

Connecting Enterprise Architecture and Information Objects Using an Enterprise Ontology

Knut Hinkelmann, Michaela Maise, Barbara Thönssen

FHNW University of Applied Sciences and Arts Northwestern Switzerland

FHNW Olten, Switzerland

knut.hinkelmann@fhnw.ch, michaela.maise@gmail.com, barbara.thoenssen@fhnw.ch

Abstract—In this paper we show how semantic metadata – derived from the enterprise architecture description - can improve both the exploitation of information in the operative business and the continuous alignment of information systems with the business. ArchiMEO, a formal representation of the ArchiMate framework, was applied to model selected parts of a large Swiss company, including metadata of concrete information objects stored in various applications. The evaluation of our approach clearly gave evidence that linking an enterprise ontology with operational databases is beneficiary: it provides an integrated view and management of enterprise entities spread over various data stores, represented in different ways and levels of granularity. Furthermore, because of its inference capabilities, when used as metadata encoding scheme an enterprise ontology can provide context-sensitive access to relevant information.

Keywords: *Enterprise Architecture; Enterprise Ontology; ArchiMate; ArchiMEO; Information Object; Information System; Metadata*

I. INTRODUCTION

In our research we tackle two closely related problems of information management: the continuous alignment of the information architecture to the business requirements and the effective access to relevant information which can be distributed over several information systems.

The enterprise information infrastructure comprises many different information systems to serve the various business needs. Due to on-going changes in business requirements the information infrastructure must be continuously adapted. On the other hand side, improvements and innovations in information systems can provide new opportunities in the way of doing business. However, since such adjustments are impeded by complex and nontransparent IT system landscapes it is difficult to foresee impacts of modifications.

Enterprise architecture descriptions aim at reducing complexity of this mutual alignment of business and IT by providing models representing business, application and infrastructure entities as well as relations among them. This transparency allows the enterprise architect to identify the potential impact of changes in the architecture. With typical enterprise architecture descriptions, however, it is not possible that a system actively checks the impacts of a proposed change.

To solve a business problem or to provide a service, information from different information systems have to be

combined. An explicit link between information objects and business processes, for example, would allow to access and integrate all data relevant for a specific activity or service. But although the information systems, their content as well as their relation to processes and services is described in an enterprise architecture, it cannot be used for this issue, because the enterprise architecture description is separated from the information systems themselves.

Our research investigates how enterprise architecture descriptions can be represented as an ontology which is machine understandable and cognitively adequate for humans. The enterprise ontology can be regarded as a metadata encoding scheme (cf. Dublin Core) describing the content and structure of enterprise objects stored in distributed information systems.

This enterprise ontology can support information management. Its inference capability allows to automatically identify which information systems or objects are affected by a change of the enterprise architecture. Furthermore, the enterprise ontology is used to represent the metadata of information objects which are stored in the information systems. These metadata describe the content and structure of information systems and can be regarded as a metadata encoding scheme

In a case study we customized the ArchiMEO meta model [1], a formal representation of the ArchiMate framework, to the requirements of a large Swiss company. Its concepts were linked to information objects stored in the company's various operational applications. This research focused on the conceptual integration of the ontology, information objects and their metadata whereas the technical implementation of a connection between ontology and information systems was out of scope.

The paper is organized as follows: In section II we provide related work to set the framework for our research. Sections III to VI introduce our approach in detail, starting with the requirement analysis based on interviews and a case study (section III), followed by a suggestion of the metadata concept and the content of the enterprise ontology (section IV). In section V we describe the implementation of our approach and in section VI the evaluation phase. The paper ends with a conclusion given in section VII.

II. THEORETICAL BACKGROUND

According to [2] the most dominant problem, stated in scientific as well as in popular science on enterprise management, is complexity and how it can be managed. Dietz claims that because of the complexity of enterprises a conceptual model is needed that "only shows the essence of the operation of an enterprise" and therefore "the model abstracts from all realization and implementation" [2, p. 8].

Enterprise Architectures can contribute to dealing with change and complexity [3]. There are a wide variety of Enterprise Architecture Frameworks [4], two of the most prominent are the Zachman Framework [3] and TOGAF [5].

Basically, there are two different notions of enterprise architecture. One perception is as a high level abstraction (of reality) with the purpose of reducing complexity and increasing stakeholder's understanding and communication (amongst others by [2], [3], [6]). The other, more recent notion of enterprise architecture focuses on integration for example [7], [8] and [9].

According to ISO/IEC/IEEE 42010 an "Architecture Description (AD) is an artifact that expresses an Architecture. Architects and other system stakeholders use Architecture Descriptions to understand, analyze and compare Architectures, and often as 'blueprints' for planning and construction" [10]. Though Chen et al. refer to work of the Open Group for practical principles [6], there is still no clear definition of what a good enterprise architecture description is. Johnson et al. explain this by showing the various purposes an enterprise model can be used for [11].

Kang et al. [12] criticize Enterprise Architecture (descriptions) for the lack of detailed models of the components, the vagueness of the modeled relationships between the components and the lack of a model for implementation. Also for [13] expressiveness and formality are missing from enterprise architecture models and, because of this there is only a limited degree of automation. Therefore the authors introduce "an approach of increasing the level of automation of BPM [Business Process Management] by representing the various spheres of an enterprise using ontology languages and Semantic Web Services frameworks" [14 p. 2].

As the investigated research showed, enterprise architecture description cannot be regarded without considering the 'language' it is expressed with. According to the ISO/IEC/IEEE 42010 standard an Architecture Description Language (ADL) is *any form of expression* for use in Architecture Descriptions [10]. More expressive ADL allow for a broader and utilization. In the specification of the ArchiMate specification we can read: "As information systems play a more active role in the management and operations of an enterprise [...] departing from their traditional role as simple repositories of data, information systems must now [...] not only answer queries with what is explicitly represented in their Enterprise Model, but must be able to answer queries with what is implied by the model." [19, p 123]. To live up to these expectations, enterprise ontologies seem to be an appropriate way to represent enterprise architecture knowledge.

