area, understanding its beginnings on it. Since the word ontology has broad utilization in different areas and a specific meaning on each of them, the focus was also to look for a definition applied to information science field that could integrate the key factors present on the specific documentation for the area.

As a result for this point, the necessary background to continue with more specific tasks was achieved.

2. *Ontology Applications*. After having a background on what ontologies are, we moved on to ontology applications. It was necessary to understand the power of the subject of study, how is useful and its current uses. Studying this part

gave us an idea of the many applications of ontologies on information sciences.

3. *Specific Ontologies.* Here we checked how a materialized ontology looks like. To do this, very well known ontologies were reviewed, like the Person ontology, Pizza ontology or Wine ontology, all of them available (to open as URI) in Protégé. Several documentation is available regarding these “famous” ontologies, also several manuals and tutorial guides showing how to construct a new ontology or how ontologies works.
4. *Enterprise ontologies.* Definitions, applications and examples of ontologies corresponds to the general information needed to start refining our research. It was time to narrow the research to a more specific topic in our area, enterprise ontologies. Here, well known and prestigious ontologies were reviewed, like EO, TOVE and Dietz ontology (described in Chapter 2 on section 2.2). The point of this was to have a clear notion of how the enterprise is modelled on ontologies that are already accepted as correct. As a result we had an idea of the terms, relations and other necessary elements in modelling enterprises. After this, with more specific knowledge, was time to move on the specific topic of the present work, agent and context based ontologies.
5. *Agent and Context based Ontologies.* This was the most specific topic studied on this stage. Corresponds to read and understand the conceptual framework detailed in [29]. This was the starting point to begin formalizing the ontology because all terms, relations and semantics were obtained from that document (meetings also provide a source of information). Additionally, a UML informal organizational meta-model was studied in order to have a graphical and more “physical” notion of the ontology. The document and the model were fundamental and essential information source and all the future development relies on the correct understanding of them.

Roles played . Analyser.

Stage input . Framework main document [29], background papers and the informal organizational meta-model.

Stage output . No physical outputs.

3.2.2 Information Gathering

In this stage we gather specific data and information, we filter it in order to have what is useful to us and finally we structure it in a way that is more understandable for our work and in accordance with formats used later.

Here, the information obtained from the previous stage is organized in a way that makes the formalization easier. Essentially, two lists and two correspondent tables were used to organize information. The first list containing essential concepts from the framework and the second one the relation between concepts. Likewise, the first table contains the essential terms and its related terms (constructed from the first list) and the second table contains the essential relations and the terms involved on them.

Roles played . Analyser.

Stage input . Framework main document [29], background papers and the informal organizational meta-model.

Stage output . List containing essential concepts from the framework (Figure 4.1), list containing relations between terms (Figure 4.3), table containing the essential terms and its related terms (Figure 4.2) and table containing the essential relations and the terms involved on them (Figure 4.4).

3.2.3 Informal Conceptualization

As formalization begins with a conceptual map that later will be imported in Protégé, the previous lists and tables are used to draw informal pieces of “conceptual maps” in paper. The idea is start working from very little pieces of the ontology to step by step visualize the entire set of concepts and relations.

Thus, on this stage, we draw a set of sketches containing little pieces of an early conception of the conceptual mas. Here, sketches are small and does not contain more than 10 entities and a few amount of relations among them.

Roles played . Analyser, designer.

Stage input . List containing essential concepts from the framework (Figure 4.1), list containing relations between terms (Figure 4.3), table containing the essential terms and its related terms (Figure 4.2) and table containing the essential relations and the terms involved on them (Figure 4.4).

Stage output . Paper sketches representing essential terms and its relations (Appendix B).

3.2.4 Conceptual Map Formalization

This stage corresponds to the first part of the formalization of the ontology. Here, we constructed conceptual maps using all the information revised and produced from previous stages. The idea was to export the conceptual maps into OWL files and then import them into Protégé, merge them and continue working on the resulting ontology.

To facilitate understanding, development and visualization, the ontology was divided into two conceptual maps, each of them containing different terms. The first

conceptual map contains terms related with Context and Agent and the second one Action, Interaction and Resources.

Roles played . Analyser, designer.

Stage input . List containing essential concepts from the framework (Figure 4.1), list containing relations between terms (Figure 4.3), table containing the essential terms and its related terms (Figure 4.2), table containing the essential relations and the terms involved on them (Figure 4.4), paper sketches representing essential terms and its relations (Appendix B), framework main document [29] and the informal organizational meta-model.

Stage output . Two conceptual maps representing all the terms and relations of an early version of the ontology (Figure 4.7, Figure 4.8, Figure 4.9 and Figure 4.10).

3.2.5 Conceptual Map - Protégé Transition

In this stage, conceptual maps created on the previous stage, are exported to two OWL files and imported into Protégé. Here, both OWL ontologies are merged in one containing the entire set of terms and relations.

Before start working with the merged ontology a deep revision is needed due to possible export/import errors. This is also key phase because we need the conceptual maps to be updated and as much consistent with the OWL ontology as possible. So for each issue that we found on the OWL ontology we have to come back to the conceptual maps and modify them. We did this several times until an “acceptable” OWL version was achieved. From this point, changes are made directly into the OWL ontology.

Roles played . Analyser.

Stage input . Two conceptual maps representing all the terms and relations of an early version of the ontology (Figure 4.7, Figure 4.8, Figure 4.9 and Figure 4.10).

Stage output . OWL ontology (early version).

3.2.6 Ontology Refinement

This is the stage in which the ontology is carefully revised, either to correct errors or to add new features to it. This is a critical stage on the methodology because its final result will be the final version of the ontology.

Key activities for this stage are meetings with the client and documentation review since the ontology need to be continuously reviser, approved and eventually modified.

Roles played . Analyser, designer.

Stage input . OWL ontology (early version), framework main document [29] and the informal organizational meta-model.

Stage output . OWL ontology (final version).

3.2.7 Ontology Population

On this stage examples were incorporated to the final version of the ontology. These examples were obtained from [29], Chapter 3. The point is to have a solid and already tested source of comparison to be able to check results obtained from rule addition and query placement into the ontology.

Specifically, information regarding activities, actions, action types, agents, context resources and objects was incorporated to the ontology. This examples gave a basis

solid enough to start adding rules and testing queries.

The following corresponds to an example of the information introduced to the ontology (obtained from [29], Chapter 3, part of “*Prof. Smith’s payment context*”). This example shows one phrase, containing one action. Below, the phrase is divided showing its element separately.

Prof. Smith request Alice the payment of a course recently finished.

- *Individuals*: Prof. Smith.
- *Action type*: Request (communicative action).
- *Embedded action*: (a1) **Alice pay a course recently finished.**
 - *Individual*: Alice.
 - *Action type*: Pay (non communicative action).
 - *Object*: a course recently finished.

Roles played . Analyser.

Stage input . OWL ontology (final version) and framework main document ([29]).

Stage output . OWL ontology (final version) with examples.

3.2.8 Rule addition and Query Testing

This is the last stage of the methodology. Here, we receive the ontology already populated with examples and we added rules and develop queries in order to answer questions and to obtain desire information from it.

In Protégé, we added rules to infer new information starting from the existent. These rules associate classes and relations to obtain new relations between entities

increasing the expressiveness of the ontology.

Queries posted on the ontology, combined with rules, give us new useful information that was not explicitly present on the ontology. These two elements are necessary to achieve the objectives of the present work.

As an example, the following corresponds to a rule developed on this stage and introduced to the ontology. This rule set an action as performed by a specific agent

```
"communicativeAction(?a), sentBy(?a, ?s) -> perform(?s, ?a)"
```

Also, as an example, the following corresponds to a query constructed on this stage and posted on the ontology. This query returns all commitments of a given agent.

```
"commitment and (isEmbeddedIn some (communicativeAction and sentTo value AGENT))"
```

Roles played . Analyser, designer.

Stage input . OWL ontology (final version) and framework specification [29].

Stage output . Set of rules and queries.

Figure 3.1 shows the methodology used in this work to formalize the ontology.

The outputs of the previous stages are presented on the next chapter.

During all the formalization process regular meetings occurred between client and provider in which results were checked and goals adjusted. The documentation review task was used to support this process which provide guidelines to make changes and make adjustments. Ontology formalization results and its evaluation can be found

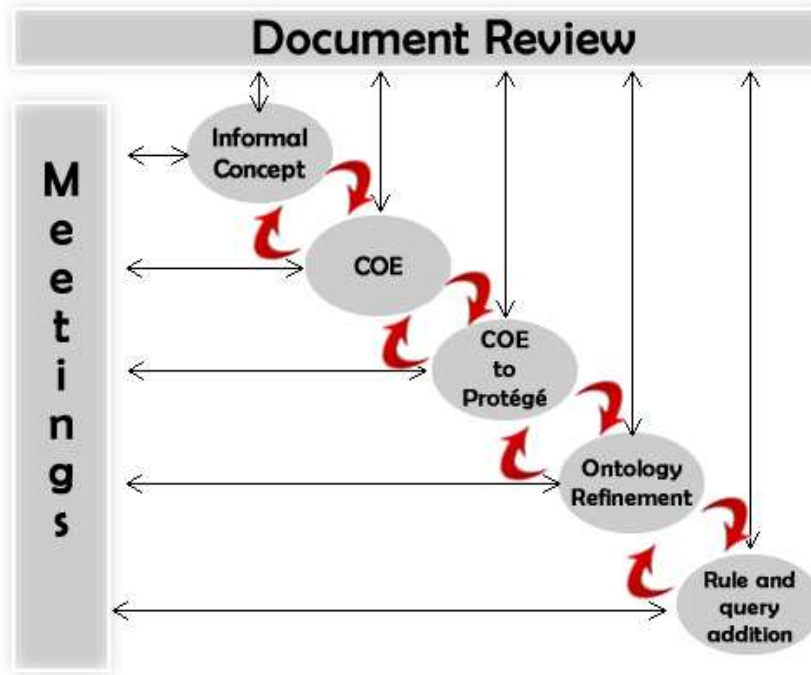


Figure 3.1: Formalization methodology.

on the next Chapter.

On this chapter we described the methodology that we followed to accomplish the formalization. This methodology has an iterative nature and is based on regular meetings with the author of the framework and the continuous revision of main documents. The main output of this methodology corresponds to the formalized ontology containing a set of example cases and a set of rules and queries.

On the next chapter we show the intermediate results of every stage (i.e. the outputs of every stage) and the final output resulting from the formalization process. Also we evaluate these results comparing them with the source from which we obtained the example cases.

Chapter 4

Formalization Results

In this chapter we show the results and evaluation of the ontology formalization process leaded by the methodology explained in Chapter 3. Here, methodology intermediate results (stage outcomes), final results and its evaluation are presented.

4.1 Methodology Results

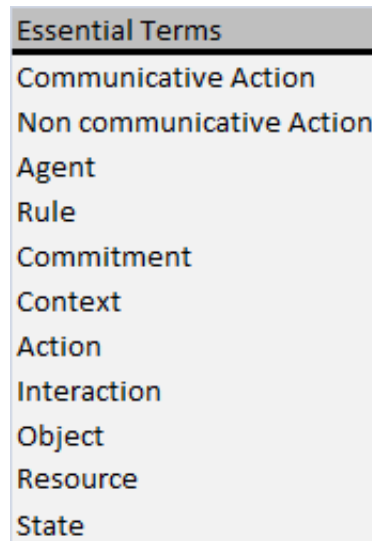
We have divided our results in terms of the formalization methodology presented on the previous chapter. Thus, for every stage on the methodology we present its outcome. We move through the different stages showing intermediate results because is a form of showing these results in a more structured and ordered way.

4.1.1 Documentation Review Outcome

The output of this stage is not physical and correspond to the necessary background to understand ontologies and specifically enterprise ontologies and agent-context based ontologies to continue with the present work.

4.1.2 Information Gathering Outcome

Two lists and two tables are the outputs of this stage of the methodology. Figure 4.1 correspond to the list containing the essential terms obtained on early revisions of the documentation.



Essential Terms
Communicative Action
Non communicative Action
Agent
Rule
Commitment
Context
Action
Interaction
Object
Resource
State

Figure 4.1: List I: Essential Terms.

Figure 4.2 correspond to the associated table containing terms and related terms.

Figure 4.3 correspond to the list containing the essential relations obtained on early revisions of the documentation.

Term	Related Terms
Communicative action	Agent, Commitment, Context, Action.
Non-communicative Action	Agent, Context, Action, Object, Resource
Agent	Communicative Action, Non-communicative Action, Rule, Commitment, Context, Action, Interaction, State
Rule	Agent, Commitment, State
Commitment	Communicative Action, Agent, Context, Action, Interaction, State
Context	Agent, Action, Resource, State
Action	Agent, Commitment, Interaction, Object, Resource
Interaction	Agent, Context, Communicative Action
Object	Agent, Resource, Non-communicative Action
Resource	Non-communicative Action, Agent, Context, State Action, Object
State	Agent, Context, Action, Commitment

Figure 4.2: Table I: Terms and its related terms.

- | Essential Relations |
|---------------------|
| Perform Action |
| Has Action |
| Has Commitment |
| Belongs to Context |
| Has Resource |
| Has State |
| Has Object |
| Has Embedded Action |

Figure 4.3: List 2: Essential Relations.

Figure 4.4 correspond to the associated table containing relations and related terms.

Relation	Related Terms
Perform Action	Agent and Action
Has Action	Context and Action
Has Commitment	Agent and Commitment
Belongs to Context	Action and Context
Has Resource	Action and Resource
Has State	Agent and State
Has Object	Action and Object
Has Embedded Action	Communicative Action and Non-communicative Action

Figure 4.4: Table II: Relations and its related terms.

Information contained on the previous lists and tables corresponds to the first approximation to the formalization of the ontology and it is a superficial but useful view of the framework that allowed us to start constructing the ontology.

4.1.3 Informal Conceptualization Outcome

In this stage, simple and small figures were drawn in paper trying to represent the information of Figure 4.2 and Figure 4.4. We can see in Figure 4.5 an example of these sketches representing terms related with actions, communicative actions and resources, and in Figure 4.6 the correspondent sketch conceptual map. These sketches were drawn imitating the conceptual map representation.

