Linked Enterprise Models and Objects providing Context and Content for creating Metadata

A rule based approach for creating metadata of enterprise objects using an enterprise ontology.

A Thesis Presented to the Institute for Information Systems of the University of Applied Sciences Northwestern Switzerland FHNW in Partial Fulfilment of the Requirements for the Degree of Master of Science in Business Information Systems,

by **Andreas Martin**, July 2010.

Supervisors:
Prof. Dr. Knut Hinkelmann
Barbara Thönssen, M.A.

University of Applied Sciences Northwestern Switzerland
School of Business

Abstract

Even the smallest enterprise has to manage so much information and documents, that a system for arranging these things is needed; even if it is a small binder. Now when we think about the amount of information which today exists in a company, we have surely to say, that information and knowledge management is not done by only one binder – the companies nowadays need something more sophisticated.

What companies nowadays need is information about information – metadata. If metadata is available, then the finding and filing process can be dramatically improved. But if the metadata is not available, it needs to be created – and this has to be done in most of the cases by hand. Would it not be great to have an automatic approach? This thesis introduces an approach for creating metadata in an automatic way based on rules and a formal description of an enterprise.

We often hear the statement that a company has the information available – “We have the information in our systems.” But it is the question how the information is available. The Linked Enterprise Models and Objects (LEMO) approach gives the possibility to formalise the information in an enterprise. And not only the information, LEMO tries to make the relationships / links between different enterprise objects, documents, people, customers, money, almost everything in an enterprise explicit and machine process able using an ontology called enterprise model ontology (EMO). This EMO can be seen as context description of an entire enterprise. And this context can be used to create metadata using rules.

The thesis provides beside the EMO and LEMO approach a demonstrator who shows the possibility of creating metadata using the mentioned ontology and rules. The whole approach comes accompanied by an application scenario based on a real world case.

Keywords: Linked Enterprise Models and Objects, Enterprise Model Ontology, Enterprise Ontology, Metadata, Semantic Rules, Semantic Web, Enterprise Architectures

Acknowledgments

I would like to take here the chance to thank all people who supported me in this work and my master studies.

First I would like to express my thanks to my supervisors and colleagues Prof. Dr. Knut Hinkelmann and Barbara Thönssen for investing a lot of time for many valuable discussions concerning this thesis and the good support.

Special thanks go to Roman Brun, my fellow student and colleague. It was a great pleasure for me to study alongside with Roman and doing many group works. I would like to thank Roman for his trust and endurance for doing the literature review and the LEMO approach of this thesis together.

I would like to thank to my interviewees Andreas Kyriakakis, Felix Akeret and Hanspeter Hebeisen for giving me an insight into practice.

Last but not least, final thanks go to my mother Dora and my father Hanspeter who supported and always believed in me during my whole studies, and Beatrix for her love.

Table of Contents

List of Figures

List of Tables

1 Introduction

Knowledge is one of the main assets in a company. Discovery of enterprise knowledge assets is one of the most important and critical processes in an organisation. Knowledge is in every company available – the question is in which from it exists. Knowledge is in people's minds, in documents, databases or web pages – and we often hear the statement that the knowledge is available. But the question is again in which way the knowledge is available – by asking people, searching and reading documents or by accident.

The main objective of this thesis is to make knowledge accessible and reusable in a formalised way. Of course this is a huge claim and this thesis will not deliver for every knowledge discovery problem a solution. But this thesis will start with an inventory of an enterprise which provides a structure and an overview about everything in an enterprise. This main enterprise structure containing all enterprise objects (this everything's) is the basis for an enterprise knowledge repository which allows finding and reusing of enterprise knowledge.

The thesis will finally show how such an enterprise repository can be used to create new knowledge, or to be more precise, knowledge about knowledge, also called metadata.

1.1 Background information and topic introduction

With the accelerating innovation cycle in a globalized world, organisations need to adapt their IT solutions to the changed business requirements more frequently and unexpectedly. Various decisions about business and technology development have to be considered in a systematic manner (Schekkerman, 2003). Enterprise Architecture Frameworks (EAF) aim to support that complex tasks but fall short in several respects. According to (Dietz, 2006), a conceptual model for an enterprise is needed to be coherent, comprehensive, consistent, concise and essential. In an enterprise often co-exist various - sometimes even overlapping - models and systems for various purposes. There exist business process models alongside with workflow management tools, IT infrastructure models, organisation models, customer relationship tools, internal control systems, enterprise resource planning (ERP) and document management systems (DMS) just to mention a few of them. Even if the relations between these various components are described in an Enterprise Architecture they are invisible from an operational point of view. Apart from migration descriptions, e.g. from business process models to executable workflows, or APIs between IT-systems, e.g. ERP and DMS, no holistic model is available. Additionally, relationships between different perspectives and aspects are not formalised nowadays (Jonkers et al., 2004) but highly important (Langenberg & Wegmann, 2004). Another drawback of currently available Enterprise Architectures (EA) is their inflexibility.

As the goal of an EA is to make the complex interdependencies between business and IT explicit, they are not designed for enabling an enterprise to identify trends, pinpoint where action is needed, support integration and show potential improvement (Anaya & Ortiz, 2005; Hanschke, 2010). Having such a functionality, an enterprise would become more agile as changes and their dependencies could be monitored on an operational level and made transparent for the management.

Most of the existing enterprise architecture frameworks use natural languages for their descriptions which leads to more problems. Natural language leaves room for interpretation, there is no common understanding of the application domain (Bertolazzi, Krusich, & Missikoff, 2001).

Modelling an EA in an ontological representation addresses existing problems by using a formal representation (Kang, Lee, Choi, & Kim, 2010). EA solution vendors like (TopQuadrant Inc., 2009) too claim that there is a need for formal representations of an enterprise architecture in an ontology. (TopQuadrant Inc., 2009) sees the advantage of an ontology based approach in the feasibility of reliable aggregation of models and “the ability to capture and relate distributed models at multiple levels and from multiple viewpoints […]”. Therefore an enterprise architecture is needed that is explicitly and formally described thus it can be used on an operational level. Describing an enterprise in an ontological manner helps to achieve a common understanding among various stakeholders, to solve ambiguity and therefore create a 'business approved' model (Hinkelmann, Merelli, & Thönssen, 2010). "Moreover, the explication and consolidation of implicit ideas and assumptions […] transcend the mere development of a shared understanding. Ontologies allow for the reuse of domain knowledge." (Janusch et al., 2008:p.11). Missing reusability has been recognized as weakness of traditional systems (Geerts & McCarthy, 2000). This approach introduces an enterprise ontology to support enterprise's agility from an operational point of view.

1.2 Problem statement concerning metadata

The management of business objects such as files, tasks, processes, etc. in a company is not easy. One big aspect is the filing and retrieval of certain business objects. Usually the filing and retrieval can be assisted by the creation or usage of metadata about business objects. The retrieval of digital assets benefits from available metadata. The main problem is how to come to this metadata. One way is the usage of metadata created by people. But the main challenge of human-created metadata is the potentially high cost of production in time, money, errors and inconsistency that can occur (Malaxa & Douglas, 2005). Automatic metadata generation promises that it can help to reduce these efforts (Greenberg, Spurgin, & Crystal, 2005). (Greenberg, Spurgin, & Crystal, 2005:p.1) derived from the results of automatic indexing developments, that "[…] automatic metadata generation is more efficient, less costly, and more consistent than human-oriented processing". The problem is now how to create metadata in an automatic way and based on which information. The mentioned enterprise architecture approach and enterprise ontology could provide the needed knowledge base to create metadata.

1.3 Thesis statement, definitions, limitations and research questions

1.3.1 Research questions

At the beginning of every research work there are some main questions. Here in this thesis it is not different from this general "rule". The following questions accompany this work.

- How to describe formally a whole enterprise?
- How to use an enterprise repository to create metadata?
- How to define good metadata?

1.3.2 Research objective

The goal of thesis is the generation of metadata for enterprise business objects based on an enterprise ontology. The first milestone of this project is to develop an enterprise ontology for an enterprise repository. There exist several enterprise architecture models and upper ontologies which propose different domains for the representation of it. As an upper ontology describes the very general concepts that are the same across all domains it can be used as fundament for the enterprise repository. Therefore the first objective of the thesis is to assess enterprise architecture frameworks and upper ontologies to develop an upper ontology that covers the needed domains for the thesis.

The mentioned enterprise repository, including an upper ontology with instances, provides the basis for the metadata generation. This metadata creation, which can happen either manually or automatic, will be triggered by the actual context and is clearly defined by rules. In other words the acquisition of the metadata of the business objects (this thesis focuses on files) will be implemented based on rules which deliver information about the context of a business object: what and how (the actual rule), when and where (time, position, process state, etc.), who (person, service). This thesis should deliver an upper ontology, an exemplary use case and a partly defined enterprise ontology, context rules and a possible (technical) architecture as a description as well as an implementation.

1.3.3 Thesis statement

Content and context of an enterprise can be represented in an enterprise ontology. This enterprise ontology depicts the general concepts and their relations, which correspond to the model elements of various model types. This ontology is used for rule based metadata creation and intelligent retrieval of enterprise objects.

The following five sub-goals describe the thesis statements in more detail:

1. An enterprise ontology containing the content and the context of an entire enterprise has to be developed which serves as basis for an enterprise repository.
2. The enterprise ontology reflects theory by considering relevant literature of enterprise architecture, upper ontologies and information management systems.

3. The enterprise ontology reflects the practice by conducting interviews to the topic of metadata management and creation.
4. The developed enterprise ontology can be used to create metadata for enterprise objects in general and in particular for a simple application scenario.
5. The metadata creation will be exemplary shown by developing a prototype with the aim of reusing as much as possible code and open source software.

1.3.4 Assumptions

The thesis assumes that the reader of it is familiar with terms like ontology including the related technologies.

1.3.5 Delineations and limitations

As described in the sub-goals of thesis statement the metadata creation will be assessed based on an application scenario and a demonstrator.

- The application scenario and demonstrator of the thesis will focus on electronic files containing text.
- The thesis will not focus on information extraction. The demonstrator will show if it is possible to use rules and an enterprise knowledge base to create new metadata under the assumption to get appropriate input data.

1.4 Brief chapter overview

The thesis is divided into ten chapters. The thesis begins with the introduction including the thesis statement and sub-goals. These two elements influence the other parts dramatically – as it will be explained in the research methodology, the thesis statement predefines the path of this thesis. After introducing the work, the literature review part will be written in chapter two. Chapter three, the research methodology, contains the description how the research work is performed. The proposed interviews will be written down in section four. In section five a new approach on linked enterprise models and objects will be introduced. And in section six the approach concerning metadata will be described. Chapter seven will come up with an application scenario. And an implementation of the approach will be introduced in chapter eight. The whole approach will be evaluated using the application scenario in chapter nine. And finally chapter ten concludes the whole work and gives an outlook to the future work.

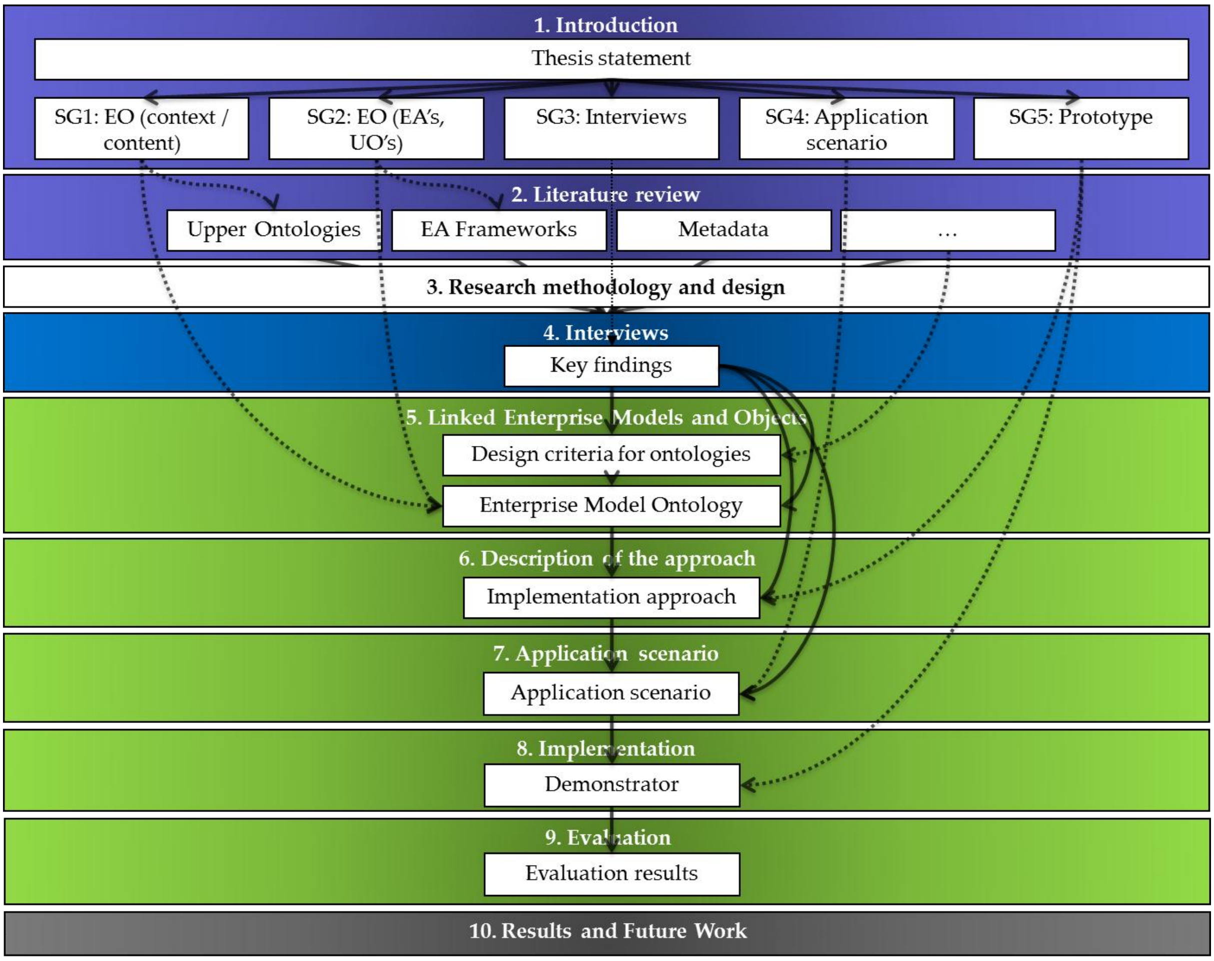

Figure 1: Chapter overview

2 Literature Review

2.1 Enterprise Context and Domain

2.1.1 What is context?

A first formal definition of context was introduced by (McCarthy, 1993). He tries to introduce context as abstract entities. But his work did not final in a universal definition of context.

Apart from the formal definition, context makes it humans possible to interact with humans. Or as (Dey, 2001) explains it in more detail: Humans "[…] are able to *use implicit situational information*, or context, to increase the conversational bandwidth." (Schilit, Adams, & Want, 1994) identified three important questions (aspects) of context: "[…] where you are, who you are with, and what resources are nearby". (Dey & Abowd, 1999) describe context more in a situation centric way. The question is, if a piece of information can describe the *situation of an entity*, "[…] then that information is context." They define an entity as "[…] a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."(Dey & Abowd, 1999).

They also differentiate between *primary and secondary context*. "**Location, identity, time, and activity** are the primary context types for characterizing the situation of a particular entity." (Dey & Abowd, 1999). This primary context can be used to "[…] find secondary context (e.g., the email address) for that same entity as well as primary context for other related entities (e.g., other people in the same location)." (Dey & Abowd, 1999).

A broader definition of context was introduced by (Leppänen, 2007). He concluded context as "[…] a whole, composed of things connected to one another with contextual relationships. A thing gets its meaning through the relationships it has with the other things in that context." (Leppänen, 2007). Further (Leppänen, 2007) defined seven contextual domains for specifying the contextual phenomena: purpose, actor, action, object, facility, location, and time.

(Dey & Abowd, 1999) defined the field of context- aware applications as follows: "Context-aware applications look at the **who's, where's, when's and what's** (that is, what the user is doing) of entities and use this information to determine **why** the situation is occurring. An application doesn't actually determine why a situation is occurring, but the designer of the application does."

Based upon the definitions of (Dey & Abowd, 1999; Leppänen, 2007; Schilit, Adams, & Want, 1994) context is defined for this work as follows:

> *Context describes the situation of an entity through the identification of the purpose, the primary and secondary context types. The primary context types are represented by the interrogatives, who (object, identity), what (action), when (time), where (location). These primary context types can used to find secondary context types (facilities, resources, etc.) and related entities or vice versa.*

2.1.2 What is a domain?

In this paper the term domain is used with the definition of the domain of discourse, which often comes along with the universe of discourse. The term universe of discourse was first introduced by (De Morgan, 1846). (Bergman & Paavola, 2003) defined it based on (Peirce, n.d.) as: "The universe of discourse is the aggregate of the individual objects which 'exist,' that is, are independently side by side in the collection of experiences to which the deliverer and interpreter of a set of symbols have agreed to refer and to consider." (Bergman & Paavola, 2003) The statement of (Regoczei & Plantinga, 1987) brings it to the point: "The domain of discourse is defined by the discourse."

(Gruber, 1993a) uses the terms domain and universe of discourse related to ontology as follows: "When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, we can describe the ontology of a program by defining a set of representational terms. In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names are meant to denote, and formal axioms that constrain the interpretation and well-formed use of these terms." (Gruber, 1993a).

Based on these explanations we define domain of discourse as follows: The domain of discourse is an agreed set of objects being discussed in a specific discourse.

2.2 Metadata

Metadata is data about data – e.g. information like author or date of a given document. The International Federation of Library Associations and Institutions (IFLA) define metadata like this:

> *"Metadata is data about data. The term refers to any data used to aid the identification, description and location of networked electronic resources. Many different metadata formats exist, some quite simple in their description, others quite complex and rich."*(IFLA, 2005).

There exists several definitions about the term metadata and the topic is always under discussion. This is the same for the metadata standards – different standards are available. The following section will introduce as representative the often mentioned metadata standard Dublin Core.

2.2.1 Dublin Core

Dublin Core goes back to a discussion during a web conference in 1994 about the need for a document or resource description as it is used in a library for printed books. A first representation of such a standard was proposed in a workshop held at Dublin, Ohio USA in 1995. It was clear the standard should provide a set of descriptors which were easy to create and understand. Dublin Core provides now as an ANSI standard a simplified set of elements that can be used to describe resources of different disciplines (Breitman, Casanova, & Truszkowski, 2007).

Element	Definition
Title	object name
Creator	person/people responsible for the intellectual property of the object
Subject	main topic
Description	description of the object contents
Publisher	agent or agency responsible for making the object available
Contributor	person/people that made significant contributions to the object
Date	publication date

Type	object type, e.g., fiction, reference, novel, poem
Format	physical manifestation of the object, e.g., executable file, PDF
Identifier	an unambiguous reference to the resource within a given context
Relation	reference to a related resource
Source	a Reference to a resource from which the present resource is derived
Language	language of the intellectual content
Coverage	spatial location and temporal duration of the object
Legal rights	legal rights information about the object in question

Table 1: Elements of Dublin Core (Breitman, Casanova, & Truszkowski, 2007).

The Dublin Core set as depicted in Table 1 provides a simple standard for describing different digital resources.

2.3 Enterprise Architectures

The term 'enterprise architecture' consists of two parts, 'enterprise' and 'architecture'. An architecture is "The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution." (IEEE, 2000), whereas an enterprise can be defined as "any collection of organisations that has a common set of goals and/or a single bottom line." (Lankhorst, 2005).

Since the late 80ties various methods and frameworks for enterprise architecture have already been developed, starting with the Zachman Framework (Zachman, 1987). Some of the wellknown EAs are, the Open Group Architecture Framework TOGAF (The Open Group, 2009), ARIS (Scheer, 2000), the Enterprise Architecture Framework of plugIT (Wache et al., 2010), the Business Process Management Systems BPMS (Karagiannis, 1995) and the 'Best-Practice Enterprise Architecture' proposed by (Hanschke, 2010).

The main objective of modelling enterprise architectures is to describe the interaction between business and information systems (Grembergen, 2004; Lankhorst, 2005; Ross, Weill, & Robertson, 2006). It facilitates the vision of integrated systems (OMG, 2003), i.e. an enterprise integration. According to (Bernard, 2009), another important point is the completeness of the overall approach, i.e. the framework covers the whole enterprise architecture. According to (TopQuadrant Inc., 2009), the next generation of EA solutions will be based on ontological representations in order to better aggregate and analyse information and increase semantic interoperability.

Enterprise architectures have various benefits which are largely qualitative (Hanschke, 2010). Therefore it is important to identify the stakeholders, their viewpoints, and to understand their needs in order to convince anyone of the improvement potential. They need to be able to identify with the enterprise architecture: The management board wants the corporate goals to be integrated, the controller needs a controlling instrument and a process owner wants to do some analysis. In order to fulfil the stakeholders needs, enterprise architecture must consider the various viewpoints of them. (Lankhorst, 2005; Scheer, 2000; Wache, Eichner, Koutsoukou, & et.al., 2009; Zachman, 1987) inter alia, consider this requirement.

2.3.1 Zachman

The Zachman Framework was introduced by John A. Zachman in 1987 as an enterprise architecture framework (Zachman, 1987). The framework provides different views and representations of an enterprise and a classification scheme for organizing the several aspects and perspectives of the enterprise. These views are arranged in two dimensions - the column represents different aspects and the row represents different perspectives which are based on different roles (see Figure 2). The aspects are named based on the fundamentals of communication. The interrogatives What (data), How (function), When (time), Who (people), Where (network), and Why (motivation), build the basis for the concise description of complex ideas (Zachman, 2008). The corresponding roles to the perspectives are named planner, owner, designer, builder and subcontractor.

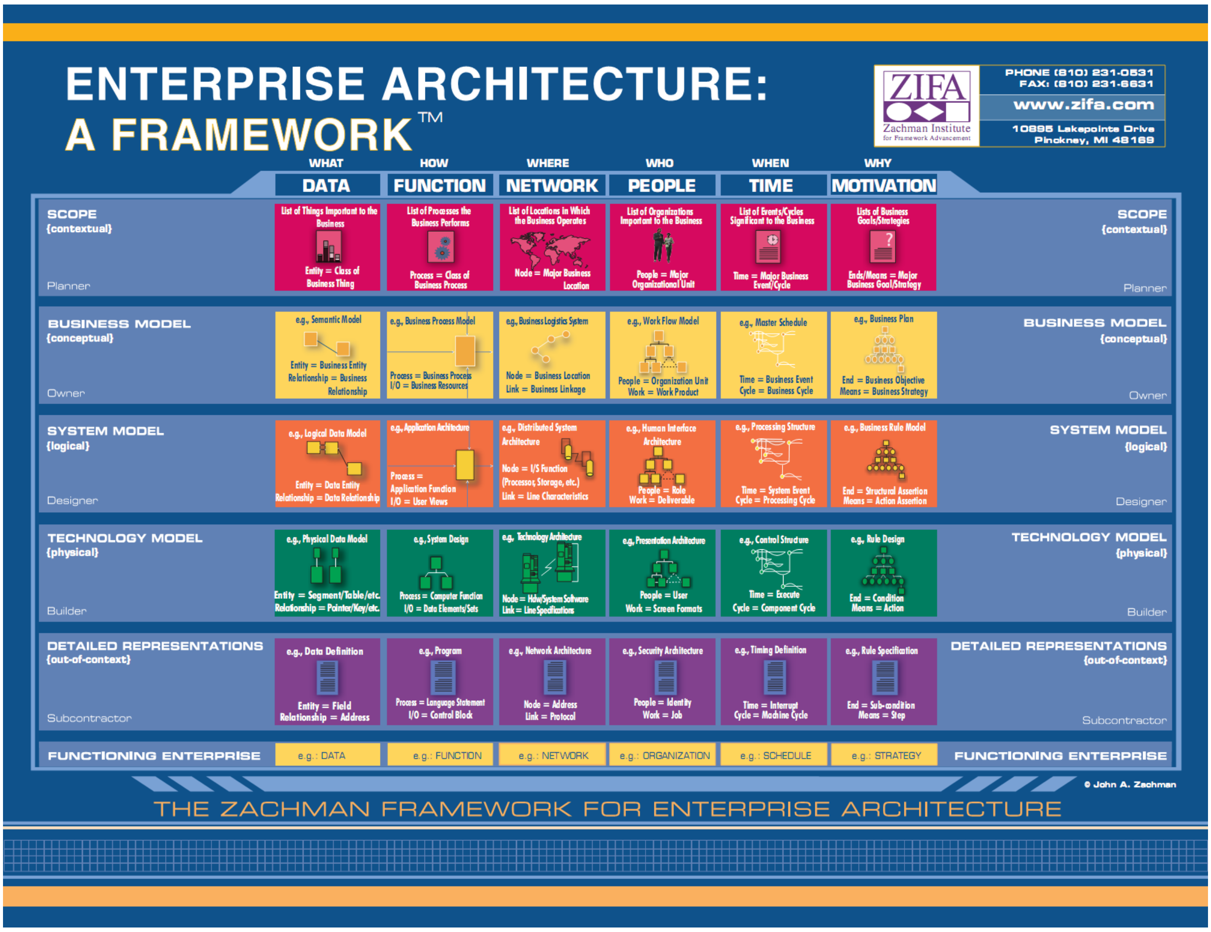

Figure 2: The Zachman framework for enterprise architecture (Zachman, 2003).

2.3.1.1 Critique from literature

(Lankhorst, 2005) sees the advantage of the Zachman Framework in the fact that the framework is easy to understand and tries to address companies as a whole. (Kang, Lee, Choi, & Kim, 2010) argued, that the framework has some drawbacks in modelling detailed components. They further criticize that the relations between the individual components are not elaborated in detail. (Lankhorst, 2005) goes a step further. He criticize in his book, that the roles especially planner and subcontractor are arbitrarily labelled and relations between some cells are not precise enough. (Schönherr, 2004) depicts in his book the missing integration or comprehension of an existing infrastructure. He points out that this drawback is a significant problem when running an integration project. The critics of (Rohloff, 2005) goes in a other direction. He complains that there are too many views included in the Zachman Framework and this would make it difficult to implement it. Further on he argued that not all views would be described in detail and the relationships would not be transparent. "The framework is a mix of views, domains of enterprise architecture, and different stakeholders" (Rohloff, 2005). (Hanschke, 2010) sees the lack of adequate tool support, guidelines or methodology for designing and rolling out an architecture as main drawback. (Chen & Pooley, 2009) identified some difficulties when trying to use the Zachman Framwork for information system development. They complain that the framework would not provide a metamodel for each cell and it would not recommend a sequence for software engineering. They also see the lack of dependencies or relationships among the cells (Chen & Pooley, 2009). (Bernard, 2009) tries in his work to map different enterprise architectures to an exemplary complete enterprise architecture approach, with the result, that it was not possible to relate all introduced artefact a relationships to the Zachman Framework.

2.3.2 The Open Group Architecture Framework (TOGAF)

The Open Group Architecture Framework (TOGAF) is a generic framework for developing architectures to meet different business needs (The Open Group, 2009). It can be used in conjunction with other frameworks such as Zachman or ARIS (Ivanov, 2010) and tailored to specific needs/scopes, e.g. an architecture domain, a vertical scope in the enterprise (level of detail) or time periods (project schedule).

The current 9th version of TOGAF consist of the following main components as shown in Figure 3:

- An Architecture Capability Framework addressing the organisation, processes, skills, roles and responsibilities.
- The Architecture Development Method (ADM) described below.
- The Architecture Content Framework considers an overall enterprise architecture composed of four interrelated architectures: Business Architecture, Data Architecture, Application Architecture and Technology (IT) Architecture. The individual architectures are described in detail and relations among them are defined.
- The Enterprise Continuum and Tools is a model for structuring a virtual repository and comprises of various reference models to show the development from a foundational to an enterprise specific architecture.

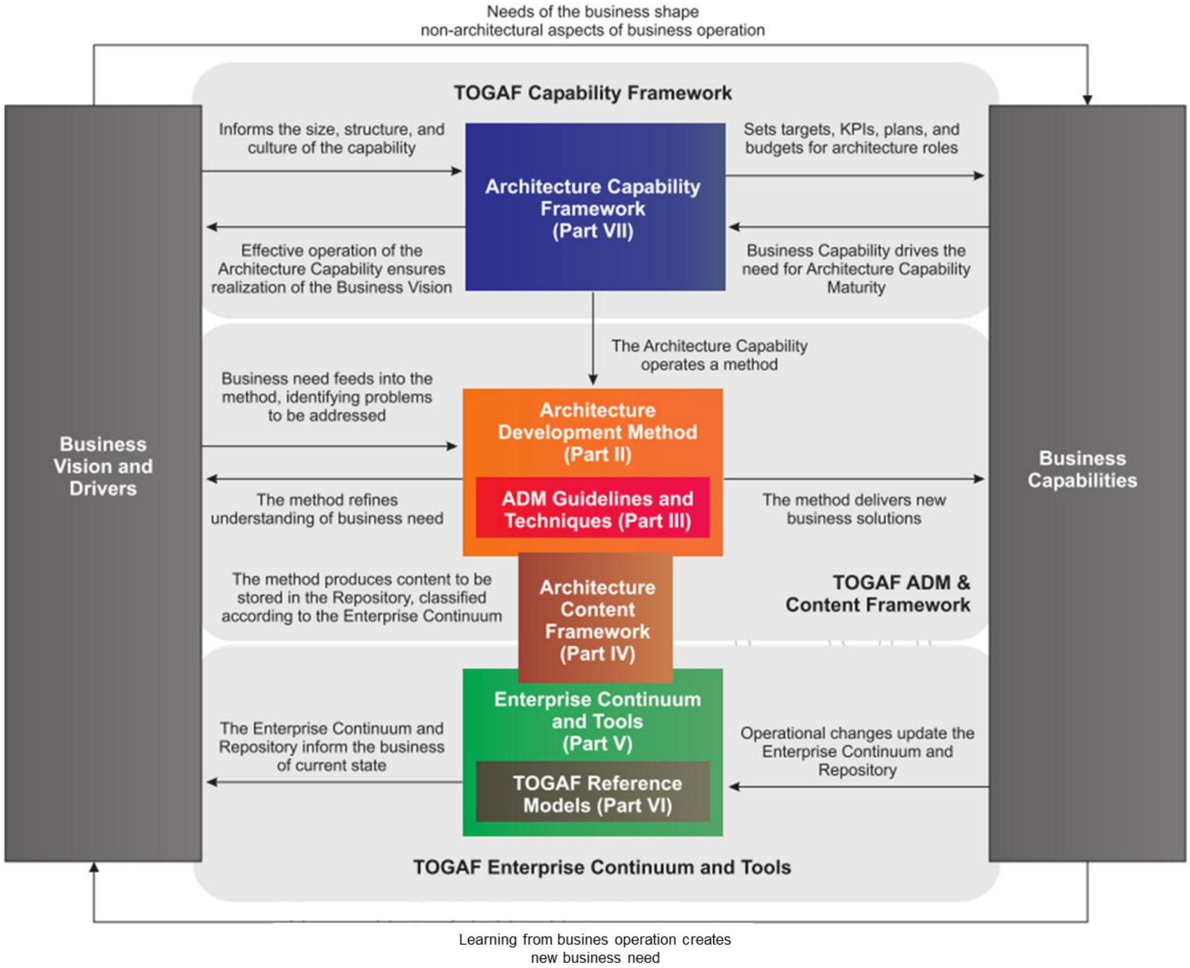

Figure 3: TOGAF 9 (The Open Group, 2009:p.4)

The Architecture Development Method (ADM) is an iterative sequence of steps to develop an enterprise-wide architecture (see Figure 4). It consists of eight phases. In phase A, the objectives and responsible persons are identified. In phases B to D, the desired state of the business-, information systems- and technology-architecture are specified. Phase E describes needed transformations to achieve a desired state. Phase E develops the detailed implementation and migration plan which is monitored in phase G. Finally a change management process in phase H collects requirements and ascendancies serving as foundation for the next cycle.