Enterprise Architecture Modeling (and description) and enterprise ontology modeling have long been developing independently and only recently started to be merged [15].

Allemang et al. [16] built the Federal Enterprise Architecture Reference Model Ontology (FEA-RMO). Even though they provide a lot of insights about dealing with problems, modeling an EA as ontology it is limited with respect to general use and content. Kang et al. criticize that "Although FEA-Reference Model Ontology (FEA-RMO) [...] is proposed in order to share meanings of FEA reference models, it is nothing but the model which describes FEA reference models with Web Ontology Language (OWL). It is only for FEA reference models and is short of concrete method to share common meanings of Enterprise Architecture components" [13, p. 1457].

In order to address the problem of matching a citizen's needs with available public services, Goudos et al. [17] created an ontology based on the Governance Enterprise Architecture (GEA). The approach lacks some important points: the GEA is not implemented by a Public Administration and it does not consider various knowledge levels. In [8] we consider this the crucial point, as an EA is not necessarily used throughout a company as monolithic construct. That is especially true with respect to the increasing number of virtual enterprises.

Enterprise ontologies, described by RDF or OWL, can be queried with SPARQL, an RDF language developed by the World Wide Web Consortium (W3C) [18]. This query language was amongst others applied by [17] to make queries from the GEA ontology (OWL file as knowledge base) or by [19] to retrieve RDF data from the knowledge base. SPARQL provides several query forms, e.g. the SELECT clause which defines variables that appear in query results and the WHERE clause which contains triple patterns (similar like RDF triples with subject-predicate-object) that are matched against the data [20].

ArchiMate is a standard modeling language for describing enterprise architecture [21], [22]. The ArchiMate language aims to support enterprise architects in describing, analyzing and visualizing the relationships among business domains in an unambiguous way. ArchiMate provides a graphical representation of its language elements based on UML class diagram but customized and limited to a small set of modeling constructs in the interest of simplicity of learning and use. The possibility of extending and refining ArchiMate is particularly important because of the heterogeneity of models present in an enterprise, with varying degrees of maturity and the difficulty to determine their interrelations and the need to do so [23]. Hence, ArchiMate provides a sound basis for describing enterprise architecture but lacks of machine processability.

However, there is no common agreement on which architecture layers, which artifact types and which dependencies constitute an enterprise architecture [24]. Aier et al. [25] provide an overview on enterprise objects companies describe in their enterprise architecture. Besides business processes, descriptions of applications and information objects are considered predominantly important. The ArchiMate framework, for examples, defines information objects on the

business, application and technology layer as business objects, data objects and artifacts, respectively.

Although the Enterprise Architecture Description allows for representing information objects, companies do not use them for operational purpose such as combining information from different information systems to solve a business problem. The descriptions remain separate from the information objects they represent. Thus, although master data management (MDM) has percolated up as a corporate priority and an increasingly important trend [26] using Enterprise Architecture Description has not been taken into account.

For our research we applied the General Methodology of Design Science Research [27] which comprises five phases:

- the problem awareness
- suggestion
- development
- evaluation and
- conclusion.

Data collection for the problem awareness and the evaluation is based on interviews with practitioners and a case study approach.

The five phases are described in detail in the subsequent sections of this paper.

III. AWARENESS OF THE PROBLEM

The awareness phase consisted of two sub-phases. First, in order to identify and understand the problems in practice and requirements on how to address these problems, literature was reviewed and interviews were conducted with practitioners who work at a large Swiss company operating in several countries. They were carried out in a semi-structured manner to increase problem awareness and to find out how processes or the organization can be supported with an enterprise architecture description, represented in an enterprise ontology. Focus of the questions was on generic problems that could occur in various areas of information management. Results of the interviews were the identification of

- different categories of problems as well as
- several application scenarios that could serve as a basis for the case study to be considered in our approach.

Then, in order to prove our approach under real conditions a detailed intensive study of the requirements of an organizational unit was conducted that stresses factors contributing to its success or failure. Based on the case study domain-specific information objects and their relations were identified. For this purpose further interviews were conducted with employees of the respective organizational unit.

A. Interviews

1) Interview Participants:

Interviewees were selected from various departments of a Swiss company using purposive sampling [30], i.e. necessary qualities of interviewees were defined and then, based on this, interviewees were selected. The interviewees should be

concerned with implementation, management or maintenance of information systems from various areas. Required qualities of interviewees are, for example, knowledge of information systems that are in use in a specific area or awareness of both operative and strategic activities in a specific area.

To get a broad overview of problems in the enterprise, we chose interviewees who work in different areas (e.g. information systems management/administration, master data, finance, human resources, or sales) and with different types of information systems (e.g. Enterprise Resource Planning ERP, Customer Relationship Management CRM, or Document Management System DMS). The starting point for finding appropriate interviewees were documentations and information about applications and organization structure (e.g. organization charts for departments, units, divisions etc.) and a database of information systems (in the following referred to as IS database) that are used in the company.

2) Interview Questions:

The interview questions focused on generic problems that could occur in various areas – e.g. frequency of adaptations in the information systems landscape due to new requirements, identification of systems that are affected by changes of another system or redundancies or inconsistencies of data.

3) Interview Results:

Finding needed information is a problem that occurs in various areas such as sales or human resources (e.g. recruiting process). According to [31] in Switzerland almost 20% of working time is lost for searching for information. Statistics from the USA show even worse results [32].

Information is distributed in several systems and often no guidelines exist about what has to be stored where. Furthermore, descriptions of the content of these information systems are not standardized and knowledge of experts (e.g. project managers) is not externalized but locked in their heads.