These sketches are very basic and sometimes contain errors that were solved in later methodology stages. This is just one sketch of many that were drawn. More

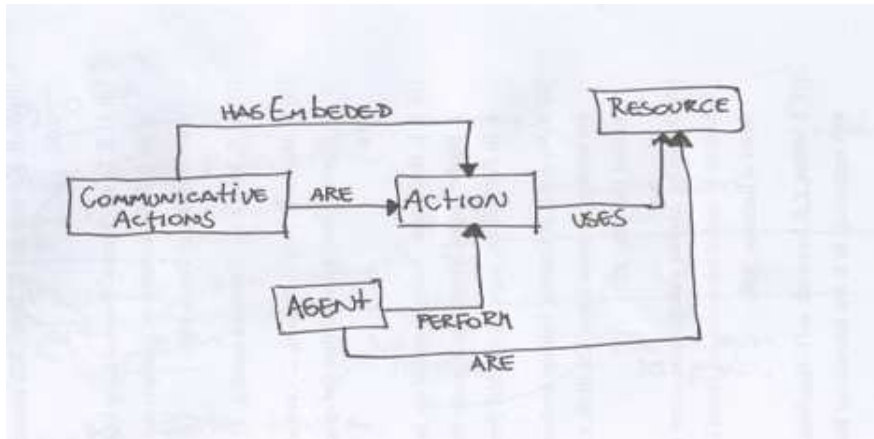


Figure 4.5: Paper sketch with conceptual map representation.

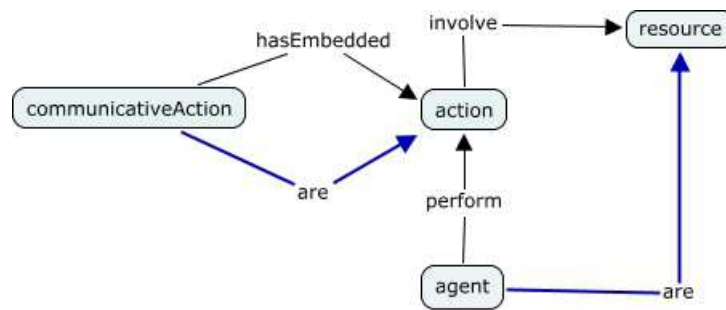


Figure 4.6: COE Conceptual map representing the sketch.

sketches can be found on Appendix B.

What was done on this stage was an important information source for the next stage in which we formalize all sketches using COE conceptual map tool.

4.1.4 Conceptual Map Formalization Outcome

The outcomes of this stage corresponds to two conceptual maps representing the ontology. Figure 4.7 and Figure 4.8 corresponds to the Action, Interaction and Resource Conceptual Map and Figure 4.9 and Figure 4.10 corresponds to the Agent and Context Conceptual Map.

These conceptual mas were constructed in a iterative way and do not represent the final state of the ontology.

4.1.5 Conceptual Map - Protégé Transition Outcome

The outcome of this stage is a single OWL ontology containing all terms and relations. This is not the final version of the ontology, on the contrary is just the final version of the conceptual maps exported to OWL format which is ready to start working with the selected ontology formalization tool (Protégé).

Figure 4.11 shows the expanded class hierarchy of an early version of the ontology. This version was improved on the next stage. Final version of the ontology is shown on the next section.

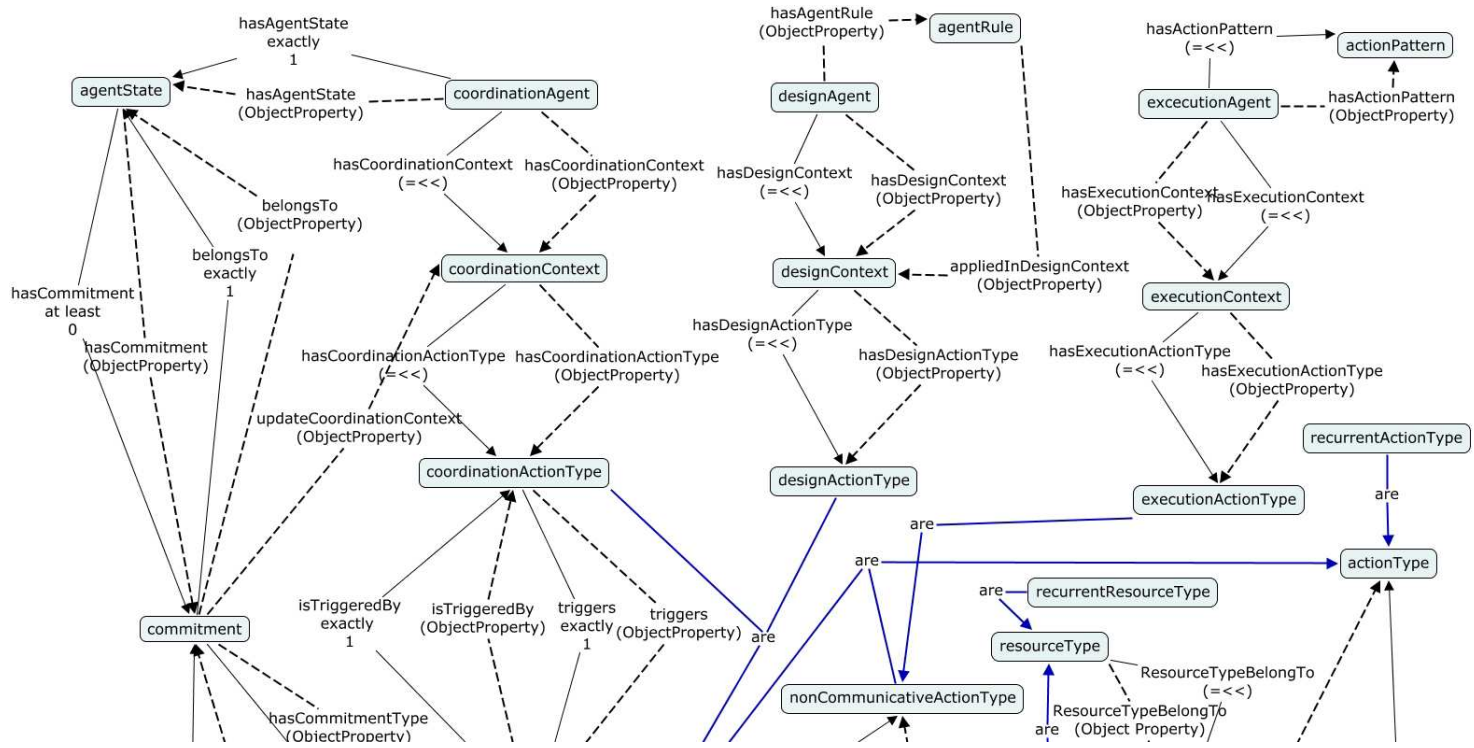
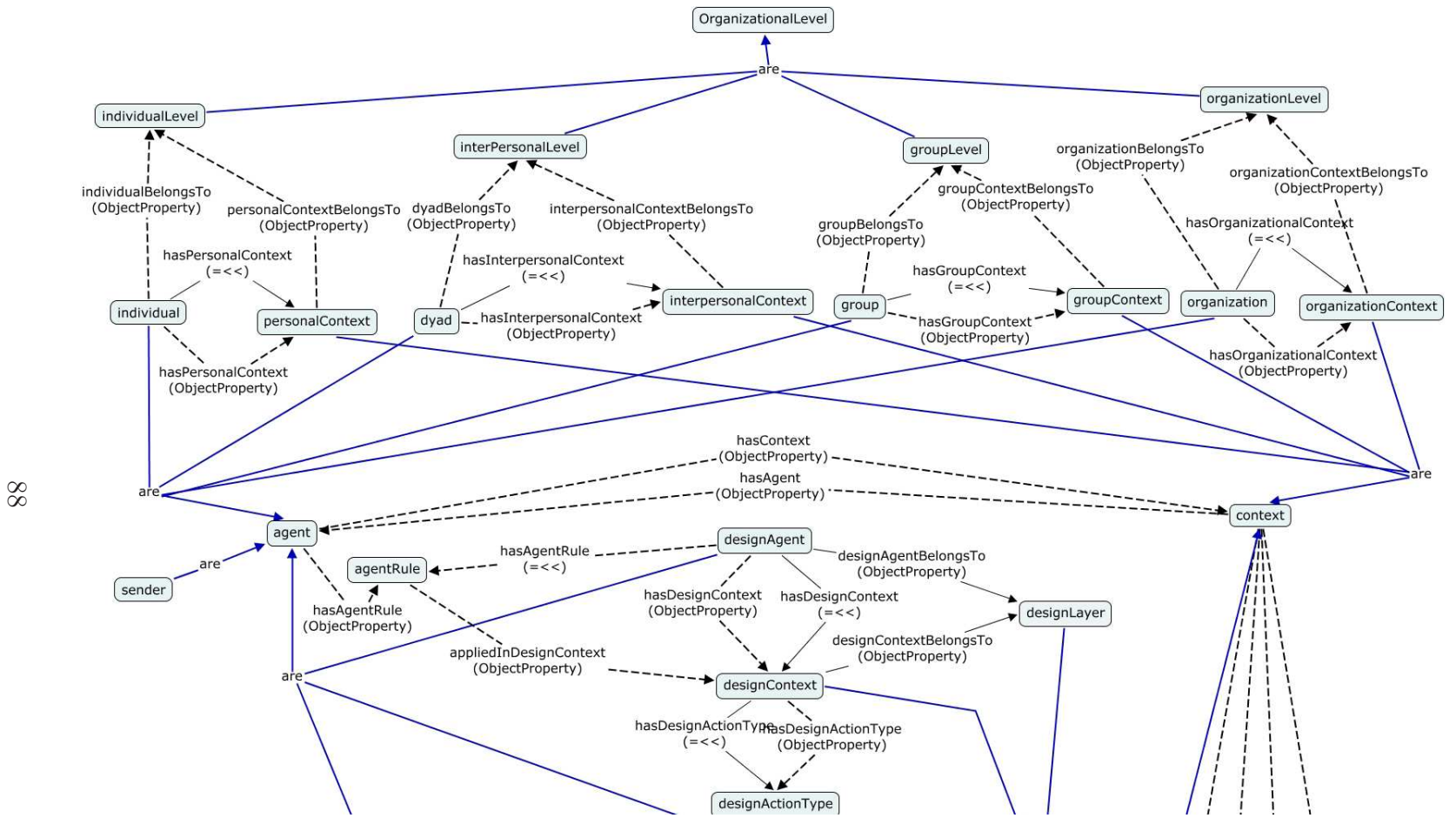


Figure 4.7: Action, Interaction and Resource Conceptual Map (Part I).



88

Figure 4.9: Agent and Context Conceptual Map (Part I).

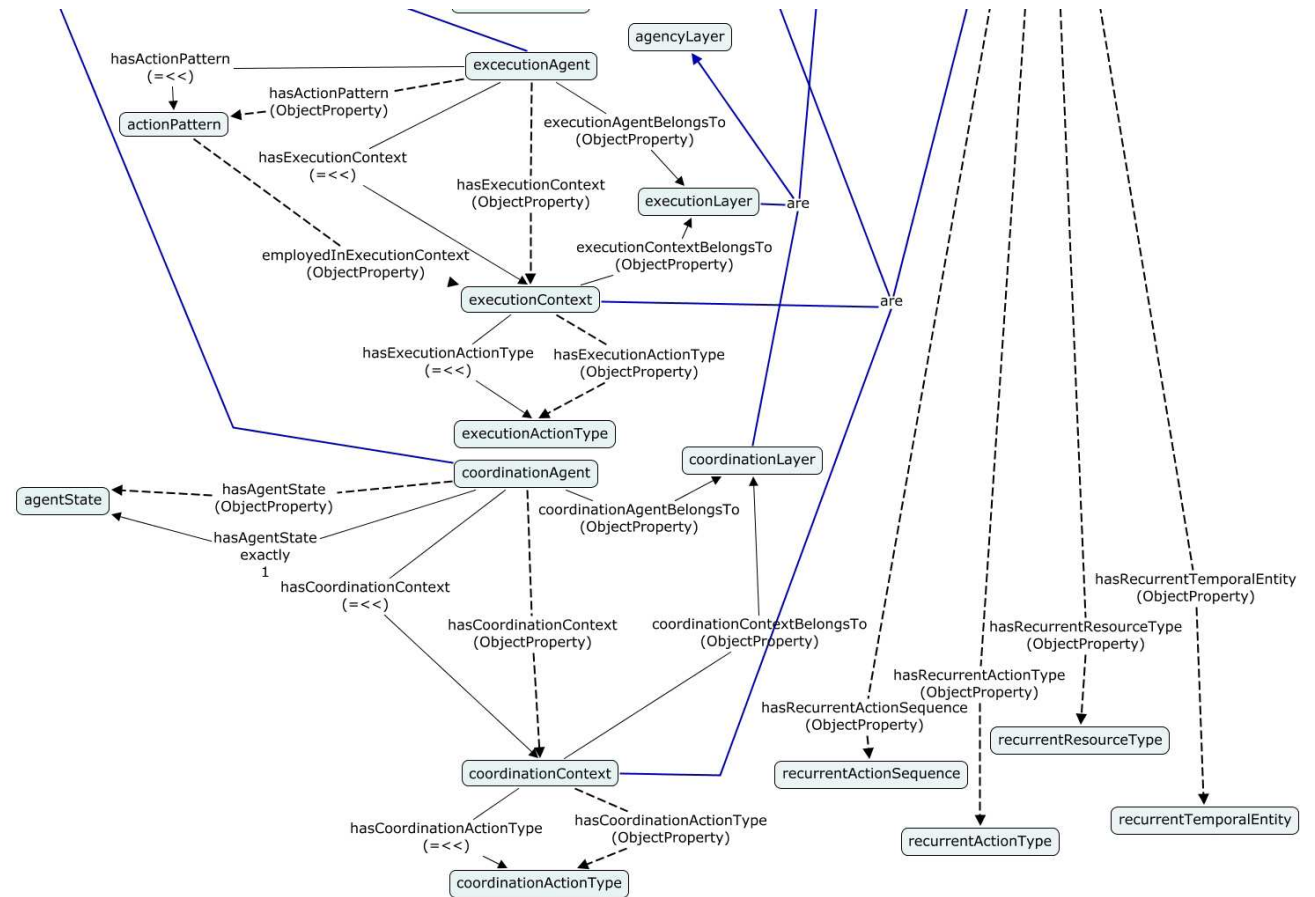


Figure 4.10: Agent and Context Conceptual Map (Part II).