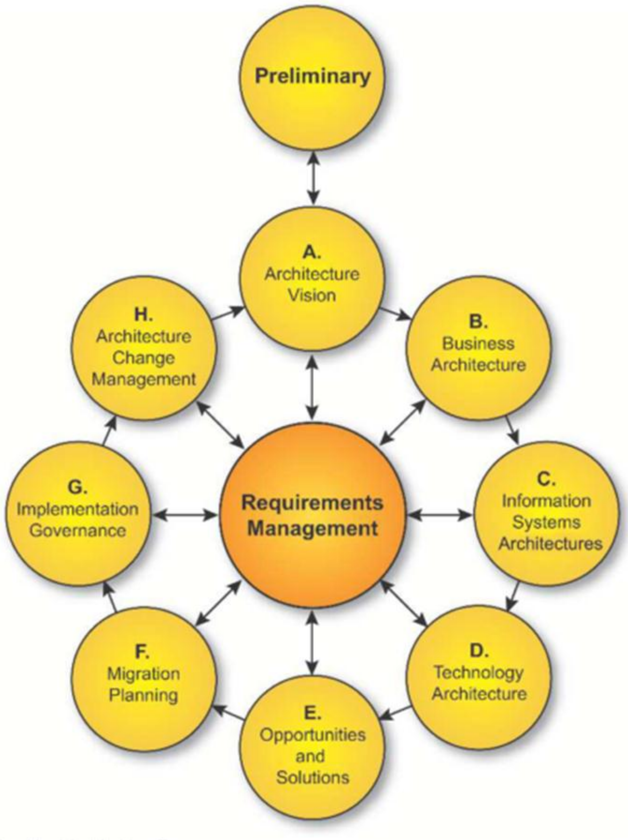

Figure 4: TOGAF Architecture Development Method (The Open Group, 2009:p.54)

2.3.2.1 Critique from literature

TOGAF is a huge framework with a documentation of hundreds of pages, not only considering an enterprise architecture itself but mainly its development and maintenance. While this can have advantageous, also lot of effort and time is needed to fully understand and implement it; even if it is only used for a project or a specific enterprise focus. Likewise the Pragmatic Enterprise Architecture Framework (PEAF) states that TOGAF is "very complex and large therefore difficult […] to adopt" (Pragmatic EA Ltd, 2010:p.11) and proposes a pragmatic and more simple simple approach.

Despite its comprehensiveness, according to (Sessions, 2006), TOGAF fails to manage the complexity of todays systems. Session proposes a 'partitionend iterative approach' to address the complexity. Instead of following a top-down approach, single business partitions should be iterative completely modelled till the whole enterprise is covered. Another drawback also belonging to most other enterprise architecture frameworks is the lack of formalism.

2.3.3 ARIS architecture for Information Systems

ARIS (Architecture of Integrated Information Systems) Architecture was introduced by (Scheer, 1991) as integration concept. It introduces five main views illustrated in the so called ARIS house (concept) – the main picture of the ARIS architecture which is the main entry point to the ARIS Toolset designed in 1992 by Scheer. The framework consists of five views (organisation, data, control, function and output) where every view is separated into three levels (requirements definition, design specification and implementation description). The origin purpose of the ARIS house is a concept for describing business processes (Scheer, 2000).

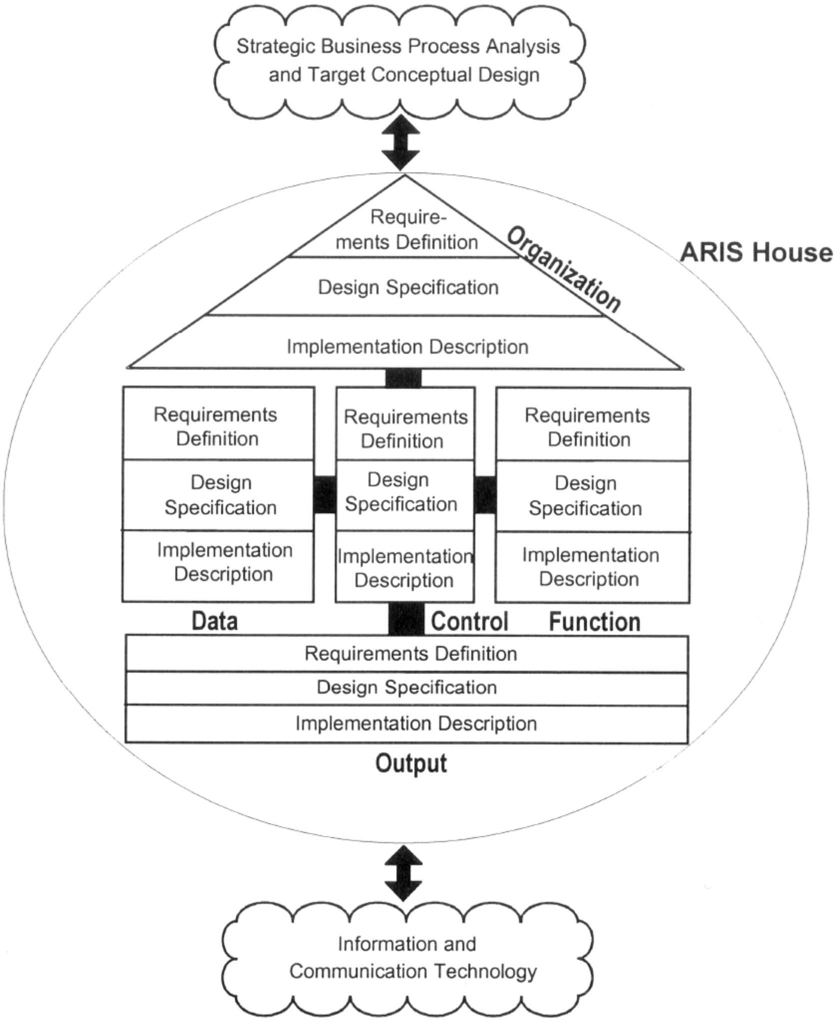

Figure 5: ARIS house (Scheer, 2000)

The organisational view contains of elements like organisational chart as part of the requirements definition. On this level typical elements are organisational unit, location, position and a concrete job holder. On the design specification the communication structure is modelled and the implementation description contains communication services. The data view contains entity- relationship models (requirement definition), relational model, tables or indexes (all in design specification) and files, software and access rights (implementation description). The functional view describes the functions and their relations in a company. Functions can be seen as operations for achieving the goals of an organisation. Elements of the functional view are functional trees (requirement definition), component and behaviour diagrams (both in design specification) and programming libraries (implementation description). The control view represents the processes of an organisation. Typical elements are event-driven process chains (requirement definition), graphical user interfaces (design specification) and software (implementation description). Finally the output view represents the output of the processes like services, financial and non- financial kind. ARIS uses the concept product for any kind of output. (Scheer, 1991; Scheer, 2000; Scheer, 2002)

2.3.3.1 Critique from literature

(Lankhorst, 2005) complains the semantic shortcoming of ARIS. "This is […] the case for the corresponding object models which are specified in a rudimentary meta-model. For this reason, ARIS lacks a solid formal foundation and is of limited use for the design of (application) architectures. The graphical notation of ARIS is unambiguous, but rather extensive, with quite a learning curve. While ARIS allows for various perspectives on the enterprise (the data view, the control view, the process/function view, and the organisation view); the integration of these aspects is somewhat lacking. Therefore, the tool does not guarantee the overall integrity of interrelated models. The tailorability of ARIS is limited to business modelling, and more precisely to organisational, functional, and process modelling; ARIS is not extensible." (Lankhorst, 2005:p.36). (Jonkers et al., 2004) finding goes in the same direction. ARIS has "[…] a clear focus on business process modelling and organisation modelling." And (Braun & Winter, 2005:p.68) sees the drawback that "[…] ARIS does not explicitly mention a strategy layer."

2.3.4 Business Process Management Systems (BPMS)

The BPMS paradigm (Karagiannis, 1995) proposes a procedure for Business Engineering. The management cycle consists of five sub-processes 'Strategic Decision Process', 'Re-Engineering Process', 'Resource Allocation Process', 'Workflow Management Process' and 'Performance Evaluation Process'. As described in (Wache, Eichner, Koutsoukou, & et.al., 2009), these sub-processes correspond to different perspectives.

The BPMS method (Karagiannis, Junginger, & Strobl, 1996) describes a universal modelling technique and a closed circle method for business process mangement starting from strategic decisions, the design of business processes, their implementation, execution and evaluation.

The management suite of (BOC-Group, 2010) which implemented the approach offers a set of different model types. Originally started with modelling types for business process management, further model types as Strategy and Performance Management, Supply Chain Design, and IT Architecture were integrated (see Figure 6).

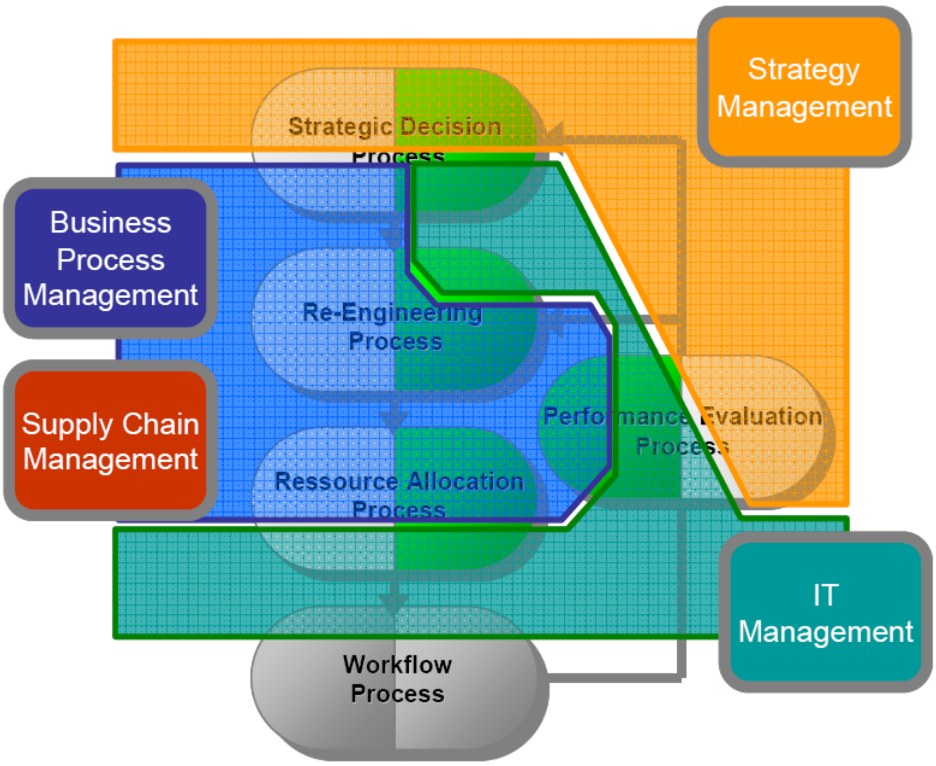

Figure 6: Model type families in the BPMS approach (BOC-Group, 2010)

According the modelling hierarchy (Karagiannis & Kühn, 2002), for each model type a meta model is defined, derived from a common meta² model which explains on a high level how elements of modelling types are connected.

2.3.4.1 Critique from literature

As the BPMS does not represent such a well-known approach as e.g. Zachman, much less citations of it can be found. (Neubauer & Stummer, 2007) criticise that existing BPM systems, to which the BPMS approach also belongs to, "do not provide methodical support for the valuation and selection of efficient solutions for IT investments." Meaning the approach lacks of appropriate procedures to measure potential benefits of new information technologies because an alignment between IT and the corporate strategy is missing.

(Ravesteyn, 2009) describes that it is unclear how BPMS approaches as the one of (Karagiannis, 1995) can be implemented and what business value it brings. It can be agreed that BPMS approaches should be implementable and show the business value, but the described on actually does. The approach has been implemented, is able to show the business benefit[1] and captures the specific context of an organisation by modelling it.

2.3.5 Best Practice Enterprise Architecture

(Hanschke, 2010) proposes a 'Best Practice Enterprise Architecture' that is based on the personal experience of the author and ready to use 'out of the box'. It highlights the importance of the relationship between business and IT and focuses on strategic management of IT landscapes for which enterprise architecture is the basis. The author states that most existing enterprise architecture approaches are "not always practical and ready for use 'out of the box' [but] require you to wade through reams of documentation and then make your own appraisal of which parts are relevant or applicable for you" (Hanschke, 2010:p.58). Therefore this framework has been developed to define an own enterprise architecture framework through simple configura-

[1] http://www.adonis-community.com/ "hitting the milestone of 10'000 users". Accessed on April 9th, 2010

tion. According to it an open source software for strategic management of IT landscapes called 'iteraplan' (http://www.iteraplan.de) has been developed.

Figure 7 represents the approach which includes four sub-architectures: Business Architecture, Application Architecture, Technical Architecture and Infrastructure Architecture. The business architecture describes the business structure and contains of business processes, business functions, products, business units and business objects. Between the elements interrelationships can exist. The enterprise architecture describes the enterprise's architecture landscape with the elements Applications, Interfaces and Information Objects. It serves as a bridge linking the business architecture with the technical and infrastructure architectures by describing the data which is used and exchanged by applications. The technical architecture contains enterprise specific technology standards, reference architectures and samples, software engineering tools, IT components and more. It is able to present planned standards whereas the infrastructure architecture describes infrastructure elements on which the applications are currently running.

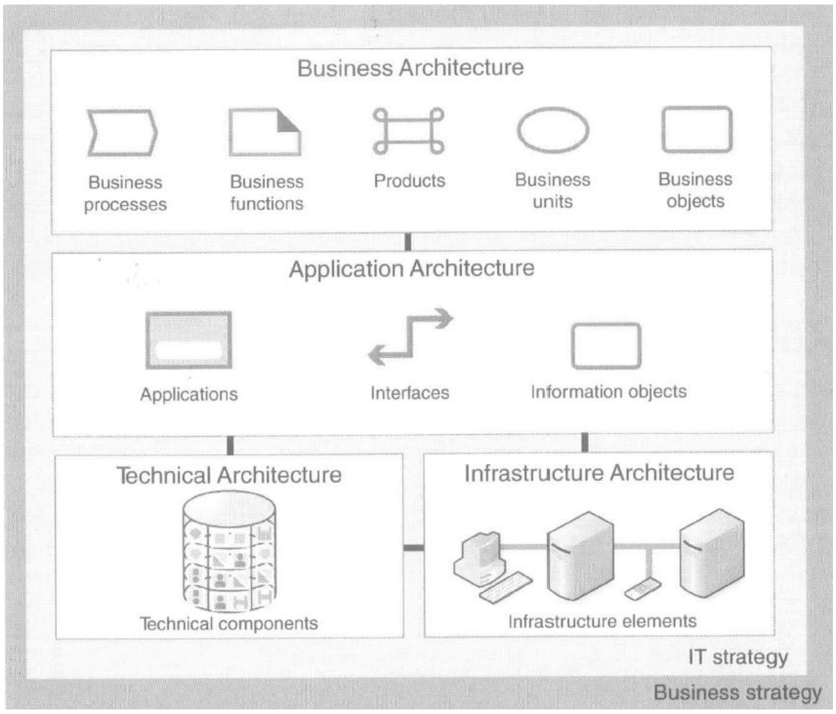

Figure 7: Best-practice enterprise architecture (Hanschke, 2010:p.66)

For the four sub-architectures, Hanschke gives examples of possible graphical representations. With respect to granularity, hints and guidelines are given to achieve a complete and up-to-date model which is still manageable. Furthermore, guidelines for the personalisation of the approach are stated. Additional, a method for strategic planning and steering of an IT landscape is defined based on the aforementioned best practice enterprise architecture.

The IT landscape management helps in "shaping the future application landscape to fit your enterprise goals and business requirements." (Hanschke, 2010:p.106). It is a tool to manage the complexity of the IT landscape and adjusts the application landscape to the strategic goals.

2.3.5.1 Critique from literature

As the proposal has just been published no critique related to it could be found yet. However, based on the whole literature review and the analysis of the approach I want to mention the following critique: Already indicating by the name of the book of Hanschke 'Strategic IT Mangement', the focus lies more on the IT side. Only a brief section investigates in a business landscape management and another in how an enterprise architecture can contribute to the strategic evolution of the IT landscape. Therefore the usage of the approach for my work keeps within limits. Another critique is related to the following statement "By omitting building blocks and relationships you do not need, you can derive your own enterprise architecture quite simply." (Hanschke, 2010:p.67). In my opinion this can be dangerous as important relationships might be lost with such an approach and it also somehow contradicts the statement of an 'out-of-the-box' solution. The importance of indicating relationships is also mentioned by (Karagiannis, Ronaghi, & Fill, 2007:p.97), stating that there is often a "missing linkage between the explicit definition of the business objectives and the IT infrastructure", meaning the composition of the enterprise architecture is not complete in terms of business engineering.

2.3.6 Enterprise Architecture Framework of plugIT

The object of the EU-FP7-ICT project plugIT is to "develop concepts, tools and methods summarised within the 'Next Generation Modelling Framework' (NGMF) that allows experts from both business and IT domains to use modelling languages that fit to their concrete needs." (BOC-Group, 2010:p.17). Figure 8 illustrates the four dimensions of the NGMF, namely perspectives, aspects, formalisation and language families.

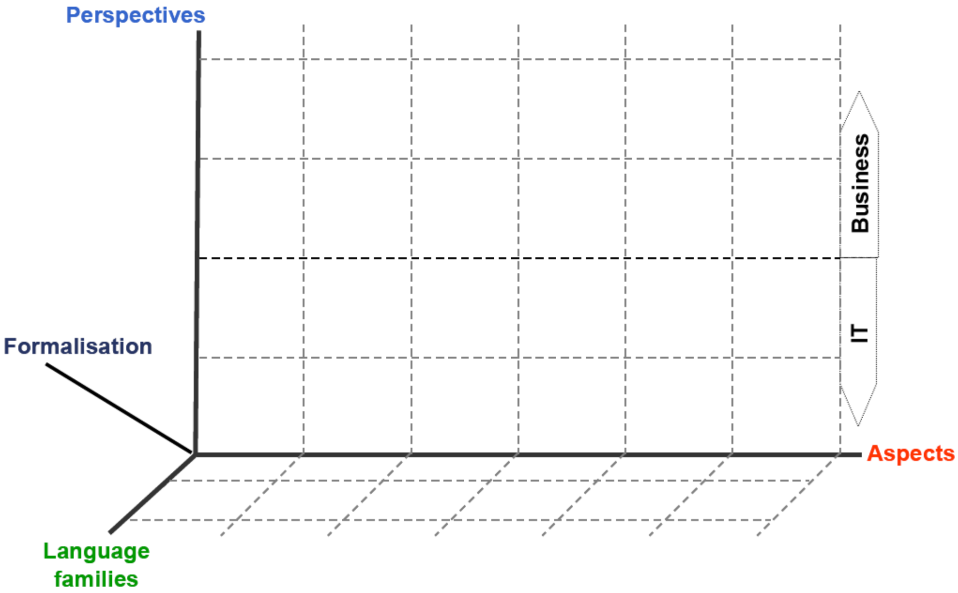

Figure 8: The dimensions of the classification framework (Wache, Eichner, Koutsoukou, & et.al., 2009:p.16)

As the purpose of the NGMF is the alignment of business and IT, they propose a business *perspective* containing the strategy and business view and an IT perspective containing the systems and technology view. The *aspect* dimension has great similarity to Zachman's framework with the distinction that the time aspect has been replaced by products. The proposed aspects are data/knowledge, process, organisation, applications, products and motivation. The *formalisation* dimension helps to identify the level of formalisation of a modelling language. They distinguish between the levels formal, semi-formal and informal. Further classification for the formal part is proposed by characterising the expressive power of a language as well as the type of it: graphical, textual or both. The fourth dimension classifies the *language families* which comprise of modelling languages (offered by e.g. standardisation bodies as OMG[2], W3C[3] or OASIS[4]). (Wache, Eichner, Koutsoukou, & et.al., 2009:p.24) define a language family as "a set of modelling languages that base on a common 'philosophy' or methodology" and a modelling language as follows: "A modelling language consists of modelling elements, relations and attributes that can be used together. It thus

[2] http://www.omg.org

[3] http://www.w3.org

[4] http://www.oasis-open.org

corresponds to a model type or diagram type. The proposed NGMF can be used to explain the relationship between perspectives and aspects and shows that some cells are not considered for modelling nowadays.

The framework at the current stage serves to classify existing language families and model types. It shows how translation between different models could be possible with semantic integration. For the alignment metamodells shall be used.

2.3.6.1 Critique from literature

As the approach has just recently being published and the project is on-going no critique related to it could be found yet. From the description of the approach no justification could be found why Zachman's time aspect is neglected.

2.4 Enterprise Ontologies

2.4.1 What is an ontology?

Depending on the definition of ontologies, they can be highly informal if expressed in natural language, semi-formal if a restricted form of a natural language is chosen or formal using defined terms, formal semantics and theorems (Gómez-Pérez, Fernández-López, & Corcho, 2004).

A well accepted definition of an ontology according to (Studer, Benjamins, & Fensel, 1998:p.186) is the following: "An ontology is a formal, explicit specification of a shared conceptualisation". The definition unites a few of the advantages of ontologies: A *formal* specification is machine readable (not just machine processable), can be validated and there is no room for interpretation. *Explicit* indicates that concepts, functions and axioms are explicit defined. *Shared* conceptualization indicates a common understanding about the concept, the 'modelled world', and assists the alignment between business and IT. Another claimed favour is reasoning. It is essential for the design and maintenance of ontologies in order to achieve high quality which includes meaningfulness, correctness, minimal redundancy and consistency checking among other criteria. Reasoning can also be use for mining (Hinkelmann, Merelli, & Thönssen, 2010). Another aspect is querying of ontologies, justification and proofs (Horrocks, 2005).

As (Gómez-Pérez, Fernández-López, & Corcho, 2004) mention, there are important connections between knowledge modelling components (elements), the knowledge representation paradigms (description logics, logic) and the languages used to implement the ontologies. Ontologies could even be modelled with UML or the Entity-Relationship (ER) notation. However, the opinions about the applicability of it differ. (Cranefield & Purvis, 1999) applies UML with few extensions and the use of the Object Constraint Language (OCL) as an ontology representation language; (Gómez-Pérez, Fernández-López, & Corcho, 2004) identified several drawbacks. It is therefore more suited to use a specifically for ontologies developed mark-up language as e.g. RDF(S) or OWL. The choose of an ontology representation language also depends on the aforementioned reasoning. When considering automatic reasoning the trade-off between the representational power of a formalism and the solvability of reasoning with it needs to be considered (Levesque & Brachman, 1985).

There exist various types of ontologies which have been categorized inter alia by (Lassila & McGuinness, 2001; Mizoguchi, Vanwelkenhuysen, & Ikeda, 1995; van Heijst, Schreiber, & Wielinga, 1997). This work will use upper ontologies and domain ontologies. Upper ontologies are further described in section 5. Domain ontologies are reusable in a specific domain and provide concepts and their relations. The concepts are usually specializations of concepts defined in upper level ontologies (Mizoguchi, Vanwelkenhuysen, & Ikeda, 1995). In general, upper ontologies are more reusable, but less usable than domain ontologies, and vice versa.

2.4.2 TOVE

The TOVE Enterprise Modelling project had the goal to create the next generation Enterprise Model (Common Sense Enterprise Model). Apart from the main goal the result of the TOVE project were two foundational ontologies (Activity, Resource) and four business ontologies (Organization, Product and Requirements, ISO9000 Quality, Activity-based Costing) (Born et al., 2008).

"TOVE Common Sense Model of Enterprise included three levels: reference model with typical business functions (finance, sales, distribution, and administration), generic model (with such concepts as time, causality, space, resources), and concept model (e.g. role, property, structure)."(Filipowska, Kaczmarek, & Markovic, 2008:p.2)

(Fox, Barbuceanu, Grüninger, & Lin, 1995) defined common sense as the (…) "ability to deduce answers to queries that require relatively shallow knowledge of the domain". The approach of (Fox, Barbuceanu, Grüninger, & Lin, 1995:p.124) was to create an ontology beginning with the requirements. They continue with the definition of the terminology of the ontology, by defining objects, attributes, and relations. Later they specified the definitions and constraints. And lastly, they tested it by providing competency questions with the Prolog axioms.

One of the major definitions is the ontology of time and action that is used to represent the behavior of the organization. "An important component of representing behavior is the ability to temporally project, that is, to determine the possible set of future states given a current state." (Fox, Barbuceanu, Grüninger, & Lin, 1995:p.129)

Figure 9 shows the basic terminology of the organisation ontology which was developed as part of the TOVE Project. It focuses on organisation structure, roles, authority and empowerment.

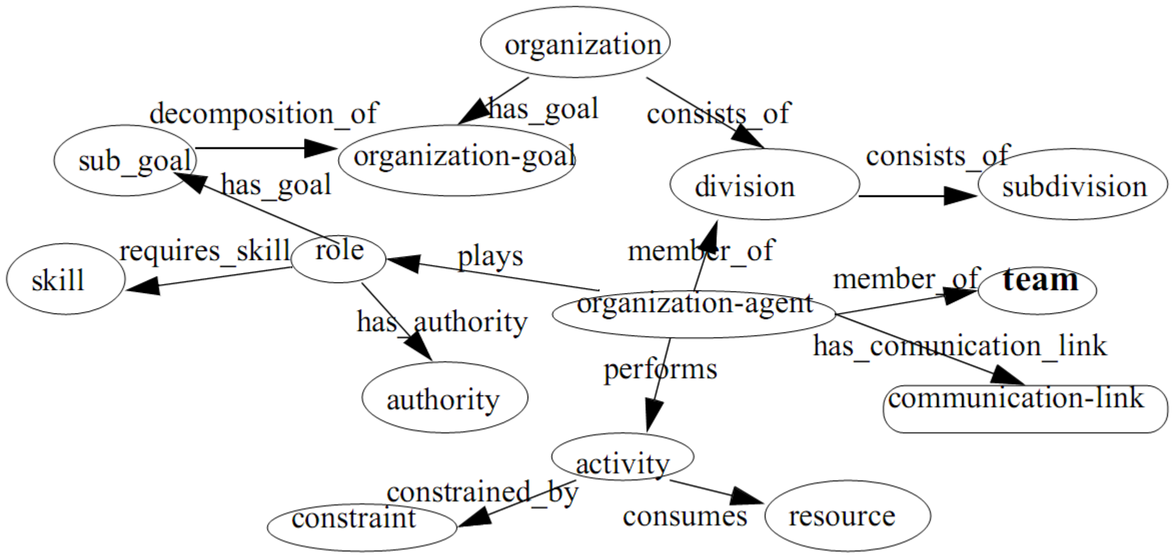

Figure 9: Organizational object taxonomy (Fox, Barbuceanu, Grüninger, & Lin, 1995:p.141)

2.4.2.1 Critique from literature

(Born et al., 2008) complains that the granularity of the ontologies would be inconsistent and that would make it inoperable to use. (Filipowska, Kaczmarek, & Markovic, 2008:p.2) came to the same findings: "[…] the granularity of developed ontologies may be perceived inconsistent and that hampers their potential application."

2.4.3 The Enterprise Ontology

The Enterprise Ontology (EO) consists of a collection of terms and definitions relevant to business enterprises defined by (Uschold, King, Moralee, & Zorgios, 1998). The work of (Uschold, King, Moralee, & Zorgios, 1998:p.32) includes "[…] the purposes, the process and results of identifying and creating natural language definitions for all the terms and [their] experiences in converting these into formal definitions".

As shown in Table 2 the ontology differentiates five mayor sections:

- Activity, Plan, Capability, and Resource
  Activity is the first central term of the ontology. Activity gives the possibility "[…] to capture the notion of anything that involves actual doing, in particular including action. An activity can have happened in the past and may be happening in the present. (Uschold, King, Moralee, & Zorgios, 1998:p.43)"
- Meta Ontology and Time
  "The basic concept of the Meta-Ontology is ENTITY. This is in a sense the catch-all for all other concepts. (Uschold, King, Moralee, & Zorgios, 1998)"
- Organisation
  "Central to the Organisation section are concepts of LEGAL ENTITY and ORGANISATIONAL UNIT […]. Both of these refer to things which have a "gestalt", whether or not they are composite. (Uschold, King, Moralee, & Zorgios, 1998)"
- Strategy
  "The central concept of the Strategy section is PURPOSE. PURPOSE captures two related notions. One, is the intended reason for EXECUTING an ACTIVITY SPECIFICATION, i.e. what a PLAN is for. (Uschold, King, Moralee, & Zorgios, 1998:p.44)"
- Marketing
  "The central concept of the Marketing section is SALE. A SALE is an agreement between two LEGAL ENTITIES for the exchange of a PRODUCT for a SALE PRICE. Normally the PRODUCT is a good or service and the SALE PRICE is

monetary, however other possibilities are included. (Uschold, King, Moralee, & Zorgios, 1998:p.46)"

ACTIVITY etc.	*ORGANISATION*	*STRATEGY*	*MARKETING*	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	

Table 2: Overview of the Enterprise Ontology (Uschold, King, Moralee, & Zorgios, 1998:p.41)

2.4.3.1 Critique from literature

(Born et al., 2008) complains that an efficient application of the Enterprise Ontology would be hampered due to the fact that the ontology was fist modelled in a natural language and the transformed into a semi- formal Ontolingua. And (Grüninger, Atefi, & Fox, 2000:p.387) remarks that "[…] the Enterprise Ontology is not completely axiomatized within logic, it cannot be used to support automated reasoning".