It is also difficult to identify which systems are affected by changes (e.g. when a new customer was added in an ERP system it needs to be determined in which other systems information about customers is kept). In general, business processes are not harmonized with respect to the information systems that are used. One business process is related to the Enterprise Resource Planning (ERP) system to retrieve customer information; the other process is related to a Customer Relationship Management System (CRM) for the same purpose.

Within the company the sales department stores similar data objects in a CRM and an ERP system, which leads to double data collection and hence to decreasing data quality due to redundancies and maybe inconsistencies.

B. Case Study

1) Selecting the Case Study

The decision for a scenario for the case study was made according to various criteria. The scenario should include different types of data (structured data vs. unstructured data, transactional vs. aggregated data) and several types of information systems covering various use cases. Furthermore,

different types of users (end user, system administrator, IS project manager, etc.) should be involved.

For the case study we considered four scenarios in the HR department, accounting, manufacturing and sales, respectively. Finally, the decision was taken in favor of a sales scenario because it best meets the criteria. The sales scenario covers various different types of information systems (customer relationship management CRM, enterprise resource planning ERP, document management DMS, and management information systems MIS) holding heterogeneous types of data (structured data, unstructured data, transactional data, aggregated data). There are different types of users (system administrators, end users, "power user"). The number of applications is broader in the sales department than in the scenarios considered for HR, accounting or manufacturing. The processes in the accounting scenario were mainly supported by the ERP system and thus hardly met the criteria of covering different types of data and information systems. The HR scenario was not selected for this study because it was using only few and very specific applications.

Based on the results of the requirements analysis the sales scenario was selected as case study. The case study (containing several use cases) was set up during the suggestion phase by conducting some further interviews as described in this chapter.

2) Scope of the Case Study

The *case study* aimed at finding out tasks of the *people*, information systems used to accomplish the tasks, major information objects, attributes that are necessary to describe the information objects and typical queries for retrieving information objects. Most important was to find out the reasons for the problems in finding the needed information.

In the sales department, various information systems are used (CRM, ERP, DMS, MIS, and some company specific applications) which all contain relevant information objects (e.g. sales order, customer quotation, potential demand, customer invoice, product, deployed product, customer, customer visit).

In addition to the analysis of information systems and process documentation, interviewees from the sales department were selected with purposive sampling. They were concerned with information systems as users who need to retrieve data for daily business, as power users or system owners who are also in charge of supporting system development from the business side. These are the perspectives and roles covered by the interviewees:

- Sales engineers are responsible to sell services or products and to identify potential customer needs (e.g. maintenance service, modernization, or retrofits).
- Sales managers communicate between factories and sales offices in terms of strategic reconciliation of product prices.
- Market managers monitor the marketing channel management, identify channel partners and conclude contracts.

- The administration manager is in charge of order handling, controlling and reporting of financial figures to other areas such as accounting.
- The system administrator initializes IS implementations and system upgrades, launches new releases and communicates with external suppliers.

3) Results of the Case Study

The interviewees mentioned several problems, for example:

- retrieving documents that need to be attached to customer quotations is very time consuming since it is often difficult to find relevant customer documents in DMS;
- it is complicated to retrieve information about deployed products (stored in a dedicated system for deployed products) which often must be enhanced by additional information from other sources;
- due to the amount of information systems it is difficult for sales people to find their way around;
- information need to be collected from several applications which is time consuming.

C. Use Cases

The following use cases emerged from the interviews and were selected as case study components.

1. Sales engineers for services have to identify products deployed at customer's sites that may need maintenance services.
2. Market managers have to identify potential demands (stored in a dedicated application) that belong to his/her area of responsibility (i.e. specific product group, customers from specific country) which are due within a specified time frame and come from customers with good payment morale. Furthermore, in order to prepare customer quotations for identified potential demands, appropriate customer documents need to be retrieved.
3. In order to avoid double data creation, information systems managers have to find information objects that are created in more than one information system. In some cases, data are required in several information systems. Hence, an interface should be set up between two systems so that data needs to be created in only one system and be copied to another other system.
4. If an interface is set up, employees who are responsible for creating data must be identified and informed about the system change.
5. For sales people, in the CRM system links to reports in Business Warehouse should be set up. For that reason, relevant reports which are interesting for sales people need to be identified.

These use cases cover different perspectives (business, processes, applications), stakeholders (sales people, IS managers, process managers), information systems (e.g. ERP, DMS, CRM, MIS) and selected relevant information objects

(related to structured data such as product or potential demand and unstructured data, e.g. customer documents).

IV. SUGGESTION: CONCEPTS AND RELATIONS FOR THE ONTOLOGY

For the information objects determined in the case study a metadata concept was defined during the suggestion phase. The concept models properties that describe the information object in order to make it searchable. In this step, standards provided by the Dublin Core Metadata Initiative (DCMI) were applied.

Afterwards, concepts and relations that describe information objects derived from the case study in an ontology were identified by asking competency questions [28].

A. Metadata Concept

The metadata concept was modeled as UML (Unified Modeling Language) class diagram with classes and attributes representing information objects and their properties.

For setting up the metadata concept, standards provided by the Dublin Core Metadata Initiative (DCMI) [33] were applied - for example properties of the Dublin Core Metadata Element Set (DCMES), Syntax Encoding schemes (SES) to determine the structure of property values and Vocabulary Encoding Schemes (VES) to specify vocabularies that may be used as property values of resources, i.e. information objects [34].

In the following an example is provided derived from the use case about the identification of potential demands and the preparation of customer quotations. For this use case the following information objects are relevant.

Potential demand: possible need of a customer in the future which is recorded in a dedicated information system.

Product: a good that is offered by the company to satisfy the customer's needs which is recorded in product master data and several other systems.

Customer: party that buys products of the company. Customers are recorded in customer master data and the ERP and CRM systems.

Customer document: a document (e.g. PDF file) which contains information that is relevant for the customer and can be attached to customer quotations. Documents are stored in a Document Management System (DMS).