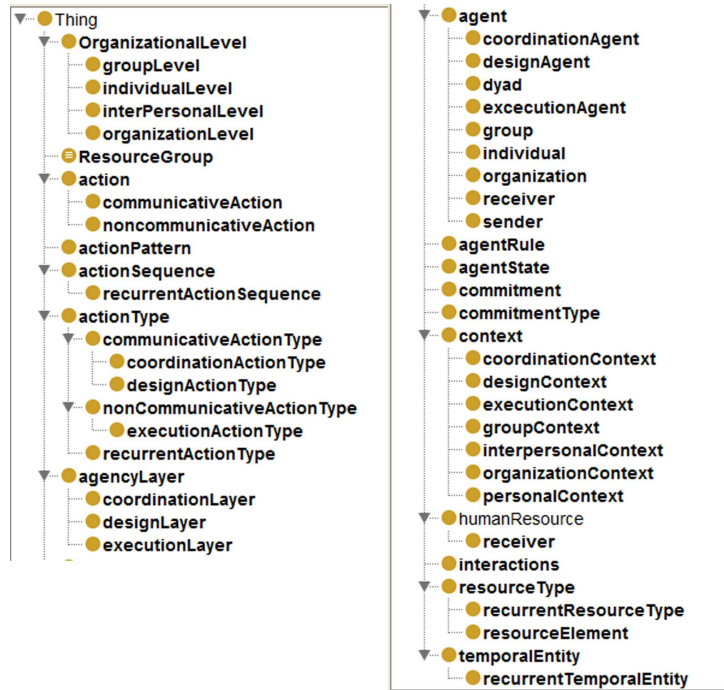


Figure 4.11: Early version of the ontology.

4.1.6 Ontology Refinement Outcome

The output of this stage corresponds to the formalized final ontology product of a long and iterative refinement process. This ontology contains all the terms and relations necessary to accomplish the objectives of this work and was accepted by the client.

Figure 4.12 shows the collapsed class hierarchy of the formalized ontology. In this screen capture we can see the main terms of the ontology which corresponds to the top of the hierarchy.

Figure 4.13 and Figure 4.14 shows the expanded class hierarchy of the formalized ontology. Here we can see the complete set of ontology classes and its hierarchy. As



Figure 4.12: Collapsed class hierarchy.

an example, on the figure we can appreciate the following:

- Class *action* has three direct subclasses: *communicativeAction*, *nonCommunicativeAction* and *embeddedAction*. Here, *embeddedAction* is a similar class with *commitment* and *nonCommunicativeAction* with *executionAction*. This corresponds to the type of actions that this ontology can represent.
- Class *actionType* has three direct subclasses: *recurrentActionType*, *communicativeActionType* and *nonCommunicativeActionType*. In turn, *communicativeActionType* has as subclasses *coordinationActionType* and *designActionType* and *nonCommunicativeActionType* has as subclass *executionActionType*.
- Class *agencyLayer* has three subclasses: *coordinationLayer*, *designLayer* and *executionLayer*. Each of these subclasses contains its correspondent agent and context sub-classifications.
- Class *agent* shows its eight types of agents: *coordinationAgent*, *designAgent*, *dyadicAgent*, *executionAgent*, *groupAgent*, *individualAgent*, *organizationalAgent*

and *sender*. This corresponds to the type of agents that are studied with this ontology.

- Class *context* shows its six type of context: *coordinationContext*, *designContext*, *executionContext*, *groupContext*, *interpersonalContext*, *organizationContext* and *personalContext*. This corresponds to the types of context recognized on this approach.
- Class *organizationalLevel* shows its four direct subclasses: *groupLevel*, *individualLevel*, *interpersonalLevel* and *organizationLevel*. This corresponds to the four levels in which the organization is conceived on this approach.
- Class *people* shows its four direct subclasses: *person*, *dyad*, *group* and *organization*. This corresponds to the different levels in which people is organized on this ontology, in complexity ascendant order.

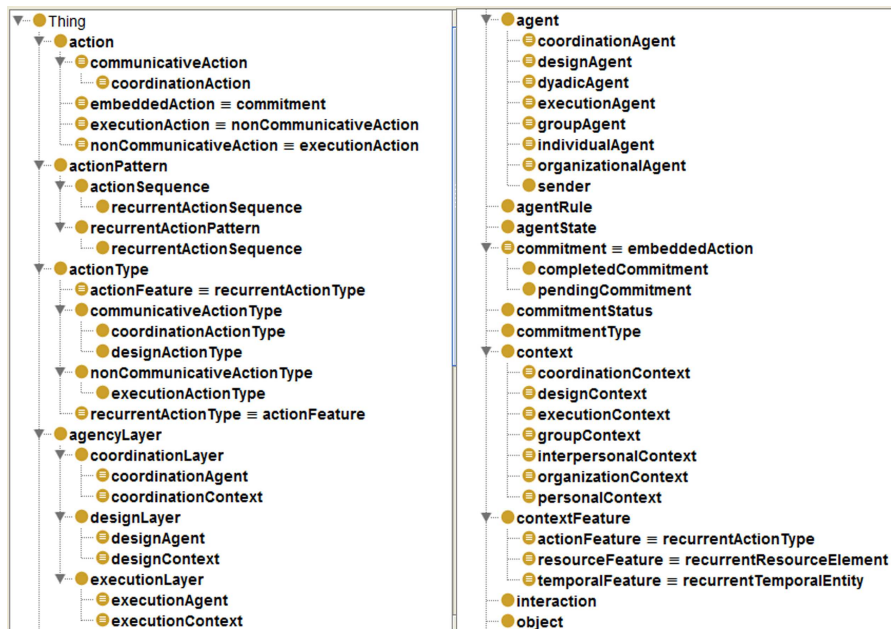


Figure 4.13: Expanded class hierarchy part 1.

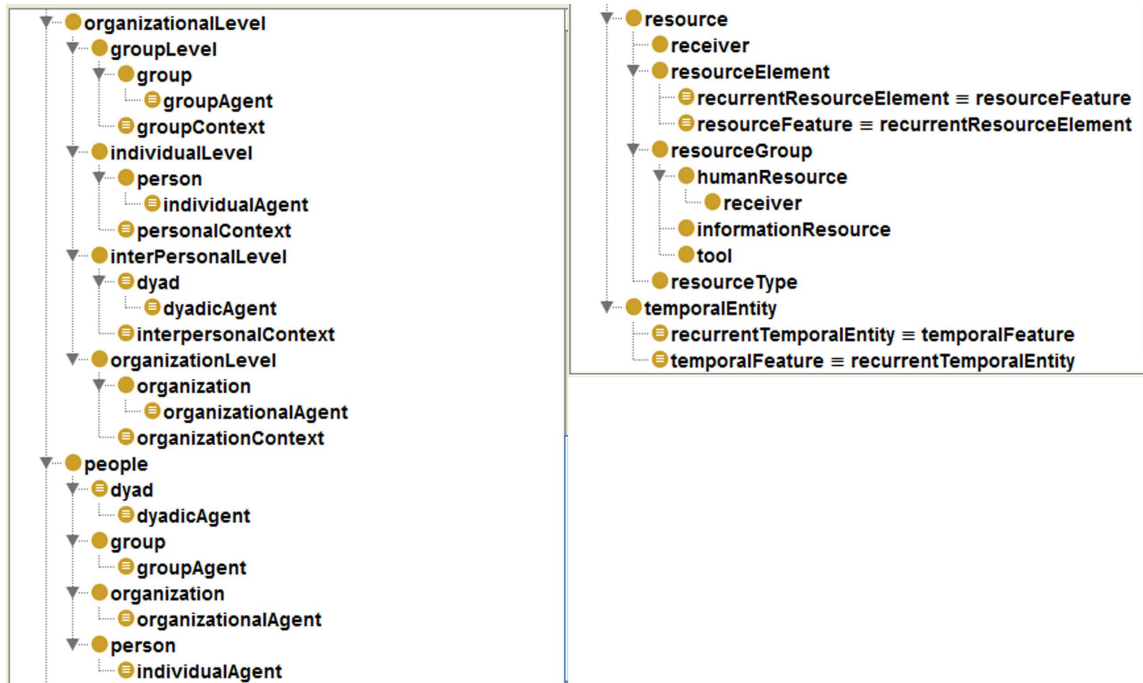


Figure 4.14: Expanded class hierarchy part 2.

Figure 4.15 and Figure 4.16 shows the description of some of the classes of the ontology. These description consists on equivalent classes (classes or expressions that have the same meaning or properties than the one described), superclasses (classes or expressions on upper levels on the hierarchy) and inherited anonymous classes (classes or expressions that are inherited by the described class).

Figure 4.15 shows class description for *communicativeAction*, *nonCommunicativeAction* and *commitment* in parts a, b and c, respectively. Similarly, Figure 4.16 shows class description for *executionAgent*, *executionContext* and *personalContext* in parts a, b and c, respectively.

Figure 4.17 shows the complete set of object properties of the ontology. Here we can highlight, for example, that property *executedBy* is equivalent to property *sentBy* and property *featureOf* has three sub-properties: *actionFeatureOf*, *resourceFeatureOf*

Description: communicativeAction

Equivalent classes +

- **action**
and (hasActionType some communicativeActionType)

Superclasses +

- **action**
- **hasEmbeddedAction some action**
- **sentBy some sender**
- **sentTo some receiver**

Inherited anonymous classes

- (hasActionType some actionType)
and (hasActionType exactly 1 actionType)
- **performedBy exactly 1 agent**

Part a.

Description: nonCommunicativeAction

Equivalent classes +

- **executionAction**
- **action**
and (hasActionType some nonCommunicativeActionType)

Superclasses +

- **action**
- **hasObject some object**

Inherited anonymous classes

- **nonCommunicativeAction**
- (hasActionType some actionType)
and (hasActionType exactly 1 actionType)
- **performedBy exactly 1 agent**

Part b.

Description: commitment

Equivalent classes +

- **embeddedAction**

Superclasses +

Inherited anonymous classes

- (hasActionType some actionType)
and (hasActionType exactly 1 actionType)
- **performedBy exactly 1 agent**
- **hasStatus max 1 commitmentStatus**
- **action**
and (isEmbeddedIn some communicativeAction)
- **commitment**

Part c.

Figure 4.15: Example 1 Class descriptions.

Description: executionAgent

Equivalent classes +

- agent
and (perform some (action and (hasActionType some executionActionType)))

Superclasses +

- agent
- executionLayer
- hasActionPattern some actionPattern
- hasContext some executionContext

Inherited anonymous classes

- perform some action
- hasContext some context

Part a.

Description: executionContext

Equivalent classes +

- context
and (hasActionFeature some executionActionType)

Superclasses +

- context
- executionLayer
- hasAgent some executionAgent

Inherited anonymous classes

- hasAgent max 1 agent
- hasAgent some agent

Part b.

Description: personalContext

Equivalent classes +

- context
and (hasAgent only person)

Superclasses +

- context
- individualLevel

Inherited anonymous classes

- hasAgent max 1 agent
- hasAgent some agent

Part c.

Figure 4.16: Example 2 Class descriptions.

and *temporalFeatureOf*.



Figure 4.17: Ontology object properties.

Some of the information that can not be seen on the figure are the inverse properties. These are shown in Figure 4.18. A inverse property of a given property is the one that relates the same entities but on the opposite way. e.g. In Figure 4.18 we can see the property *hasAgent*. This property relates entities *context* and *agent* (“*context hasAgent agent*”). Its inverse property is *hasContext* which relates the same entities (*context* and *agent*) but in a opposite way (“*agent hasContext context*”).

On Appendix A we can find an extract of the complete OWL ontology formalized on this process. The final version of the ontology is available on the digital support

Property	Inverse Property
belongToGroup	hasResourceTypes
actionFeatureOf	hasActionFeature
resourceFeatureOf	hasResourceFeature
temporalFeatureOf	hasTemporalFeature
finishSequence	finishedByAction
hasAction	partOfcontext
hasAgent	hasContext
hasCommitment	stateElementOf
hasEmbeddedAction	isEmbeddedIn
hasFeature	featureOf
hasObject	objectOf
hasPerson	partOfDyad
hasReply	replyOf
hasResourceElement	resourceElementOf
initiateSequence	initiatedByAction
isTriggeredBy	triggerCommitmentType
perform	performedBy

Figure 4.18: Object inverse properties.

(CD) that comes with the present document and on the WEB following the URI

‘‘http://www.semanticweb.org/ontologies/2011/Agent-Context_Ontology.owl’’

4.1.7 Ontology Population Outcome

In this stage we took examples form [29] (Chapter 3) and we introduced on the ontology via Protégé. On this document we found sentences with which we worked to structure them in a way that their introduction into Protégé were more direct. On Appendix C we can find an example of these sentences representing the “*Prof. Smith’s payment*” context with the views of agents Alice, Prof. Smith, Alice’s Boss and Luisa.

Figure 4.20 and 4.21 corresponds to the full set of examples introduced to the

ontology. The column *code* correspond to the actions name, *actionType* corresponds to the action type associated to the action, *Object* corresponds to the description of the object, *On* correspond to the number (name) of the object, *Resource(s)* corresponds to the resource(s) that the action needs for its execution and finally *Context* corresponds to the context number (name).

The correspondence between contexts numbers in Figures 4.20 and 4.21 (names on the ontology) and context names (presents in [29]) is given by the following table.

Id	Name	Number	Agent
a1	Prof. Smith's payment	1	Prof. Smith
	Prof. Smith's payment	2	Alice
	Prof. Smith's payment	3	Alice's Boss
	Prof. Smith's payment	4	Luisa
a10	POSI meetings	5	Alice
	POSI meetings	6	Miguel
a11	TR (POSI Sub-contractor)	7	Alice
	TR (POSI Sub-contractor)	8	Luisa
a5	Travel Arrangements	9	Alice
	Travel Arrangements	10	Luisa
	Travel Arrangements	11	Aida
a7	Project Management courses	12	Alice
a3	POSI budget training & supervision	13	Alice
	POSI budget training & supervision	14	Luisa

Figure 4.19: Correspondence between ontology contexts and document contexts.

After this stage, the ontology is almost complete. The final stage will incorporate rules and will allow us to answer the questions stated in Chapter 1.

N	Code	Agent	actionType	Object	O n	Resource(s)	Context
1	action01	prof. smith	request (c)				1
2	action01-1	alice	pay (nc)	course x	o1	e-mail	2
3	action02	alice	check (nc)	prof. smith payment requirements	o2	excel, payment requirement records	2
4	action03	alice	request (c)				2
5	action03-1	prof. smith	send (nc)	course grades and the corresponding report	o3	e-mail	1
6	action04	alice	inform (c)				2
7	action04-1	prof. smith	know (nc)	will proceed when requirements are sent	o4	e-mail	1
8	action05	prof. smith	inform (c)				1
9	action05-1	alice	know (nc)	reason for not sending requirements	o5		2
10	action06	prof. smith	request (c)				1
11	action06-1	alice	pay (nc)	without payment requirements	o6	e-mail	2
12	action07	prof. smith	promise (c)				1
13	action07-1	alice	expect (nc)	requirements for date d	o7	e-mail	2
14	action08	alice	analyze (nc)	payment request and reason given	o8		2
15	action09	alice	request (c)				2
16	action09-1	alice's boss	analyze (nc)	payment request and reason given	o9	phone	3
17	action10	alice's boss	suggest (c)				3
18	action10-1	alice	know (nc)	payment acceptance	o10	phone	2
19	action11	alice's boss	inform (c)				3
20	action11-1	alice	know (nc)	prof. smith is a good professor	o11	phone	2
21	action12	alice	accepts (c)				2
22	ac12-1	prof. smith	know	payment request of prof. smith	o12	e-mail	2
23	action13	alice	order (c)				2
24	action13-1	luisa	pay (nc)	course x to prof. smith	o13		4
25	action14	alice	inform (c)				2
26	action14-1	prof. smith	know (nc)	payment is ordered	o14	e-mail	1
27	action15	alice	verify (nc)	last posi meeting minute	o15	word, posi minute file	5
28	action16	alice	inform (c)				5
29	action16-1	miguel	know (nc)	last posi meeting is ok	o16	e-mail	6
30	action17	alice	order (c)				7

Figure 4.20: Examples introduced to the ontology (Part 1).