2.4.4 A Context-Based Enterprise Ontology

(Leppänen, 2005; Leppänen, 2007) proposes a context-based enterprise ontology which provides "basic concepts for conceiving, structuring and representing things within contexts and/or as contexts". It extends a core (upper) ontology which includes several ontologies with a 'centered' semiotic ontology (Leppänen, 2005). Based on relevant theories of meaning, semiotic, semantics[5], pragmatics[6], the activity theory[7] and some contextual approaches, Leppänen proposes seven contextual domains (see Figure 10):

- Purpose - these concepts refer to the goals and reasons of someone or something.
- Actor - human and other active parts (but not tools which belong to the facility domain), which act in a context.
- Action - refers to events, operation and deeds which belong to an action structure.
- Object - distinguishes between material and informational aspects, where the latter are of interest. It can be a message, a decision, a program code, etc.
- Facility - something which is used for an action, a tool or a resource.
- Time - refers to temporal aspects in context.
- Location - parts of space used by someone or something and can be physical or logical.

Leppänen calls it the seven S's scheme: For Some purpose, Somebody does Something for Someone, with Some means, Sometimes and Somewhere.

---

[5] Semantics is "the philosophical and scientific study of meaning in natural and artificial languages." Encyclopædia Britannica, 2010. Web. 6 Mar. 2010 http://search.eb.com/eb/article-9110293

[6] Pragmatics is "the study of the use of natural language in communication; more generally, the study of the relations between languages and their users. It is sometimes defined in contrast with linguistic semantics." Encyclopædia Britannica, 2010. Web. 6 Mar. 2010 http://search.eb.com/eb/article-9473100. "Pragmatics studies how people comprehend and produce a communicative act or speech act in a concrete speech situation which is usually a conversation." (Shaozhong, 2000)

[7] The relationship between human agent and objects of environment is mediated by cultural means, tools and signs. The object of activity theory is to understand the unity of consciousness and activity. Web. 6 Mar. 2010 http://carbon.ucdenver.edu/~mryder/itc_data/act_dff.html

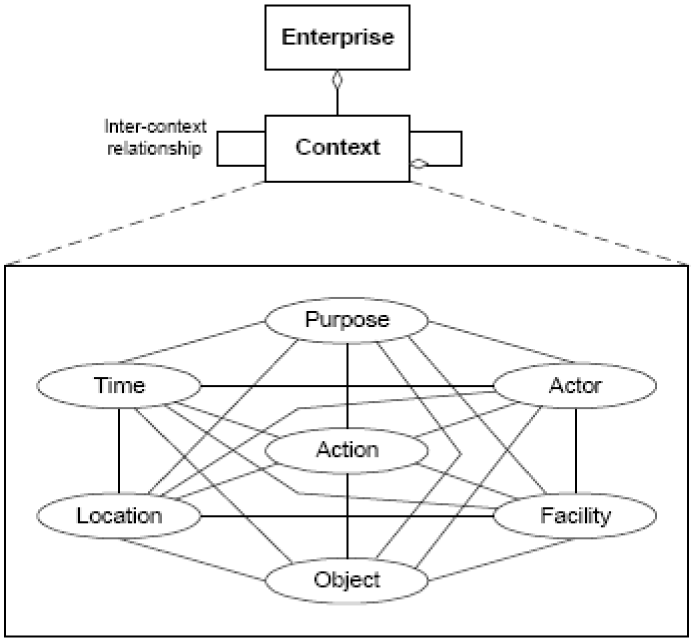

Figure 10: An overall structure of the context- based enterprise ontology (Leppänen, 2005:p.18)

It should serve as a groundwork for the specification, analysis and integration of more specific views.

(Leppänen, 2007:p.273) argues, that human beings must "need to know about contexts where the things appear, have appeared, and/or are to be appeared, and also about the things related to them in those contexts" and therefore context plays an important role. According to Leppänen, context has mainly been ignored in existing enterprise ontologies.

Furthermore, (Leppänen, 2007:p.274) states that "a context is a whole, composed of things connected to one another with contextual relationships. A thing gets its meaning through the relationships it has with the other things in that context." The approach implements this statement by relating the seven domains with each other, shown on Figure 11. With these binary and n-ary inter-domain relationships the things get embedded into their contexts.

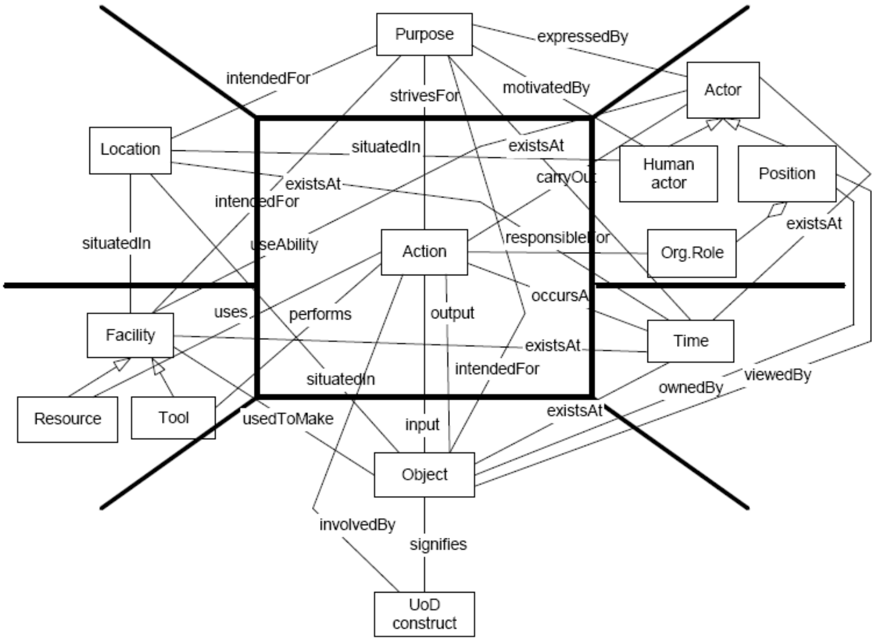

Figure 11: Overview of inter-domain relationships (Leppänen, 2005:p.22)

All in all the approach could be used as a frame to analyse other ontologies with respect to context, especially as "it is surprising how ignored a contextual view is in current enterprise ontologies." (Leppänen, 2005:p.17).

2.4.4.1 Critique from literature

(Filipowska, Kaczmarek, & Markovic, 2008) state that to the best of their knowledge, no commonly accepted model (among which they also count Leppännen's approach) could be reusable for various domains regarding organisational ontologies. They propose the need of a coherent ontology stack which provides the necessary semantics, considers flexibility and is extendible. They propose a modular approach for ontology development (consisting of an organisational structure ontology, organisational units ontology, business roles ontology, etc.) instead of a single organisational ontology. However, Filipowska and colleagues focus on the organisational domain and not the whole enterprise. Nevertheless they indicate some vacancy in the objective of being a groundwork covering the whole enterprise. Regarding a modular approach it can be commented that Leppänen actually follows a such one by proposing different domain ontologies. Furthermore Filipowska et al. assert that the approach of Leppänen is among others not fully consistent and operational. What certainly can be agreed, is that no implementation is available for Leppännen's approach.

Although it was not the intention of Leppänen to develop an operational approach, (Borch & Stefansen, 2004) state that it is not feasible to neglect operational issues as for operational purposes ontologies tend to over-abstract. Real-world scenarios should be examined and realistic applications implemented. Among others, (Staab, 2002) states that UML - in which Leppänen's approach is modelled - is helpful in ontology development especially when discussing issues with domain experts. Nevertheless it can not be used to reach the objective of a formal ontology as it has to be translated again; integration of UML into ontological tools is still in a development stage (Gómez-Pérez, Fernández-López, & Corcho, 2004).

2.4.5 ContextOntology

The approach of (Thönssen & Wolff, 2010) consists of seven areas in a context model called ContextOntology. The seven areas are business motivation, organisational structure, IT infrastructure, process knowledge, domain knowledge, business rules and information objects. The approach has been developed based on literature review about context and the identified 'six dimensions of change' which are dynamism, adaptability, flexibility, awareness, treasuring and gardening to meet the requirements for context formalization. As shown in Figure 12, different areas are defined and all are stored in an ontology.

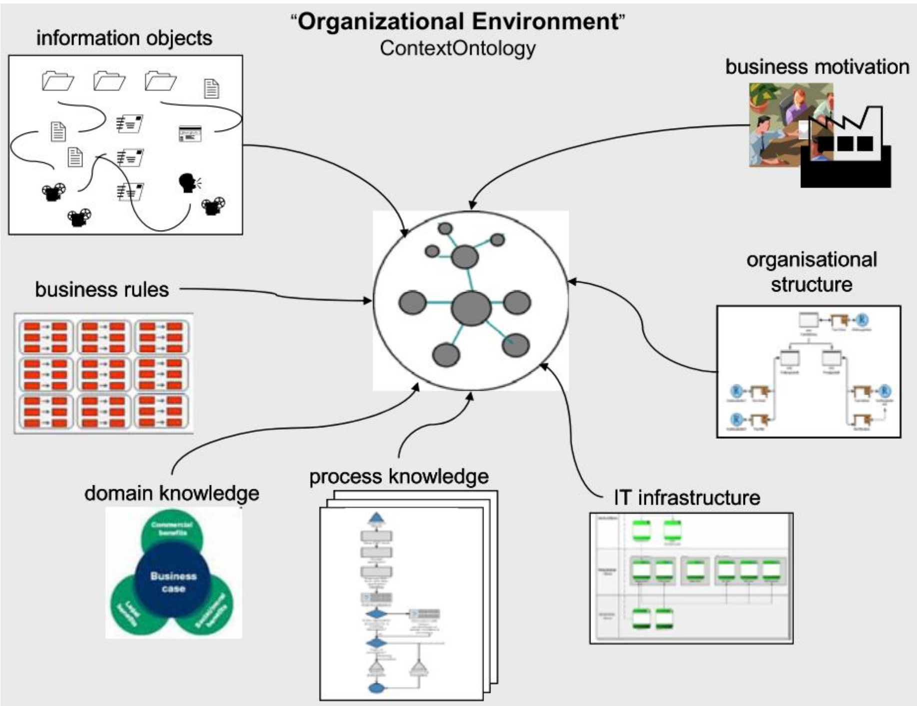

Figure 12: The seven areas of the ContextOntology (Thönssen & Wolff, 2010:p.9)

For each area different levels of abstraction have to be considered. The ReferenceContext, representing the meta model of a model type, the RealizationContext representing the model itself and the ManifestationContext representing the instances of it.

For the approach, the modelling framework ATHENE has been used to implement parts of the ContextOntology and relations among the elements.

2.4.5.1 Critique from literature

The approach has just recently being published, therefore no critique could be found. But the following critiques also hold for the ATHENE ContextOntology. It has not been proved in a practical environment and is at the current state not an 'out-of-the-box' solution as (Hanschke, 2010) claims to be one. Also models for process knowledge, business rules, organisational structure, IT system models and a business motivation model have been implemented, the requirement of being operational (Filipowska, Kaczmarek, & Markovic, 2008) is not fulfilled yet. In the description of the approach an explanation of the "domain knowledge" was not found. Likewise a description about the relations among the areas is not given (but intended).

2.5 Information Management Systems

The knowledge of an enterprise must be captured somewhere. Today a lot of knowledge is still encoded in algorithms and data structures but knowledge should be consistently managed by explicitly and formally describing it. Enterprises have built information management systems (IMS) to systematically structure and store their knowledge. Knowledge engineers try to make this hidden knowledge accessible to the computer but also closer to human languages by using knowledge-based systems (Sowa, 2000). IMS give an enterprise means for information and knowledge management, which try to order and structure their knowledge in order to tackle typical knowledge management problems: How to manage information sources? How to represent knowledge? How to find knowledge? How to make employees aware of knowledge? How to continuously maintain and mature knowledge?(Rehäuser & Krcmar, 1996).

Following (Krcmar, 2004), IMS serve as connective link between the business and the IT and supports business process management by building easily accessible and readable information out of weakly structured information in a consistent way. They enable a flexible access to information and a demand meeting assistance for each employee by offering different viewpoints. In order to achieve these requirements, information resources must be managed, maintained and meta-data management is needed. There exist different implementation approaches for the modelling and organisation of an IMS. These are entity-relationship models, data base management systems, taxonomies, thesaurus or ontologies, just to mention a few.

The requirements to an IMS can be summarised as follows: "Supplying the decision makers with relevant information, guaranteeing high information quality […] and organizing the information management as a cross-sectional function of the enterprise" Translated from (Krcmar, 2004:p.51).

2.5.1.1 Design criteria enhancements from IMS

IMS have the following requirements which need to be fulfilled by an enterprise ontology.

- Different viewpoints for different employees accessing the information are needed
- Information that is retrieved must be relevant

2.6 Conclusion

The literature review shows that there exist multiple enterprise architecture frameworks. The EA frameworks have in common that they are described through various perspectives and aspects, whereas the latter ones are in many cases based on the Zachman Framework, trying to cover the whole enterprise. The EA frameworks have in common that they are described through various views and aspects, whereas the latter ones are in many cases based on the Zachman Framework. Most of them are highly abstract and somehow remote from real-life practice. Agility and ad-hoc use is not given. Beside them, various approaches of enterprise upper ontologies have been proposed. Information systems store and managed the data and are often used in conjunction with an EA solution.

Most of the enterprise architecture frameworks are too rigid and inflexible to support an agile enterprise. They have clear but complex procedures how an enterprise architecture has to be built and structures how an enterprise should be represented. But as perceived in the interviews, contemporary enterprises model their specific enterprise architecture with which they identify themselves. Some of them combine multiple approaches and even integrate enterprise specific concepts. This might be an obstacle for reusability, e.g. by a merger of two enterprises. However, if these two enterprises use the same general concepts, but still having their specific enterprise architecture, reusability is still guaranteed. The use of an (enterprise) upper ontology (see section 5.5) and an application ontology offers a way to distinguish between the two distinct 'worlds'. By relating different approaches to the same upper concepts, an enterprise is able to combine multiple approaches. The various enterprise architectures, upper ontologies propose different aspects and the interviewed enterprises even again use specific ones. The list of aspects proposed for the enterprise upper ontology considers the differents viewpoints (see section 5.5), integrates them and is still extensible.

With exception of the Enterprise Architecture Framework of plugIT (Wache, Eichner, Koutsoukou, & et.al., 2009), the relations among the domains of enterprise architectures are only implicitly given, but not explicitly described. Missing or unclear relations make an architecture a loosely construct with a missing business - IT alignment. As it is the intention of ontologies to show the relations among concepts, they are described in all mentioned enterprise upper ontologies. Therefore these approaches have mainly been considered in the development of the approach.

A missing linking between business and IT also leads to an incomprehensible overall architecture, not understandable for all stakeholders. The nonexistence of views and the nonadaptability of frameworks are two reasons for a lack of understanding. Another is the use of natural language. The lack of semantics causes communication problems between humans, between systems or between human and system. (Kang, Lee, Choi, & Kim, 2010), there is no common under-standing of the application domain (Bertolazzi, Krusich, & Missikoff, 2001). This problem is often not handled by enterprise architectures but by upper ontologies. Therefore an enterprise upper ontology has been developed which considers different views, which is adaptable and forces semantic descriptions.

The already conducted research in upper ontologies covers a substantial area of knowledge, however its usefulness is limited as the results are not serialized in any contemporarily recognized ontology language standard (Janusch et al., 2008). Some of the proposed enterprise upper ontologies have not been implemented in an ontology language yet and only parts of the Enterprise Ontology (Uschold, King, Moralee, & Zorgios, 1998) and of TOVE (Fox, Barbuceanu, Grüninger, & Lin, 1995) have been applied according to literature. However, EA solution vendors like (TopQuadrant Inc., 2009) claim that there is a need for formal representations of an enterprise architecture in an ontology. Implementing the enterprise architecture in an ontology language also fulfils the requirements from information systems (see section 2.5). Ontologies are proposed as one possibility of managing data and fulfil the need of information systems.

Hence the enterprise ontologies are able to cover additional requirements not covered by enterprise architecture but fall short in other aspects, e.g. the comprehensiveness. This is where the LEMO approach fits in, combining the advantages of enterprise architectures and enterprise ontologies. It has the demand of being a holistic basis for an enterprise repository covering aspects and perspectives as applied by the more comprehensive enterprise architectures. The approach is implemented as an ontology in a contemporary modelling language; relationships among the enterprise objects can be simply modelled and the ontology is easily extensible. Having such relationships implemented leads to enterprise being able to react much faster as needed actions for a change can be pinpointed. Using RDF(S) as modelling language leads to a formal (and explicit) description of the enterprise. Considering multiple level/types of the ontology leads to reusable enterprise models. Including perspectives as enterprise architectures do, assist a common understanding about the enterprise and implementing it as an ontology forces a common understanding. Using the technology of ontologies enables and even more detailed and adopted view on the enterprise. Furthermore by using an ontology the approach is flexible also assisted by the different levels/types of ontologies used. The LEMO approach is introduced and described in chapter 5 (Linked Enterprise Models and Objects). The next chapter introduces the 3 Research methodology and design for this work.

3 Research methodology and design

The purpose of this Chapter is to introduce the research methodology and design at the thesis at hand.

3.1 Introduction

According to the thesis statement it is assumed that the context and content of an enterprise can be represented in an enterprise ontology including the relations, general concepts, model elements and model types. Finally the thesis assumes that the ontology can be used as knowledge base to create metadata. To "prove" this thesis statement the appropriate research methodology has been chosen. The enterprise ontology should reflect the theory on one side and practice on the other side. Back to research design an inductive and deductive approach needs to be defined. The detailed definitions were made according to (Saunders, Lewis, & Thornhill, 2007). They provide the theoretical background for this work. The sub-goals (SG) of the thesis (see section 1.3.3 (Thesis statement)) provide hints for choosing the right research methods:

- SG 1 & 2 say that the enterprise ontology should consider the content and context of an entire enterprise. This will be elaborated by doing literature review as a theoretical basis.
- SG 3 tries to put the thesis additionally into a practical view. The enterprise ontology should reflect also the practice by conducting interviews.
- SG 4 & 5 give hints about the evaluation of the approach. The enterprise ontology should be evaluated based on a simple application scenario and implemented in a demonstrator.

3.2 Research design

Research design describes the used research philosophy, methodology and approach including how the data has been collected and analysed.

The research ‘onion’ (see Figure 13) from (Saunders, Lewis, & Thornhill, 2007) gives a nice overview over the different aspects of a research design and the possible philosophies, approaches and strategies. The research ‘onion’ introduces layers, starting from the philosophy, over the approach, the strategy, the choices, the time horizons to the research techniques and procedures.

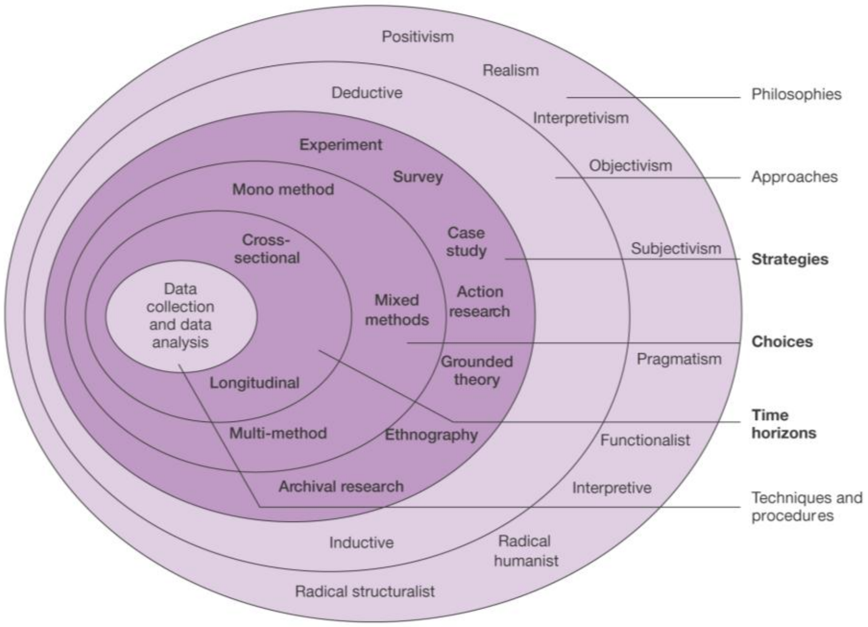

Figure 13: The research ‘onion’ (Saunders, Lewis, & Thornhill, 2007)

The research ‘onion’ gives the toolset to answer the research questions and elaborate the findings according the thesis statement and sub-goals. According to that, the thesis statement delivered the requirements to the appropriate research philosophy, research approach and research strategy.

3.2.1 Research philosophy

According to (Saunders, Lewis, & Thornhill, 2007) there are three major ways of thinking about research philosophies:

The first one is epistemology. “[It] concerns what constitutes acceptable knowledge in a field of study.” (Saunders, Lewis, & Thornhill, 2007). It contains positivism, realism and interpretivism. *Positivism* deals with the social reality, meaning that a result of this approach “[…] can be law-like generalisations similar to those produced by the physical and natural scientists”(Remenyi & Williams, 1998:p.32). The *realism* philosophy defines “that what the senses show us as reality is the truth. […] The theory of realism is that there is a reality quite independent of the mind." (Saunders, Lewis, & Thornhill, 2007). The last research philosophy of epistemology is *interpretivism*. Interpretivistc researchers argues that "[…] the social world is far too complex to lend itself to theorising by definite 'laws' the same way as the physical sciences, […] it is necessary for the researcher to understand differences between humans in our role as social actors. This emphasises the difference between conducting research among people rather than objects such as trucks and computers." (Saunders, Lewis, & Thornhill, 2007:p.106).

The second one is the ontology and it "[…] is concerned with nature of reality. To a greater went than epistemological considerations, raising questions of the assumptions researchers have about the way the world operates and the commitment held to particular views." (Saunders, Lewis, & Thornhill, 2007:p.108). One aspect of ontology is *objectivism*. "This portrays the position that social entities exist in reality external to social actors concerned with their existence." The second aspect is *subjectivism*. "[It] holds that social phenomena are created from the perceptions and consequent actions of those social actors concerned with their existence." (Saunders, Lewis, & Thornhill, 2007:p.108).

It is often hard to decide in which research philosophy a whole research work, a thesis etc. takes place and "the debates on both epistemology and ontology have had a competitive ring to them. The debate is often framed in terms of a choice between either the positivist or the interpretivist research philosophy." (Saunders, Lewis, & Thornhill, 2007:p.120). It is in practice unrealistic to choose one approach for an entire work. *Pragmatism* argues that the most important determinant of the research philosophy adopted are the research questions. "One approach may be 'better' than the other for answering particular questions." (Saunders, Lewis, & Thornhill, 2007:p.110).

The last one is Axiology as "[…] a branch of philosophy that studies judgements about value. Although this may include values [...] in the fields of aesthetics and ethics, it is the process of social enquiry […]."(Saunders, Lewis, & Thornhill, 2007:p.110).

Applied research philosophy for the thesis:

Based on the thesis statement and sub-goals it can be seen, that it is obvious hard to decide for a particular research philosophy. Figure 14 tries to give a basis for decision about the research philosophy. The figure from (De Villiers, 2005) is adopted in a way that it shows the used research methods based on the thesis sub-goals.

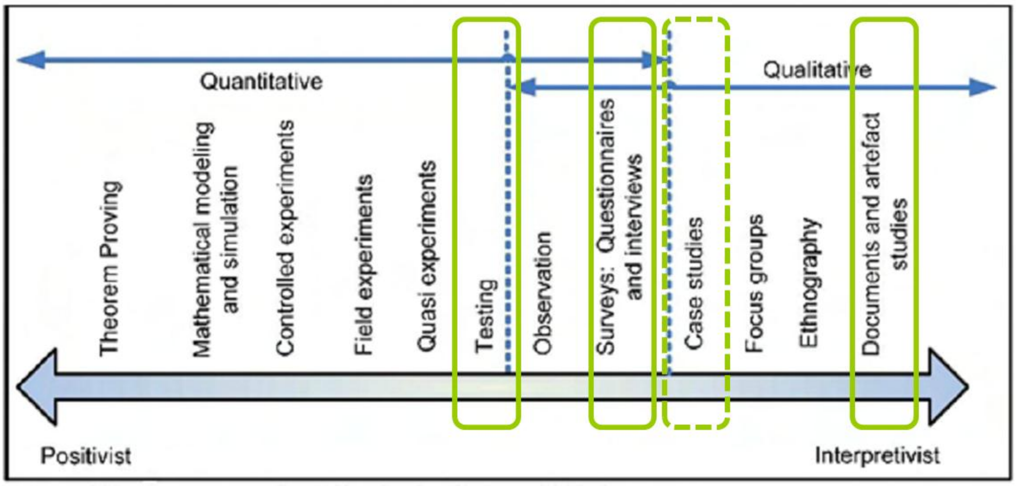

Figure 14: Research methods / strategies [adapted from (De Villiers, 2005)]

It can be seen that a final decision between positivism and interpretivism cannot be done for the entire work. Therefore the pragmatism points a good way out of the dilemma. Pragmatism says that every subgoal of the thesis statement can follow independent research philosophies. It is therefore obvious that this thesis follows pragmatism as research philosophy.

3.2.2 Research approach

Back to (Saunders, Lewis, & Thornhill, 2007), they distinguish between two research approaches. The deductive as testing theory starts with deducting a hypothesis, expressing the hypothesis in operational terms and testing it, examining the specify outcome and if needed modifying the theory (Robson, 2002). The inductive approach has its origin in social sciences. It begins with an observation with the goal to identify patterns based on cause effect relationships. The data collection could be done for example by making interviews. After making this data collection the analysed data will be used to formulate a hypothesis and ending up with a theory.

<u>Applied research approaches:</u>

Again considering the thesis statement and the sub-goals we are able to define the applied research approaches. The thesis says that theory work, interviews and a confirmation by a demonstrator will be done. Mapping that to the research approach it can be seen that the thesis uses inductive research (making interview) and deductive research (theory work and confirmation with demonstrator) as well.

3.2.3 Research strategy

The research 'onion' lists the following research strategies: experiment, survey, case study, action research, grounded theory, ethnography and archival research. Of course this is not an all-embracing list and the following word will only describe the most promising ones for this thesis.

Action research is defined by (Avison, Lau, Myers, & Nielsen, 1999:p.94) as following: "Action research combines theory and practice (and researchers and practitioners) through change and reflection in an immediate problematic situation within a mutually acceptable ethical framework". (Saunders, Lewis, & Thornhill, 2007) identified four common sub- themes of action research. First, it is research in action and conducted with those who experience the issues directly. Secondly, there exists a partnership between practitioners and researchers whereas the research can act as internal or external consultant. Thirdly, an iterative process of diagnosing, planning, taking action and evaluating is taken place described in the action research spiral (see Figure 15). Fourthly, the gained knowledge should be used to inform other contexts (e.g. in the organisation).

(Robson, 2002:p.178) defines *case study* as "[…] a strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon within its real life context using multiple sources of evidence". (Saunders, Lewis, & Thornhill, 2007), sees case study as an explanatory and exploratory research using various data collection techniques. A case study gives the ability to generate answers to the question 'why? as well as the 'what?' and 'how?' questions.

Experiment can be seen as a "[…] classical form of research that owes much to the natural sciences." (Saunders, Lewis, & Thornhill, 2007:p.136) Experiments tend to be used in exploratory and explanatory research and tries to answer the 'how?' and 'why?' questions (Saunders, Lewis, & Thornhill, 2007). (Saunders, Lewis, & Thornhill, 2007) give a summery what experiments could involve typically: Experiments will start with a definition of a theoretical hypothesis. A selection of samples from known populations will be allocated to conditions, the experimental group and the control group. The variables will then be manipulated and controlled.

Applied research strategy:

Based on the pragmatic research philosophy the thesis- sub-goals (SG) decides for the research applied strategy.

- SG 1 & 2 say that the enterprise ontology should consider the content and context of an entire enterprise. SG 3 tries to put the thesis and the enterprise ontology additionally into a practical view. As mentioned this will be done by literature review and conducting interviews.
- SG 4 & 5 say that the enterprise ontology should be evaluated based on a simple application scenario and implemented in a demonstrator.

The sub-goals are mapped to the research strategies as follows: All sub-goals could be reached by using the action research strategy as an iterative approach. In principle it is possible to elaborate the key findings and results in a close partnership with practitioners. But early discussions with the application partners show that it is not possible to work during the whole time together – because of the time frame and the available resources. But nevertheless a part of the thesis could be done in action research manner – SG 1 to SG 3 and slightly SG4 will be done in cooperation with different companies. This will mainly be done by conducting interviews, considering literature review and retrieving feedback from interviewees including hints for creating an application scenario. Even this mentioned application scenario (SG 4) can be seen as a part of a case study approach. The application scenario should reflect a real life case and should serve the basis for evaluating the entire approach. Of course an application scenario is only a part of a big case study, but nevertheless it will provide a basis for the demonstrator (SG5). This demonstrator can be seen as an experiment approach where the variables, in this case the variables could be seen as the ontology, will be manipulated using rules.

3.3 Methodology

3.3.1 Data collection

The data collection of this work includes a literature survey as secondary data on the one side and interviews as primary data on the other side.

In order to get secondary data and an overview about current research state, different research approaches concerning enterprise architectures and ontologies has been identified. Six enterprise architecture frameworks (ARIS, BPMS, Best Practice Enterprise Architecture, NGMF, TOGAF and Zachmann), four upper ontologies developed for enterprise architectures (TOVE, Enterprise Ontology, Context-Based Enterprise Ontology, ContextOntology) and information management systems have been analysed. Various other approaches are integrated in the critique sections of each description.

The primary data has been gathered by conducting interviews at three different companies dealing with metadata. These companies were a consulting company and two software providers one from the long term archiving word and to other working in the document and contract management sector.

3.3.2 Research instruments and methods

As described in the research strategy, the **action research strategy** will be applied which can be pictured in an action research spiral as described in (Saunders, Lewis, & Thornhill, 2007). Each cycle consists of four steps: Diagnosing (data gathering, fact finding and analysis), Planning, Action taking and Evaluation. Subsequent cycles take previous evaluation into account. The following steps follow the action research spiral as defined in Figure 15:

The 'Context and Purpose' at the beginning is equatable to the thesis statement. The following steps are performed.

1.1 Getting an overview of existing literature based on the thesis statement and sub-goals.

1.2 Choosing relevant literature about enterprise architecture (frameworks), upper ontologies, information systems and metadata.

1.3 Analysis of the selected enterprise architectures (frameworks), enterprise ontologies and information systems.

1.4 Writing of critique and development of the ontology design criteria list.

Outcome: a) Overview of existing literature, b) a first version of the design criteria list and c) requirements and aspects which are needed for the enterprise model ontology.

2.1 Finding of interview partners (inductive approach).

2.2 Planning of interviews and set up of a list of questions based on the knowledge gained in the first cycle but also independent from any theory.

2.3 Conducting interviews.

2.4 Affirmation of the gathered information from the interviews in a second meeting if possible, enlargement of the requirements to the enterprise model ontology and application scenario. Development of the enterprise model ontology in an ontological way.