Customer quotation: an offer to a customer for a certain product at a certain price. It is captured in the ERP and CRM systems.

The information objects and some of their properties are represented in a UML class model (Unified Modeling Language by the Object Management Group) shown in Fig. 1. UML terms correspond to DCMI terms as follows: Classes can be found in both UML and DCMI, they correspond to information objects. Classes group UML objects or DCMI resources (instances of classes). UML attributes listed within a class are DCMI properties for which its values are described using a Syntax Encoding Scheme. Connections between classes represent DCMI properties for which its values are determined

by a Vocabulary Encoding Scheme; this means that resources of a specific class are allowed as property values to describe a resource.

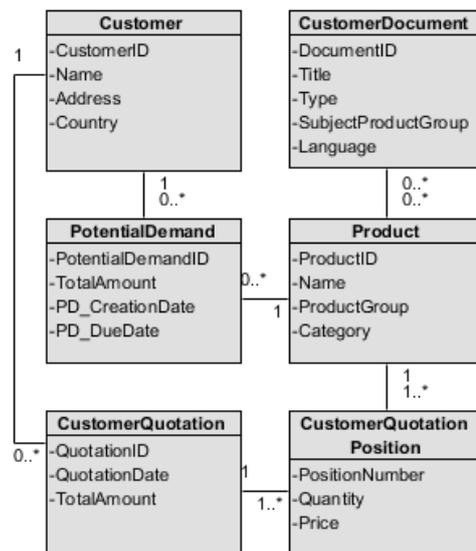


Fig. 1. Selected information objects and their properties

The properties illustrated in Fig. 1 can hold any values which fulfill the specific syntax constraints.

Connections between classes represent DCMI properties for which its values are determined by a Vocabulary Encoding Scheme; this means that resources of a specific class are allowed as property values to describe a resource. For example, the information object's properties CustomerID, QuotationID, ProductID and PotentialDemandID can be covered with the identifier element from DCMES, and PD_CreationDate and PD_DueDate with the date element.

The numbers written next to the end of the connections (by the class that contains resources to be used as property values) indicate how often a property could be used – e.g. a customer quotation contains one or many customer quotation positions and each position is related to exactly one product. The attributes described above are needed to find information objects that fulfill some criteria, e.g. potential demands for a certain product or product group which were recorded after a specific date and before a certain due date and which belong to customers from only one particular country. Ontology queries of chapter VI Evaluation contain some of the properties introduced in Fig. 1.

B. Identification of Concepts and Relations for the Ontology

For the development phase, the case study with its use cases built the basis for ontology modeling, respectively for the domain-specific ontology enhancement. In order to identify required concepts and relations, the method of [28] was applied: for each use case one or two competency questions (CQ) were formulated; the CQs were described according to a schema which contains the components input, assumptions, constraints and query; for each CQ it is explained briefly how the solution to the question is used (rationale) and what needs

to be known to answer the question (decomposition); finally, answers to the CQs indicate the needed concepts and relations.

In the following paragraphs we give examples of competency questions for two use cases: The second use case is about the identification of potential demands and the preparation of customer quotations. The fifth use case that deals with reports from a business intelligence system (Business Warehouse) to be provided to sales people (introduced above).

C. Use Case 2: potential demands and customer quotations

In the sales department, potential demands from customers in the future are captured in a dedicated system (Potential Demand and Quotations, PDQ). This is done, for example, when a sales manager during a customer visit learns about a planned project of the customer. Potential demand entries in PDQ comprise, among others, customer, product, and date when a certain demand will be due.

Based on potential demands recorded in the PDQ, the sales person has to identify potential demands for his/her area (certain product group, customer country) where the customer has good payment morale and should receive a quotation. Good payment morale means that the customer does not have too many overdue invoices. For this study, customer payment morale was assumed to be given, but it can be inferred based on information from the ERP system (e.g. days overdue of customer's open invoices). Identifying relevant potential demand is treated with CQ1

CQ1: Given potential demands and some constraints (product group, customer country, potential demand due date, potential demand creation date, and customer payment morale) for which potential demands a quotation should be prepared?

- *Rationale:* The solution to the question is used to identify potential demands from a specific responsibility area (product group, customer's country) that are due at a specific date and come from customers with a good payment morale (i.e. exclude customers with too high amount of overdue invoices) in order to prepare quotations. The solution of this CQ is used as input to the subsequent CQ.
- *Input:* potential demands
- *Assumptions:* Customer's payment morale can be good or bad
- *Constraints:*
 - Potential demand for products of a certain product group
 - Customer country is Switzerland
 - Potential demand due date is between 01.01.2013 and 31.03.2013
 - Potential demand creation date is after 01.01.2011
 - Customer's payment morale is good

- *Query:* What are the potential demands (e.g. described by product, customer, customer country, potential demand value, potential demand due date and creation date) that should be followed up with a quotation supposing the assumptions and the constraints?
- *Decomposition:* In order to answer the question, the following things need to be known – potential demand has due date and creation date, potential demand is related to product, potential demand has value, product is part of product group, potential demand is from customer, customer is located at country, and customer has payment morale.

After the potential demands were identified in CQ1, relevant documents that are available to be attached to the customer quotation (response to a potential demand) must be found in the document management system (DMS) (e.g. technical descriptions, installation guides, or catalogues). This question is covered by CQ2.

CQ2: Given a potential demand and some constraints (document language, document category, related product, related product group, and document group) which documents are available to be attached to the quotation?

- *Rationale:* The solution to the question is used to identify documents that might be relevant for the quotation.
- *Input:* potential demand
- *Assumptions:* Document categories are installation guide, technical description, test report, catalogue, technical drawing, etc.
- *Constraints:*
 - Document language is German
 - Document category is installation guide, technical description, test report, catalogue or technical drawing
 - Document is related to product of potential demand
 - Document is related to product group of product of potential demand
 - Document belongs to document group customer documents
- *Query:* Which documents (e.g. described by title, related product, related product group) are available supposing the assumptions and the constraints?
- *Decomposition:* In order to answer the question, the following things need to be known – document belongs to category, document has language, document is related to product or product group, document has title, document belongs to document group, and results of CQ1.