N	Code	Agent	actionType	Object	O n	Resource(s)	Context
31	action17-1	luisa	pay (nc)	march-april it support	o17	e-mail, march-april invoices	8
32	action18	alice	send (nc)	software cds for installation	o18	snail mail, cds	7
33	action19	alice	request(c)				9
34	action19-1	luisa	search (nc)	travels folder	o19		10
35	action20	alice	request (c)				9
36	action20-1	luisa	register (nc)	invoice data in travels file	o20	excel, travels folder, travels file	10
37	action21	alice	request (c)				9
38	action21-1	luisa	compare (nc)	invoice value with budgeted value	o21	excel, travels budget file	10
39	action22	alice	answer				9
40	action22-1	aida	know	information about ticket payment	o22		11
41	action23	alice	give (nc)	plane tickets	o23		9
42	action24	alice	request (c)				9
43	action24-1	travel age	give (nc)	ticket invoice	o24	phone, ticket information	
44	action25	alice	order (c)				9
45	action25-1	luisa	make (nc)	make other travel arrangements	o25		10
46	action26	alice	print (nc)	pm course proposal file	o26	word, pm course proposal file	12
47	action27	alice	store (nc)	pm course proposal in short courses folder	o27	word, course folder	12
48	action28	alice	inform (c)				13
49	action28-1	luisa	know (nc)	how to verify posi budget	o28	excel, posi budget file	14
50	action29	alice	inform (c)				13
51	action29-1	luisa	know (nc)	how to attach student form and check	o29	word, student form file	14
52	action30	alice	inform (c)				13
53	action30-1	luisa	know (nc)	how to send student form and check	o30	student form, rcheck	14
54	action31	alice	inform (c)				13
55	action31-1	luisa	know (nc)	where to store posi budget	o31	folder history editions for budgets	14
56	action32	alice	schedule (nc)	posi meeting	o32	outlook, alice's boss, prof. jm	5

Figure 4.21: Examples introduced to the ontology (Part 2).

4.1.8 Rule addition and Query Testing Outcome

On this section we divide the outcomes in rules and queries.

Rule addition outcome

The following corresponds to the set of rules developed on this stage of the methodology. Also we show the effect that the addition of those has on the ontology. The information is presented as Protégé screen shots containing the previous state of the ontology (before the introduction on the rule) and the state after the rule is introduced with the reasoner activated. This results are just a part of the entire effects that the has rule on the ontology and corresponds only to selected examples chosen for illustrative purposes.

In figures showing the state of the ontology after the addition of the rule, (representing Protégé screen shots), elements surrounded by dotted line corresponds to reasoner inferences.

a) `"communicativeAction(?a), sentBy(?a, ?s) -> perform(?s, ?a)"`

This rule set an action as performed by a specific agent (the action must be a communicative action and sent by this agent).

Figure 4.22 show the state of the ontology before the addition of rule a). As we can see (on the left part of the figure) there is no information about action *action01* being performed by any agent or (on the right part of the figure) about agent *Prof. Smith* performing any action.

Figure 4.23 show the state of the ontology after the addition of rule a). Here (on the left part of the figure), inferred information shows that action *action01* is

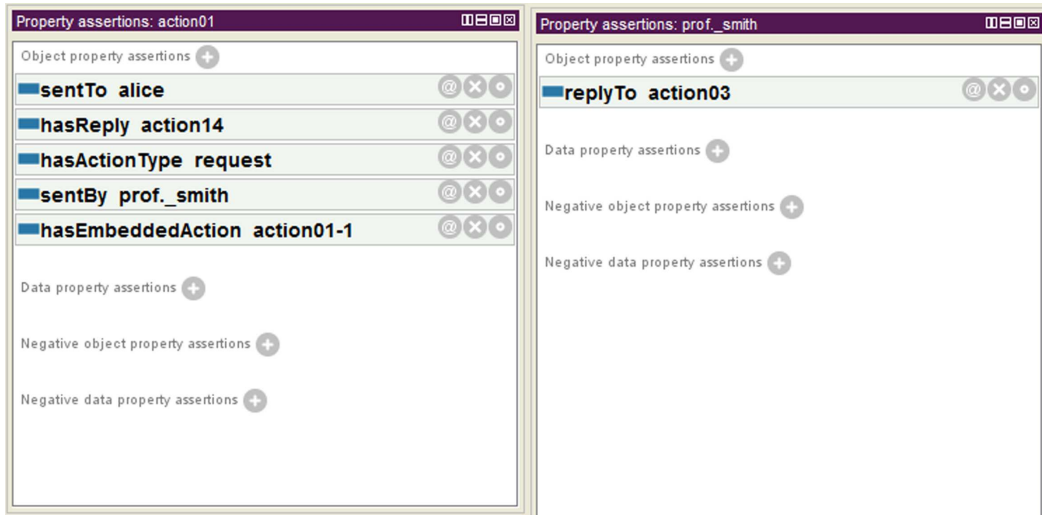


Figure 4.22: Ontology before introduction of rule a).

performed by (*performedBy*) agent *Prof. Smith*. Additionally (on the right of the figure) we can see that agent *Prof. Smith* performs action *action01*.

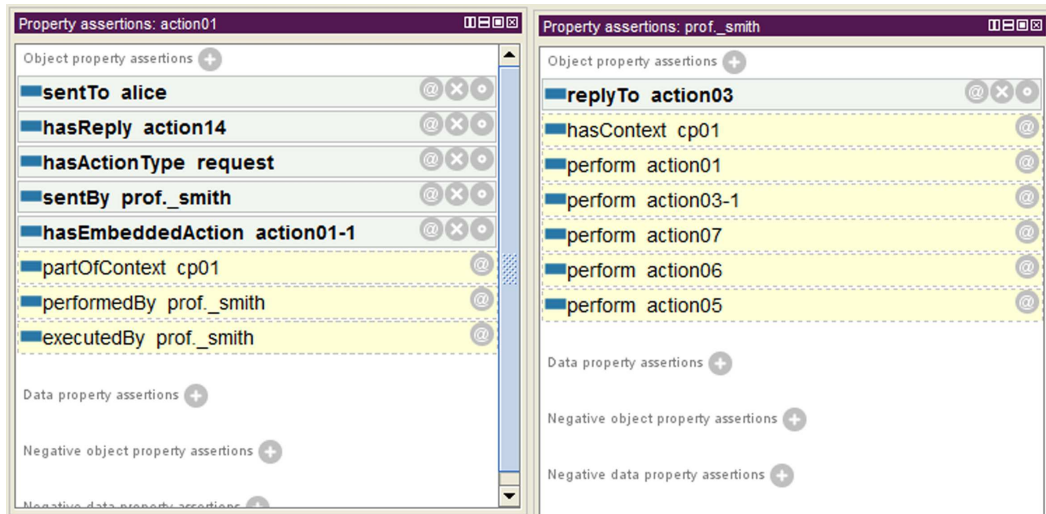


Figure 4.23: Ontology after introduction of rule a).

b) `"hasEmbeddedAction(?a, ?ea), replyTo(?r, ?a) -> completedCommitment(?ea),`

```
perform(?r, ?ea)"
```

This rule set an action as performed by a specific agent (the action must be embedded in an action who the agent reply). Also set this action (embedded action) as completed commitment.

Figure 4.24 show the state of the ontology before the addition of rule b)As we can see (on the left part of the figure), action (*action01*) has as type just *action*, nothing else. Also, on the right part of the figure we can see information about agent *Prof. Smith*, but no information about him performing actions is present.

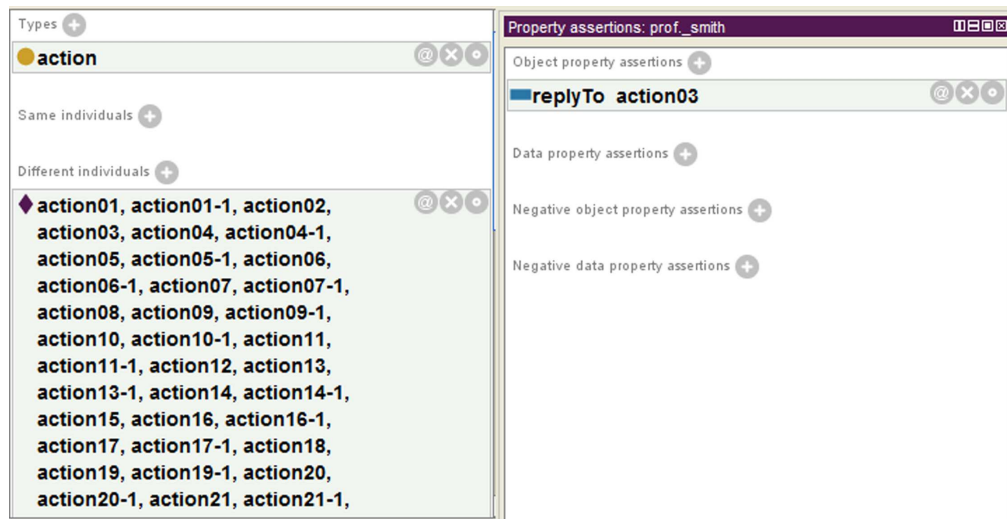


Figure 4.24: Ontology before introduction of rule b).

Figure 4.25 show the state of the ontology after the addition of rule b). On the left part of the figure we can see the inferred information showing that action *action01* has as type *completedCommitment* which is one of the aims of rule b). Additionally (on the right of the figure), inferred information shows us that agent *Prof. Smith* performs action *action01*.

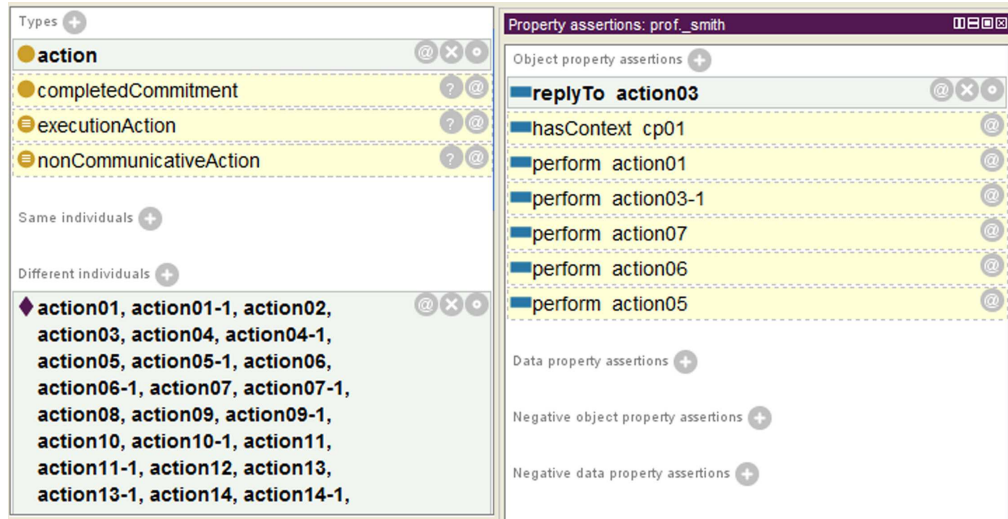


Figure 4.25: Ontology after introduction of rule b).

c) `"nonCommunicativeAction(?nca), hasActionFeature(?c, ?at),
 hasActionType(?nca, ?at), hasAgent(?c, ?ag), hasObject(?nca, ?o),
 hasResourceFeature(?c, ?re), involveElement(?o, ?re), perform(?ag, ?nca)
 -> hasAction(?c, ?nca)"`

This rule associates a non-communicative action to a context. Context and action are associated via action types, agents and resources. This is one of the rules to accomplish objective 8.(a) from Chapter 1.

Figure 4.26 show the state of the ontology before the addition of rule c). Here, context *cp02* has no actions associated.

Figure 4.27 show the state of the ontology after the addition of rule c). Here, on the left side we can see action *action02* associated (*partOfContext*) with context *cp02*. On the middle and right part of the figure we can see part of the association made

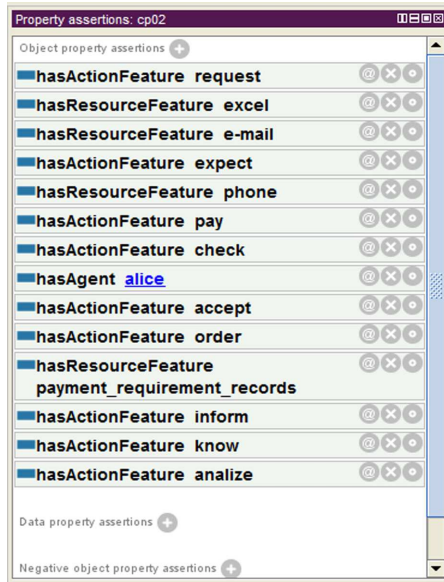


Figure 4.26: Ontology before introduction of rule c).

by the rule and the reasoner, here the object (*o02*) of action *action02* involve elements (resources) that belongs to context *cp02*, so, according to rule c), the association is made.