Outcome: a) Insight into practice, b) an updated design criteria list, c) a first version of the enterprise model ontology and e) basis for the application scenario.

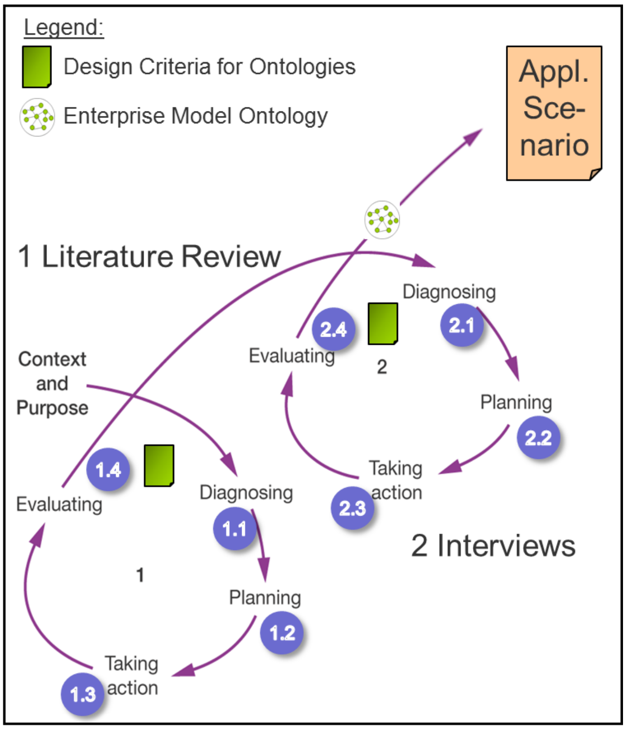

Figure 15: The action research spiral [adapted from (Saunders, Lewis, & Thornhill, 2007)]

The **application scenario strategy** (case study strategy) will be applied using the results from action research phase. If possible it is planned to get real data from interviewed companies as a basis for creating an application scenario which provides the possibility to evaluate the ontological approach. The application scenario should provide a plausible context when metadata could be created in an enterprise. It is clear that also assumptions and limitations need to be made in the scenario, but nevertheless it should reflect the real life as much as possible. To have the possibility to get real data it is needed to create the application scenario in annonymized manner. Therefore the application scenario will use fictive terms and additional concepts.

In the end the **experimental strategy** should give an answer if the introducd thesis statement can be fulfilled. The experiment will be done by describing and implementing a simple demonstrator which should serve the application scenario.

Finally it is possible to say that this thesis follows a multi- method approach doing literature survey (action research) and experiments (case study and demonstrator).

3.4 Summary

The thesis statement and sub-goals are the starting point of this document at hand. They define the adopted research methodologies and followed research approach. Figure 16 gives a nice overview of the adopted research methods and approaches in this thesis.

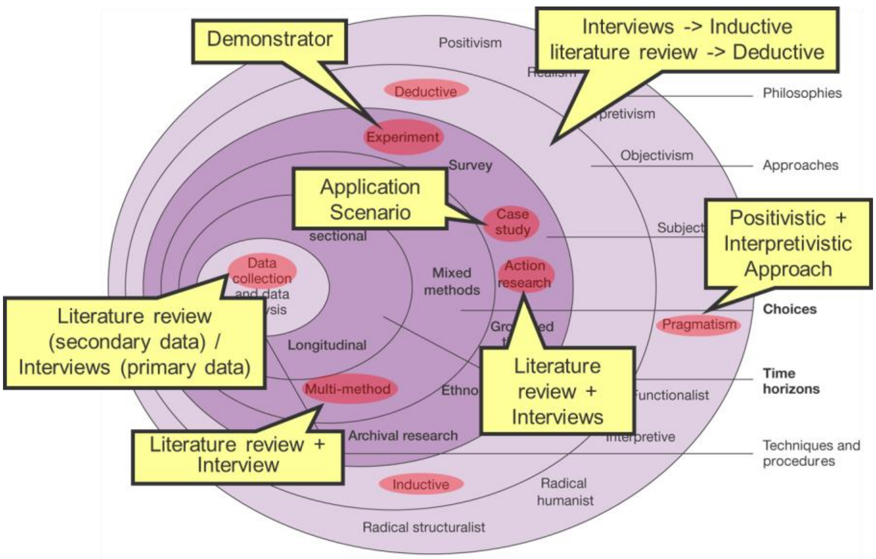

Figure 16: Overview of the used research methodology and design [adapted from (Saunders, Lewis, & Thornhill, 2007)]

As mentioned many times the thesis statement and sub-goals trigger the decision of the research methods and approaches. Based on the nice overview from (Saunders, Lewis, & Thornhill, 2007) it is possible to summarise the chosen approach as follows:

- The thesis follows the research philosophy based on pragmatism. This gives the possibility to define for every sub-goal its own research methodology and instruments.
- The thesis follows an inductive (making interviews) and deductive (theory work) research approach.
- The thesis chooses the following three research strategies: action research containing literature review (secondary data) and interviews (primary data), case study implemented as application scenario and experiment represented as software- demonstrator.

4 Conducted Interviews

The following interviews were made at an IT solution company, a consulting company and a software provider company. All companies dealing a lot with metadata.

The interview questions are based on the research questions and verbalized in a general way as follows:

- How to describe formally a whole enterprise?
- How to use an enterprise repository to create metadata?
- How to define good metadata?
- What is good metadata?
- What is a good metadata standard?

4.1 The interviews

4.1.1 IT solution and consulting company

scope Solutions AG is a producer of Archive Information Systems (AIS) and provides consulting services in records management. The company is located in Basel and has currently 15 employees. The main products are the archive information system scopeArchiv and scopeOAIS. The software provides the basis for a hybrid management of physical and digital records in an archive based on the ISO-20652[8] standard. scopeArchiv is currently used by around 70 customers with 1‘000 workplaces in several European countries.

The interview was done with Felix Akeret, CEO of the scope Solutions AG.

[8] ISO-20652: ISO-20652 is a standard of the International Organisation for Standardization (ISO). "[The standard] identifies, defines and provides structure to the relationships and interactions between an information producer and an archive. It defines the methodology for the structure of actions that are required from the initial time of contact between the producer and the archive until the objects of information are received and validated by the archive." (International Organization for Standardization (ISO), 2006).

Context: The context scope mainly deals with is the archiving business. And that is the focus of the interviewee's questions. scope lays its great emphasis on the two typical characteristics of archiving: durability and restorability.

Customer: The customers of scope are in general autonomous. Of course they have to fulfil some requirements from the business and government side, but they are often empowered to define their own archiving structure.

Product: scopeArchiv allows the entry into the OAIS conform digital archiving including full-text and descriptor searching. Paper and electronic files are managed in a single system and all common formats can be archived. Furthermore it is able to import data and metadata from other stems like DMS or AIS etc. One of the major features of the scope system is that the levels of description (department, fonds, series, dossier, document etc.) are freely definable and nameable. scope customers are able to define own field types which are subsumed in sets (also called forms). The customer is able to assign or structure the metadata in a hierarchical manner and define apart from a standard own field types.

Requirements from the customer: A typical scope customer wishes a standard on the one hand and the full flexibility to define own structures on the other hand. A customer wants to use a standard as a common sense approach in the beginning and the possibility to enhance the system with own definitions. Two major user groups have their own specific requirements to the system. An archivist is interested in a fast and economical recording of artefacts. The end-user is particularly interested in any exact and meaningful information and artefact retrieval.

Implementation on customer side: A customer without an existing archiving system will usually begin with using a standard. After some time the customer realizes that own definitions and structures apart from the standard are necessary. This is one point where scope can play its experience and the strengths of the system. Customers who want to make a shift to the scope system from an existing system are beholden to ascertain the existing structures in workshops. Usually the existing structure will be reproduced by own field types in the scope system. Several customers take the shift to a new system as a chance to revise and restructure the existing metadata according to a standard.

The importance of a standard: The AIS of scope is able to handle the major standards in the archiving industry. scope uses standards of the International Council on Archives (ICA)[9], ISAD(G)[10] and ISAAR(CPF)[11] quite often in their implementations. Apart from the classical and widely distributed ICA standards the customers of scope wishes the implementation of digital standards as the Encoded Archival Description (EAD)[12], METS[13] or Dublin Core[14]. EAD is one recommendation of the Swiss Office of Coordination KOST[15] for archiving related metadata like dossiers, documents and classification system. For technical related metadata like authenticity or technical source information, KOST recommends the use of the PREMIS[16] standard which is also supported by the AIS of scope. KOST tries to standardise the archiving landscape of Switzerland by giving recommendations on file formats and the men-

---

[9] ICA: International Council on Archives (ICA) a decentralized organisation governed by a General Assembly and administered by an Executive Board; ICA works closely with inter-governmental organisations such as UNESCO and ICCROM.

[10] ISAD(G): General International Standard Archival Description, it standard provides general guidance for the preparation of archival descriptions. “It is to be used in conjunction with existing national standards or as the basis for the development of national standards.” (International Council on Archives (ICA), 2000) ISAD(G) is standard of ICA.

[11] ISAAR(CPF): International Standard Archival Authority Record for Corporate Bodies, Persons, and Families, a standard of the ICA. “This standard provides guidance for preparing archival authority records which provide descriptions of entities (corporate bodies, persons and families) associated with the creation and maintenance of archives.” (International Council on Archives (ICA), 2004)

[12] EAD: Encoded Archival Description is an standard based on a Document Type Definition (DTD) for encoding archival finding aids using Extensible Markup Language (XML). The standard is maintained by the Library of Congress (LC) in partnership with the Society of American Archivists. (Encoded Archival Description Working Group, 2003)

[13] METS: “Metadata Encoding and Transmission Standard (METS) is a data encoding and transmission specification, expressed in XML, that provides the means to convey the metadata necessary for both the management of digital objects within a repository and the exchange of such objects between repositories (or between repositories and their users). […] The METS XML schema was created in 2001 under the sponsorship of the Digital Library Federation (DLF).” (Digital Library Federation, 2010:p.15)

[14] Dublin Core: Dublin Core (DC) provides a schema of metadata elements through which resources (documents, books, videos, images, etc.) can be described. The creator of these schemas is the Dublin Core Metadata Initiative (DCMI). DCMI is an open organization, incorporated in Singapore as a public, not-for-profit Company. (Dublin Core Metadata Initiative, 2010)

[15] KOST: German abbreviation for office of coordination for permanent archiving for electronic documents (Koordinationsstelle für die dauerhafte Archivierung elektronischer Unterlagen).

[16] PREMIS: “The PREMIS (Preservation Metadata: Implementation Strategies) working group, comprised of international experts in the use of metadata to support digital preservation activities. […] First and foremost is the Data Dictionary, a comprehensive, practical resource for implementing preservation metadata in digital archiving systems.” (PREMIS Editorial Committee, 2008:p.1)

tioned metadata standards. scope is able to handle this standard and has experience in implementing such a standard. But from the scope's experience from customer side, it is possible to draw a clear picture: most customers of scope are willing to use a standard but on the other side they do not want abdicate from defining own structures and field types.

Vision: scope depicts the vision of a document or artefact which is somehow the central entry point to entire knowledge of an organisation. Further on, scope expects the usage of standards as in the past. Some customers are willing or obligated to follow a standard entirely and some customers can't do without their own defined adaption to a standard or own field types and structures.

4.1.1.1 Key findings

1. The possibility for defining domain and enterprise specific metadata types and structures is an important aspect.
2. Standard plays its role but not the intended role – a standard is often only used as template. Some companies follow a standard and some companies are not willing to follow any standard.
3. There are several standards on the market. There exist several initiatives to implement a standard without resounding success – there is no right one.
4. The question what good metadata is cannot be answered in general. Considering the scope's context, each customer has its own definition of good metadata. Some customers consider the strict compliance of a standard; others follow their own perceptions of good metadata.

4.1.1.2 Derived requirements for thesis

1. The thesis should elaborate a basic structure based on literature review without making a claim to be all-embracing.
2. The thesis should consider the need for creating domain and enterprise specific concepts and relations.
3. There is no need to figure out the so-called best metadata standard or enterprise architecture rather giving the possibility to include different standards and architectures in a generic way.

4.1.2 Consulting company

BCP Business Consulting Partner AG is a consulting company focussing on information technology, project- , process- and strategic- management. The company is located in Basel and has about 18 consultants.

The interview was done with Hanspeter Hebeisen a consultant with many years of experience in different fields of IT and management. The interview does not only consider one particular project of the BCP AG it is based on the experience of the consultant concerning metadata management, analysis and gathering including management and operational aspects. The provided information, experience and examples of BCP customers are confidential. Therefore this interview will only provide a generalized view.

Customers: The customers of BCP are distributed in several industry sectors like private sector, governmental sector and health care.

Services: The services of BCP starting from IT related services like: IT strategy, evaluation and implementation of IT solutions, IT system integration, data warehousing; over business related services like: business process management, strategy management and project management in general.

Requirements concerning metadata: Based on the consulting experience of the interviewee four major requirements to metadata can be derived – data quality, metadata expressiveness and processability, homogeneity and the possibility of an improvement- process implementation. Data quality is an important issue when setting up a new project – problems like redundant data and wrong relationships occurs in nearly all legacy systems. But also new implementations are not secure of having insufficient data quality. Expressiveness and processability is another key issue in a metadata related project. Good expressiveness and processability leads to good accuracy in information retrieval. Homogeneity and unity is the third key issue. It is not in every case possible to follow a predefined structure – e.g. time pressure can lead the user of a metadata system to non- adherence of the given structure. The fourth important requirement is the need of continuous improvement- process. A Metadata schema needs care; therefore it is needed to define in advance an improvement process based on data mining approaches and predefined “improvement” rules.

Several systems provide mechanisms which try to avoid redundant data. But there is still the metadata schema which defines how the data is structured and in most of the cases this schema is customer specific. This means that the schema will be defined by the customer – sometimes a standard is required by law or regulation, but it is mostly the case that the customer is the one who decides how a schema should look like. Of course the role of the consultant is to give advices how a schema could look like and a good consultant should have the experience to give the costumers the security that they follow the right strategy, use the right tools and follow the right methodology. But in the end it is in the majority of the cases the specific project with the specific requirements and the customers' needs who "defines" the data schema.

Metadata identification: In the majority of the projects the business process of the company delivers the entry point for metadata creation. This means that a project starts with the analysis of the process landscape if available. If not available, in most of the cases the processes will be somehow captured. Because a consultant and the customer should have a common understanding about what is actually done in the company. This gives the basis for a deeper analysis about the data needed during the relevant processes.

Metadata gathering: There are usually two ways for starting the metadata gathering task. If there is already metadata available in the company, the existing data needs to be analysed during a workshop. In most of the cases some parts of the structure needs to be revised during iterative sessions doing analysis, definition and review. If there is no metadata available a customer can start with a standard if available and if it suites to the customer's needs. But in most of the cases a standard does not suite the customers' needs or the customer is willing to define its own metadata definition from scratch. In most of the cases a thesaurus is a good starting point for defining a metadata schema. This thesaurus will be created during several brainstorming sessions in consideration of existing data if available.

4.1.2.1 Key findings

1. Metadata quality, metadata expressiveness and processability, metadata homogeneity and a metadata improvement process are the main metadata requirements.

2. In most of the cases a customer defines its own metadata schema – a standard is often only used as a template. It is the customers' needs and its specific requirements who "define" the metadata schema.
3. Business processes are the "entry point" to the relevant metadata.

4.1.2.2 Derived requirements for thesis

1. The thesis should elaborate a basic structure but not a standard, which is all-embracing.
2. Again, there is no need to figure out the so-called best metadata standard rather giving the possibility to include different standards in a generic way.
3. The thesis should consider the business process methodology.

4.1.3 Software provider company

Symfact AG is a software provider for Contract Lifecycle Management (CLM).

The interview was done with Andreas Kyriakakis, CEO of the Symfact AG. The company is located in Sugiez, Switzerland and has several offices in the USA and UK. Symfact has about over 100 customers in 20 different countries.

Customers: The customers of Symfact are located in different sectors like real estate, chemical and life sciences, retail and consumer, services, energy, finance, manufacturing, communication, insurance and administration.

Products: Symfact provides a single technology platform for several business tasks. ContractX provides the possibility to author, approve and manage contracts. ConceptX manages and optimises intellectual property, like brands, patents, logos and copyrights. MonitorX provides corporate housekeeping. ControlX provides the management of documents and relationships for the SME/SMB. Apart from the standard product Symfact provides also customer specific solution based on the single platform. Symfact sees its unique feature of its technology platform that it is completely XML based. Metadata can be seen as one of the main "information" in the products of Symfact. For example the product ContractX manages the data of a contract using metadata definitions.

Metadata identification: Symfact identifies metadata based on the documents of the customer. This will be usually done by analysing the actual documents and highlight the potential metadata. After creating a potential set of metadata, Symfact is organizing several workshops with the customer. During the workshop they decide which metadata should be implemented in the software. Apart from doing analysis based on the actual contract, it can be the case, that the customer provides a strict requirement specification. It can happen that a customer needs to follow a standard by law or regulation, but it is mostly the case that the customer decides how a metadata schema should look like without considering a standard. Quite often it is the case that Symfact provides their own best practice metadata schema which will be modified by the customer during workshops. An important aspect for the metadata set is the actual type of the document – Symfact is able implement different metadata sets for different system serving different purposes.

Metadata gathering: After identifying the metadata the data will be entered into the system. This will be done using data import approaches. It is also possible that some data needs to be entered by hand when the needed information is not electronically available.

4.1.3.1 Key findings

1. In most of the cases a customer defines its own metadata. It is the customers' needs and its specific requirements who "define" the metadata schema.
2. The actual document type defines the needed metadata.

4.1.3.2 Derived requirements for thesis

1. Again, the thesis should elaborate a basic structure and not standard, which has to be all-embracing.
2. Again, there is no need to figure out the so-called best metadata standard rather giving the possibility to include different standards in a generic way.
3. The thesis should consider the possibility to create several metadata sets for several document types.

4.2 Summary

4.2.1 Summary of the key findings

The first interesting finding is that the need for defining domain and enterprise specific metadata types and structures is seen as an important aspect. And the view, that business processes are the "entry point" to the relevant metadata confirms the assumption the processes and tasks are an important aspect for identifying metadata.

The question what good metadata is cannot be answered in an easy way. In most of the cases, the customer has its own definition of good metadata. Some customers consider the strict compliance of a standard; others follow their own perceptions of good metadata. But there are some general requirements concerning metadata. Metadata quality, metadata expressiveness and processability, metadata homogeneity and a metadata improvement process are the main metadata requirements.

The question if there is the so called best metadata standard available can be answered as follows: There are several standards on the market. There exist several initiatives to implement a standard without resounding success there is no right one.

The last interesting finding is that every document type "defines" its own metadata set. That means that the interviewed company uses different metadata sets for different document types.

4.2.2 Summarised requirements for the thesis

1. The thesis should elaborate a basic structure based on literature review without making a claim to be all-embracing.
2. The thesis should consider the need for creating domain and enterprise specific concepts and relations.
3. There is no need to figure out the so-called best metadata standard or enterprise architecture rather giving the possibility to include different standards and architectures in a generic way.
4. The thesis should consider the business process methodology.
5. The thesis should consider the possibility to create several metadata sets for several document types.

5 Linked Enterprise Models and Objects

The Linked Enterprise Models and Objects (LEMO) approach is a new way to view an enterprise. It leads to a view point how things are related or linked in an enterprise. It makes the relations explicit and (re-)usable. The LEMO approach uses two central concepts to describe an entire enterprise. The first one is the Enterprise Object. It can be everything which exists in an enterprise like persons, skills, buildings, documents, machinery, etc. The other one is the Enterprise Model which uses and relates Enterprise Objects to describe the enterprise or a part of it.

The Linked Enterprise Models approach is inspired by the Linked Data sub-topic of the Semantic Web introduced by (Berners-Lee, 2006). The idea of Linked Data is that with the usage of linked data it is possible to find related, other (linked) data.

Linked Enterprise Models and Objects (LEMO) are implemented using an ontological representation, the Enterprise Model Ontology (EMO) which is described in the next section. The EMO consists of the super concepts enterprise ontology, aspects, perspectives, model types and modelling languages. The enterprise ontology links the enterprise objects (the first part of the LEMO) and the enterprise model consists of linked model types and modelling languages with aspects and perspectives (the second part of the LEMO).

5.1 Analysis of research findings of enterprise architectures

For the mentioned EAs it has been compared which **aspects** are proposed by each of them in order to understand which aspects might be relevant in contemporary enterprises. As there is no generally agreed set of aspects and some aspects seem to be missing in one or the other approach, a list of aspects is proposed:

Facility, Function, Information Object, Motivation, People and Product.

Account includes all the financial aspect; Facility includes information resources, tools and assets; Function are processes; Information Objects is all about data; Location distinguishes between physical and logical location; Motivation is all about strategy, mission and vision; People can be either internal or external; Product also include services and Time includes time point, time interval and time line. Table 3 depicts a comparison of the proposed aspects related to existing EAs. As it is shown, all aspects from the existing EAs are covered. Furthermore the aspects Account and Location have have been identified during the analysis of existing enterprise ontologies.

	Enterprise Architecture Frameworks					
LEMO	**ARIS**	**BPMS**	**Hanschke**	**EAF of plugIT**	**TOGAF**	**Zachman**
Information Object	Data		Inf. Object Bus. Objects	Data / Knowledge	Data Architecture	Data
Motivation		Strat. Mgmt		Motivation	Business Architecture	Motivation
People	Organisation		Bus. Units Bus. Objects	People/Org.	Architecture Cap. Framew.	People
Facility	Function	IT Mgmt	Applications	Application	Application / Tech. Arch.	Network
Function	Control	BPM SCM	Bus. Proc./ Functions	Process		Function
Product			Products Bus. Objects	Product		

Table 3: EA comparison regarding Aspects

Proposed by many frameworks are **perspectives**. They are needed to fulfill the different demands and views of the various stakeholders. All considered EAs mainly distinguish between business and IT. The number of business and IT level is the only differentiation. Following the Enterprise Architecture Framework of plugIT (which considered other approaches) we use the following perspectives:

Strategy, Business, Systems and Technology.

5.2 Design criteria for ontologies

As mentioned above, to define design criteria for an enterprise ontology three types of sources have been used: interviews with enterprise representatives, six well known Enterprise Architecture Frameworks and literature on existing enterprise ontologies. In addition analysis of EA frameworks and further work from EA and UO has been considered. The design criteria can be grouped into the following four topics: content of the ontology, quality of the ontology, requirements to the system where the ontology is used and common understanding.

5.2.1 Content

- *Completeness*: Does the ontology cover all aspects of an enterprise? (Bernard, 2009) Does the ontology provide a holistic view from the enterprise? (Lankhorst, 2005)
- *Minimality*: Does the ontology contain the minimum number of concepts? (Fox, Barbuceanu, Grüninger, & Lin, 1995). An ontology or framework should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities (i.e. there exists a common understanding and agreement about a minimum number of concepts needed to serve as basis for an enterprise repository). (Gruber, 1993b)
- *Adaptability*: A framework must be adaptable and extensible to implement different standards or even an enterprise-specific one. Furthermore it should be able to handle a trade-off between complexity, detail and simpleness, cost-effectiveness: Modelling as less as possible but as much as needed.

5.2.2 Quality

- *Clearness*: Is the representation easily understood by the users? Does the representation "document itself?" (Fox, Chionglo, & Fadel, 1993) *Perspicuity*: An ontology or framework should effectively communicate the intended meaning of defined terms. The representation has to consider relevant standards. Definitions should be objective and independent of social or computational context. (Gruber, 1993b)
- *Precision*: Does the core set of concepts not overlap in meaning? (Fox, Barbuceanu, Grüninger, & Lin, 1995)
- *Unambiguousness*: (Concise Oxford English Dictionary, 2008) defines unambiguous as "not open to more than one interpretation".
- *Formal*: Is the ontology machine readable? According to "Formal refers to the fact that the ontology should be machine readable, which excludes natural language." (Studer, Benjamins, & Fensel, 1998:p.46)
- *Consistency*: An ontology is consistent of coherent, stable - "unchanging in standard over time" (Concise Oxford English Dictionary, 2008). The term is closely related to perspicuity.

5.2.3 System requirements

- *Extensibility*: Can the representation be extended to encompass new concepts? (Fox, Chionglo, & Fadel, 1993)
- *Various model types*: The following model types are used in enterprise where interviews have been conducted: Balanced Scorecard, The Value Chain (the dynamic view of the process map), Business Processes, Input/Output, Regulations & Controls, External Influences, Products & Services, Legal & Compliance, Risk Management, Knowledge, Communication, Ressources, Applications, IT Architecture, Data, further Utilities & Tools, Organisation Charts, Employees and Roles, Partners (customers, supplier, counterparty, institution/authority), References (templates, checklists, laws & regulations) and a Glossary.
- Human factor considered: The different views of the stakeholders should be represented and not the one of a modelling standard (Staab, 2002). This includes also the requirement from IMS that different viewpoints for different employees are needed.
- *Connected to operative data*: The modelled enterprise should be connected to operative data. For examples customer are stored in a CRM from which the data should be taken.
- Relevant information: Based on the given context, only relevant information should be retrievable. This requirement is demanded by information management systems (IMS).

5.2.4 Common understanding

- *Generality*: To what degree is the representation shared between diverse activities and domain? (Fox, Chionglo, & Fadel, 1993) An ontology or framework should make as few claims as possible about the world being modelled (Gruber, 1993b)
- *Transparency*: A framework must give an overview and transparency about the enterprise and assist a common understanding of business and IT people. Furthermore it should also indicate which relations among the concept have to be modelled.

5.3 Aspects for an enterprise ontology based on analysis of research findings of enterprise ontologies

For this work a categorization based on (Guarino, 1997) is proposed, illustrated in Figure 17. It distinguishes between four types of ontologies, all together belonging to the enterprise ontology. On the top,

> *a **top-level ontology**, describes "very general concepts like space, time, matter, object, event, action, etc. which are independent of a particular problem or domain" (Guarino, 1997:p.146).*

The enterprise upper ontology, enterprise domain ontology and the application ontology further refine the top-level ontology.

> *The **enterprise upper ontology** describes the most general and reusable concepts of each domain and their relations.* (Pease, 2007)

For example, the concepts 'process' and 'controlConstruct' for the process domain, 'organisational unit' and 'role' for the organisation domain, 'document type' and 'document form' for the information object domain and so on.

> *The **enterprise domain ontology** describes the concepts specific to a domain by specializing the concepts introduced in the top-level and upper ontology. (Guarino, 1997:p.146).*

In this work the term domain is used with the definition of the domain of discourse, which often comes along with the universe of discourse. Based on descriptions of (Bergman & Paavola, 2003; De Morgan, 1846; Gruber, 1993; Peirce, n.d.; Regoczei & Plantinga, 1987), it is defined as follows:

> *The **domain of discourse** is an agreed set of objects being discussed in a specific discourse.*

Specific to each application is the application ontology.

> *The **application ontology** again specializes the domain ontology with respect to an enterprise idiosyncrasy and is therefore not reusable.* (van Heijst, Schreiber, & Wielinga, 1997)

A domain or top-level ontology can be associated to multiple application ontologies.

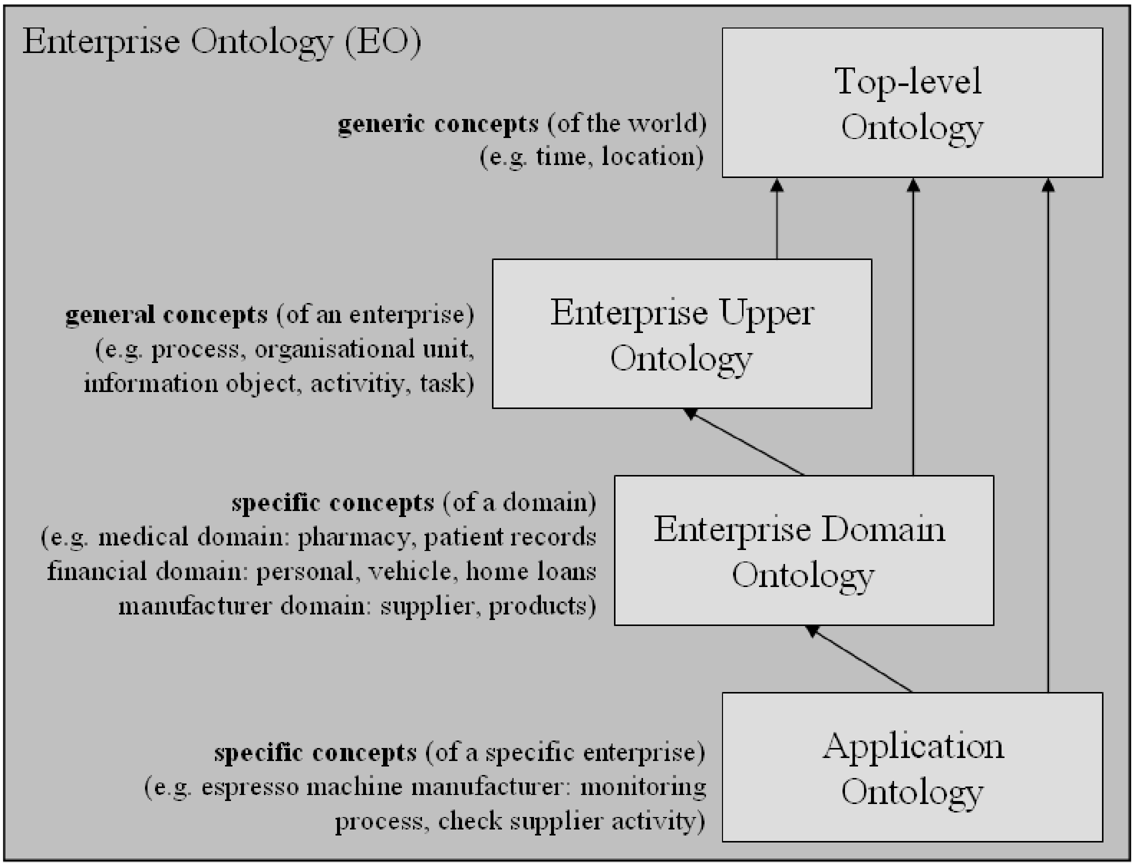

Figure 17: Ontology categorization

This categorisation guarantees reusability of concepts of the domain ontology an enterprise is situated in, but still gives the possibility to refine the ontology to the enterprises specific needs. As no widely accepted categorisation has become established yet, the examined enterprise ontologies use their own categorization and therefore cannot be categorized according to Figure 17.

5.4 The Enterprise Model Ontology

The following Figure 18 gives an overview on how the abovementioned design criteria for ontologies, the Enterprise Ontology (EO), the Model Types and the Aspects & Perspectives relate to each other, called the Enterprise Model Ontology.