D. Use Case 5: Business Warehouse reports

The other use case to be described here in detail deals with sales managers who mainly use the CRM system for retrieving information required for their work. However, there is additional interesting information available in the Business Warehouse (BW) reports. These reports should be made easily accessible for sales persons by providing a link to the BW reports in the CRM. For this purpose, the IS manager has to find out which data objects are contained in the BW reports and which of those could be interesting for the sales team. Based on the results of this search, links to identified BW reports would be implemented in the CRM system.

In CQ3, data objects relevant for sales experts from a sales department are determined. A data object is assumed to be relevant if it realizes a business object that is accessed by a business function of the sales department. One constraint is set in order to limit data objects only to specific related business functions (e.g. only sales controlling, key account management, market management, but no functions that are in charge of clerical tasks). Another constraint restricts data objects to those with granularity “aggregated data”, as business warehouse reports do not contain transactional data.

CQ3: Given an organizational unit and some constraints (business function, data object granularity) which data objects are relevant?

- *Rationale:* The solution to the question is used to identify business objects in the area of organizational unit sales department. The solution of CQ3 is used as input to CQ4.
- *Input:* organizational unit
- *Assumptions:* Data objects are relevant if they realize business objects accessed by business functions which belong to the organizational unit
- *Constraints:*
 - Business objects are accessed by business functions key account management, strategic account management, or sales controlling
 - Data objects have granularity “aggregated data”
- *Query:* Which data objects (e.g. described by label, accessing business function) are interesting supposing the assumptions and the constraints?
- *Decomposition:* In order to answer the question, the following things need to be known – business function is assigned to organizational unit, business function accesses business object, business object is realized by data object, data object has granularity.

Afterwards, based on data objects retrieved from CQ3, the next competency question CQ4 determines which application functions (i.e. business warehouse reports) have access to a data object. Since for this purpose only the BW reports are of interest (but not application functions of other application components), the application component BW is set as a constraint.

CQ4: Given a data object and some constraints (application component) which application functions access the data object?

- *Rationale:* The solution to the question is used to identify application functions (i.e. BW reports) that cover relevant data objects identified in CQ3.
- *Input:* Data object
- *Assumptions:* BW report is an application function
- *Constraints:* Application functions are assigned to application component Business Warehouse
- *Query:* Which application functions (e.g. described by label, accessed data object) are relevant supposing the assumptions and the constraints?
- *Decomposition:* In order to answer the question, the following things need to be known – results of CQ3, data object is accessed by application function, and application function is assigned to application component.

V. DEVELOPMENT: ONTOLOGY IMPLEMENTATION

For implementation in the development phase we re-used the enterprise ontology ArchiMEO [29]. The concepts, relations and instances identified with competency questions were integrated into ArchiMEO to build a domain-specific version called ArchiMEO SALE. Some concepts and relations already existed and were reused; missing ones were added to ArchiMEO. ArchiMEO is publicly available on <https://github.com/ec-ikm/ArchiMEO> under a Creative Commons Attribution-ShareAlike 3.0 Unported License. Concepts, relations and concrete instances were created in the ontology software TopBraid [35] and formally modeled with RDFS 3.0 (Resource Description Framework Schema).

RDF statements have the form of triples: a triple consists of a subject, a predicate and an object; subject and object are classes (or concepts) and predicate refers to the property (or relation) between subject and object [36]. Properties can be distinguished into data type properties used to connect instances with data values and object properties to connect instances with instances [37]. Data type properties correspond to Syntax Encoding Schemes and object properties to Vocabulary Encoding Schemes as described in chapter IV.

Names of subjects, predicates and objects start with prefixes. In ArchiMEO they differ depending on the development phase where classes or properties originate from. For example the prefix “top:” is used for generic classes and properties of the top level ontology (e.g. location, time, or event), “archi:” namespace is used to mark classes and properties that come from ArchiMate (e.g. business object, application component, or business actor), “eo:” refers to enterprise ontology classes and properties, “elements:” is applied for properties according to DCMES. Classes and properties that are needed for the domain-specific sales ontology, but are not yet available within ArchiMEO are added with prefix “sale:”.

Names of classes (subjects or objects) can be distinguished from names of properties based on the part of speech which is used and on the first letter after the prefix. Class names are nouns and start with a capital letter. Property names are small sentences containing a noun, a verb and another noun. They can be compared to predicates of a triple indicating how a specific subject and a particular object are connected to each other.

During the ontology implementation, after instances were created for the concepts, also results expected from ontology evaluation queries (conducted in the evaluation phase according to competency questions) were defined.

The example introduced above is continued in the following paragraphs. Both competency questions for the second and fifth use case are depicted as UML class diagram showing required classes and properties, as well as ArchiMEO parent classes. These diagrams include several elements – blue and grey boxes represent classes, data type properties are listed as attributes within classes, association relationships denote object properties, specialization relationships indicate subclasses and parent classes. Blue classes are applied in the use cases and contain instances. Grey classes depict superior classes in the hierarchy of ArchiMEO. Attributes of classes representing data type properties show prefixes in the first attribute. Subsequent attributes have the same prefix as the first one.