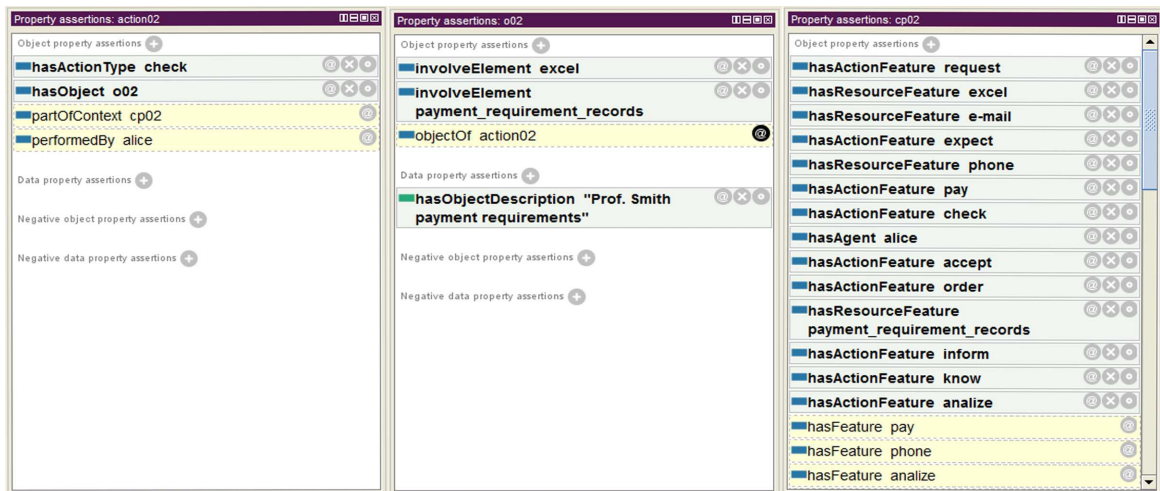


Figure 4.27: Ontology after introduction of rule c).

```
d) "nonCommunicativeAction(?nca), hasActionFeature(?c, ?at),
hasActionType(?nca, ?at), hasAgent(?c, ?ag), hasObject(?nca, ?o),
hasResourceFeature(?c, ?re), involveElement(?o, ?re), sentTo(?ca, ?ag),
isEmbeddedIn(?nca, ?ca) -> hasAction(?c, ?nca)"
```

This rule associates a non-communicative action to a context. Context and action are associated via action types, agents, embedded actions and resources. This is one of the rules to accomplish objective 8.(a) from Chapter 1.

Figure 4.28 show the state of the ontology before the addition of rule d). Here, context *cp01* has no actions associated.



Figure 4.28: Ontology before introduction of rule d).

Figure 4.29 show the state of the ontology after the addition of rule d). Here, on the left side and middle of the figure we can see some of the element of the non-communicative action *action03-1* that allows rule d) to associate it with a given context (like resources, type of actions and agent). On the right of the figure we can see that context *cp01* has associated (*hasAction*) the non-communicative

action *action03-1*.

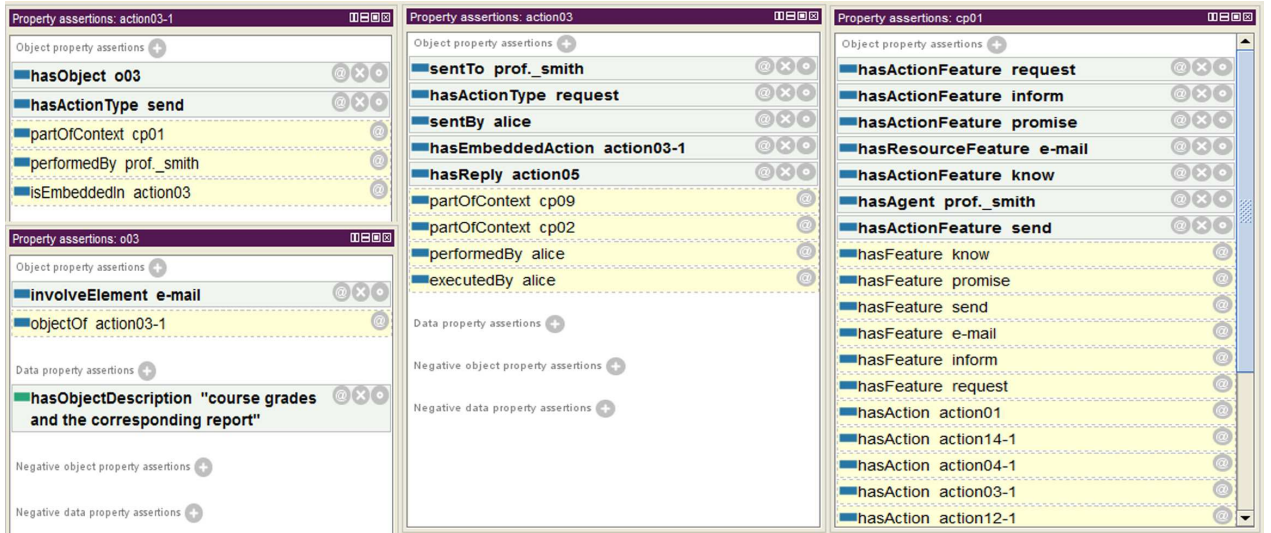


Figure 4.29: Ontology after introduction of rule d).

e) "hasActionFeature(?c, ?at), hasActionType(?a, ?at), hasAgent(?c, ?ag), hasEmbeddedAction(?a, ?a1), sentBy(?a, ?ag) -> hasAction(?c, ?a)"

This rule associates a communicative action to a context. Here, context and action are associated via action types, agents and embedded actions. This is one of the rules to accomplish objective 8.(a) from Chapter 1.

Figure 4.30 show the state of the ontology before the addition of rule e). Here, context *cp09* has no actions associated.

Figure 4.31 show the state of the ontology after the addition of rule e). Here, on the left right side of the figure we can see the communicative action *action13* and context *cp09* bounded by the relation *partOfContext* and *hasAction*.

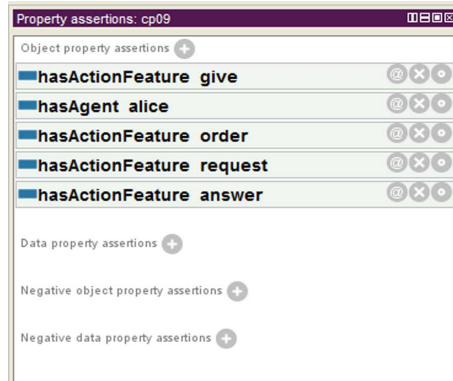


Figure 4.30: Ontology before introduction of rule e).

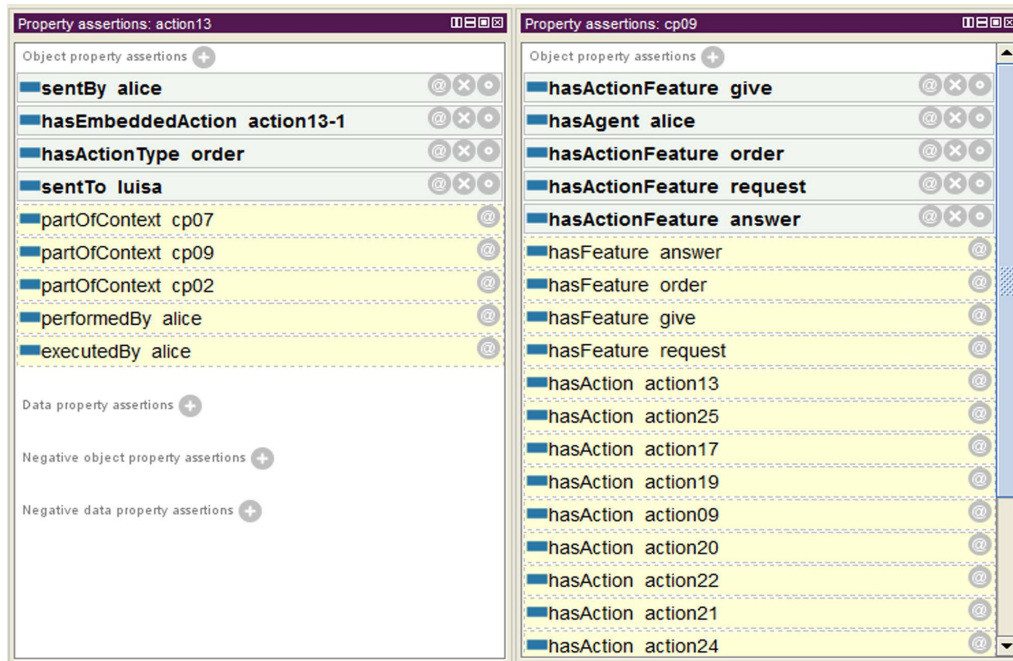


Figure 4.31: Ontology after introduction of rule e).

```
f) "communicativeAction(?ac), hasActionFeature(?c, ?at1),
hasActionFeature(?c, ?at2), hasActionType(?ac, ?at1),
hasActionType(?ea, ?at2), hasAgent(?c, ?ag), hasEmbeddedAction(?ac, ?ea),
hasObject(?ea, ?o), hasResourceFeature(?c, ?re), involveElement(?o, ?re),
perform(?ag, ?ac) -> hasAction(?c, ?ac)"
```

This rule associates a communicative action to a context. Here, context and action are associated via action types, agents, embedded actions and resources. This is one of the rules to accomplish objective 8.(a) from Chapter 1.

Figure 4.32 show the state of the ontology before the addition of rule f). Here, context *cp03* has no actions associated.

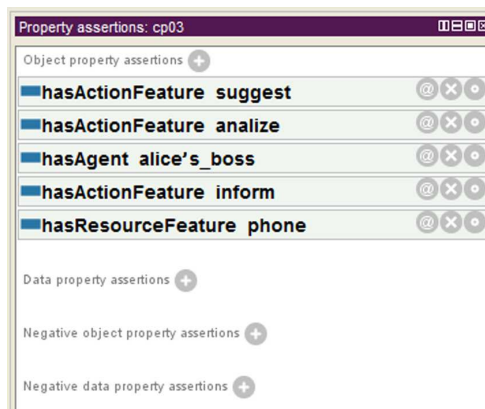


Figure 4.32: Ontology before introduction of rule f).

Figure 4.33 show the state of the ontology after the addition of rule f). Here, on the 3 parts of the figure we can see how the rule bound and make the association between context *cp03* and the communicative action *action10*. Resources, the agent and action types are used by the rule to associate action and context and the result is shown on the right side of the figure which indicates that context *cp03*

has as action (*hasAction*) action *action10*.

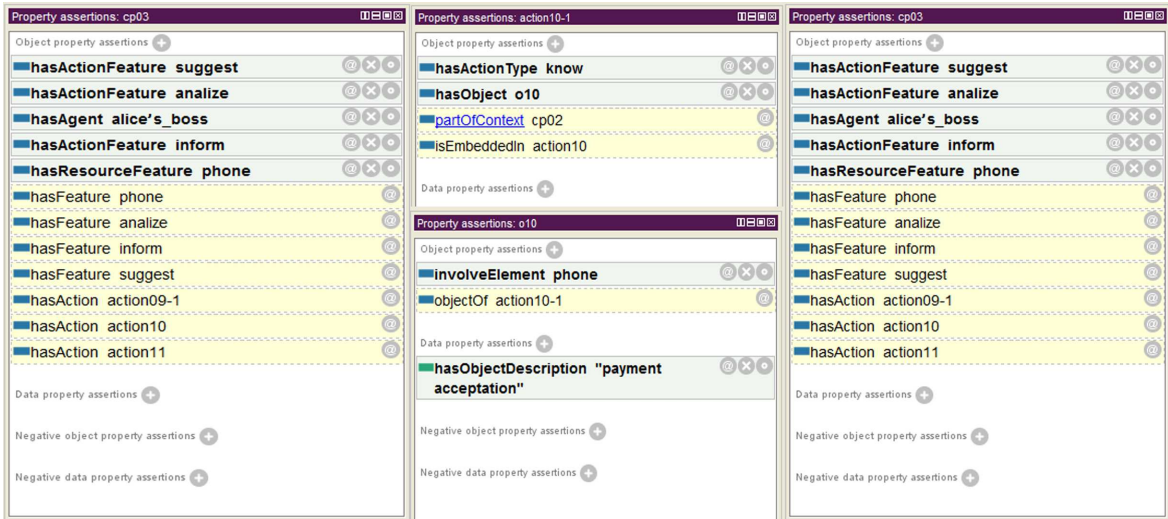


Figure 4.33: Ontology after introduction of rule f).

Query testing outcome

The following corresponds to the set of queries constructed on this stage of the methodology. Also we show the results obtained from the ontology when these queries are applied on it. Results are presented as Protégé screen shots showing the specific query and the returned instances which corresponds to the answer to the query.

1. "commitment and (isEmbeddedIn some (communicativeAction and sentTo value AGENT))"

This query returns all commitments of a given agent (on the query, replace AGENT for the name of the given agent). This query accomplish objective 8.(b) from Chapter 1.

Figure 4.34 show the result of query 1 replacing *AGENT* by *alice*. The result corresponds to a set of actions which corresponds to Alice commitments.

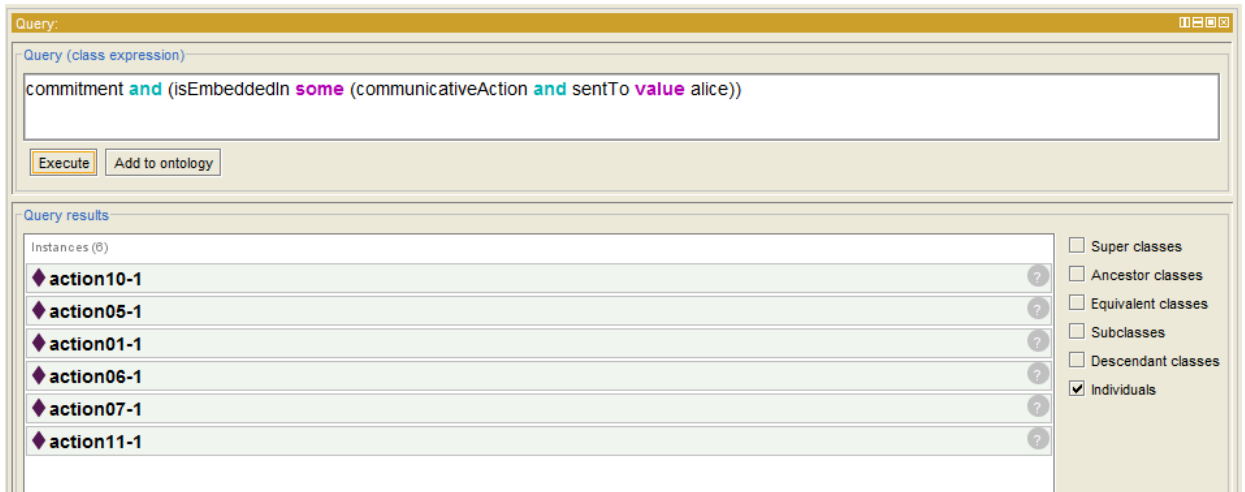


Figure 4.34: Ontology results of query 1.

2. "completedCommitment and (isEmbeddedIn some (communicativeAction and sentTo value AGENT))"

This query returns only completed commitments of a given agent (on the query, replace AGENT for the name of the given agent). This query accomplish objective 8.(c) from Chapter 1.

Figure 4.35 show the result of query 2 replacing *AGENT* by *alice*. The result corresponds to the set of actions being the completed commitments of Alice. As we can see this is a sub set of Alice commitments.