The EO (see section 5.5) has been developed based on the analysis of existing upper ontologies and has also been evaluated against the design criteria for ontologies. The EO starts with the top concept 'EnterpriseObject' which is then refined into multiple concepts as described below. Each of the concept is used in one to multiple model type(s). A model type is used to for modelling a part of a domain. For each model type several modelling languages might exist. For example, a process model is a model type for which different modelling languages as BPMN, EPC, etc. exist. A model type belongs to one or multiple Aspect(s) and Perspective(s) as indicated by the arrows.

Based on the literature review about EA Frameworks, needed aspects and perspectives were defined, indicated by the dashed arrows. Together with the conducted interviews a first list of model types has been listed.

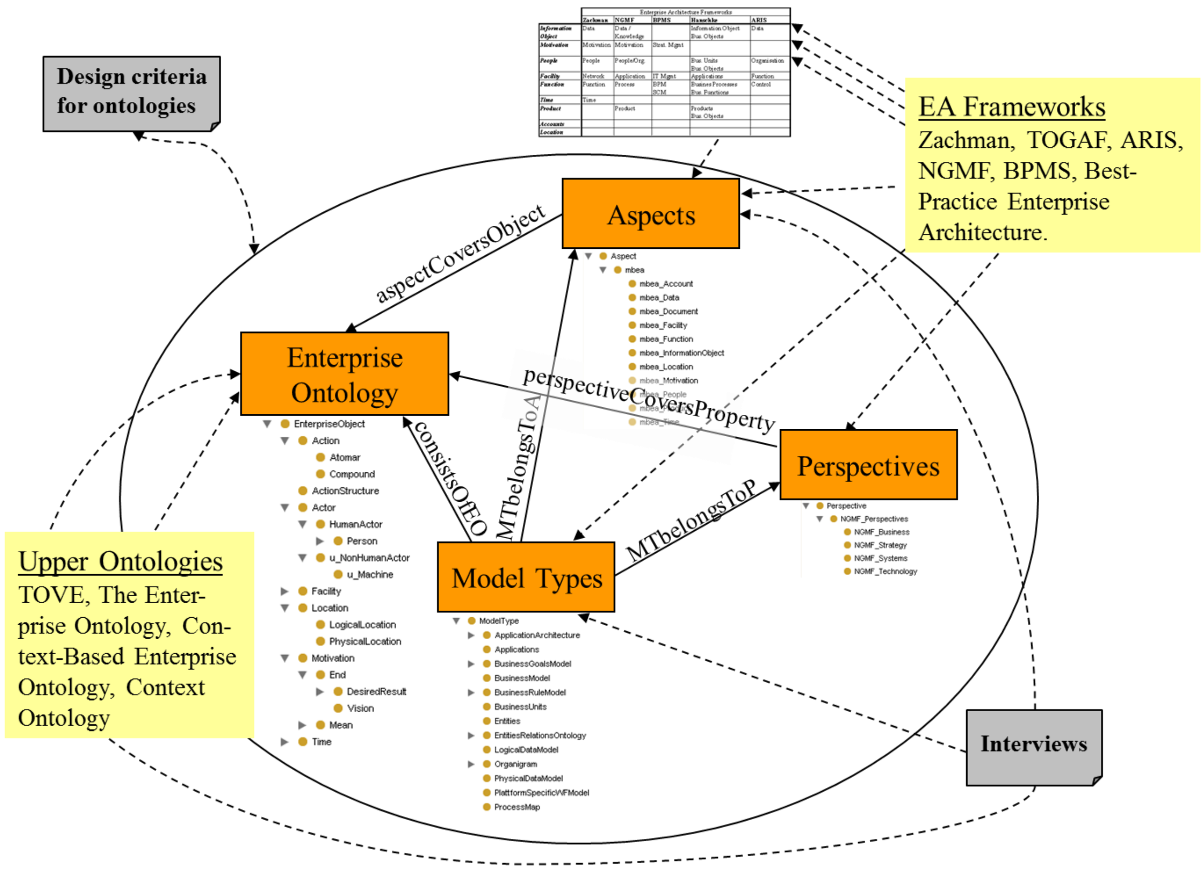

Figure 18: The Enterprise Model Ontology

The following model types have been taken into account and related to the aspects and perspectives. They will be refined in later stages. The model types in brackets come from the requirements gather in the interviews:

- ProcessModel (The Value Chain, Business Processes)
- OrganisationalModel (Organisation Charts, Employees and Roles, Partners, RACI Model)
- StrategyModel (Balanced Scorecard, External Influences)

- ProductModel (Products & Services)
- InformationModel
- InfrastructureModel (IT Architecture)
- DataModel (Data, Input/Output, References, Data structure diagram, Entity-relationship model)
- SoftwareModel (Applications)
- FacilityModel (Ressources, further Utilities & Tools)
- Regulations&ControlModel (Regulations & Controls, Legal & Compliance, Risk Management)

The model types are related to the EO and additionally serve as a link between aspects and perspectives. With this relation it can be verified whether the proposed concepts of the EO are sufficient to model an enterprise architecture with its aspects and perspectives or whether some concepts are not needed, missing or need to be redefined.

The main objective of the work is the development of the EO. The model types serve as a link between aspects and perspectives. As they are related to the aspects and the perspectives it can be verified whether the proposed concepts of the EO are sufficient to model an enterprise architecture and its aspects and views or whether some concepts are not needed, missing or need to be redefined.

For the Zachman and the plugIT approach, their aspects and perspectives have been exemplarily listed in the ontology and linked to the model types. So, using e.g. a SPARQL query it can be asked which model type is applicable for e.g. the perspective Business and the aspect Product of the plugIT which would return the model type 'Product Model'. This shows that an arbitrarily framework can be integrated into the ontology. Once modelled, it also shows which enterprise objects are needed to model it.

5.4.1 The relationships between the model types and the aspects/perspectives

The main objective of the work is the development of the EO. The model types serve as a link between aspects and perspectives. As they are associated to the aspects and the perspectives it can be verified whether the proposed aspects and perspectives are sufficient to integrate all model types.

The LEMO Aspects and the LEMO Perspectives are integrated into the ontology and linked to the model types. The same is done for the Zachman and the EAF of plugIT approach. So, using a SPARQL[17] query (SPARQL is a query language for RDF) it can be asked which model type is applicable for e.g. the perspective Business and the aspect Product of the LEMO which returns the model type 'Process Model'.

Get Model Type for perspective / aspect

```
SELECT DISTINCT ?modeltype
WHERE {?m rdfs:subClassOf ch_fhnw_emo:ModelType.
   ?a rdfs:domain ?modeltype.
   ?a rdfs:subPropertyOf ch_fhnw_emo:modelTypeBelongsToAspect.
   ?a rdfs:range ch_fhnw_emo:LEMO_Function. //Choose the aspect
   ?p rdfs:domain ?modeltype.
   ?p rdfs:subPropertyOf ch_fhnw_emo:modelTypeBelongsToPerspective.
   ?p rdfs:range ch_fhnw_emo:LEMO_Business. //Choose the perspective }
```

Modeltype
ch_fhnw_emo:ProcessModel

Figure 19: Result searching for a modeltype for the aspect LEMO_Motivation and the perspective LEMO_Business

If one is also interested in the modelling languages that can be used, the query can be enhanced. The following query uses this time the NGMF instead of the LEMO approach to show that an arbitrarily framework can be integrated into the ontology, linked with the model types and the queried.

Get additional Modelling Languages of Model Type for perspective / aspect

```
SELECT DISTINCT ?modelType_and_ModellingLanguage
WHERE {?modelType_and_ModellingLanguage rdfs:subClassOf ?modelType.
   ?modelType rdfs:subClassOf ch_fhnw_emo:ModelType.
   ?a rdfs:domain ?modelType.
```

[17] SPARQL Protocol and RDF Query Language http://www.w3.org/TR/rdf-sparql-query/ (Accessed on 2010-07-12)

```
?a rdfs:subPropertyOf ch_fhnw_emo:modelTypeBelongsToAspect.
?a rdfs:range ch_fhnw_emo:NGMF_Motivation.
?p rdfs:domain ?modelType.
?p rdfs:subPropertyOf ch_fhnw_emo:modelTypeBelongsToPerspective.
?p rdfs:range ch_fhnw_emo:NGMF_Business.}
```

ModelType_and_ModellingLanguage
ch_fhnw_emo:PRR
ch_fhnw_emo:SBVR
ch_fhnw_emo:BusinessRuleModel

Figure 20: Result searching for a modeltype for the aspect LEMO_Motivation and the perspective LEMO_Business

5.4.2 The relationships between aspects and enterprise objects

As there might exist enterprise objects which are not used in a model type but nevertheless associated to an aspect, this relation must also explicitly be modelled. One could argue, that an enterprise object which is not modelled is of no interest for an enterprise. Nevertheless it could be of interest to see which enterprise objects exist for an aspect as a new model type has to be developed or for any other reason. Therefore the relation *aspectContainsEnterpriseObject* is implemented. The refinement of this relation remains future work.

5.4.3 The relationships between perspectives and enterprise objects

The business process model is used by various stakeholders, first the business people working for the particular business process, the managers of a business function up to the board of directors which are interested in the process but also by the more technical oriented people in order to understand which IT resources are needed for the business process and how they are applied. They might all be interested in the same action, but as a matter of course they are interested in different properties of this action: An action

- *strivesForEnd* and *endorsesMeans* which is interesting for the **strategic** perspective
- *hasAssociatedBusinessRule*, *hasRoleRequirement* and *isPerformedByActor* interesting for the **business** perspective

- *usesFacility* and *producesFacility* which can be of interest for the **system** or **technology** perspective.

Therefore a perspective is associated to properties of enterprise objects with the relation *perspectiveCoversProperty*. The described example is illustrated in the following figure.

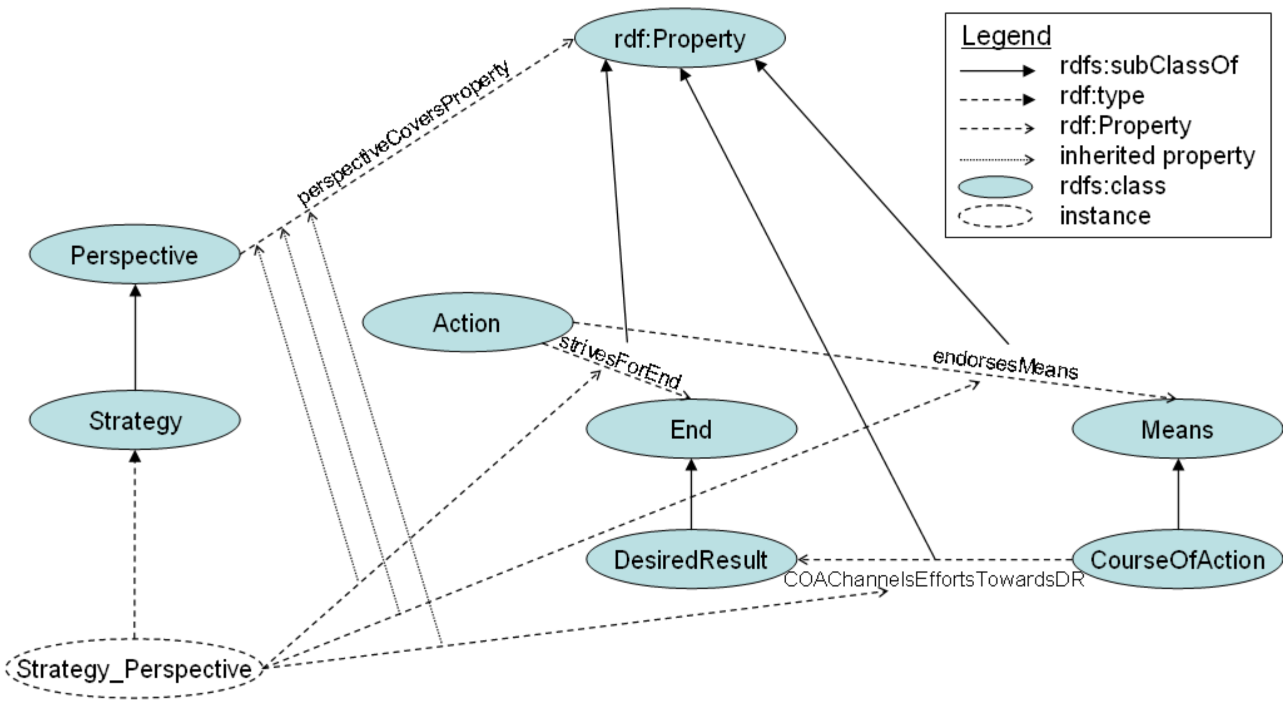

Figure 21: Overview for the given example

The following SPARQL Query is defined which returns all properties of the strategy perspective for an action.

```
SELECT DISTINCT ?relevantProperty
WHERE {ch_fhnw_emo:LEMO_Strategy_Perspective
ch_fhnw_emo:perspectiveCoversProperty
?relevantProperty. //Choose the perspective
?relevantProperty rdfs:domain ch_fhnw_emo:Action.} //Choose the
enterprise object
```

The SELECT Statement retrieves properties of an enterprise object for a specific perspective.

RelevantProperty
ch_fhnw_emo:actionEndorsesMeans
ch_fhnw_emo:actionStrivesForEnd

Figure 22: Query-Result searching for all relevant properties of the enterprise object action for the perspective strategy

As the result shows, only the properties belonging to action (i.e. not *courseOfAction-ChannelsEffortsTowardsDesiredResult*) are returned, belonging to the strategy perspective.

Such a perspective can now be assigned to a role. As a matter of course, these perspectives can be defined in much more detail. The description above gives an example how to implemented it, the completion of it for the whole EMO remains future work.

5.4.4 Glossary and Non-Categorized Object (NCO)

The interview partner claimed that a glossary is of great support for an enterprise and frequently used. The EMO inclues all enterprise objects which need to be defined. As the EMO serves as knowledge base, the glossary description should also be given in it. Therefore to each concept an additional label is added called 'glossaryDescription'. In such a way, this information can be fetched from the ontology and displayed in a more user friendly way. As an example the concept Location is tagged with a label 'glossaryDescription' (Hint: Multiple labels cannot be used in the used version 3.4.1 of Protégé in RDF projects.):

```
<rdfs:Class rdf:about="&ch_fhnw_emo;Location">
    <rdfs:label xml:lang="name">Location</rdfs:label>
    <rdfs:label xml:lang="glossaryDescription">Location is a place or
position of an actor or a facility.</rdfs:label>
   <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
```

The ontology can then be queried with the following code:

```
SELECT DISTINCT ?concept ?label
WHERE { ?concept rdfs:label ?label.
FILTER sameTerm(?concept, ch_fhnw_emo:Location). //for the example only
retrieving the concept Location
FILTER (lang(?label)="glossarydescription").}
```

Concept	Label
ch_fhnw_emo:Location	"Location is a place or position of an actor or a facility."@glossarydescription

Figure 23: Query-Result searching for the glossary entry of the concept location

It is the idea that any enterprise object can be categorized according to the enterprise ontology. However, there might be some concepts which are needed (e.g. for the glossary) but unclear where it should be integrated. For such concepts a direct sub-concept of Thing called non-categorized object (NCO) is used. It can be seen as a temporary place where for putting concepts. As soon as it becomes clear where to integrate them, they are rearranged.

5.5 The Enterprise Ontology (EO)

As described in section 5.3, for this work it is distinguished between the four types of ontologies top-level ontology, enterprise upper ontology, enterprise domain ontology and application ontology. Figure 24 shows the relationship between the four enterprise ontology types embedded into the enterprise ontology and the model types with the related aspects and perspectives.

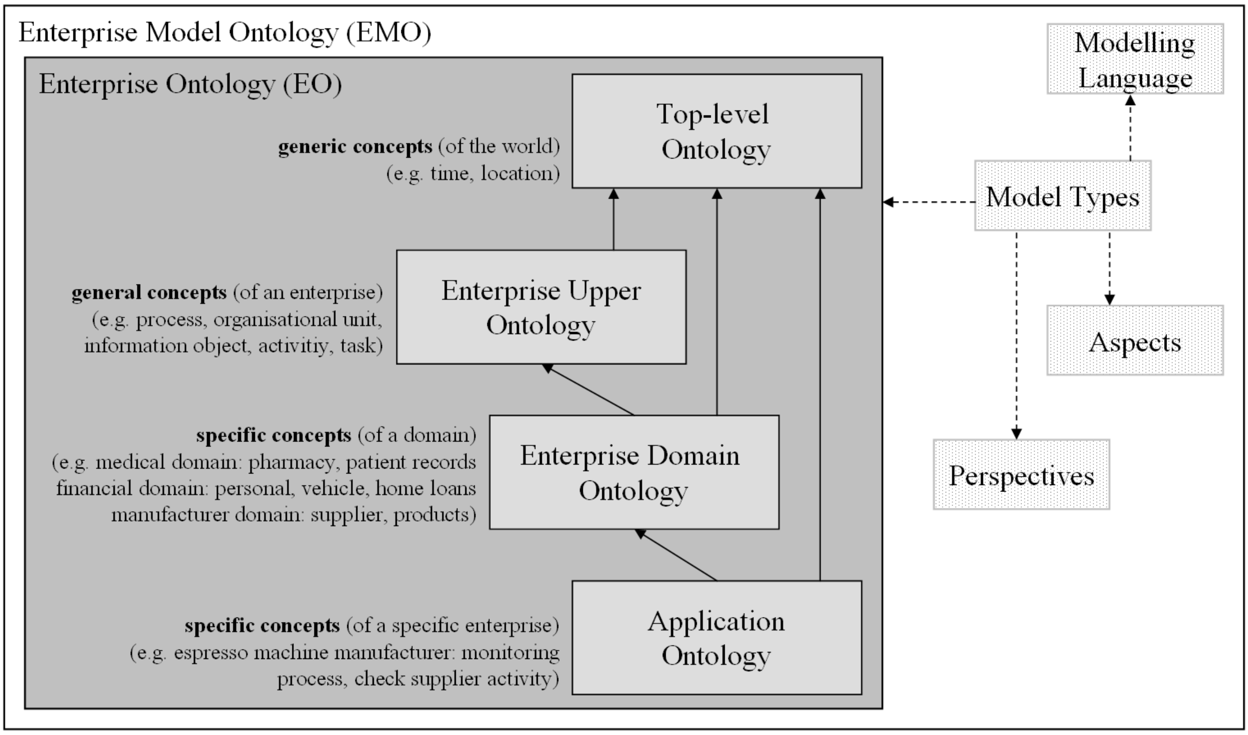

Figure 24: Ontology categorization

In the EO described in the following, the concepts time and location belong to the top-level ontology. Action, action structure, actor, facility, role, authority, skill and motivation belong to the upper ontology. However, refinements of the upper ontology belong to the domain ontology as well.

To determine what concepts and relations have to be defined in an enterprise ontology three sources are used: Interviews with practitioners to get a first list of important entities and relations that they regard as important. Literature review on a) several architecture frameworks to identify the main constituents to be represented in an enterprise architecture and on b) existing enterprise ontologies which lead to concept and relations and additionally give input to the way of structuring the ontology. Figure 25 illustrates the basic concepts and their relations of the EO. For a better readability, not all described relationships are visualized.

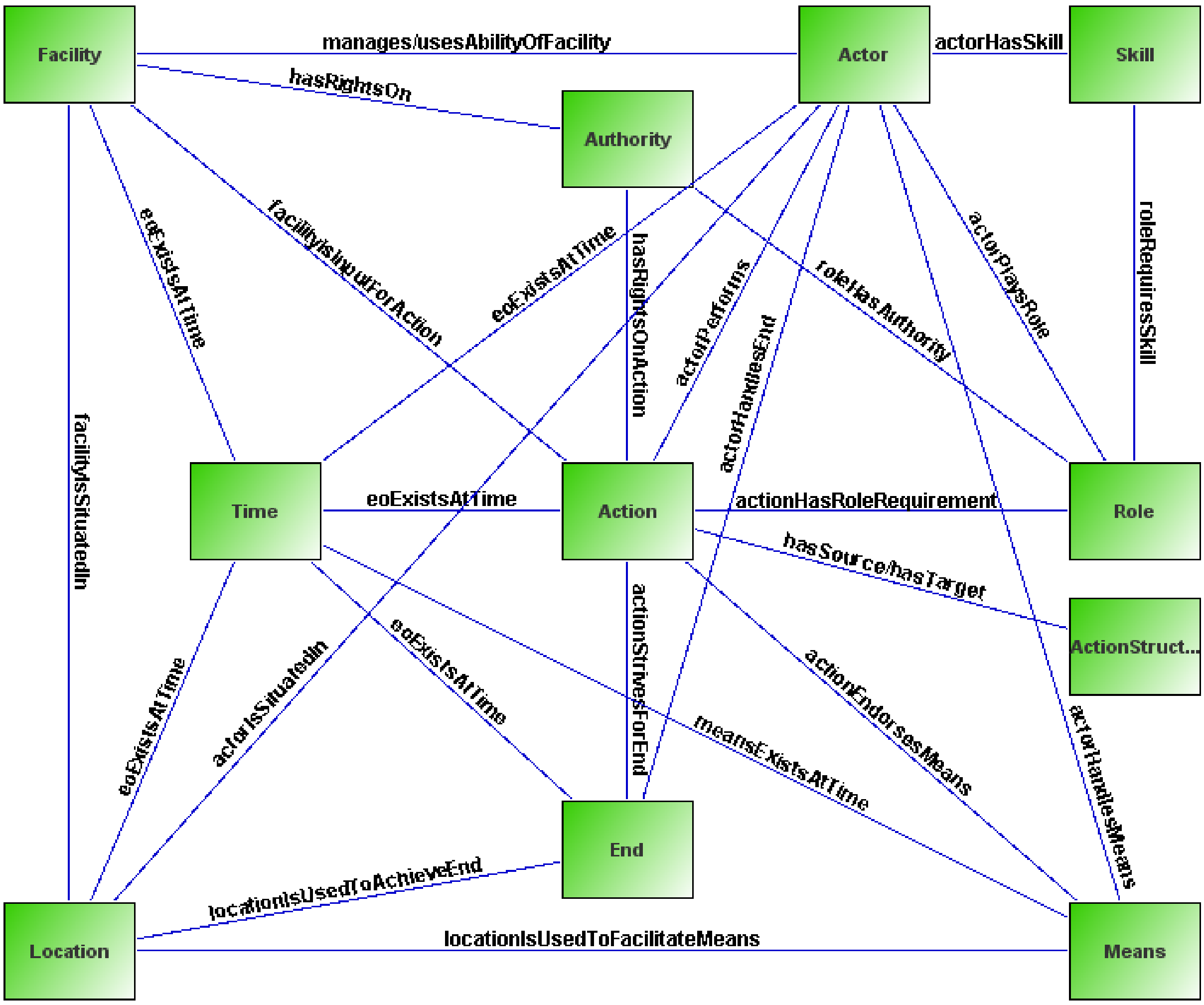

Figure 25: The enterprise objects of the EO and their relationships

An **action** is "the process of doing something to achieve an aim." (Concise Oxford English Dictionary, 2008) and "comprises all those concepts and constructs that refer to deeds or events"(Leppänen, 2007). An action produces an output when executed. The output is any facility, e.g. a product, a resource, a tool or an asset (*actionProducesOutputFacility*). An action endorses the end, e.g. when an activity is directed towards achieving a desired result (*actionStrivesForEnd*) or following a strategy (*actionEndorsesMeans*). Furthermore an action has certain requirements against the role executing it (*actionHasRoleRequirement*) and is performed by an actor (*actionIsPerfomedByActor*).

An **action structure** defines the control structure, e.g. the sequence of actions, the message flow, an association or data association. It is among others associated to action, which is either the source (*hasSource*) or the target (*hasTarget*) of the relation.

A **facility** "contains all those concepts and constructs that refer to the means by which something can be accomplished, i.e. something, which makes an action possible, more efficient or effective." (Leppänen, 2007:p.284). It can be an asset, a tool, a resource, a product or a component. A facility is situated in a location (*facilityIsSituatedInLocation*). A facility (mainly a resource) is used as input for an action (*facilityIsInputForAction*), to facilitate means (*facilityIsUsedToFacilitateMeans*) and to achieve and end (*facilityIsUsedToAchieveEnd*). Further, a facility is managed by an actor (*facilityIsManagedByActor*).

An **actor** can be a legal entity, a collection, a natural actor or a machine actor. Actors are "active parts in a context and perform, own, communicate, borrow, send, receive etc. objects in the contexts" (Leppänen, 2007:p.275). An actor is a member of a division (*actorIsMemberOfLegalEntity*) or of a collective (*actorIsMemberOfCollective*) and plays a certain role (*actorPlaysRole*). It performs action (*actorPerformsAction*) for which it needs/has skills (*actorHasSkill*). An actor is situated in a location (*actorIsSituatedInLocation*) and further manages facilities (*actorManagesFacility*) - which is the inverse slot of *facilityIsManagedByActor* - and uses abilities of facilities (*actorUsesAbilityOfFacility*) following a certain means (*actorHandlesMeans*) to achieve ends (*actorHandlesEnd*).

The concept **skill** represents the ability to do something well with expertise or dexterity.

Role defines one or more job functions in an organisation. Authority is needed for the role to achieve its goals (*roleHasAuthority*). "A role may be a generalized or specialized (*roleHasSpecializedRole*) role of another." (Fox, Barbuceanu, Grüninger, & Lin, 1995:p.143). A role belongs to a process to achieve goals (*roleHasProcess*). "[…] Resources may be allocated to a role for disposition under its authority" (Fox, Barbuceanu, Grüninger, & Lin, 1995) (*roleHasResource*). A Role needs skills for realizing job functions (*roleRequiresSkill*). The hierarchy of roles is represented by a supervisor relationship (*roleHasSupervisorRole*).

Authority gives something the right of using facility (*authorityHasRightOnFacility*) and to execute actions (*authorityHasRightOnAction*) (Fox, Barbuceanu, Grüninger, & Lin, 1995).

Motivation gives a reason or reasons for doing something (Concise Oxford English Dictionary, 2008). Motivation is divided into Ends and Mean. "**Ends** are about what an enterprise wants to be. […] **Means** are about what an enterprise has decided to do in order to become what it wants to be." (Object Management Group, 2008:p.12-13).

Location is a place or position of an actor or a facility. "The location can be physical, like a room or a building, or logical, like a site in a communication network." (Leppänen, 2005:p.21). Following Leppänen, the physical location consists of a spatial thing which is placed in a region. A region can be an area or a point. A specific location is needed to facilitate means (*locationIsUsedToFacilitateMeans*) and to achieve an end (*locationIsUsedToAchieveEnd*).

Time is defined by (Concise Oxford English Dictionary, 2008) as "the indefinite continued progress of existence and events in the past, present, and future, regarded as a whole." The concepts of time are merged from the proposals of (Leppänen, 2007; Uschold, King, Moralee, & Zorgios, 1998) which base on existing terms Allen's work (Allen & Lehrer, 1992). The direct sub-concepts of time are time interval, time point, time zone, time element and calendar element. The time interval is divided into convex interval and non-convex interval. The convex interval has a start time point and an end time point, the non-convex interval is used for "representing regularly recurring events. For example, 'every Wednesday in September'" (PREMIS Editorial Committee, 2008:p.2) time point is particular point in time belonging to a time zone, a calendar element (which includes calendar day, calendar weekday, calendar month and calendar year) and a time element (which includes time hour, time minute and time second).

All described concepts are sub-concepts of the **Enterprise Object** (not illustrated in Figure 25). It is the highest concept of the EO and has a few attributes. First it is associated to the time concept (*enterpriseObjectExistsAtTime*), e.g. a location is situated in Zürich in December 2010 or an authority is handed over for the period from the 2nd of July till the 28th of August 2010. It further possesses the relations *enterpriseObjectHasName* and *enterpriseObjectHasIdentifier* to uniquely identify an enterprise object. Last but not least it is also associated to the EMO (*modelTypeConsistsOfEnterpriseObject*, *aspectCoversObject*, *perspectiveCoversProperty*) and in that way links the EO with the other main constituents, namely model types, aspects and perspectives, of the EMO

5.6 Validation of the EMO with the design criteria

The design criteria for ontologies have been described in section 4.1. How the critieria have been fulfilled is described in this section.

Regarding the ***content***, completeness and can be verified by checking whether the concepts of the enterprise ontology are sufficient for the definition of the model types and the corresponding aspects and views. As described above, the aspects and perspectives of the enterprise architecture Zachman and plugIT have been exemplary linked to model types which are related to the Enterprise Ontology. This shows that an arbitrary framework could be used in conjunction with approach and adaptibility therefore is given.

The ***quality*** aspects can be verified as follows: Regarding *perspicuity* it can be checked whether standards are considered. As described by (McGuinness, 2002) already a simple ontology contains a controlled vocabulary which provides an *unambiguous* interpretation of terms. Therefore, an ontology is usually unambiguous. As the enterprise ontology is described in RDF(S) and it is attempt of this language to be unambiguous (Miller, 2001), this criterion should be fulfilled but needs further verification for a technically and methodologically sound proof. Through the usage of RDF(S), the ontology needs to be described in a *formal* way and therefore it is machine readable and the consistency of an RDF(S) ontology can be verified with a *con-*

sistency checker as Pellet[18] (Parsia & Sirin, 2004) which can be integrated into Protégé, ConsVISor[19] (Baclawski, Kokar, Waldinger, & Kogut, 2002) or FaCT++[20] (Tsarkov & Horrocks, 2006) and an RDF/XML Validator as the rdf:about Validator[21]. The ontology is succesfully validated with Pellet and rdf:about. To validate *clearness* and *precision* an application into a real life scenario over a longer period of time is needed and following remains future work.

System requirements. As the enterprise is implemented as ontology in the modelling tool ATHENE, new concepts can easily be added and obsolete ones removed, achieving *extensibility* of the architecture. As listed in section 5.1, the proposed list considers all the *model types* demanded. However, only a part of them could already be implemented. To consider various views of stakeholders a framework including different perspectives is implemented also claimed by IMS. Therefore, the *human factor* is partially considered, but as for the criteria clearness and precision further investigation is needed to validate whether this requirement is fulfilled. A connection to operative data is certainly necessary but out of scope for this work. The last system requirement defines that only *relevant information* should be retrievable. The exemplary queries applied to the EMO indicate that they can be used as filter to only retrieve relevant information. As described by (Krcmar, 2004) an ontology as an IMS is one possible implemenation approach.