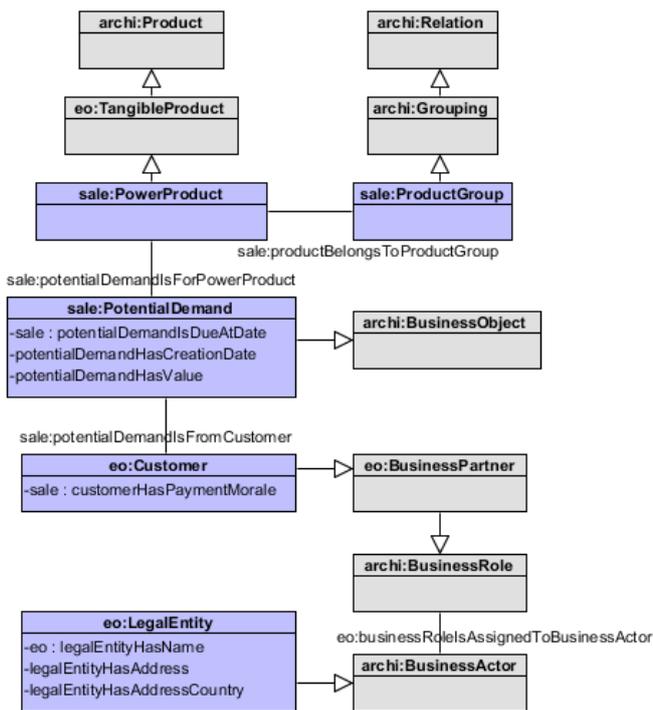


Fig. 2. Competency question 1, illustrated as UML class diagram

A. Use Case 2: potential demands and customer quotations

To find the solution to CQ1, the classes and properties illustrated in Fig. 2 were required: existing classes eo:Customer and eo:LegalEntity were reused and additional classes

sale:PowerProduct, sale:PotentialDemand and sale:ProductGroup were added.

The second part of this use case, respectively CQ2 is required to find documents that are related to products of identified potential demands or corresponding product group and fulfill other conditions such as document language, document category, or document group.

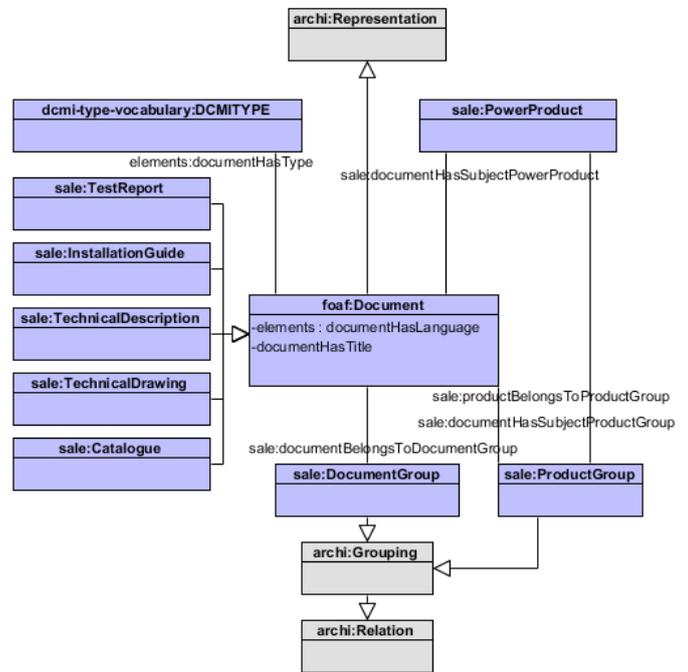


Fig. 3. Competency question 2, illustrated as UML class diagram

As depicted in Fig. 3, ArchiMEO provides the class foaf:Document under which for ArchiMEO SALE several sub-classes (that represent document category) were added – sale:TestReport, sale:InstallationGuide, sale:TechnicalDescription, sale:TechnicalDrawing, and sale:Catalogue. For CQ2 the class sale:DocumentGroup was added as a further sub-class of archi:Grouping to model specific groups of documents. In this use case, only the document group “customer document” was required. However, in other cases further document groups may be necessary, for example, internal sales documents to mark those documents that are confidential and must not be published (e.g. price lists), or documents from human resources department.

B. Use Case 5: Business Warehouse reports

In order to solve CQ3 the classes eo:OrganisationalUnit, archi:BusinessFunction, archi:BusinessObject (respectively its sub-classes, e.g. eo:Sale), and archi:DataObject (resp. sub-class eo:DataSet) are required. A business function “is defined as a unit of internal behavior that groups behavior according to, for example, required skills, knowledge, resources, etc. and is performed by a single role within the organization”[21].

These classes are all provided by ArchiMEO, so no additional classes were needed, see Fig. 4. This use case applies class archi:BusinessFunction in order to set a constraint

on certain business functions (e.g. sales controlling and key account management) that are assigned to the organizational unit (sales department).

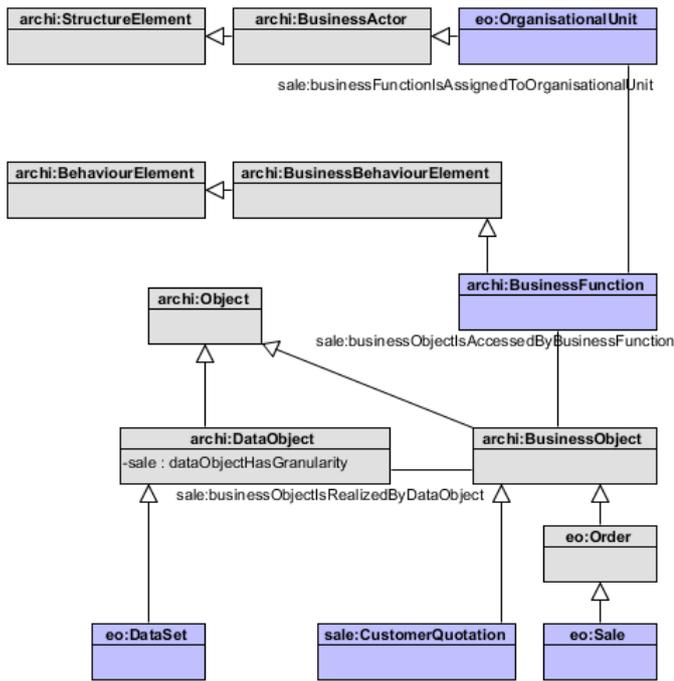


Fig. 4. Competency question 3, illustrated as UML class diagram

In order to connect the classes `eo:OrganisationalUnit`, `archi:BusinessFunction` and `archi:BusinessObject`, the properties `sale:businessFunctionsAssignedToOrganisationalUnit` and `sale:businessObjectsAccessedByBusinessFunction` were created.