3. "executionAction and (isEmbeddedIn some (communicativeAction and sentTo value AGENT)) or (performedBy value AGENT)"

This query returns the execution context of a given agent (on the query, replace AGENT for the name of the given agent). This query accomplish objective

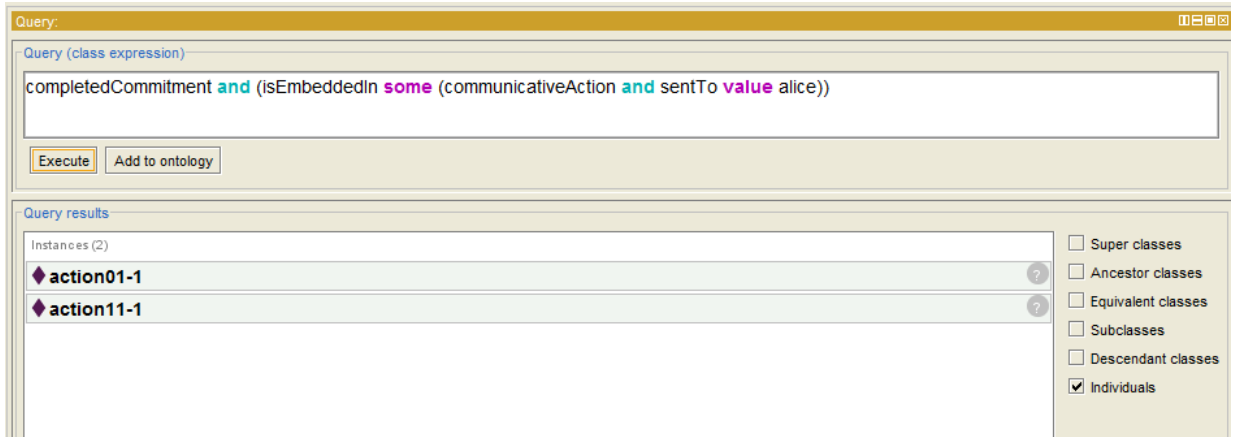


Figure 4.35: Ontology results of query 2.

8.(d) from Chapter 1.

Figure 4.36 show the result of query 3 replacing *AGENT* by *alice*. The result corresponds to the set of execution actions (i.e. Execution context) of the agent Alice.

4. "communicativeAction and (hasActionType some communicativeActionType) and (sentBy value AGENT)"

This query returns the coordination context of a given agent (on the query, replace AGENT for the name of the given agent). This query accomplish objective 8.(e) from Chapter 1.

Figure 4.37 show the result of query 4 replacing *AGENT* by *alice*. The result corresponds to the set of coordination actions (i.e. Coordination context) of the agent Alice.

5. "communicativeAction and hasReply some communicativeAction"

This query returns all the actions that interact with another actions. Alternatively, we can replace "*some communicativeAction*" by "*value ACTION*" to

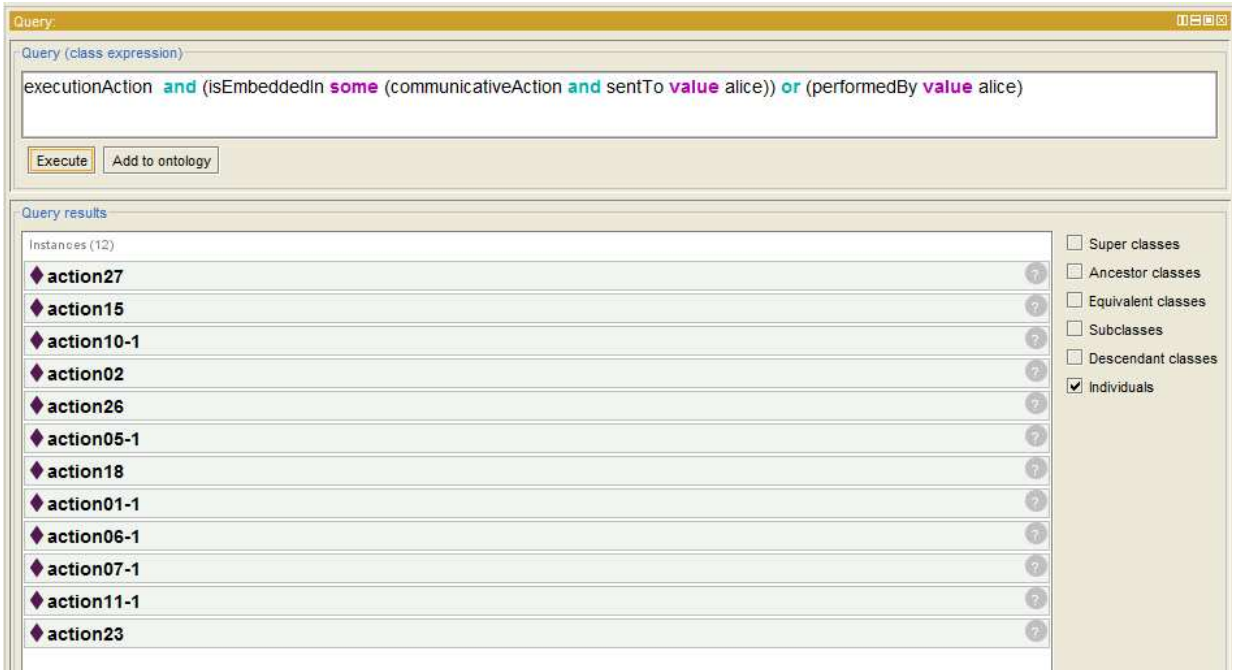


Figure 4.36: Ontology results of query 3.

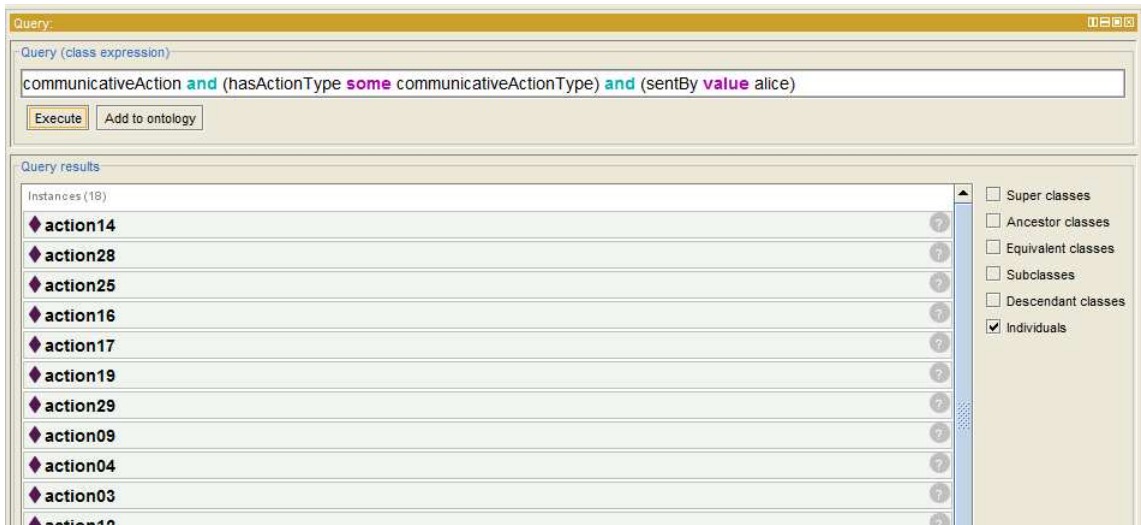


Figure 4.37: Ontology results of query 4.

have a specific query which returns communicative actions that interact with action *ACTION*.

Figure 4.38 show the result of query 5. The result corresponds to the set of communicative actions that interacts with other communicative actions. This is a general query and can be written more specific asking by the action that we want to know, i.e. which set of actions interact with action *action01*, for example.

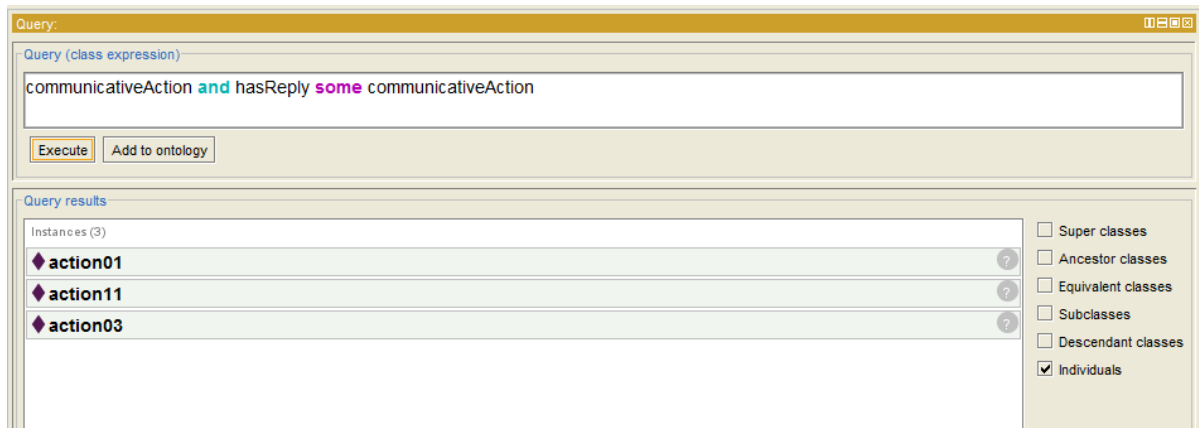


Figure 4.38: Ontology results of query 5.

6. "context and hasAction some (action and isEmbeddedIn some (communicativeAction and partOfContext value CONTEXT))"

This query returns the inter-personal contexts given a specific personal context. This query accomplish objective 8.(f) from Chapter 1.

Figure 4.39 show the result of query 6 replacing *CONTEXT* by *cp05*. The result corresponds to the set of context that has actions that are embedded in some communicative action which is part of context *cp05* (inter-personal context of

context *cp05*).

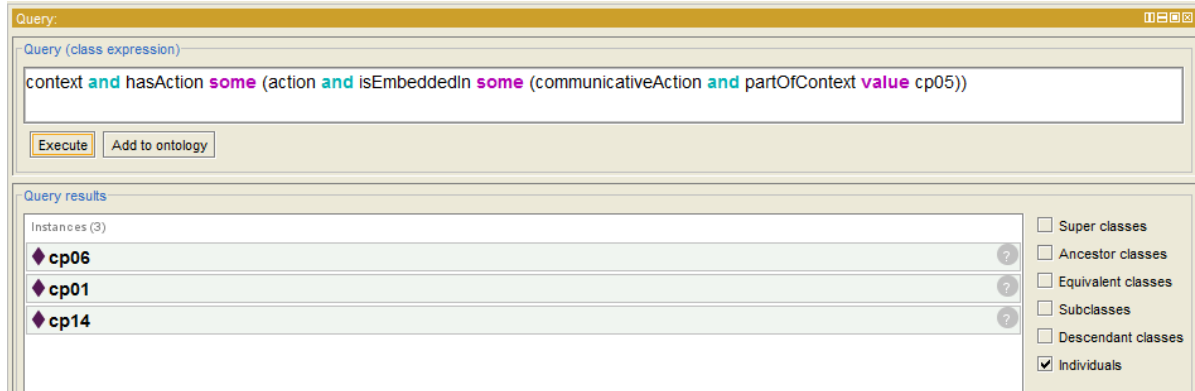


Figure 4.39: Ontology results of query 6.

7. "agent and hasContext some (context and hasAction some (action and isEmbeddedIn some (communicativeAction and partOfContext some (context and hasAgent value AGENT))))"

This query returns the inter-personal interactions given a specific agent. This query accomplish objective 8.(g) from Chapter 1.

Figure 4.40 show the result of query 7 replacing *AGENT* by *prof. smith*. The result corresponds to the set of agents (in this case just one) which has some context which in turn has some action which is embedded in some communicative action of some context which has as agent *prof. smith* (inter-personal interactions of *prof. smith*).

The aim of the previous two sections was just to show the behavior of the ontology when we added rules, the power that rules have, and how we can used them to obtain information that is already on the ontology but not in a explicit way. Also we wanted to show the power that queries have, the way of use them and how the

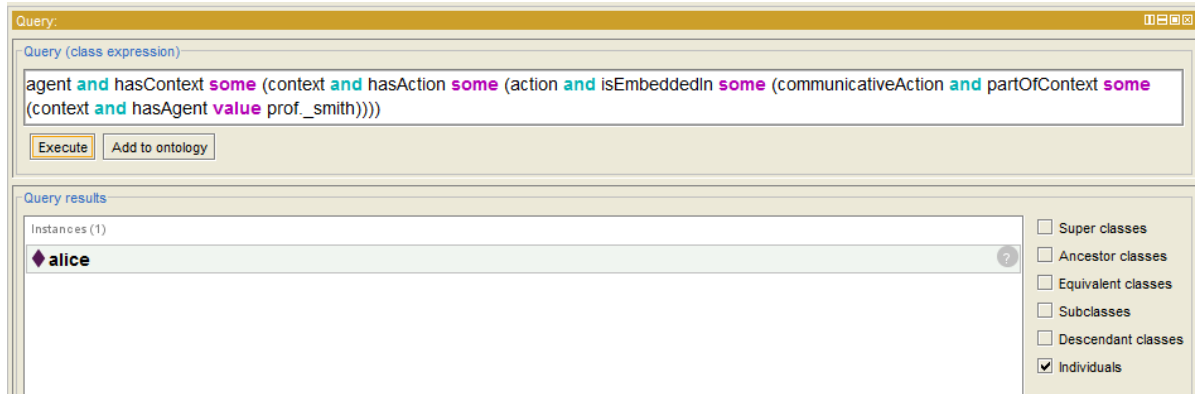


Figure 4.40: Ontology results of query 7.

ontology answer to them.

The results presented on the previous sections are just a part of the entire set of result that we obtained from the ontology via rules and queries, using the reasoner. The next section shows the evaluation to the complete set of results.

4.2 Results Evaluation

In this section we evaluate the work done by making a comparison between the results obtained from rules and queries and the results that should be obtained according with the examples in [29].

Action-Context Association Evaluation

In this part the aim was to associate a given action to its respective context. The results obtained are shown in Figure 4.41. Here, column “correctly associated” corresponds to the set of actions that were correctly associate to the given context, column “missing actions” corresponds to the actions that belongs to the context but were not associated by the rule and column “wrongly classified” corresponds to the action that were classified by the rule but don’t belong to the context.