The following criteria could not be verified yet but it is itended to do so soon: Minimality, Connected to operative data, Precision, Clearness, Generality and Transparency. The following table gives an overview about the design criteria and whether they are fulfilled, partially fulfilled, not fulfilled or not validated in the EMO:

[18] Pellet http://clarkparsia.com/pellet (Accessed on 2010-07-24)

[19] ConsVISor http://173.14.188.57:8080/consvisor/ (Accessed on 2010-07-05)

[20] Fact++ http://owl.man.ac.uk/factplusplus/ (Accessed on 2010-07-24)

[21] rdf:about http://rdfabout.com/demo/validator/ (Accessed on 2010-07-12)

	Fulfilled	*Partially fulfilled*	*Not fulfilled*	*Not validated*
Content				
- Adaptability	✓			
- Completeness	✓			
- Minimality	✓			
Quality				
- Clearness				*
- Consistency	✓			
- Formal	✓			
- Perspicuity	✓			
- Precision				*
- Unambiguousness				*
System requirements				
- Connected to operative data				*
- Extensibility	✓			
- Human factor considered		~		
- Relevant information	✓			*
- Various model types		~		
Common understanding				
- Generality				*
- Transparency				*

Table 4: Evaluation based on design criteria

6 Description of the approach

In the above sections we have seen that it is possible with usage of an ontology to organise enterprise information in a formal way. The thesis shows also how it is possible to capture all relevant enterprise information based on the enterprise architecture approach. Now the knowledge base, the enterprise model ontology is ready to use to create new metadata. The question is how to create the metadata? Now the enterprise information is formally available. Or in other words expressed, the links between the enterprise objects are explicit. In order to create metadata an executing enterprise object (an actor) and execution description (a description of an action) is needed. As described in the thesis statement and sub-goals, this will be done implementing a rule based approach.

6.1 Ways for creating metadata

Researchers have found out that automatic metadata generation can provide useful results and it could be possible that automatic metadata creation is more efficient, less costly, and more consistent than human-oriented metadata creation. (Greenberg, Spurgin, & Crystal, 2005) And in fact, research indicates that automatic metadata creation can produce acceptable results (Giles, Manavoglu, & Fox, 2003; Liddy et al., 2002; Takasu, 2003). And (Greenberg, 2004) found out, that metadata extraction and metadata harvesting are two important methods of automatic metadata creation based on digital resources. "Metadata extraction involves the mining of resource content […]. Metadata harvesting relies on machine capabilities to collect tagged metadata previously." (Greenberg, Spurgin, & Crystal, 2005:p.2).

There exists several ways for creating metadata based on ontologies. Document annotation as described in (Eriksson, 2007) is one promising approach for knowledge acquisition. It gives the possibility to annotate text phrases and link it to an ontology which can be seen as a way to create metadata. The CREAM approach of (Handschuh, Staab, & Ciravegna, 2002) and (Handschuh, Staab, & Studer, 2003) goes in the same direction; they use an annotation approach and an information extraction method in combination.

Based on this and the literature review concerning enterprise ontologies and architectures in section 2 (Literature Review) we can derive the following ways for creating metadata:

1. Manual metadata creation: If we have no automatic system providing the creation, then the only way for metadata creation is adding metadata by hand. As seen from (Schwartz, 2001) most effective results can be achieved by integrating manual and automatic metadata creation approaches.
2. Automatic information extraction: Automatic information extraction is used when the content of a resource will be automatically gathered as described in (Greenberg, 2004). What is defined here as automatic information extraction is sometimes called automatic metadata extraction in the metadata terminology. But this term leads to misunderstanding because metadata extraction stands sometimes for metadata harvesting.
3. Metadata harvesting: Metadata harvesting deals with the possibility to collect existing metadata which is already embedded in a resource (Greenberg, 2004). In some literature metadata harvesting is called metadata extraction (see for example (The Apache Software Foundation, 2010)).
4. Context-based metadata creation: The thesis statement says that it is possible to create metadata considering the context. Mapping that back to the LEMO approach, it is possible to say that metadata could be created using an enterprise ontology which contains all information and objects in a linked way. Through the usage of this ontology or in other word this context information one is able to create metadata using predefined rules.
5. Application / Data Integration: Application and data integration goes in the same direction as context-based metadata creation. There we say that existing information stored in third party systems can be used to create metadata.

As already mentioned and showed by (Schwartz, 2001), is makes often sense to combine the mentioned approaches for creating metadata in order to receive better results.

6.2 Preliminary thoughts on rules

As defined in the thesis statement this approach will be implemented using rules. Dealing with ontologies, software and business the following rule descriptions come into consideration:

1. Semantic Web Rule Languages: When dealing with rules in semantic environment, semantic web rules come into consideration. The perhaps most prominent semantic web rule representation is the Semantic Web Rule Language or short SWRL. SWRL is "[…] based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language." (Horrocks et al., 2004). An other prominent representation is the Jena Rule language, which is based on the Jena Generic Rules Reasoner API (Reynolds, 2010).
2. Business Rule Languages: Rules were also mentioned in a business context during the interviews (see section 4.1.2 (Consulting company)). In this context the rules action as business rules. Business rules provides an additional value "[…] on top of a general purpose Rule Engine by providing business user focused systems for rule creation, management, deployment, collaboration, analysis and end user tools." (JBoss Community, 2010).
3. Domain Specific Rule Languages: Is derived from the Domain Specific Language (DSL) approach. A DSL gives the possibility to create for a specific domain a specific (programming) language (Behrens et al., 2010). Based on this it could also make sense to provide a domain specific rule language to the user of a metadata system in order to have a rule language with reduced complexity.
4. Programming Languages: And finally, program code can also be seen as a way for creating rules.

6.3 Preliminary thoughts on the system architecture

Based on the common software layer architecture (Green & Miller, 2007; Schussel, 1996) of having a presentation layer, a business layer and a data access / persistence layer we can derive the following layers for this metadata creation approach:

6.3.1 User- centric Layer

Derived from the presentation layer we can describe the user- centric layer as follows.

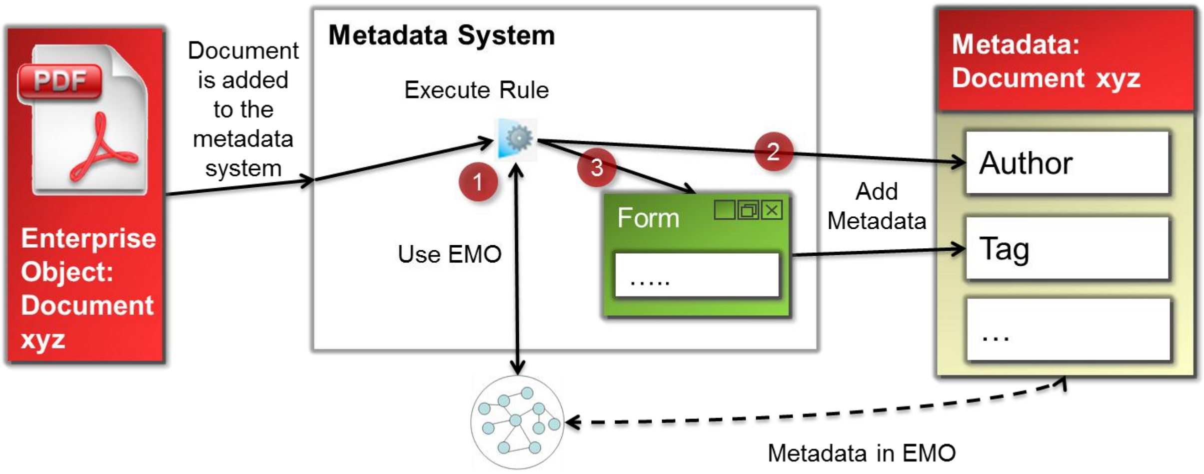

Figure 26: Exemplary rule execution on the user- centric layer

This user centric layer deals mainly with actions made by the user. When we take the example showed in Figure 26 we can see how and when rules can be executed on this layer. Assuming there is a metadata system to which the user adds a document. The system tries to analyse the document using one or more metadata approaches described in section 6.1 (Ways for creating metadata). Further we assume that the information and the existing metadata is somehow extracted and now a context- based metadata approach will be executed as follows (please follow the numbers in Figure 26):

1. The defined rules will be executed and the ontology will be queried in order to find relevant concepts or instances. On the user-centric layer it is possible to use semantic web rule languages like SWRL or Jena Rules to define rules. But it is even possible to execute business rule languages or domain specific rule languages directly on the user-centric layer. And of course it is possible to execute program code which can be also seen as a sort of predefined rules.
2. After executing the rules the metadata system is then able to assign new metadata to a document stored in the ontology.
3. It could also happen that some information is not available. Then the system could execute some user-request rules which create for example a dialog

which asks the user for entering additional information in order to assign new metadata to the document.

6.3.2 Functional Layer

Derived from the business layer we can describe the functional layer. Actually this layer is indeed very similar to the business layer concerning software architecture.

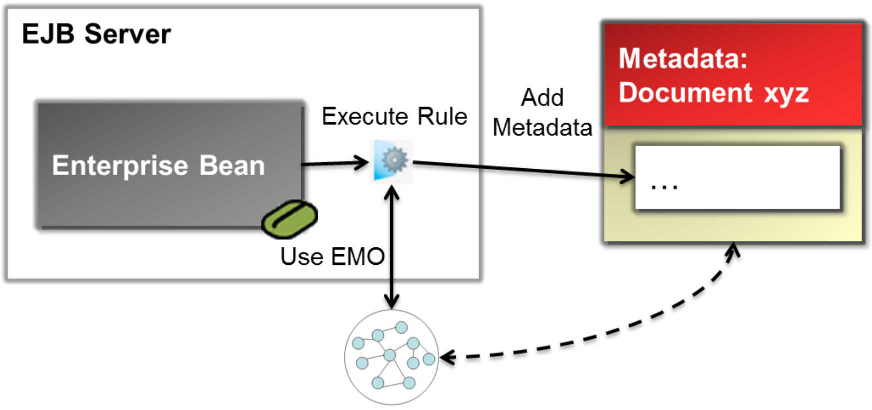

Figure 27: Exemplary rule execution on the functional layer

Figure 27 shows how rules can be executed on the functional layer. Assuming we have an enterprise bean which has new or changed information about an object or an entity. This changes or new information can be “recognized” using rules based on rules languages like business rule languages, programme code or even semantic web rule languages (when the enterprise beans contains ontology related information). After executing the rules and querying the ontology, the result can be added as metadata to an enterprise object, in this case the document.

6.3.3 Semantic Layer

The semantic layer is derived from data access / persistence layer. It is the layer where rules based on semantic web rule languages will be executed.

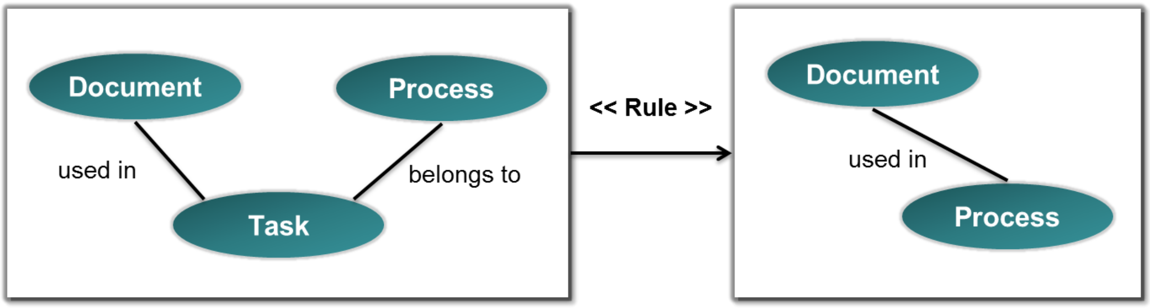

Figure 28: Exemplary rule definition on the semantic layer

Figure 28 shows how rules based on semantic web rule languages could look like. These rules can be executed using a semantic web rule language engine like the Jena Generic Rules Reasoner API (Reynolds, 2010) to derive new knowledge or metadata.

6.4 Preliminary thoughts on querying the ontology

As mentioned in the architecture examples in section 6.3 (Preliminary thoughts on the system architecture) the ontology needs to be accessed. There are two common ways to access an ontology represented as RDF:

1. Accessing by query: The most prominent query language is SPARQL. It is a query language defined for RDF and the name stands for SPARQL Protocol and RDF Query Language. "SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware."(Prud'hommeaux & Seaborne, 2008).

2. Accessing by model API's: The second possibility for accessing an ontology is the usage of ontology framework like the Jena Ontology API (Dickinson, 2009). Such a model can be used for "[…] creating resources, properties and literals and the Statements which link them, for adding statements to and removing them from a model, for querying a model and set operations for combining models." (Dickinson, 2009).

6.5 The implemented approach

After considering all relevant parts, the demonstrator is developed as follows (see Figure 29):

- When adding a document to the metadata system, the existing metadata will be harvested using a metadata harvesting component.
- Then the content of the document needs to be accessed an analysed using an automatic information extraction approach.
- After analysing the content, the initial findings will be added to the enterprise ontology. This will be done on the functional layer using an Ontology API, SPARQL and program code.
- In order to enhance the metadata findings, the user is requested to create metadata rules; this has to be done only once and the rules will be stored after creation in the ontology or somewhere else. In the case of the demonstrator the rules will be created based on semantic web rule languages.
- The created rules will be executed on the semantic layer using an Ontology API and rules based on semantic web rule languages. The result of the execution will be added as new metadata linked to the document.

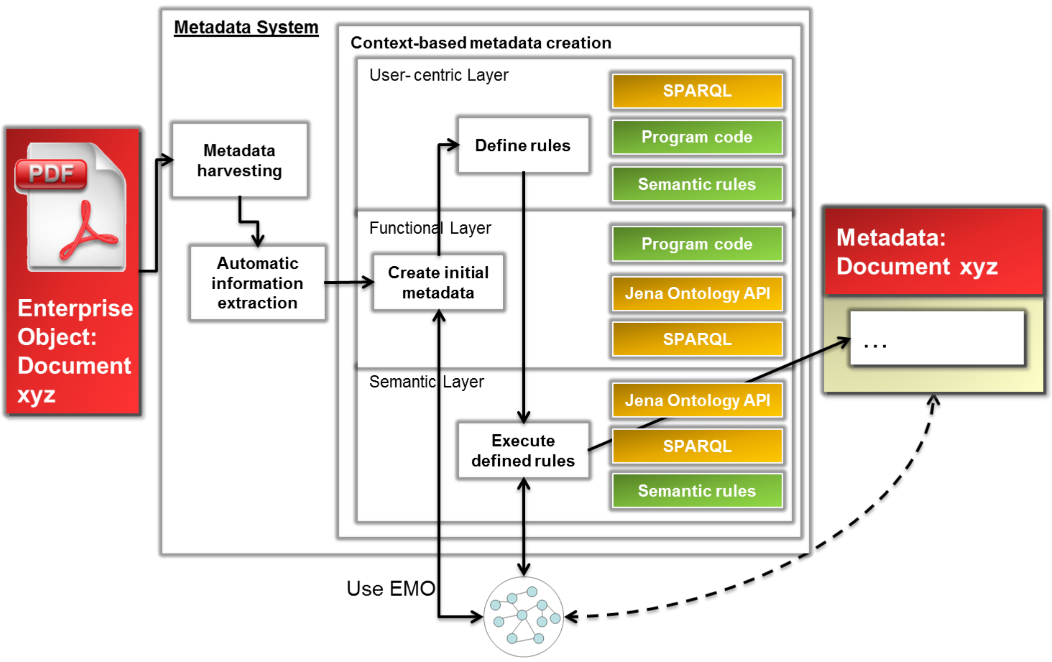

Figure 29: Conceptual overview of the implemented approach

7 Application Scenario

The following application scenario is based on a real use case. It is here presented in a completely anonymized form using fictive companies, products and people. The name of every object is given in an arbitrary way – similarities to real things are unintentional and completely random.

The application scenario deals with a contract about a product delivery. We assume that the fictive SwissSoft AG is selling the content management system SwissCMS to the SwissBikes AG. Now after signing the contract SwissSoft would like to "import" the contract into their enterprise repository including automatically created metadata.

7.1 Data

7.1.1 The contract

The exemplary contract is a software license agreement between SwissSoft AG and SwissBikes AG. In contains information about the parties, product and services to be provided, payment, warranty, licence, duration of the licence, Proprietary rights, coping, training, termination and law. The whole contract is shown in (Appendix A: Exemplary contract of the application scenario). The contract includes also information about the people who signed the contract.

7.1.2 EMO Data

As mentioned in the short description about the application scenario we assume that SwissSoft will "import" the signed contract into their enterprise repository containing the EMO. Additionally we assume that SwissSoft is implementing the LEMO approach – meaning, that SwissSoft is in possession of an enterprise repository running the enterprise ontology. Figure 30 gives an overview about the enterprise repository entries concerning the SwissSoft AG. This figure was created using the software

StarUML[22] (an open source UML modelling tool; indicates a generalization, indicates an association and relates an instance to its class; indicates an inferred association between instances)

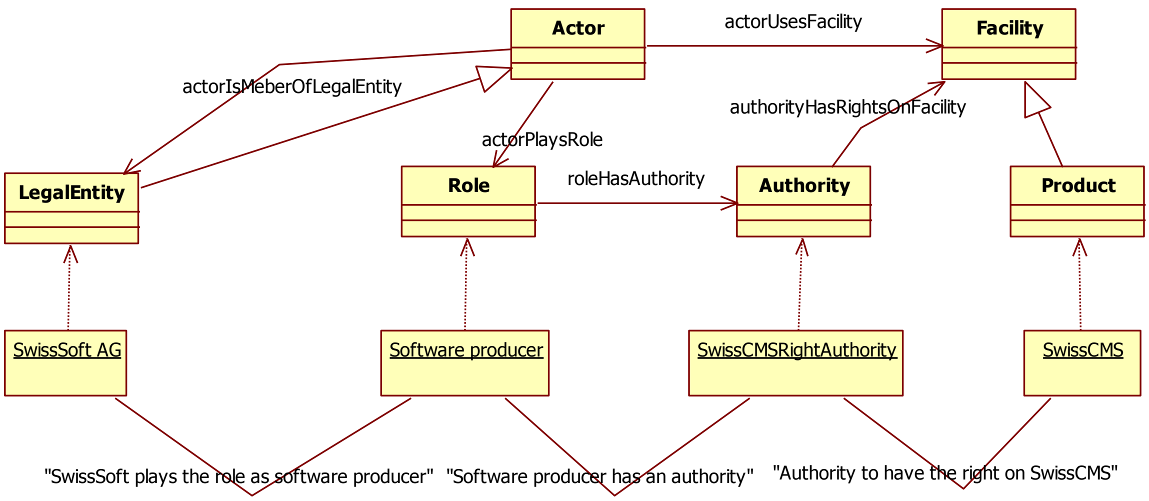

Figure 30: Application scenario: SwissSoft (Legal Entity, Role, Authority, Product)

The enterprise ontology of SwissSoft consists of the following RDF entries (cf. Figure 30) concerning SwissSoft's legal entity, role, authority and product information:

SwissSoft is a Legal Entity and plays the role of a software producer.

```
<emo:LegalEntity rdf:ID="SwissSoft_AG">
    <emo:actorPlaysRole rdf:resource="#SoftwareProducer"/>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >SwissSoft AG</rdfs:label>
</emo:LegalEntity>
```

Through the role as software producer, SwissSoft has a right on their product using the authority relation:

```
<emo:Role rdf:ID="SoftwareProducer">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Software producer</rdfs:label>
    <emo:roleHasAuthority rdf:resource="#SwissCMSRightAuthority"/>
</emo:Role>
```

[22] StarUML - Open Source UML/MDA Platform http://staruml.sourceforge.net/en/ (Accessed on 2010-07-09)

The authority gives SwissSoft the right on a facility – in this case it is their own software SwissCMS:

```
<emo:Authority rdf:ID="SwissCMSRightAuthority">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >SwissCMSRightAuthority</rdfs:label>
    <emo:authorityHasRightsOnFacility rdf:resource="#SwissCMS"/>
</emo:Authority>
```

As mentioned above SwissSoft has a product called SwissCMS:

```
<emo:Product rdf:ID="SwissCMS">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >SwissCMS</rdfs:label>
</emo:Product>
```

The enterprise ontology of SwissSoft consists of the following RDF entries (cf. Figure 31) concerning the people and their roles in the SwissSoft AG:

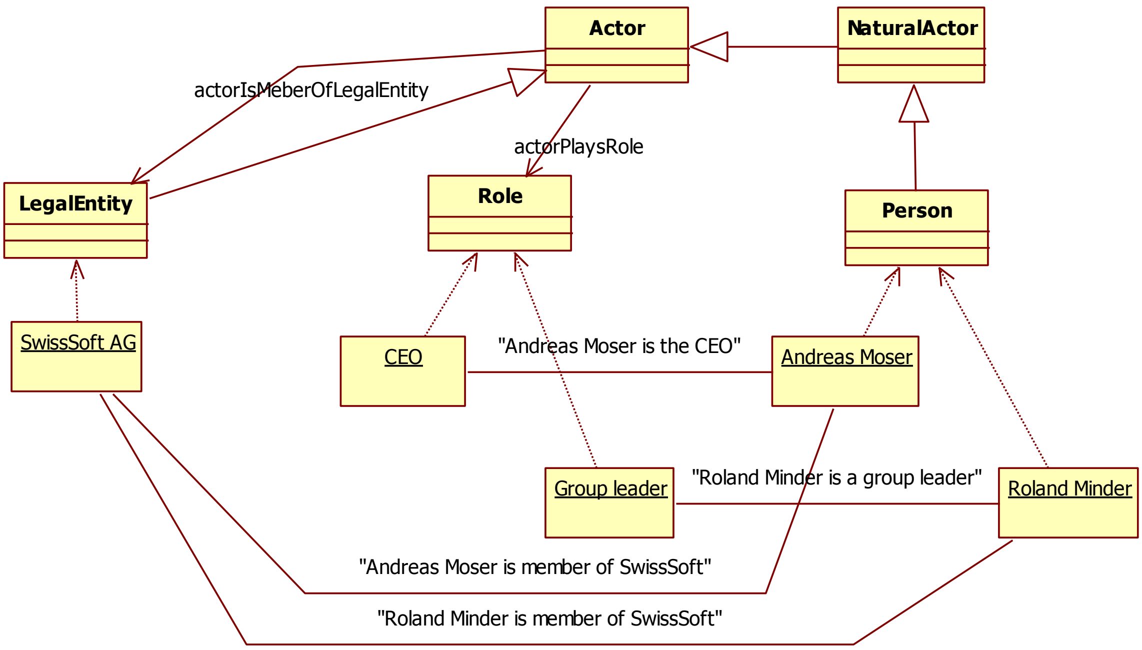

Figure 31: Application scenario: SwissSoft (Legal Entity, Role, Person)

Andreas Moser is a member (or employee) and the CEO of the SwissSoft AG.

```
<emo:Person rdf:ID="Andreas_Moser">
    <emo:actorIsMemberOfLegalEntity rdf:resource="#SwissSoft_AG"/>
    <emo:actorPlaysRole rdf:resource="#CEO"/>
```

```
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Andreas Moser</rdfs:label>
</emo:Person>
```

The role CEO is described as follows:

```
<emo:Role rdf:ID="CEO">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >CEO</rdfs:label>
</emo:Role>
```

Roland Minder is member of the SwissSoft AG and plays the role of a group leader.

```
<emo:Person rdf:ID="Roland_Minder">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Roland Minder</rdfs:label>
    <emo:actorIsMemberOfLegalEntity rdf:resource="#SwissSoft_AG"/>
    <emo:actorPlaysRole rdf:resource="#Group_Leader"/>
</emo:Person>
```

The role group leader is described as follows:

```
<emo:Role rdf:ID="Group_Leader">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Group leader</rdfs:label>
</emo:Role>
```

The enterprise ontology of SwissSoft consists of the following RDF entries (cf. Figure 33) concerning of the information about the SwissBikes AG:

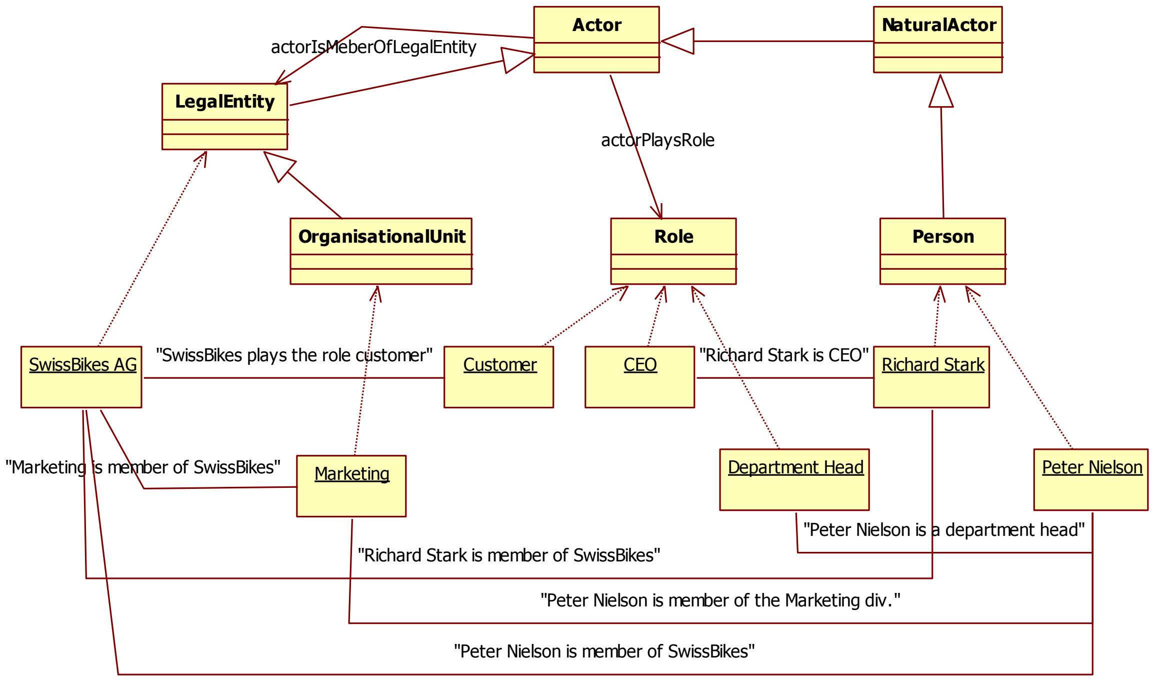

Figure 32: Application scenario: SwissBikes (Legal Entity, Org. Unit, Role, Person)

Swiss Bikes AG is a Legal Entity and plays the role as customer of SwissSoft.

```
<emo:LegalEntity rdf:ID="SwissBikes_AG">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >SwissBikes AG</rdfs:label>
    <emo:actorPlaysRole rdf:resource="#Customer"/>
</emo:LegalEntity>
```

The role customer is defined as follows:

```
<emo:Role rdf:ID="Customer">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Customer</rdfs:label>
</emo:Role>
```

In SwissBikes exists an organisational unit called marketing.

```
<emo:OrganisationalUnit rdf:ID="Marketing">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Marketing</rdfs:label>
    <emo:actorIsMemberOfLegalEntity rdf:resource="#SwissBikes_AG"/>
</emo:OrganisationalUnit>
```

Richard Stark is the CEO of SwissBikes.

```
<emo:Person rdf:ID="Richard_Stark">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Richard Stark</rdfs:label>
    <emo:actorPlaysRole rdf:resource="#CEO"/>
    <emo:actorIsMemberOfLegalEntity rdf:resource="#SwissBikes_AG"/>
</emo:Person>
```

The role CEO is described as follows (same definition as SwissSoft):

```
<emo:Role rdf:ID="CEO">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >CEO</rdfs:label>
</emo:Role>
```

Peter Nielson is department head of the SwissBikes division marketing.

```
<emo:Person rdf:ID="Peter_Nielson">
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Peter Nielson</rdfs:label>
    <emo:actorIsMemberOfLegalEntity rdf:resource="#SwissBikes_AG"/>
    <emo:actorPlaysRole rdf:resource="#Department_Head"/>
    <emo:personIsMemberOfOrganisationalUnit rdf:resource="#Marketing"/>
  </emo:Person>
```

The role department head is defined as follows:

```
<emo:Role rdf:ID="Department_Head">
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Department Head</rdfs:label>
</emo:Role>
```

7.2 EMO – Contract ontology

In order to implement the application scenario the enterprise model ontology needs be enhanced in a way that contracts can be represented. The enhancements were made considering the work of (Kabilan, Johannesson, & Rugaimukamu, 2003) and (Kabilan & Johannesson, 2003).

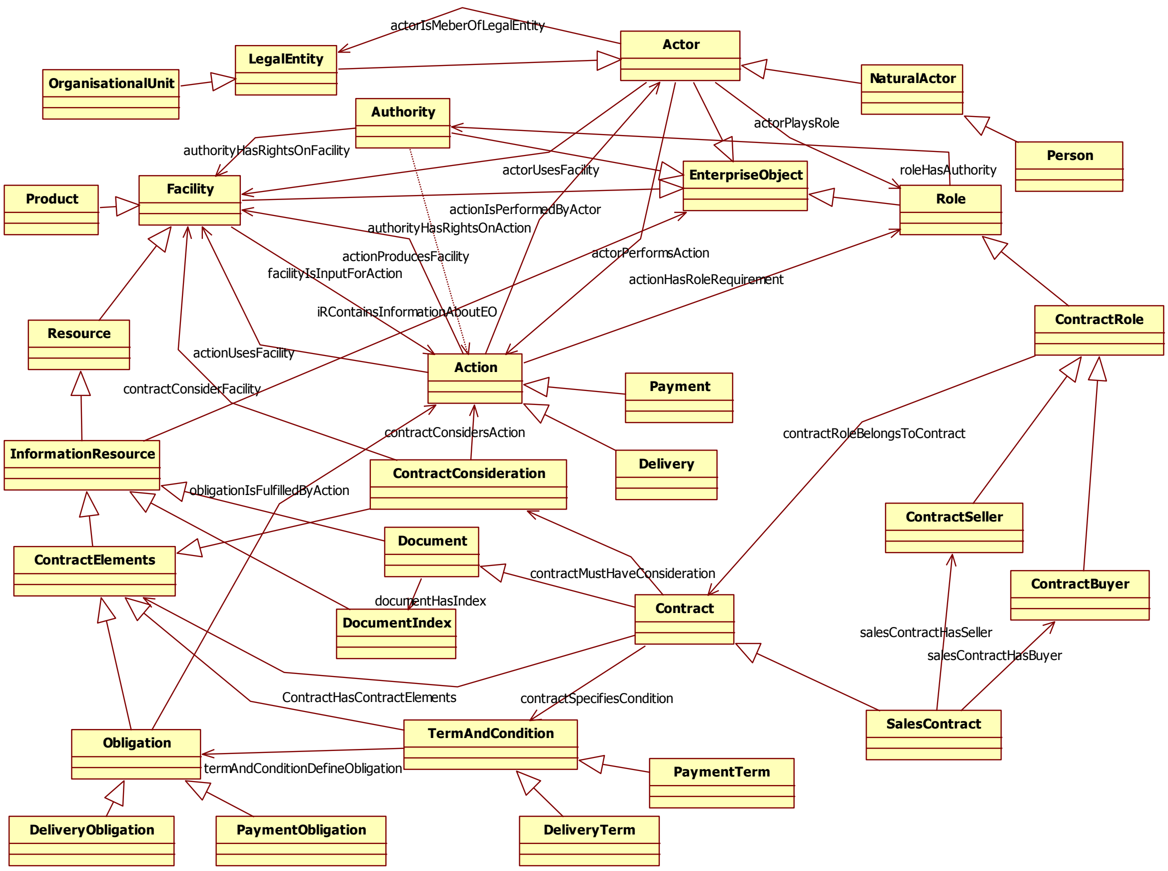

Figure 33: EMO including contract ontology

Figure 33 shows the enterprise model ontology including the concepts of the contract ontology. The integration was made by analysing every proposed concept in (Kabilan, Johannesson, & Rugaimukamu, 2003). It was not possible to integrate the proposed ontology one to one into the EMO, because the ontology from (Kabilan, Johannesson, & Rugaimukamu, 2003) has the purpose to represent one contract – meaning the contract is somehow the starting point of the ontology. This is the opposition to the EMO ontology were an enterprise or the enterprise object is the starting point.