The second part of this use case is depicted in Fig. 5. It covers CQ4 and requires the classes `archi:DataObject`, `archi:ApplicationFunction` and `archi:ApplicationComponent`.

Business Warehouse reports are considered to be application functions which, according to [21], access and operate on data objects. Application functions are performed by an application component (in this case Business Warehouse) that is defined as “a modular, deployable, and replaceable part of a system that encapsulates its contents and exposes its functionality through a set of interfaces” [21]. Also, for this use case no new classes were created, but only the object property `sale:applicationFunctionsAssignedToApplicationComponent` was added as a sub-property of the generic `archi:isAssignedTo` property.

VI. EVALUATION

The evaluation of our approach was based on the competency questions determined within the case study. In order to proof the concept, for each use case evaluation queries were performed on ArchiMEO SALE implemented in TopBraid using the RDF query language SPARQL. Afterwards the query results were compared to expected results defined during the ontology implementation. All actual query results have met the expected results.

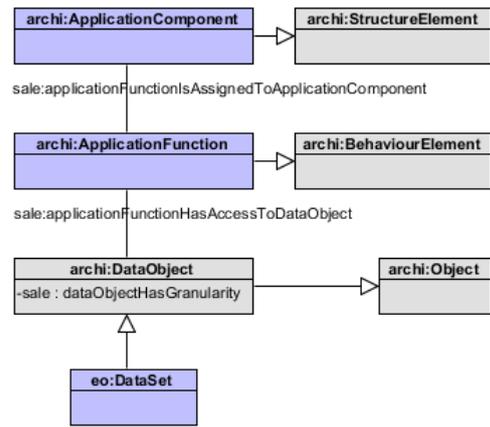


Fig. 5. Competency question 4, illustrated as UML class diagram

Finally, in order to evaluate the artifact w.r.t. its usefulness, the results were presented to practitioners previously interviewed for the case study. They confirmed that an enterprise architecture describing the various information systems and its data, represented in an ontology would be beneficiary. Instead of distributing queries over several different information systems one complex query can be executed. Hence, time for searching the needed information can be reduced significantly.

To continue the example use cases about the identification of potential demands in order to prepare customer quotations and about the identification of BW reports relevant for sales people, the SPARQL queries are explained in the following paragraphs.

A. Use Case 2: potential demands and customer quotations

Competency questions CQ1 and CQ2 were combined into one single query for this use case. Instances of class `sale:PotentialDemand` were given as input. The query result delivered instances of class `sale:PotentialDemand` and certain sub-classes of `foaf:Document` (`sale:TestReport`, `sale:InstallationGuide`, `sale:TechnicalDescription`, `sale:TechnicalDrawing`, `sale:Catalogue`).

The SPARQL query (illustrated in Fig. 6) contains an inner query covering CQ1 and an outer query for CQ2. The results, potential demand instances fulfilling the constraints of the inner query, are given as input to the outer query. Constraints of the inner query are defined by FILTER keyword for several variables and by giving a concrete value to property `sale:productBelongsToProductGroup`. Constraints of the outer query also use the FILTER keyword for some variables and assign a value to property `sale:documentBelongsToDocumentGroup`. Furthermore, there is a constraint connecting two property value assignment with the UNION keyword. The latter is necessary to find documents that belong either to a product of product group “Circuit Breakers” or directly to product group “Circuit Breakers”.

```

SELECT DISTINCT ?potentialDemand ?demandValue ?powerProd ?document ?subclass
?docTitle ?docLanguage ?docType
WHERE {
  ?document rdf:type ?subclass .
  ?document elements:documentHasType ?docType .
  ?document elements:documentHasTitle ?docTitle .
  ?document sale:documentBelongsToDocumentGroup sale:CustomerDocuments .
  ?document elements:documentHasLanguage ?docLanguage .
  ((?document sale:documentHasSubjectPowerProduct ?powerProd .
  ?powerProd sale:productBelongsToProductGroup sale:CircuitBreakers)
  UNION
  {?document sale:documentHasSubjectProductGroup sale:CircuitBreakers})
  FILTER ( ?subclass = sale:TechnicalDescription || ?subclass = sale:TechnicalDrawing ||
  ?subclass = sale:Catalogue || ?subclass = sale:InstallationGuide ||
  ?subclass = sale:TestReport)
  FILTER (?docLanguage = "EN" || ?docLanguage = "DE")
}
}
(SELECT *
WHERE {
  ?potentialDemand sale:potentialDemandIsDueAtDate ?dueDate .
  ?potentialDemand sale:businessObjectHasCreationDate ?creationDate .
  ?potentialDemand sale:potentialDemandHasValue ?demandValue .
  ?potentialDemand sale:potentialDemandsForPowerProduct ?powerProd .
  ?powerProd sale:productBelongsToProductGroup ?productGroup .
  ?powerProd sale:productBelongsToProductGroup sale:CircuitBreakers .
  ?potentialDemand sale:potentialDemandsFromCustomer ?customer .
  ?customer eo:businessRolesAssignedToBusinessActor ?legalEntity .
  ?legalEntity eo:legalEntityHasAddressCountry ?country .
  ?customer sale:customerHasPaymentMorale ?paymentMorale .
  FILTER ((?dueDate < "2013-03-31"^^xsd:date) && (?dueDate > "2013-01-01"^^xsd:date)) .
  FILTER (?creationDate > "2011-01-01"^^xsd:date) .
  FILTER (xsd:string (?country) = "Switzerland") .
  FILTER (xsd:string (?paymentMorale) = "good" .)
})