Context	Correctly classified	Missing actions	Wrongly Classified
cp01	action01, action03, action04 action05, action06, action07 action12-1, action14-1		
cp02	action01-1, action02, action03, action04, action06-1, action07-1 action09, action10-1, action11-1, action12, action13, action14	action05-1, action08-1	action16, action17, action19, action20, action21, action24, action25, action28, action29, action30, action31
cp03	action09-1, action10, action11		
cp04		action13-1	
cp05	action15, action16, action32		action04, action14, action28, action29, action30, action31
cp06	action16-1		
cp07	action17, action18		action13, action25
cp08	action17-1		
cp09	action19, action20, action21, action22, action24, action25	action23	action03, action09, action13, action17
cp10	action20-1, action21-1		action19-1, action25-1
cp11		action22-1	
cp12	action26, action27		
cp13	action28, action29, action30, action31		action04, action14, action16
cp14	action28-1, action29-1, action30-1, action31		

Figure 4.41: Actions associated to context.

To evaluate this aspect we calculated the percentage of association success. To work in a consistent way, we used the same formula used in Chapter 5 of [29], specifically in the part of quantitative comparison between clusters and context (Figure 128).

Figure 4.42 shows, in the diagonal, the number of actions correctly classified and in the rest of the row the number of action incorrectly classified that belongs to other context. Column “Success” show the success of the classification calculated from the correctly classified among the total. We can see that on 50% of the classification (7 out of 14 contexts) we achieved a 100% of accuracy. The average success achieved on this point was 68%.

Context	cp01	cp02	cp03	cp04	cp05	cp06	cp07	cp08	cp09	cp10	cp11	cp12	cp13	cp14	Total	Success
cp01	8														8	1,00
cp02		12			1		1		5				4		23	0,52
cp03			3												3	1,00
cp04				0											0	0,00
cp05		2			3								4		9	0,33
cp06						1									1	1,00
cp07		1					2		1						4	0,50
cp08								1							1	1,00
cp09		3					1		6						10	0,60
cp10										4					4	1,00
cp11											0				0	0,00
cp12												2			2	1,00
cp13		2			1								4		7	0,57
cp14														4	4	1,00
	8	20	3	0	5	1	4	1	12	4	0	2	12	4	76	0,68

Figure 4.42: Success on actions associated to context.

Execution Context Evaluation

In this part we evaluate the results achieved obtaining the execution context of a given agent. Figure 4.43 shows the classified, missing and wrongly classified execution actions (which corresponds to the execution context) per agent. As we can see on the figure, all the actions were correctly classified (29 out of 29). In the case of agent “alice” just one action was missing on the classification (*action08*), this was the only case of missing actions. Finally, we had a 97% of correctness (counting the missing action) but a 100% of correctness on the classification (a classification without errors).

Coordination Context Evaluation

In this part we evaluate the results achieved obtaining the coordination context of a given agent. Figure 4.44 shows the classified, missing and wrongly classified coordination actions (which corresponds to the coordination context) per agent. As we can see on the figure, all the actions were correctly classified (24 out of 24), a clean classification without errors or missing actions. We obtained a 100% of correctness.

Agent	Classified actions	Missing actions	Wrongly classified
Alice	action01-1, action02, action05-1, action06-1, action07-1, action10-1, action11-1, action15, action18, action23, action26, action27	action08	
Prof. Smith	action03-1, action04-1, action12-1, action14-1		
Alice's Boss	action09-1		
Luisa	action13-1, action17-1, action19-1, action20-1, action21-1, action25-1, action28-1, action29-1, action30-1, action31-1		
Miguel	action16-1		
Aida	action22-1		
Total	29	1	0
		Correct among total	0,97
		Correct among classified	1,00

Figure 4.43: Execution context evaluation.

Agent	Classified actions	Missing actions	Wrongly classified
Alice	action03, action04, action09, action12, action13, action14, action16, action17, action19, action20, action21, action22, action24, action25, action28, action29, action30, action31		
Prof. Smith	action01, action05, action06, action07		
Alice's Boss	action10, action11		
Luisa			
Miguel			
Aida			
Total	24	0	0
		Correct among total	1,00
		Correct among classified	1,00

Figure 4.44: Coordination context evaluation.

Agent commitments evaluation

Here, the results of obtaining all the commitment are presented. Figure 4.45 presents the classified, missing and wrongly classified commitments (represented by actions) of each agent in the ontology. As we can see in the figure all the commitments (actions) were correctly classified (23 out of 23), a clean classification without errors or missing actions. We obtained a 100% of correctness on this point.

Agent	Classified actions	Missing actions	Wrongly classified
Alice	action01-1, action05-1, action6-1, action07-1, action10-1, action11-1		
Prof. Smith	action03-1, action04-1, action12-1, action14-1		
Alice's Boss	action09-1		
Luisa	action13-1, action17-1, action19-1, action20-1, action21-1, action25-1, action28-1, action29-1, action30-1, action31-1		
Miguel	action16-1		
Aida	action22-1		
Total	23	0	0
		Correct among total	1,00
		Correct among classified	1,00

Figure 4.45: Agent commitments evaluation.

Agent completed commitments evaluation

Here, the results of obtaining the completed commitment of agents are presented. Figure 4.46 presents the classified, missing and wrongly classified completed commitments (represented by actions) of each agent in the ontology. As we can see in the figure all the completed commitments (replied actions) were correctly classified (3 out of 3), a clean classification without errors or missing actions. We obtained a 100% of correctness on this point.

Agent	Classified actions	Missing actions	Wrongly classified
Alice	action01-1, action11-1		
Prof. Smith	action03-1		
Alice's Boss			
Luisa			
Miguel			
Aida			
		Correct among total	1,00
		Correct among classified	1,00

Figure 4.46: Agent completed commitments evaluation.

Inter-personal Context Evaluation

This part shows the evaluation of the results obtaining inter-personal contexts. This was the aspect in which worst result were obtained. Figure 4.47 shows the classified contexts, from them, in the following columns we can see the ones that were correctly classified, wrongly classified and the contexts that are missing on the classification. We calculated a 25% of correctness in the classification considering corrects, wrongs and missing. A second percentage was calculated considering only the classified contexts (corrects and wrongs), in this case we obtained a 60% of correctness.

Inter-personal Interactions Evaluation

This part shows the evaluation of the results obtaining inter-personal interactions. Figure 4.48 shows the set of agents who share inter-personal interactions with a given agent. Here, again we obtained a faithful correspondence of what is on the ontology. As we can see on the figure we obtained a 100% of correctness on the results.

Evaluation Analysis

Figure 4.49 shows a summary of the results obtained from the ontology using queries and rules. Here, we obtained a factor of success of 0.88 (average success of results obtained from queries a, b, c, d, e, f and g).

Context	Classified context	Correctly classified	Missing contexts	Wrongly classified
cp1	cp09, cp02	cp02		cp09
cp2	cp01, cp03	cp01, cp03	cp04	
cp3			cp02	
cp4				
cp5	cp01			cp01
cp6			cp05	
cp7				
cp8			co07	
cp9				
cp10			cp09	
cp11			cp09	
cp12				
cp13	cp01			cp01
cp14			cp13	
	Total	3	7	3
			Correct among total	0,23
			Correct among classified	0,5

Figure 4.47: Inter-personal context evaluation.

Agent	Classified agents	Missing Agents	Wrongly classified	
Alice	Luisa, Miguel, Prof. Smith, Alice's Boss			
Prof. Smith	Alice			
Alice's Boss	Alice			
Luisa				
Miguel				
Aida				
Total	6	0	0	
			Correct among total	1,00
			Correct among classified	1,00

Figure 4.48: Inter-personal interactions evaluation.

Objective	Description	Classifieds	Correct	Wrong	Success
a	Associate contexts with actions	76	50	26	0,68
b	Commitments of a given agent	23	23	0	1
c	Completed commitments of a given agent	3	3	0	1
d	Execution context of a given agent	29	29	0	1
e	Coordination context of a given agent	24	24	0	1
f	Inter-personal contexts	6	3	3	0,5
g	Inter-personal interactions	6	6	0	1

Figure 4.49: Evaluation Summary.

In five out of the seven cases we obtained a success factor of 1 which indicate us that the objectives were achieved. In the case of “Associate context with actions” (letter a) and “Inter-personal interactions” (letter f) we obtained the worst results of all, 0.68 and 0.5, respectively.

The previous two situations are due (in part) to two facts: (1) we are not working with time (i.e. actions does not have time for its execution) and (2) we have similar actions (i.e. performed by the same agent, with the same action type and belonging to contexts that have the same resources). The previous factors causes that rules associating actions to contexts tends to classify similar actions that belongs to different context as they are part of the same one.

Consequently, queries using the “hasAction” relation (the one that relates actions to contexts) are more likely to have errors in their results. This is the case of the letters a and f of Figure 4.49.

Figure 4.50 shows similar actions. We can see on the figure, for example, that action “*action28*” belongs to context “*cp14*” but can be classified as belonging to

contexts “*action02*” and “*action05*” because is performed by the same agent and has the same action type.

Agent	Action Type	Action	Context
Alice	request	action03,action 09	cp02
		action19, action20, action21,action 24	cp09
	inform	action14,action4	cp02
		action28,action29,action31,action30	cp14
		action16	cp05
	order	action13	cp02
		action25	cp09
action17		cp07	
Luisa	pay	action13-1	cp04
		action17-1	cp08

Figure 4.50: Similar actions.

The problems caused by the previous situation can be improved incorporating *time* to our ontology. Adding time execution to a given action will generate some level of distinctions between similar actions so the classification of actions into context would be more accurate.

Moreover, the reason for having missing classifications is due to the fact that we used rules and queries to obtain results. Different sets of rules and queries can be developed to obtain results or answers and the combination of those rules and queries sometimes can give us different results. Sometimes, it is not possible obtain all the results (all the entities) when we use a query, this is because queries (and rules) are composed by a set of relationships that relates terms on the ontology, but sometimes it is not possible to include all the relations that relate terms on this elements (for example, to associate context with actions we used 4 rules, with less than that we would not be able to have an acceptable amount of associations). On this work,

different set of queries and rules were developed and tested and the ones with better results were selected, evaluated and sowed on this document.

This Chapter shows the evaluation of the results obtained on the present work. This evaluation is presented in order to show how accomplished were the objectives stated in Chapter 1. Conclusion and future work from this work is presented on the next chapter.

Chapter 5

Conclusion and Future Work

On this work we have done a research concerning ontologies, specifically enterprise ontologies. Under this area we have centred our attention in agent-context based ontologies, specifically on the one presented in [29]. To develop this work we have studied three prestigious enterprise ontologies: EO, TOVE and Dietz ontology. Doing this, we were able to analyse key concepts, essential terms and common relations of enterprise ontologies which helped us to understand better how this type of organizations can be modelled.

Also, the work done required a formalization methodology suitable to achieve the proposed objectives. For this we analysed ontology development methodologies and checked its important stages. This methodologies gave us a general idea of what its important to consider when developing a ontology, which elements need special considerations and possible difficulties on the development process.

As a final outcome of this work we have the formalized ontology, written in OWL language and containing rules that, along with queries, allowed us to answer the questions proposed in Chapter 1. This ontology is also extendible, being possible add more entities to it or merge it with another related ontologies to cover a broader scope

of the enterprise. It is also possible to continue adding rules and queries to retrieve more information, depending on our objectives.

After concluding this work we have realized how practical it is the utilization of ontologies to model enterprises. Specifically agent-context based ontology, which covers formal and informal element of the organization. Covering these aspect of enterprises is not a trivial task, relations among individuals creates new communication channels inside the organization which are not specified by formal definitions, so it is important to have mechanisms that help us discover this relations.

Also, ontologies give us the possibility of infer new information from the one that we already have. This is a key point in using ontologies. The value of information nowadays is very high, so, having mechanisms to retrieve more, give the enterprise the possibility to adapt to changes, to restructure itself, to be more flexible. This is translated in a more competitive enterprise.

Finally, ontologies are a suitable way to model an enterprise, because they provide the flexibility needed to represent something that is always changing as the enterprise. Ontologies are highly scalable, which allow us to integrate them with another ontologies or add elements to it relatively easy. The dynamic nature of enterprises requires dynamic models to represent them and ontologies are up to this challenge.

5.1 Future Work

As future work we intermediately visualize the possibility of continuing exploiting rule and query capabilities. The aim of these really depends on the developer objectives but as an example we can create rules or queries to obtain information about resource

utilization (which are the shared resources, which resources are used by a given agent or in a given context).

Protégé also allow us to use SPARQL query language, so a future work could use this type of queries, which has different capabilities than DL queries, to obtain richer information from the ontology.

The final version of this ontology can be used along with some enterprise modelling tool. This corresponds to a practical utilization of the formalized ontology. This kind of integration can take the most of both sides helping us to achieve all the benefits of this ontology that we have described along this document.

Finally, taking advantage of the scalability of ontologies, it is also possible to integrate the ontology developed in this work to cover a wider range of the enterprise. So we can incorporate Time, Resource and Activity ontologies which will provide us a complete and new base to start working on a wider enterprise modelling.

Bibliography

- [1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services - concepts, architectures and applications*. Springer Verlag, 2004.
- [2] R. Axelrod and M. Cohen. *Harnessing complexity: organizational implications of a scientific frontier*. 2000.
- [3] E. Börger. Abstract state machines: A method for high-level system design and analysis, 2003.
- [4] J. Dietz. *Enterprise Ontology: Theory and Methodology*. Springer, May 2006.
- [5] M. Fernandez-Lopez, A. Gomez-Perez, and N. Juristo. Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium*, pages 33–40, Stanford, USA, March 1997.
- [6] Florida Institute for Human and Machine Cognition. Coe overview. <http://www.ihmc.us/groups/coe/>. [Online; accessed June-2011].
- [7] M. Fox, J. Chionglo, and F. Fadel. A common-sense model of the enterprise. In *Proceedings of the Industrial Engineering Research Conference*, pages 425–429, 1993.
- [8] M. Fox and M. Gruninger. Enterprise modeling. *AI Magazine*, 19(3):109–121, 1998.

- [9] A. Gómez-Pérez, N. Juristo, and J. Pazos. Evaluation and assessment of the knowledge sharing technology. In *Towards very large knowledge bases*, pages 289–296, Holanda, 1995. IOS Press.
- [10] T. Gruber. A translation approach to portable ontology specifications. Technical Report KSL-92-71, Knowledge Systems, AI Laboratory, 1993.
- [11] M. Grüninger and M. Fox. Methodology for the design and evaluation of ontologies. In *International Joint Conference on Artificial Inteligence (IJCAI95), Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.
- [12] M. Koivunen and E. Miller. W3c semantic web activity. Technical report, 2001.
- [13] F. López. Overview of methodologies for building ontologies. In *Proceedings of the IJCAI99 Workshop on Ontologies and Problem Solving Methods Lessons Learned and Future Trends*, 1999.
- [14] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino. Social roles and their descriptions. pages 267–277. AAAI Press, 2004.
- [15] D. McGuinness. Ontologies come of age. In H. L. W. W. Dieter Fensel, James A. Hendler, editor, *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, pages 171–196. MIT Press, 2003.
- [16] A. Mili, R. Mili, and R. Mittermeir. A survey of software reuse libraries. *Annals of Software Engineering*, 5:349–414, 1998.
- [17] NeOn Foundation. Neon toolkit overview. http://neon-toolkit.org/wiki/Main_Page, 2011. [Online; accessed May-2011].
- [18] N. Noy and D. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Knowledge Systems, AI Laboratory, Stanford University, 2001.