The ontology is enhanced by the concept *contract*, which is a document containing *contract considerations*, *terms and conditions* and *contract elements* in general, and has a sub- concept a *sales contract* which has information about the *contract roles*, a *contract seller* role and a *contract buyer* role. As a reminder, every *actor* plays a *role*. That gives us the possibility to specify the *contract seller* and *contract buyer* of a *contract*. The concept *contract consideration* specifies the *facility* (e.g. product) or *action* (e.g. service) of the *contract*. The *terms and conditions* specify in a *contract* what has to be done e.g. we have a *payment term* or a *delivery term* in a *contract*. The *obligation* is something that says how the *terms and conditions* will be fulfilled by a specific *action*. The *contract considerations*, the *terms and conditions* and the *obligations* are consolidated using the concept *contract element* which is an *information resource*. An exemplarily sub concept of the obligation could be a *delivery obligation* or a *payment obligation*. Apart from the contract extensions the enterprise model ontology was enhanced by the concept *document index*. This concept is intended to store an index (e.g. an index as an outcome of an information extraction process see section 8.2) of a *document*.

7.3 Metadata rules

As described in the thesis statement and in section 6 (Description of the approach) the metadata will be created using rules. As already mentioned the application scenario is based on a real use case, to be more precise it is based on a real contract. In order to create the metadata rules some assumptions need to be made – like the input data, the rules are enterprise specific as well. Based on the input data the following rules come into consideration – as a reminder, the rules are created from the SwissSoft perspective:

7.3.1 Rule 1: Document has relation to product

It is assumed that, if a document contains information about SwissSoft products, then we can say that the document has a relation to a product of SwissSoft.

7.3.2 Rule 2: Document is a contract

It is assumed that, if the text of the document contains information about a contract, then the document is a contract.

7.3.3 Rule 3: Contract has a creator

It is assumed that, if a contract contains information about SwissSoft products and contains information about SwissSoft employees, then we can add the employees as document creator.

7.3.4 Rule 4: Contract has a contributor

It is assumed that, if a contract contains information about a person who is member of a company who is a customer of SwissSoft, then we can add this person as contributor to the contract.

7.3.5 Rule 5: Contract is a sales contract

It is assumed that, if a contract deals with a SwissSoft product and contains information about of our customer, then we say that the contract is a sales contract.

7.3.6 Rule 6: Sales contract is about our product

It is assumed that, if the sales contract contains information about the SwissSoft product, then we can add the information to the document that the sales contract considers that product.

7.3.7 Rule 7: Sales contract has a seller and a buyer

It is assumed that, if the sales contract is about the product of SwissSoft and about one of their customers, then we add SwissSoft as seller and the customer as buyer.

8 Implementation

As proposed in the thesis statement, the LEMO / metadata approach will be prototypical implemented in a software demonstrator. The demonstrator should consider the following requirements:

- The demonstrator should be able to deal with semantic technologies like ontological models, semantic queries and rules based on semantic web rule languages.
- The demonstrator should extract existing metadata from documents.
- The demonstrator should optionally extract existing information out of the content of the documents.
- The demonstrator should store the result.

Based on the requirements the following technologies (see Figure 34) were identified:

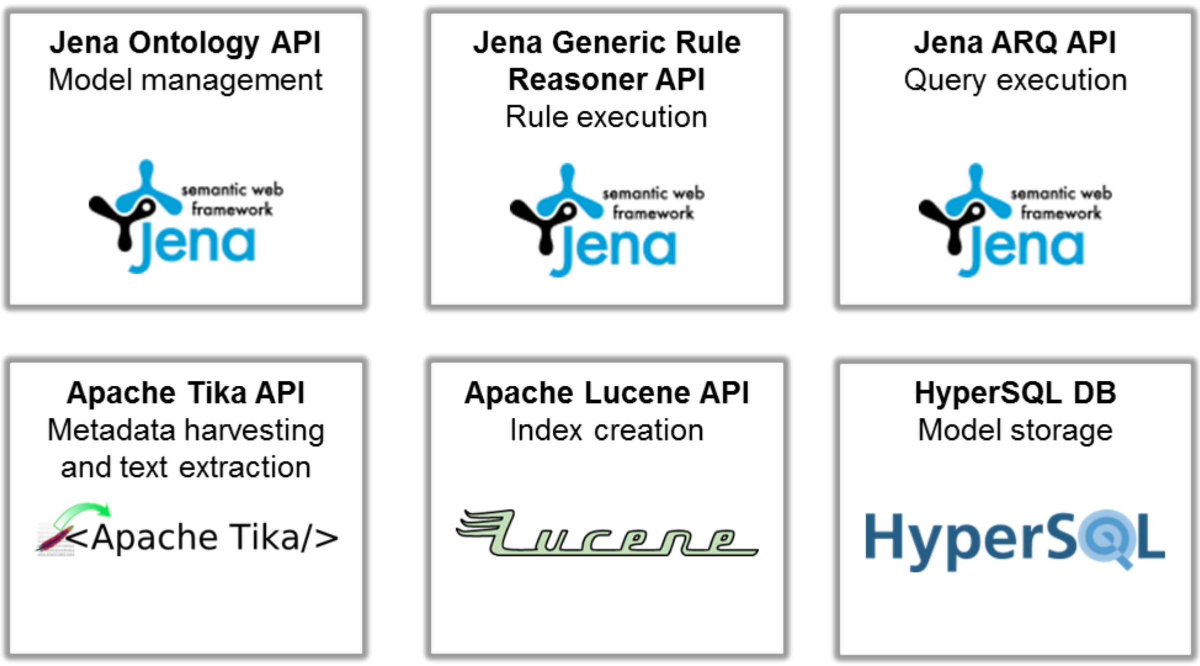

Figure 34: Used technologies for the implementation

The **Jena Ontology API** (Dickinson, 2009) gives the possibility to read and write RDF and OWL ontologies using Java. The **Jena Generic Rules Reasoner API** (Reynolds, 2010) gives the possibility to execute Jean Rules using a forward and backward chaining engine. The forward chaining rule engine is based on the standard RETE algorithm (Forgy, 1982). The backward chaining rule engine is based on logic programming (LP). The **Jena ARQ** gives the possibility to execute SPARQL queries. The **Apache Tika API** "[…]is a toolkit for detecting and [harvesting] metadata and structured text content from various documents using existing parser libraries." (The Apache Software Foundation, 2010). The **Apache Lucene API** is in general "a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform." (The Apache Software Foundation, 2009) Here in this thesis, it is used for creating an index of a document. And finally the **HyperSQL DB** (HSQL Development Group, 2010) is a SQL relational database engine written in Java. It is used as storage for the ontological models.

8.1 Explaining the demonstrator with the GUI

In order to show a proof of concept and tackle the complexity of creating rules the demonstrator called LEMO Workbench has been created. It is implemented in Java and a graphical user interface (GUI) is created using the Java library Swing.

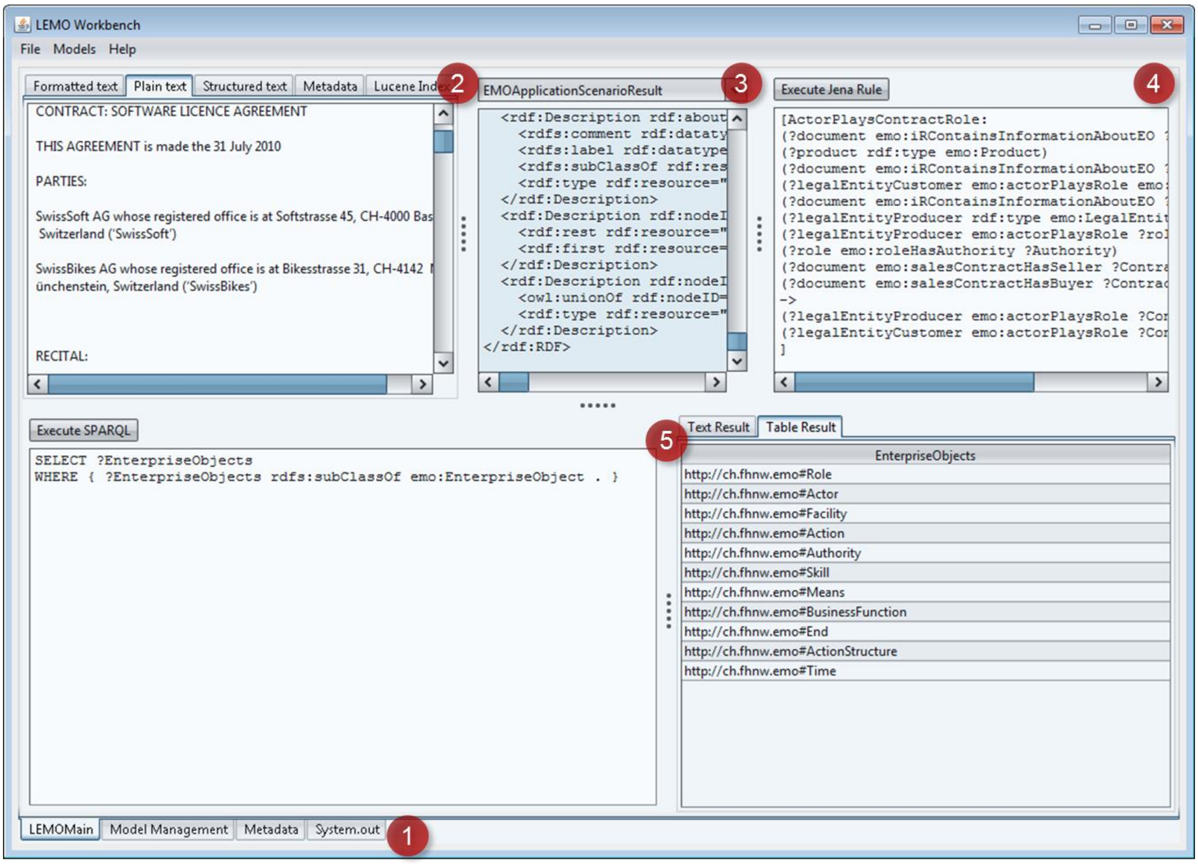

Figure 35: LEMO Workbench GUI

The LEMO Workbench consists of the following main parts (please follow the numbers in Figure 35):

1. The LEMO Workbench consists of four main tabs.
    a. The first one contains the main view of the Workbench.
    b. The second one called model management provides several functions to manage several ontological Jena models. It is possible to create and delete models from the system memory or database as well as importing models from RDF, RDFS and OWL files. The model management

tab gives also the possibility to connect to a HyperSQL Database and provides further methods for copying the models.

c. The metadata tab shows the created metadata based on the LEMO and LEMO contract ontology (see section 9 (Evaluation using an application scenario) for details).

d. The fourth tab shows every "System.out" message in a box. This tab is mainly used for debugging purposes.

2. As mentioned the LEMO Workbench uses the Apache Tika API for extracting existing metadata and text from a document. Number two of the Figure 35 points to the Apache Tika tabs: The formatted text tab presents the content of a document with format information; the plain text tab returns the content of a document as text stream; the structured text tab presents the content including structuring information like page brakes etc.; the metadata tab shows the existing metadata of the document; the lucene index tab shows the generated index using the Lucene API based on text stream of Apache Tika API. All tabs except the lucene tab are based on the Apache Tika GUI (The Apache Software Foundation, 2010) and has been modified and integrated by the author of this thesis.

3. The third number points to the model selection combo box and model output. The model selection combo box shows all Jena models which are available. If a model is selected, the model will be show in the model output text field and the selected model will be the "current" model on which all operations of the LEMO Workbench will executed.

4. The fourth number points to the rule execution section. The Jena Rules can be entered into the text box and executed by pressing the execution button. The text box provides to possibility to crate comments using double-slash (//) or slash-star-star-slash (/* */) mark as used in Java. It is also possible to execute only the selected rule.

5. The fifth number points to the SPARQL section. The text box has the same functionality as the rule execution text box. But it is made for executing SPARQL queries by pressing the execution button. The result of the executed

query can be shown as plain text output on the text result tab or as table out put on the table result tab.

8.2 Document import into the ontology

As mentioned in section 1.3.5 (Delineations and limitations) the thesis will not deal with information extraction details. Nevertheless, the documents and the content need to find their way into the ontology. Therefore an import needs to be implemented.

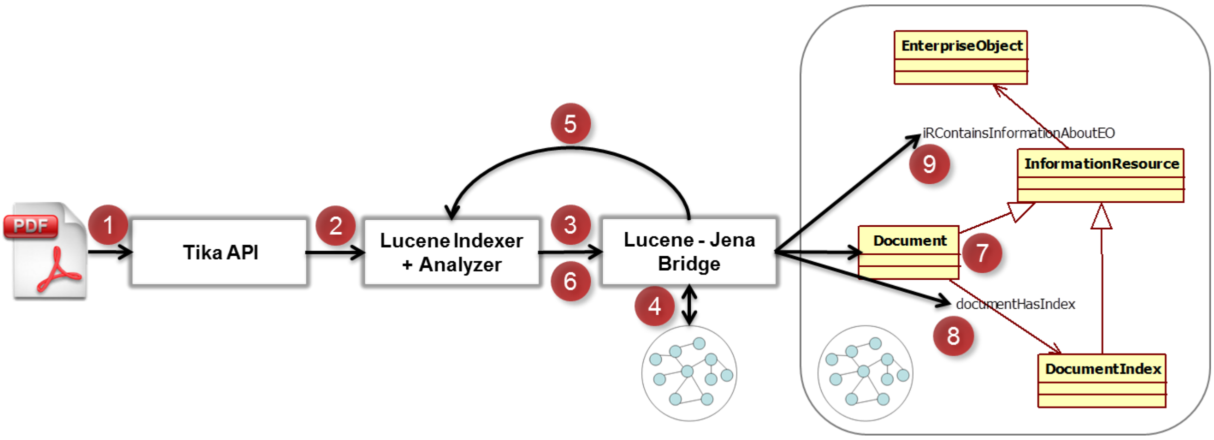

Figure 36: Document import into LEMO

Figure 36 gives an overview over the import process that can be described as follows (please follow the numbers in Figure 36):

1. A document can be imported via drag-and-drop approach into the LEMO Workbench. The document needs to be dropped into an Apache Tika tab shown in Figure 35. After dropping the document into the LEMO Workbench Tika starts with analysing the document. This means that the existing metadata and the content will be extracted.

2. The extracted document content will be analysed using the Lucene API. The text will be tokenized using a paragraph and a sentence tokenizer based on the Unicode Text Segmentation (Davis, 2009) approach originally implemented by (Pal, 2010) and modified by the author of this thesis.

3. After splitting the text into sentences, the content will be analysed using several analysers. The content will be filtered using the following filters: a stop set parser using an english stop set (Caron, 2002); the Lucene standard tokenizer

(The Apache Software Foundation, 2009) which splits words at punctuation characters, removing punctuation, hyphens and recognizes email addresses and internet hostnames; a numeric token filter will remove all numbers from the content; and a Lucene lower case filter (The Apache Software Foundation, 2009) will normalize the text to lower case. After doing that the index will transferred to the Lucene – Jena Bridge.

4. The Lucene – Jena Bridge will then try to find a corresponding label entry in the ontology based on the following SPARQL query:

```
SELECT ?labelEntry ?instance ?type
WHERE { ?instance rdfs:label ?labelEntry.
?instance rdf:type ?type.
FILTER regex(str(?labelEntry), <Java.String.instance> , "i").
FILTER (afn:namespace(?type) = 'http://ch.fhnw.emo#' ).
FILTER (afn:namespace(?instance) = 'http://ch.fhnw.emo#' ). }
```

   The result is a second index called ontology index represented as a list of strings.

5. The ontology index will be transferred back to the Lucene Indexer + Analyzer will be analysed using the same algorithm as described in point 3. This will be done in order to have the terms in the ontology index in the same "form" as in the document index.

6. The "analysed" ontology index will now be "compared" with the document index. The terms who are similar will be added to the final index and again transferred to the Lucene – Jena Bridge.

7. The Lucene – Jena Bridge will first create in the ontology an instance of the EMO class document and add the document name, provided from Tika, as identifier.

8. Then the Lucene – Jena Bridge will create an instance of the document index and will add the final index terms as properties to the document index.

9. Finally the Lucene – Jena Bridge will link the document with the Enterprise Objects according to the final index.

8.3 Short recap

Now we have seen that the needed technology is identified and implemented as demonstrator including a graphical user interface. Documents can be imported, ontological models can be created, Jena Rules and SPARQL queries can be executed. This is done according to the thesis statement and the corresponding sub-goals. Despite the exclusion in section 1.3.5 (Delineations and limitations), the demonstrator implements an information extraction method – even if it is a simple one. But it is clear that the information extraction part should be enhanced in future work.

9 Evaluation using an application scenario

After implementing the demonstrator the whole approach will be evaluated using the application scenario.

9.1 Creating Jena Rules

Before executing the approach, the defined "textual" rules in section 7.3 (Metadata rules) from the application scenario needs to be translated into Jena Rules.

9.1.1 Rule 1: Document has relation to product

The original rule "if a document contains information about SwissSoft products, then we can say that the document has a relation to a product of SwissSoft." can be translated as follows:

```
[Document_relation_to_product:
(?document rdf:type emo:Document)
(?document emo:iRContainsInformationAboutEO ?legalEntity)
(?legalEntity rdf:type emo:LegalEntity)
(?legalEntity emo:actorPlaysRole ?role)
(?role emo:roleHasAuthority ?authority)
(?authority emo:authorityHasRightsOnFacility ?product)
(?document emo:iRContainsInformationAboutEO ?product)
(?product rdf:type emo:Product)
->
(?document dc:relation ?product)
]
```

1. First we look if a document contains information about a legal entity.
2. If we find a legal entity that plays a role and this role has the authority which has the right on a facility.
3. And if this facility is a product.
4. Then we say that the document has a Dublin Core relation to that product.

9.1.2 Rule 2: Document is a contract

The original rule "if the text of the document contains information about a contract, then the document is a contract." can be translated as follows:

```
[Contract_instantiation:
(?document rdf:type emo:Document)
(?document emo:documentHasIndex ?documentIndex)
(?documentIndex rdf:type emo:DocumentIndex)
(?documentIndex emo:indexEntry ?indexEntry)
equal(?indexEntry, "contract")
->
(?document rdf:type emo:Contract)
]
```

1. If the document has an index entry which contains the term contract.
2. Then we can define that the document is as contract by associating it to the class contract.

9.1.3 Rule 3: Contract has a creator

The original rule "if a contract contains information about SwissSoft products and contains information about SwissSoft employees, then we can add the employees as document creator." can be translated as follows:

```
[Contract_creator:
(?document rdf:type emo:Contract)
(?document emo:iRContainsInformationAboutEO ?person)
(?person rdf:type emo:Person )
(?person emo:actorIsMemberOfLegalEntity ?legalEntity)
(?legalEntity emo:actorPlaysRole ?role)
(?role emo:roleHasAuthority ?Authority)
(?Authority emo:authorityHasRightsOnFacility ?product)
(?document emo:iRContainsInformationAboutEO ?product)
(?product rdf:type emo:Product)
->
(?document dc:creator ?person)
]
```

1. If the document contains information about a person who is member of a legal entity.

2. This legal entity that plays a role and this role has the authority that has the right on a facility.
3. And this facility which is a product and is mentioned in the contract.
4. Then we can add the person as creator of this document.

9.1.4 Rule 4: Contract has a contributor

The original rule "if a contract contains information about a person who is member of a company who is a customer of SwissSoft, then we can add this person as contributor to the contract." can be translated as follows:

```
[Contract_contributor:
(?document rdf:type emo:Contract)
(?document emo:iRContainsInformationAboutEO ?person)
(?person rdf:type emo:Person )
(?person emo:actorIsMemberOfLegalEntity ?legalEntity)
(?legalEntity emo:actorPlaysRole emo:Customer)
->
(?document dc:contributor ?person)
]
```

1. If the document is a contract containing information about a person.
2. If this person is member of a legal entity which plays the role of a customer.
3. Then the person will be added as contributor to the document.

9.1.5 Rule 5: Contract is a sales contract

The original rule "if a contract deals with a SwissSoft product and contains information about one of our customer, then we say that the contract is a sales contract." can be translated using forward and backward chaining (hybrid) as follows:

```
[SalesContract_instantiation:
(?document emo:iRContainsInformationAboutEO ?product)
(?document rdf:type emo:Contract)
(?product rdf:type emo:Product)
(?document emo:iRContainsInformationAboutEO ?legalEntityCustomer)
(?legalEntityCustomer emo:actorPlaysRole emo:Customer)
(?document emo:iRContainsInformationAboutEO ?legalEntityProducer)
(?legalEntityProducer rdf:type emo:LegalEntity)
(?legalEntityProducer emo:actorPlaysRole ?role)
```

```
(?role emo:roleHasAuthority ?Authority)
(?Authority emo:authorityHasRightsOnFacility ?product)
->
(?document rdf:type emo:SalesContract)
[[ContracConsideration:
(?ContractConsiderationInstanceName rdf:type
emo:ContractConsideration) <-
uriConcat(?document,".ContractConsideration",
?ContractConsiderationInstanceName),
makeInstance(?ContractConsiderationInstanceName, rdf:type,
emo:ContractConsideration, ?ContracConsiderationInstance)]
[ContractConsideration_SalesContract:
(?document emo:contractMustHaveConsideration
?ContracConsiderationInstance) <- (?ContracConsiderationInstance
rdf:type emo:ContractConsideration)]
[ContractSeller:
(?ContractSellerInstanceName rdf:type emo:ContractSeller) <-
uriConcat(?document,".ContractSeller",?ContractSellerInstanceName),
makeInstance(?ContractSellerInstanceName, rdf:type,
emo:ContractSeller, ?ContractSellerInstance)]
[ContractSeller_SalesContract:
(?document emo:salesContractHasSeller ?ContractSellerInstance) <-
(?ContractSellerInstance rdf:type emo:ContractSeller)]
[ContractBuyer:
(?ContractBuyerInstanceName rdf:type emo:ContractBuyer) <-
uriConcat(?document,".ContractBuyer",?ContractBuyerInstanceName),
makeInstance(?ContractBuyerInstanceName, rdf:type,
emo:ContractBuyer, ?ContractBuyerInstance)]
[ContractBuyer_SalesContract:
(?document emo:salesContractHasBuyer ?ContractBuyerInstance) <-
(?ContractBuyerInstance rdf:type emo:ContractBuyer)]
]
```

1. If we have a product, a customer and a document which is already a contract. And if we have a legal entity that has the right on the product, defined over the role and authority.
2. Then we can define that the contract is a sales contract by adding to sales contract type.
3. If the sales contract type is added, it is needed that the contract consideration type will be instantiated.
4. And the contract consideration instance will then be linked to document.
5. If the sales contract type is added, it is needed that the contract seller type will be instantiated.
6. And the contract seller instance will then be linked to document.

7. Finally, if the sales contract type is added, it is needed that and contract buyer type will be instantiated.
8. And the contract buyer instance will then be linked to document.

9.1.6 Rule 6: Sales contract is about our product

The original rule "if the sales contract contains information about the SwissSoft product, then we can add the information that the sales contract considers their product." can be translated as follows:

```
[ContractConsiderFacility:
(?document emo:iRContainsInformationAboutEO ?product)
(?product rdf:type emo:Product)
(?document rdf:type emo:SalesContract)
(?document emo:contractMustHaveConsideration
?ContractConsideration)
->
(?ContractConsideration emo:contractConsiderFacility
?product)
]
```

1. If the document is a sales contract which contains information about product.
2. And the document has a contract consideration.
3. Then we can add the product as contract consideration.

9.1.7 Rule 7: Sales contract has a seller and a buyer

The original rule "if the sales contract is about the product of SwissSoft and the about one of their customer, then we add SwissSoft as seller and the customer as buyer." can be translated as follows:

```
[ActorPlaysContractRole:
(?document emo:iRContainsInformationAboutEO ?product)
(?product rdf:type emo:Product)
(?document emo:iRContainsInformationAboutEO ?legalEntityCustomer)
(?legalEntityCustomer emo:actorPlaysRole emo:Customer)
(?document emo:iRContainsInformationAboutEO ?legalEntityProducer)
(?legalEntityProducer rdf:type emo:LegalEntity)
(?legalEntityProducer emo:actorPlaysRole ?role)
(?role emo:roleHasAuthority ?Authority)
```

```
(?document emo:salesContractHasSeller ?ContractSeller)
(?document emo:salesContractHasBuyer ?ContractHasBuyer)
->
(?legalEntityProducer emo:actorPlaysRole ?ContractSeller)
(?legalEntityCustomer emo:actorPlaysRole ?ContractHasBuyer)
]
```

1. If we a product.
2. And a customer.
3. And if we have a legal entity that has the right on the product, defined over the role and authority.
4. And of course the document which is already a sales contract (because it has a sales contract has… relation).
5. Then we can add the customer as buyer and the producer who has the right on the product as seller.

9.2 Running the evaluation on the LEMO Workbench

The application scenario is defined, the demonstrator is developed and the rules are translated. Now the demonstrator is ready the load the LEMO ontology including the instances from the application scenario and to execute the Jena Rules. As mentioned in section 8.1 (Explaining the demonstrator with the GUI) the demonstrator has a metadata tab which shows the created metadata related to a document.

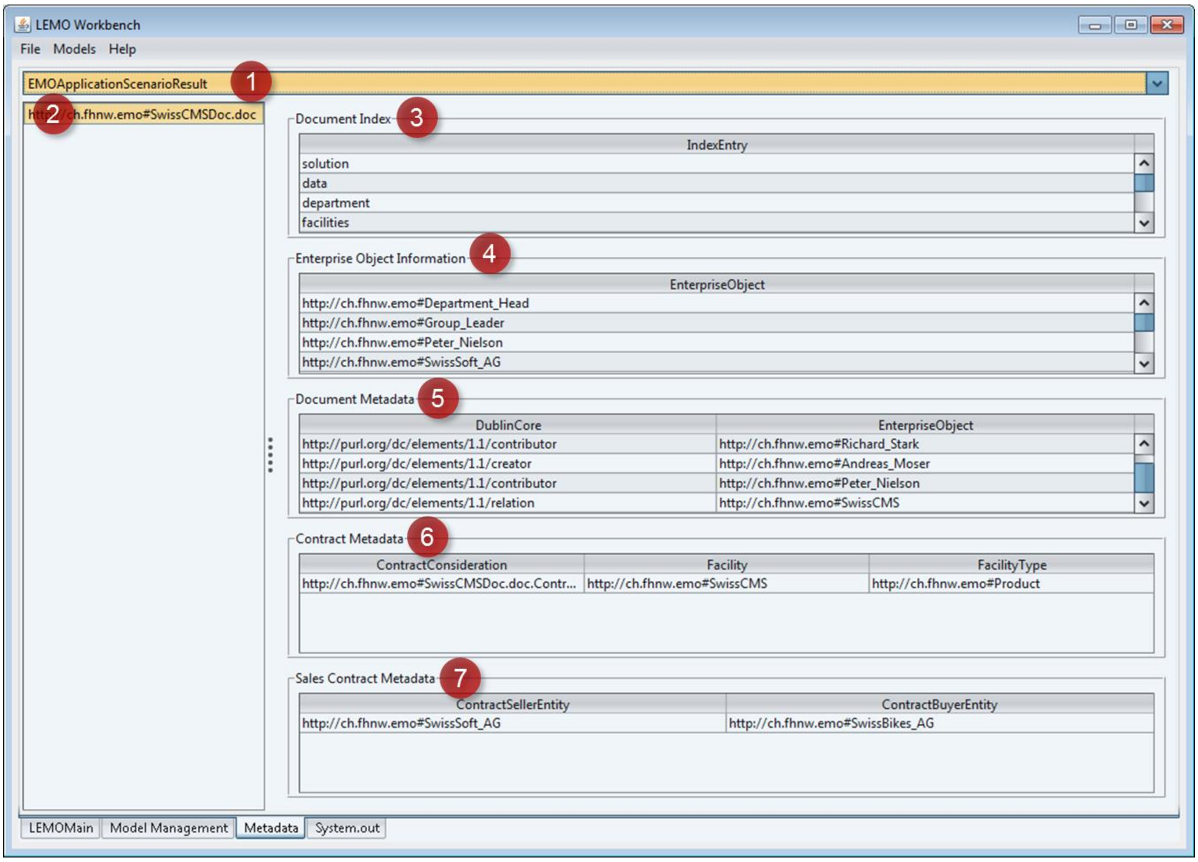

Figure 37: Evaluation results in the LEMO Workbench

Figure 37 shows the metadata tab containing the evaluation result. The functionality and the results can be explained as follows (please follow the numbers in Figure 37):

1. The available models can be selected in the LEMO Workbench using the combo box highlighted by number one.
2. Number two points to the imported documents which are available in the model.

3. The document index panel shows the index entries which are added to the ontology as explained in section 8.2 (Document import into the ontology).
4. The enterprise object information panel shows the related enterprise objects as explained in section 8.2 (Document import into the ontology).
5. The document metadata panel shows the linked enterprise objects using Dublin Core properties. This panel contains results from rule 1, 3 and 4.
6. The panel highlighted with number six is active if the document is a contract – this is the result of rule 2. And it shows the result when adding the contract consideration – this is the result of rule 5 and 6.
7. The panel highlighted with seven is active if the document is a sales contract – this is the result of rule 5. And shows the result when adding a contract seller or a contract buyer.

9.3 Recap the results

Now we have seen that it is possible to crate metadata using an ontology and rules. Again the information extraction part of the demonstrator is indeed very simple – but it is not the task of this thesis to evaluate information extraction methods. The lack of additional information and semantic delivered from the information extraction part, leads to metadata rules using many assumptions. Nevertheless, the evaluation shows us that it is possible to create metadata using an ontology and rules. And even more important we can say that it makes sense to capture all enterprise objects based on the Linked Enterprise Models and Objects approach.

10 Results and Future Work

10.1 Summary of findings

In order to summarise the thesis the requirements from the interviews are consulted.

The first requirement is that the thesis should elaborate a basic (metadata) structure. This can be seen as fulfilled. The proposed enterprise ontology allows to define own concepts and defines only a very basic structure. Through the usage of an ontology language it is possible to enhance the ontology in an easy way.