- [19] D. Oberle. Abstract semantic management of middleware. Technical report, Institute of Applied Informatics and Formal Description Methods (AIFB), University of Karlsruhe, 2006.
- [20] Semantic Web W3C. W3c semantic web frequently asked questions. <http://www.w3.org/RDF/FAQ/>, 2009. [Online; accessed April-2011].
- [21] Semantic Web W3C. W3c semantic web overview. http://www.w3.org/standards/semanticweb/#w3c_overview, 2010. [Online; accessed April-2011].
- [22] Stanford Center for Biomedical Informatics Research. Protégé overview. <http://protege.stanford.edu/overview/>, 2011. [Online; accessed June-2011].
- [23] Stefan Seedorf and Fzi Forschungszentrum Informatik and Universitt Mannheim. Applications of ontologies in software engineering. In *In 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006), held at the 5th International Semantic Web Conference (ISWC 2006)*, 2006.
- [24] TopQuadrant Inc. Topbraid composer overview. http://www.topquadrant.com/products/TB_Composer.html, 2001 - 2011. [Online; accessed June-2011].
- [25] M. Uschold. Building ontologies: Towards a unified methodology. In *In 16th Annual Conf. of the British Computer Society Specialist Group on Expert Systems*, 1996.
- [26] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–136, 1996.
- [27] M. Uschold and M. King. Towards a methodology for building ontologies. In *Proc. of IJCAI95's WS on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, 1995.
- [28] M. Uschold, M. King, S. Moralee, and Y. Zorgios. The enterprise ontology. *The Knowledge Engineering Review*, 13:31–89, 1995.

- [29] M. Zacarias. *Conceptual Framework based on Agents and Contexts for the Alignment between Individuals and Organizations*. PhD thesis, Universidad Técnica de Lisboa, Instituto Superior Técnico, 2008.
- [30] M. Zacarias, S. Pinto, R. M. aes, and J. Tribolet. A 'context-aware' and agent-centric perspective for the alignment between individuals and organizations. *Inf. Syst.*, 35:441–466, June 2010.

Appendix A

OWL Ontology extract

This appendix contains an extract of the OWL ontology formalized on the present work. This extract shows some of the most important classes and relations. We did not include the entire OWL file because it has more than 200 pages but it is available on the digital support (CD) that comes with the present document and on the WEB following the URI

‘‘http://www.semanticweb.org/ontologies/2011/Agent-Context_Ontology.owl’’

```
<?xml version="1.0"?>

<!DOCTYPE Ontology [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.semanticweb.org/ontologies/2011/2/Ontology1300791413232.owl"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

```

```

xmlns:xml="http://www.w3.org/XML/1998/namespace"
ontologyIRI="http://www.semanticweb.org/ontologies/2011/2/Ontology1300791413232.owl">
<Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#" />
<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
<Prefix name="" IRI="http://www.w3.org/2002/07/owl#" />
<Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
<Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
<Declaration>
  <Class IRI="http://localhost/default#action" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#actionPattern" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#actionSequence" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#actionType" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#agencyLayer" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#agent" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#agentRule" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#agentState" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#commitment" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#commitmentType" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#communicativeAction" />
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#communicativeActionType" />
</Declaration>

```

```

<Declaration>
  <Class IRI="http://localhost/default#context"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#coordinationActionType"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#coordinationAgent"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#coordinationContext"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#coordinationLayer"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#designActionType"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#designAgent"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#designContext"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#designLayer"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#dyad"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#executionActionType"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#executionAgent"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#executionContext"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#executionLayer"/>
</Declaration>
<Declaration>

```

```

    <Class IRI="http://localhost/default#group"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#groupContext"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#groupLevel"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#individualLevel"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#interPersonalLevel"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#interaction"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#interpersonalContext"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#nonCommunicativeAction"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#nonCommunicativeActionType"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#object"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#organization"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#organizationContext"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#organizationLevel"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#organizationalLevel"/>
</Declaration>
<Declaration>
    <Class IRI="http://localhost/default#personalContext"/>

```

```
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#receiver"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#recurrentActionSequence"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#recurrentActionType"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#recurrentTemporalEntity"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#resourceElement"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#resourceGroup"/>
</Declaration>
<Declaration>
  <Class IRI="http://localhost/default#resourceType"/>
</Declaration>
.
.
.
</Ontology>
```

Appendix B

Informal conceptualization sketches

This appendix contains the final set of selected sketches drawn during this work. We show here the sketches that were more useful to construct the formal conceptual maps.

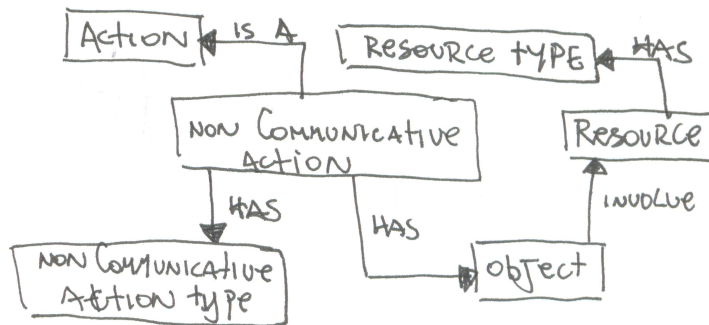


Figure B.1: Non communicative action sketch.

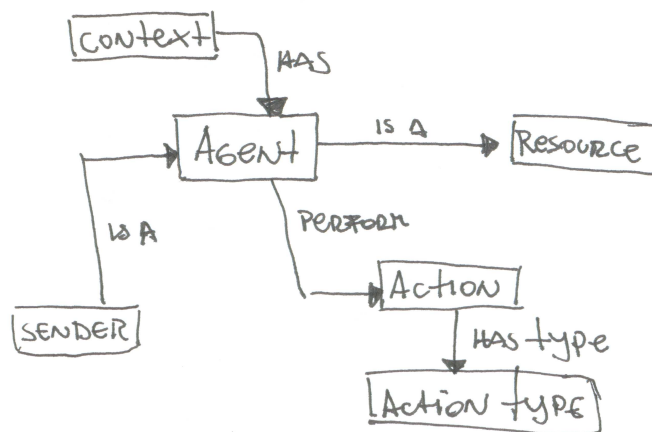


Figure B.2: Agent sketch.

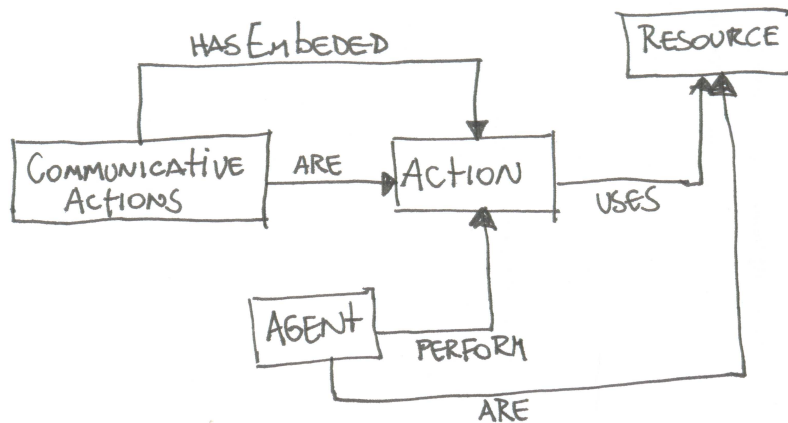


Figure B.3: Action sketch.

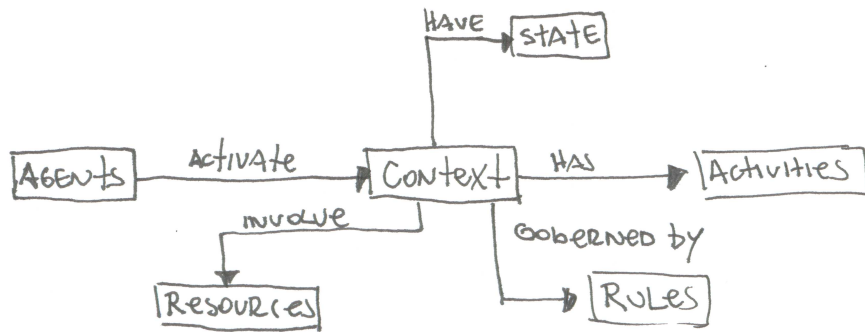


Figure B.4: Context sketch.

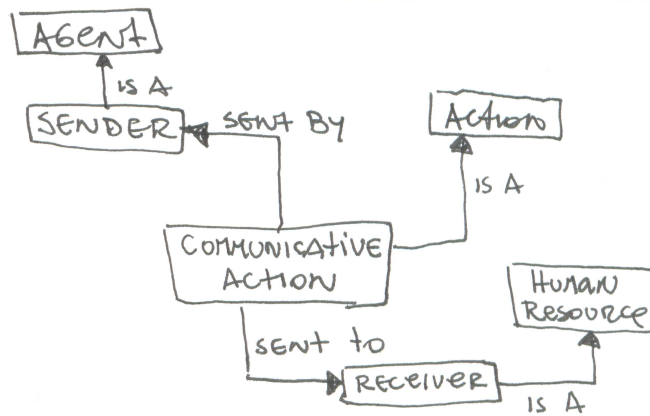


Figure B.5: Communicative action sketch.

Appendix C

Prof. Smith's payment context

The following corresponds to a part the examples introduced to the ontology. This is the “*Prof. Smith's payment*” context with the views of agents Alice, Prof. Smith, Alice's Boss and Luisa.

(a) **Prof. Smith request Alice the payment of a course recently finished.**

- *Individuals*: Prof. Smith.
- *Action type*: Request (communicative action).
- *Embedded action*: (a1) **Alice pay a course recently finished.**
 - *Individual*: Alice.
 - *Action type*: Pay (non communicative action).
 - *Object*: a course recently finished.

(b) **Alice check Prof. Smith payment requirements (course grades and report) and do not find them.**

- *Individuals*: Alice.
- *Action type*: Check (non communicative).
- *Object*: Payment requirements.
- *Embedded action*: None.

(c) **Alice request Prof. Smith to send the course grades and the corresponding report.**

- *Individuals*: Alice.
- *Action type*: Request (communicative action).

- *Embedded action:* (c1) **Prof. Smith to send the course grades and the corresponding report.**
 - *Individual:* Prof. Smith.
 - *Action type:* Send (non communicative action).
 - *Object:* Course grades and the corresponding report.
- (d) **Alice inform Prof. Smith that payment will proceed after these requirements are met.**
- *Individuals:* Alice.
 - *Action type:* Inform (communicative action).
 - *Embedded action:* (d1) **Prof. Smith know that payment will proceed after these requirements are met.**
 - *Individual:* Prof. Smith.
 - *Action type:* Know (non communicative action).
 - *Object:* Payment will proceed after these requirements are met.
- (e) **Prof. Smith inform Alice that due to a personal reason X, he will not be able to send the requirements until a date D.**
- *Individuals:* Prof. Smith.
 - *Action type:* Inform (communicative action).
 - *Embedded action:* (e1) **Alice know that due to a personal reason X, Prof. Smith will not be able to send requirements until date D.**
 - *Individual:* Alice.
 - *Action type:* Know (non communicative action).
 - *Object:* Due to a personal reason, Prof. Smith will not be able to send requirements until date D.
- (f) **Prof. Smith request the payment to be made without the requirements.**
- *Individuals:* Prof. Smith.
 - *Action type:* Request (communicative action).
 - *Embedded action:* (f1) **Alice to pay without the requirements.**
 - *Individual:* Alice.
 - *Action type:* Pay (non communicative action).
 - *Object:* Without the requirements.
- (g) **Prof. Smith promises Alice that he will send the requirements before date D.**
- *Individuals:* Prof. Smith.
 - *Action type:* Promises (communicative action).
 - *Embedded action:* (g1) **Alice expects that she will receive the requirements before date D.**

- *Individual*: Alice.
 - *Action type*: Expects (non communicative action).
 - *Object*: Will receive the requirements before date D.
- (h) **Alice analyses the reason X given by Prof. Smith.**
- *Individuals*: Alice.
 - *Action type*: Analyses (non communicative).
 - *Object*: Reason X given by Prof. Smith.
 - *Embedded action*: None.
- (i) **Alice asks her boss whether to accept Prof smith’s request.**
- *Individuals*: Alice.
 - *Action type*: Asks (communicative action).
 - *Embedded action*: (i1) **Alice’s boss answers whether to accepts Prof. Smith’s request.**
 - *Individual*: Alice’s boss.
 - *Action type*: Analyse (non communicative action).
 - *Object*: Whether to accept Prof Smith’s request.
- (j) **Alice’s boss answer that she should accept Prof. Smith’s request.**
- *Individuals*: Alice’s boss.
 - *Action type*: Answer (communicative action).
 - *Embedded action*: (j1) **Alice know that she should accept Prof. Smith’s request.**
 - *Individual*: Alice.
 - *Action type*: Know (non communicative action).
 - *Object*: She should accept Prof. Smith’s request.
- (k) **Alice’s boss inform that Prof. Smith is a very good professor that deserves the requested exception.**
- *Individuals*: Alice’s boss.
 - *Action type*: Inform (communicative action).
 - *Embedded action*: (k1) **Alice know that Prof. Smith is a very good professor that deserves the requested exception.**
 - *Individual*: Alice.
 - *Action type*: Know (non communicative action).
 - *Object*: Prof. Smith is a very good professor that deserves the requested exception.
- (l) **Alice accepts Prof. Smith’s request.**

- *Individuals*: Alice.
- *Action type*: Accepts (non communicative).
- *Object*: Prof. Smith's request.
- *Embedded action*: None.

(m) **Alice orders Luisa the corresponding payment.**

- *Individuals*: Alice.
- *Action type*: Order (communicative action).
- *Embedded action*: (m1) **Luisa pay the course to Prof. Smith.**
 - *Individual*: Luisa.
 - *Action type*: Pay (non communicative action).
 - *Object*: The course Prof. Smith.

(n) **Alice informs Prof. Smith that payment is ordered.**

- *Individuals*: Alice.
- *Action type*: Informs (communicative action).
- *Embedded action*: (n1) **Prof. Smith know that payment is ordered.**
 - *Individual*: Prof. Smith.
 - *Action type*: know (non communicative action).
 - *Object*: Payment is ordered.