Even the second requirement from the interviews can be fulfilled with the enterprise ontology. It is indeed possible to create domain specific ontologies using the EMO. This was shown by the application scenario, where the EMO was enhanced by the contract ontology.

The thesis considers business processes in several ways. In the EMO, business processes are of high importance – function is one of the core concepts of the enterprise ontology. Business processes can also be represented as a model type and they can also be used in the metadata creation part as described in future work.

As shown in the application scenario, it is possible to create several "metadata sets" for several document types. This means that it is possible to create for example for a contract a specific ontology and specific rules.

10.2 Conclusion

Several questions came up during the thesis work.

The first question that came up was, if it is possible to create a formally description of a whole enterprise. Short answer – yes it is. The Enterprise Model Ontology enables to cover many aspects of an enterprise. It is even possible to have different views on the enterprise using the aspect and perspective "filters" from the EMO. The LEMO approach provides additionally a new view on the enterprise. It makes the relationships between the enterprise objects in some way visible.

The next question was, if it is possible to use an enterprise repository to create metadata. Again it is possible to answer with yes as the enterprise repository is able to deal with ontologies. Then the LEMO can be used in combination with the developed demonstrator, the LEMO Workbench.

Even the next question, if it is possible to use the LEMO approach to create metadata, can be answered with yes. The demonstrator and the application scenario showed that is possible to create metadata with only three main things – input data (e.g. extracted content information), some rules and of course the EMO.

The last question, if it makes it sense to model everything in an enterprise ontology, cannot be answered completely. There is always a matter of cost and benefit relationship, which is different from enterprise to enterprise. But it is possible to say that it makes sense to model at least a part of the enterprise objects in an enterprise model ontology – as it is done in the application scenario.

10.3 Summary of contributions

The following contributions were made:

- First, the thesis contributes with an extensive literature review concerning enterprise architectures, ontologies and metadata creation.
- Secondly, the thesis provides a practical view concerning metadata. The interviews containing a respectable amount of practical information about how companies and their customers work concerning metadata.
- Further the thesis contributes with a new view on enterprise objects. Additionally the thesis defines new wording concerning enterprise ontologies.
- One of the most important contributions is the enterprise model ontology including all concepts for describing an enterprise.
- The description of the approach provides a concise overview of the metadata creation approaches.
- The most important contribution is the LEMO Workbench – a nice piece of software for creating, debugging and executing queries, rules and documents.

- And finally the evaluation, which shows how the whole approach can be executed.

10.4 Future work

There are four main points for future work:

10.4.1 Enhance the information extraction approach

As mentioned many times the information extraction part is not the main focus of this thesis – it is even excluded in the in section 1.3.5 (Delineations and limitations). Nevertheless it is slightly considered in the thesis, because the proposed evaluation would not have been possible.

Section 6.1 (Ways for creating metadata) shows ways how to create metadata and gives an introduction in the information extraction approaches considering metadata creation. The information extraction part needs to be enhanced – further literature review needs to be made in this field.

10.4.2 Improve usability concerning rule creation

In section 9.1 (Creating Jena Rules), we have seen how the proposed rules from the application scenario have been translated to executable rules. It is somehow obvious, that creating Jena Rules is not very user friendly. Section 6.2 (Preliminary thoughts on rules) shows further methods for creating rules in a more user friendly way. Business rules and the proposed domain specific rules are promising approaches for creating rules in a more user appealing way.

10.4.3 Consider other context input

As defined in section 2.1.1 (What is context?), context is not only the information stored in the enterprise ontology. It could also be that an other system provides a source for the actual context.

Figure 38 shows an example which provides different source of context (please follow the numbers in Figure 38).

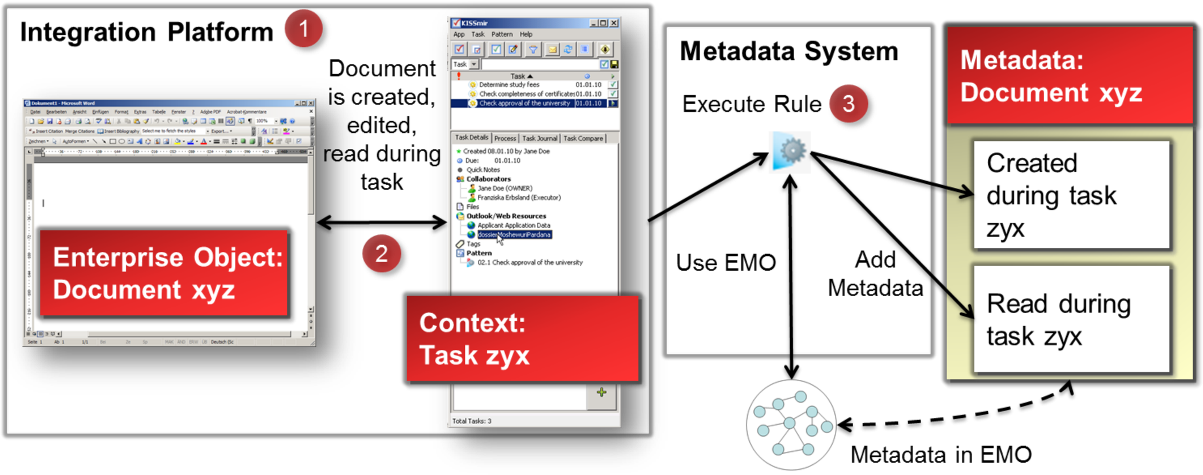

Figure 38: Example for context in future work

1. We assume that we have two systems (showed as part of the integration platform in the figure) who are integrated over a integration platform as proposed in (Martin & Brun, 2010) using a message server as described in (Nelkner, 2009). This gives the possibility that messages could be interchanged which delivers the actual context of each system.
2. Through the message exchange every systems knows from each other – and even more important the integration platform can trigger the metadata system and provides further context information.
3. The metadata system is then able to execute the rules using the additional context information.

An integration platform is one example that delivers additional context information. Another well known example could be position specific information provided from GPS devices.

10.4.4 Always improve the EMO and keeping alive

Last but not least an ontology should always be improved and adapted. Only the improvement ensures that the ontology is still based on common understanding. Therefore the author wishes that future projects will use this ontology. And even more important, enhance the ontology – adding more classes, more model types and enterprise architectures using aspects und perspectives (see (Brun, 2010)). This is the only way to keep the ontology alive. Good luck LEMO!

Bibliography

Allen, J., & Lehrer, N. (1992). DARPA/Rome Laboratory Planning and Scheduling Initiative Knowledge Representation Specification Language (KRSL).

Anaya, V., & Ortiz, A. (2005). How enterprise architectures can support integration. In *Proceedings of the first international workshop on Interoperability of heterogeneous information systems* (pp. 25-30). ACM.

Avison, D. E., Lau, F., Myers, M. D., & Nielsen, P. A. (1999). Action research. *Communications of the ACM*, *42*(1).

BOC-Group. (2010). BOC. Retrieved from http://www.boc-eu.com.

Baclawski, K., Kokar, M., Waldinger, R., & Kogut, A. (2002). *Consistency Checking of Semantic Web*.

Behrens, H., Clay, M., Efftinge, S., Eysholdt, M., Friese, P., Köhnlein, J., et al. (2010). *Xtext User Guide*. Retrieved from http://www.eclipse.org/Xtext/documentation/1_0_0/xtext.pdf.

Bergman, M., & Paavola, S. (2003). The Commens Dictionary of Preirce's Terms. Retrieved from http://www.helsinki.fi/science/commens/terms/universedisc.html.

Bernard, S. (2009). The Importance of Formal Documentation In Enterprise Architectures. *Journal of Enterprise Architecture*, *5*, 28-57.

Berners-Lee, T. (2006). Linked Data - Design Issues. Retrieved from http://www.w3.org/DesignIssues/LinkedData.html.

Bertolazzi, P., Krusich, C., & Missikoff, M. (2001). An approach to the definition of a core enterprise ontology: CEO. In *OES-SEO 2001, International Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations* (p. 14–15).

Borch, S. E., & Stefansen, C. (2004). Evaluating the REA enterprise ontology from an operational perspective. In *Proceedings of the EMOI-INTEROP'04 - Enterprise Modelling and Ontologies for Interoperability Workshop*.

Born, M., Markovic, I., Filipowska, A., Starzecka, M., Kaczmarek, M., Walczak, A., et al. (2008). Business Functions Ontology and its Application in Semantic Business Process Modelling. In *ACIS 2008 Proceedings*. 136-145.

Braun, C., & Winter, R. (2005). A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform (pp. 64-79). Klagenfurt: Gesellschaft für Informatik.

Breitman, K. K., Casanova, M. A., & Truszkowski, W. (2007). *Semantic Web: concepts, technologies and applications* (p. 327). Springer.

Brun, R. (2010). Linked Enterprise Models and Objects representing Content and Context of an Enterprise.

Caron, J. (2002). English Stop Words. Retrieved from http://onjava.com/onjava/2003/01/15/examples/EnglishStopWords.txt.

Chen, Z., & Pooley, R. (2009). Rediscovering Zachman Framework Using Ontology from a Requirement Engineering Perspective. In *2009 33rd Annual IEEE International Computer Software and Applications Conference* (pp. 3-8). IEEE. doi: 10.1109/COMPSAC.2009.107.

Concise Oxford English Dictionary. (2008). *The Concise Oxford English Dictionary*. (C. Soanes & A. Stevenson) (Twelfth ed.). Oxford University Press.

Cranefield, S., & Purvis, M. (1999). UML as an ontology modelling language. In D. Fensel, C. Knoblock, N. Kushmeric, & M. Rousset, *Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99)* (Vol. 212). Amsterdam, The Netherlands.

Davis, M. (2009). Unicode Text Segmentation. Retrieved from http://unicode.org/reports/tr29/.

De Morgan, A. (1846). On the Structure of the Syllogism. I. *Transactions of the Cambridge Philosophical Society*, *8*, 379–408.

De Villiers, M. R. (2005). Three approaches as pillars for interpretive information systems research: development research, action research and grounded theory. *SAICSIT; Vol. 150*.

Dey, A. K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing*, 4-7.

Dey, A. K., & Abowd, G. D. (1999). *Towards a better understanding of context and context-awareness* (pp. 304-307). In HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing. doi: 10.1.1.150.4833.

Dickinson, I. (2009). Jena Ontology API. Retrieved from http://jena.sourceforge.net/ontology/.

Dietz, J. L. (2006). *Enteprise Ontology Theory and Methodology*. Berlin / Heidelberg / New York: Springer.

Digital Library Federation. (2010). *METS: The Metadata Encoding and Transmission Standard*. *Cataloging & Classification Quarterly* (Vol. 40). Digital Library Federation.

Dublin Core Metadata Initiative. (2010). DCMI Specifications. Retrieved from http://dublincore.org/specifications/.

Encoded Archival Description Working Group. (2003). *Encoded Archival Description Tag Library, Version 2002*. *The Moving Image* (Vol. 3). Chicago: Society of American Archivists. doi: 10.1353/mov.2003.0022.

Eriksson, H. (2007). The semantic-document approach to combining documents and ontologies. *International Journal of Human-Computer Studies*, *65*(7), 624-639. doi: 10.1016/j.ijhcs.2007.03.008.

Filipowska, A., Kaczmarek, M., & Markovic, I. (2008). Organizational Ontologies to Support Semantic Business Process Management. In *19th Australasian Conference on Information Systems*. Christchurch.

Forgy, C. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, *19*(1), 17-37. doi: 10.1016/0004-3702(82)90020-0.

Fox, M. S., Barbuceanu, M., Grüninger, M., & Lin, J. (1995). An Organization Ontology for Enterprise Modelling: Preliminary concepts for linking structure and behaviour. *Simulating Organizations: Computational Models of Institutions and Groups*, *29*, 123-134. doi: 10.1016/0166-3615(95)00079-8.

Fox, M. S., Chionglo, J. F., & Fadel, F. G. (1993). A common-sense model of the enterprise. In *Proceedings of the 2nd Industrial and Engineering Applications of Artificial Intelligence and Expert Systems* (pp. 25-34). Berlin / Heidelberg: Springer.

Geerts, G., & McCarthy, W. (2000). The ontological foundation of REA enterprise information systems. *American Accounting Association Conference*.

Giles, C., Manavoglu, E., & Fox, E. (2003). *Automatic document metadata extraction using support vector machines*. *2003 Joint Conference on Digital Libraries, 2003. Proceedings.* (pp. 37-48). IEEE Comput. Soc. doi: 10.1109/JCDL.2003.1204842.

Green, M., & Miller, S. D. (2007). *Multitier Portal Architecture for Thin- and Thick-Neutron Scattering Experiment Support*. *International Workshop on Grid Computing Environments*.

Greenberg, J. (2004). Metadata Extraction and Harvesting: A Comparison of Two Automatic Metadata Generation Applications. *Journal of Internet Cataloging*, *6*(4), 59-82. Taylor & Francis. doi: 10.1300/J141v06n04_05.

Greenberg, J., Spurgin, K., & Crystal, A. (2005). Final Report for the AMeGA (Automatic Metadata Generation Applications) Project. *Library*.

Grembergen, W. V. (2004). *Strategies for information technology governance*. (W. V. Grembergen) (p. 390). Idea Group Inc (IGI).

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Systems Laboratory Technical Report KSL 92-71*.

Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing? In N. Guarino & R. P. Kluwer, *Formal Ontology in Conceptual Analysis and Knowledge Representation* (Vol. 43). Padova, Italy. doi: 10.1.1.91.6025.

Grüninger, M., Atefi, K., & Fox, M. S. (2000). Ontologies to Support Process Integration in Enterprise Engineering. *Computational & Mathematical Organization Theory*, *6*(Number 4), 381-394. doi: 10.1023/A:1009610430261.

Guarino, N. (1997). Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. In *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology* (p. 139–170). Springer.

Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). *Ontological engineering: with examples from the areas of knowledge ...* (p. 403). Springer.

HSQL Development Group. (2010). HSQLDB. Retrieved from http://hsqldb.org/.

Handschuh, S., Staab, S., & Ciravegna, F. (2002). S-CREAM—Semi-automatic CREAtion of Metadata. In A. Gómez-Pérez & V. R. Benjamins, *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, Lecture Notes in Computer Science (Vol. 2473, pp. 165-184). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/3-540-45810-7.

Handschuh, S., Staab, S., & Studer, R. (2003). Leveraging Metadata Creation for the Semantic Web with CREAM. In A. Günter, R. Kruse, & B. Neumann, *KI 2003: Advances in Artificial Intelligence*, Lecture Notes in Computer Science (Vol. 2821, pp. 19-33). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/b13477.

Hanschke, I. (2010). Strategic IT Management: A Toolkit for Enterprise Architecture Management.

Hinkelmann, K., Merelli, E., & Thönssen, B. (2010). The Role of Content and Context in Enterprise Repositories. In *Applied Sciences*.

Horrocks, I. (2005). Applications of Description Logics: State of the Art and Research Challenges. In F. Dau, M. Mugnier, & G. Stumme, *13th International Conference on Conceptual Structures*. Kassel, Germany.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. Retrieved from http://www.w3.org/Submission/SWRL/.

IEEE. (2000). IEEE-Std-1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems. *IEEE, http://standards. ieee. org*. New York.

IFLA. (2005). Digital Libraries: Metadata Resources. Retrieved from http://archive.ifla.org/II/metadata.htm.

International Council on Archives (ICA). (2000). *ISAD(G): General International Standard Archival Description, Second edition*. Ottawa. Retrieved from http://www.ica.org/biblio/isad_g_2e.pdf.

International Council on Archives (ICA). (2004). *ISAAR(CPF): International Standard Archival Authority Record for Corporate Bodies, Persons, and Families, Second edition*.

International Organization for Standardization (ISO). (2006). ISO 20652:2006. Retrieved from http://www.iso.org/iso/catalogue_detail.htm?csnumber=39577.

Ivanov, K. (2010). Enterprise Architecture with ARIS and TOGAF9. Retrieved from http://www.ariscommunity.com/users/koiv/2010-02-09-enterprise-architecture-aris-and-togaf9.

JBoss Community. (2010). Drools Expert (rule engine). Retrieved from http://downloads.jboss.com/drools/docs/5.0.1.26597.FINAL/drools-expert/html_single/index.html.

Janusch, W., Filipowska, A., Hepp, M., Kaczmarek, M., Kowalkiewicz, M., Markovic, I., et al. (2008). Project IST 026850 SUPER Semantics Utilized for Process management within and between Enterprises. *Security*.

Jonkers, H., Lankhorst, M., van Buuren, R., Hoppenbrouwers, S., Bonsangue, M., van Der Torre, L., et al. (2004). Concepts for modeling enterprise architectures. *International Journal of Cooperative Information Systems (IJCIS)*, 257-287.

Kabilan, A., Johannesson, P., & Rugaimukamu, D. (2003). An Ontological Approach to Unified Contract Management.

Kabilan, V., & Johannesson, P. (2003). Semantic Representation of Contract Knowledge using Multi tier Ontology. *SWDB*, *2003*, 395--414.

Kang, D., Lee, J., Choi, S., & Kim, K. (2010). An ontology-based Enterprise Architecture. *Expert Systems with Applications*, *37*(2), 1456-1464. doi: 0.1016/j.eswa.2009.06.073.

Karagiannis, D. (1995). BPMS: business process management systems. *ACM SIGOIS Bulletin*, *16*(1).

Karagiannis, D., & Kühn, H. (2002). Metamodelling platforms. *Lecture Notes in Computer Science*, 182–197. Citeseer.

Karagiannis, D., Junginger, S., & Strobl, R. (1996). lntroduction to Business Process Management Systems Concepts. *Business Process Modelling: Lecture Notes in Computer Science*.

Karagiannis, D., Ronaghi, F., & Fill, H. (2007). Business-oriented IT management: developing e-business applications with E-BPMS. *ACM International Conference Proceeding Series; Vol. 258*, 97.

Krcmar, H. (2004). Informationsmanagement 1 - Einleitung 2 - Ein Rahmen für Informationsmanagement 3 - Die Aufgaben des Informationsmanagement (4 ed., pp. 1-95; 478-489). Berlin / Heidelberg / New York: Springer.

Langenberg, K., & Wegmann, A. (2004). Enterprise Architecture: What Aspects is Current Research Targeting. *Laboratory of Systemic Modeling, EPFL*. Lausanne, Switzerland.

Lankhorst, M. (2005). *Enterprise architecture at work: modelling, communication, and analysis*.

Lassila, O., & McGuinness, D. (2001). The role of frame-based representation on the semantic web. Technical Report KSL-01-02.

Leppänen, M. (2005). *A Contextual-Based Enterprise Ontology*. *Information Systems* (p. 30). Enschede: EDOC International Workshop on Vocabularies, Ontologies and Rules for The Enterprise.

Leppänen, M. (2007). A Context Based Enterprise Ontology. In A. W., *10th International Conference on Business Information Systems (BIS 2007)* (pp. 273-286). Poznan, Poland: Springer Berlin / Heidelberg.

Levesque, H., & Brachman, R. (1985). A fundamental tradeoff in knowledge representation and reasoning. *Readings in knowledge representation*, 42–70.

Liddy, E. D., Allen, E., Harwell, S., Corieri, S., Yilmazel, O., Ozgencil, N. E., et al. (2002). Automatic metadata generation & evaluation. *Annual ACM Conference on Research and Development in Information Retrieval*, 401.

Malaxa, V., & Douglas, I. (2005). A Framework for Metadata Creation Tools. *Learning*, *1*(Interdisciplinary Journal of Knowledge and Learning Objects. Vol. 1), 151-162. doi: 10.1.1.89.8580.

Martin, A., & Brun, R. (2010). Agile Process Execution with KISSmir. In *5th International Workshop on Semantic Business Process Management collocated with 7th Extended Semantic Web Conference*. Heraklion, Greece.

McCarthy, J. (1993). *Notes On Formalizing Context* (pp. 555-560). Morgan Kaufmann. doi: 10.1.1.35.8117.

McGuinness, D. (2002). Ontologies come of age. *Spinning the semantic web: bringing the world wide web to its full potential*, 171–192.

Miller, E. (2001). An introduction to the resource description framework. *Journal of Library Administration*.

Mizoguchi, R., Vanwelkenhuysen, J., & Ikeda, M. (1995). Task ontology for reuse of problem solving knowledge. *Towards Very Large Knowledge Bases*.

Nelkner, T. (2009). *Best Practices for the Knowledge Society. Knowledge, Learning, Development and Technology for All*. *Best Practices for the Knowledge Society. Knowledge, Learning, Development and Technology for All*, Communications in Computer and Information Science (Vol. 49, pp. 41-48). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-04757-2.

Neubauer, T., & Stummer, C. (2007). Extending Business Process Management to Determine Efficient IT Investments. In *Proceedings of the 2007 ACM symposium on Applied computing* (pp. 1250-1256). New York, NY, USA: ACM.

Object Management Group. (2008). *Business Motivation Model*. *Business*. Object Management Group. Retrieved from http://www.omg.org/spec/BMM/1.0/PDF.

PREMIS Editorial Committee. (2008). *PREMIS Data Dictionary for Preservation Metadata*. *Preservation*.

Pal, S. (2010). Java Text Mining Toolkit. Retrieved from http://jtmt.sourceforge.net/.

Parsia, B., & Sirin, E. (2004). Pellet: An owl dl reasoner. In *Third International Semantic Web Conference-Poster*. Citeseer.

Pease, A. (2007). The Suggested Upper Merged Ontology (SUMO) at Age 7: Progress and Promise.

Peirce, C. S. (n.d.). *The Principles of Logical Graphics.*

Pragmatic EA Ltd. (2010). PEAF : Framework Comparison. *Framework*.

Prud'hommeaux, E., & Seaborne, A. (2008). SPARQL Query Language for RDF. Retrieved from http://www.w3.org/TR/rdf-sparql-query/.

Ravesteyn, P. (2009). A Situational Implementation Method for Business Process Management Systems A Situational Implementation Method for Business Process Management Systems. In *Proceedings of the Fifteenth Americas Conference on Information Systems,*. San Francisco, California.

Regoczei, S., & Plantinga, E. (1987). Creating the domain of discourse: ontology and inventory. *International Journal of Man-Machine Studies*, *27*(3), 235-250. doi: 10.1016/S0020-7373(87)80054-8.

Rehäuser, J., & Krcmar, H. (1996). Wissensmanagement im Unternehmen. *Managementforschung*.

Remenyi, D., & Williams, B. (1998). *Doing research in business and management: an introduction to process and method* (p. 309). SAGE.

Reynolds, D. (2010). Jena 2 Inference Support. Retrieved from http://jena.sourceforge.net/inference/.

Robson, C. (2002). *Real world research: a resource for social scientists and practitioner ...* (p. 599). Wiley-Blackwell.

Rohloff, M. (2005). Enterprise Architecture - Framework and Methodology for the Design of Architectures in the Large. In *European Conference on Information Systems (ECIS)*.

Ross, J. W., Weill, P., & Robertson, D. (2006). *Enterprise architecture as strategy: creating a foundation for business ...* (p. 234). Harvard Business Press.

Saunders, M., Lewis, P., & Thornhill, A. (2007). Research Methods for Business Students.

Scheer, A. (1991). *Architektur integrierter Informationssysteme : Grundlagen der Unternehmensmodellierung* (1 ed.). Berlin: Springer-Verlag. Retrieved from http://www.worldcat.org/title/architektur-integrierter-informationssysteme-grundlagen-der-unternehmensmodellierung/oclc/214323507&referer=brief_results.

Scheer, A. (2000). *ARIS - Business Process Modeling*. Springer.

Scheer, A. (2002). *ARIS - Vom Geschäftsprozess zum Anwendungssystem*. Springer.

Schekkerman, J. (2003). Thought Leader Business Technology Strategy & Enterprise Architecture.

Schilit, B., Adams, N., & Want, R. (1994). Context-Aware Computing Applications. In *1994 First Workshop on Mobile Computing Systems and Applications* (pp. 85-90). IEEE. doi: 10.1109/WMCSA.1994.16.

Schussel, G. (1996). Client/Server: Past, Present and Future. Retrieved from http://www.dciexpo.com/geos/dbsejava.htm.

Schwartz, C. (2001). Sorting out the Web: Approaches to Subject Access, 184.

Schönherr, M. (2004). Enterprise Architecture Frameworks. In M. Schönherr & S. Aier, *Enterprise Application Integration – Serviceorientierung und nachhaltige Architekturen.* (pp. 3-48). Berlin: Gito.

Sessions, R. (2006). A Better Path to Enterprise Architectures.

Sowa, J. F. (2000). *Knowledge representation: logical, philosophical, and computational foundations*. Brooks/Cole.

Staab, S. (2002). Wissensmanagement mit Ontologien und Metadaten. *Informatik-Spektrum*, *25*(3), 194-209. doi: 10.1007/s002870200226.

Studer, R., Benjamins, R. V., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, *25*(1-2), 161-197. doi: 10.1016/S0169-023X(97)00056-6.

Takasu, A. (2003). Bibliographic attribute extraction from erroneous references based on a statistical model. *International Conference on Digital Libraries*, 49.

The Apache Software Foundation. (2009). Apache Lucene. Retrieved from http://lucene.apache.org/.

The Apache Software Foundation. (2010). Apache Tika. Retrieved from http://tika.apache.org/.

The Open Group. (2009). *The Open Group Architecture Framework (TOGAF) 9*.

Thönssen, B., & Wolff, D. (2010). A broader view on Context Models to support Business Process Agility. In *Context* (p. 17).

TopQuadrant Inc. (2009). *Towards Executable Enterprise Models : Building Semantic Enterprise Architecture Solutions with TopBraid Suite ™*. *Representations*. TopQuadrant Inc.

Tsarkov, D., & Horrocks, I. (2006). *Automated Reasoning*. (U. Furbach & N. Shankar)*Automated Reasoning*, Lecture Notes in Computer Science (Vol. 4130, pp. 292-297). Springer Berlin Heidelberg. doi: 10.1007/11814771.

Uschold, M., King, M., Moralee, S., & Zorgios, Y. (1998). The Enterprise Ontology. *The Knowledge Engineering Review*, *13*(1), 31-89. doi: 10.1017/S0269888998001088.

Wache, H., Bergmayr, A., Feldkamp, D., Kondylakis, H., Nikles, S., Plexousakis, D., et al. (2010). plugIT. D3.2 Specification of Next Generation Modelling Framework. WP3 Specification of the Next Generation Modelling Framework.

Wache, H., Eichner, H., Koutsoukou, N., & et.al. (2009). *plugIT. State of the Art in Modelling. WP 2 Use Case and State of the Art Analysis*. *Analysis*.

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal*, *26*(3).

Zachman, J. A. (2003). The Zachman framework for enterprise architecture. *Strategy*. Retrieved from http://www.zifa.com/framework.pdf.

Zachman, J. A. (2008). John Zachman's Concise Definition of the The Zachman Framework™. Retrieved from http://www.zachmaninternational.com/concise definition.pdf.

van Heijst, G., Schreiber, A., & Wielinga, B. (1997). Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, (45), 183-292.

Authenticity Statement

I hereby declare that this thesis is my own work and confirm its authenticity where not indicated differently.

Date: July 31th, 2010

Signature Andreas Martin

Appendix A: Exemplary contract of the application scenario

CONTRACT: SOFTWARE LICENCE AGREEMENT

THIS AGREEMENT is made the 31 July 2010

PARTIES:

1. SwissSoft AG whose registered office is at Softstrasse 45, CH-4000 Basel, Switzerland ('SwissSoft')
2. SwissBikes AG whose registered office is at Bikesstrasse 31, CH-4142 Münchenstein, Switzerland ('SwissBikes')

RECITAL:

SWISSSOFT has agreed to deliver to the Customer computer software and to grant the Customer a non-exclusive licence to use such software and its associated documentation upon the terms and conditions hereinafter contained.

NOW IT IS HEREBY AGREED as follows:

1. Products and services to be provided

SWISSSOFT hereby agrees to:

(a) Grant to the Customer a non-exclusive licence of SwissCMS to use the Licensed Materials in accordance with the terms of this agreement;

(b) Deliver the Licensed Software, SwissCMS, to the Customer on the Installation Date;

(c) Provide the Program Documentation of SwissCMS;

2 Payment

2.1 The Licence Fee shall be paid within 30 days after the date of issuance of SWISSSOFT's invoice

3 Warranty

3.1 SWISSSOFT warrants that the Licensed Software will after configuration and installation provide the facilities and functions set.

4 Licence

4.1 SWISSSOFT hereby grants to the Customer a non-exclusive licence to use the Licensed Materials subject (SwissCMS) to the terms and conditions hereinafter contained

4.2 The Customer shall use the Licensed Materials for processing its own data for its own internal business purposes only.

5 Duration of Licence

The Licence shall commence on the Installation Date and shall continue until terminated in accordance with Clause 9 or as otherwise provided in this Agreement.

6 Proprietary rights

6.1 The Licensed Materials and the copyright and other intellectual property rights of whatever nature in the Licensed Materials are and shall remain the property of SWISSSOFT.

7 Copying

7.1 The Customer may make only so many copies of the Licensed Software as are reasonably necessary for operational security and use…

8 Training

8.1 SWISSSOFT undertakes to provide training in the use of the Licensed Software for the staff of the Customer.

9 Termination

9.1 The Customer may terminate the Licence at any time by giving at least 30 days' prior written notice to SWISSSOFT

10 Law

This Agreement shall be governed by and construed in accordance with the laws of Switzerland

SIGNED for and on behalf of

SWISSSOFT AG:

By:	Andreas Moser	Roland Minder
Signatures:		
Title:	CEO	Group Leader

SIGNED for and on behalf of

SwissBikes AG:

By:	Richard Stark	Peter Nelson
Signatures:		
Title:	CEO	Department Head

Appendix B: CD including code

The attached CD contains the following files:

- Folder *"/ApplicationScenario_Document/"* contains the exemplary contract of the application scenario *"SwissCMSDoc.pdf"*.
- Folder *"/ApplicationScenario_Rules/"* contains the application scenario rule file *"ApplicationScenario.rule"* which can be loaded into the LEMOWorkbench.
- Folder *"/EnterpriseModelOntology_OWL/"* contains the enterprise model ontology including application scenario data as OWL file *"EnterpriseModelOntology.owl"*.
- Folder *"/EnterpriseModelOntology_RDF/"* contains the enterprise model ontology excluding application scenario data as RDF file *"EnterpriseModelOntology.rdf"* and RDFS file *"EnterpriseModelOntology.rdfs"*.
- Folder *"/LEMOWorkbench_build/"* contains the LEMOWorkbench as executable JAR file – double-click on *"LEMOWorkbench-2.0.jar"*.
- Folder *"/LEMOWorkbench_sourcecode/"* contains the entire source code of the LEMOWorkbench including a Maven .pom file.
- File *"Martin - 2010 - LEMO creating Metadata.pdf"* thesis as PDF.
- File *"Martin - 2010 - LEMO creating Metadata.doc"* thesis as Word document.