```

Fig. 6. SPARQL query for competency questions 1 and 2

B. Use Case 5: Business Warehouse reports

For this use case two different queries were done. The first one selected all `archi:BusinessFunction` instances that are assigned to `eo:OrganisationalUnit` “SalesLVProducts” (input) – see Fig. 7. Since not all business functions need BW reports, only selected instances are used as input to the next query, for example key account management or sales controlling.

```

SELECT *
WHERE {
  ?businessFunction sale:businessFunctionIsAssignedToOrganisationalUnit ?orgUnit .
  FILTER (?orgUnit = sale:SalesLVProducts)
}

```

Fig. 7. Preparatory SPARQL query for the fifth use case

Fig. 8 illustrates the main SPARQL query for the fifth use case. It is also composed of an inner query and an outer query. Input is defined with a `FILTER` (in the inner query) containing selected `archi:BusinessFunction`. Since the inner query result does not need to deliver data objects that are related to all of the input instances, they are connected with logical OR operator “||”. The result of the inner query (instances of `archi:DataObject`) is given to the outer query as input.

In the outer query, the second constraint is set by selecting only instances that are related to instance “BusinessWarehouse” of class `archi:ApplicationComponent`.

Finally, in order to evaluate the artifact w.r.t. its usefulness, the results were presented to practitioners previously interviewed for the case study. They confirmed that an enterprise architecture describing the various information systems and its data, represented in an ontology, would be beneficiary. Instead of distributing queries over several different information systems one complex query can be

executed. Hence, time for searching the needed information can be reduced significantly.

```

SELECT DISTINCT ?applicationFunction ?dataObject
WHERE {
  ?applicationFunction sale:applicationFunctionIsAssignedToApplicationComponent ?applicationComponent .
  ?applicationFunction sale:applicationFunctionHasAccessToDataObject ?dataObject .
  FILTER (?applicationComponent = sale:BusinessWarehouse) .
}
(SELECT DISTINCT *
WHERE {
  ?businessObject sale:businessObjectIsAccessedByBusinessFunction ?businessFunction .
  ?businessObject sale:businessObjectIsRealizedByDataObject ?dataObject .
  ?dataObject sale:dataObjectHasGranularity ?granularity .
  FILTER (?businessFunction = sale:KeyAccountManagementLV ||
  ?businessFunction = sale:StrategicAccountManagementLV ||
  ?businessFunction = sale:SalesControllingLV)
  FILTER (?granularity = "aggregated data")
})

```

Fig. 8. SPARQL query for competency questions 3 and 4

VII. CONCLUSION

Generally, in order to close the gap between information objects and enterprise ontologies, it must be determined where to store both the metadata concepts and the metadata themselves and how to connect them with the enterprise ontology. This is, for example, storing the metadata and the metadata concept in a separated repository vs. integrating the concept into the ontology. There are several approaches to connect ontologies with information objects (e.g. by translating SPARQL queries to other query languages that can access corresponding information systems [38] or by mapping RDB schemas to ontologies [39]). However, this research focused on the conceptual integration whereas the technical implementation of a connection between ontology and information systems was out of scope.

Our approach shows that it is possible and beneficiary to characterize information systems and its data in an enterprise architecture description represented in an ontology. This enterprise ontology can be queried in order to access information stored in the distributed information systems.

Moreover, this paper validates that ArchiMEO provides a sound model of the enterprise architecture as it is based on the ArchiMate standard. The fact that information objects can be connected with an Enterprise Architecture does not depend on the underlying domain of the company. ArchiMEO can be regarded as an enterprise upper ontology. It is suitable as a basis and extendable by domain-specific concepts and relations for a concrete scenario (including elements like processes, applications, business actors, or business objects). So every enterprise could apply ArchiMEO to describe their enterprise architecture and metadata of information objects in order to retrieve them from the respective information systems. Therefore, the approach applied in this research study is considered as generalizable and could be followed by other enterprises to develop a domain-specific ontology containing a metadata concept for selected information objects.

This paper illustrates how the enterprise ontology can be modeled, how metadata of information objects can be described and how the metadata schema can be integrated into ArchiMEO.

REFERENCES

- [1] K. Hinkelmann, A. Martin, S. Nikles, and B. Thönssen, "ArchiMEO: Representing and Enhancing ArchiMate in RDFS 3.0." 2013.
- [2] J. L. G. Dietz, *Enterprise Ontology. Theory and Methodology*. Berlin Heidelberg: Springer-Verlag, 2006.
- [3] J. A. Zachman, *The Zachman Framework For Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing*. 2003.
- [4] D. Matthes, *Enterprise Architecture Frameworks Kompendium*. Heidelberg: Springer, 2011.
- [5] The Open Group, *TOGAF Version 9.1*. Van Haren Publishing, 2011.
- [6] D. Chen, G. Doumeingts, and F. Vernadat, "Architectures for enterprise integration and interoperability: Past, present and future," *Computers in Industry*, vol. 59, pp. 647–659, 2008.
- [7] R. Woitsch, D. Karagiannis, D. Plexousakis, and K. Hinkelmann, "Business and IT alignment: the IT-Socket," *Elektrotechnik und Informationstechnik*, vol. 126, no. 7–8, pp. 308–321, Jul. 2009.
- [8] K. Hinkelmann, E. Merelli, and B. Thönssen, "The Role of Content and Context in Enterprise Repositories," in *2nd International Workshop on Advanced Enterprise Architecture and Repositories - AER 2010*, 2010, no. Cmd.
- [9] K. Valtonen, S. Mäntynen, M. Leppänen, and M. Pulkkinen, "Enterprise Architecture Descriptions for Enhancing Local Government Transformation and Coherency Management," *2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops*, pp. 360–369, Aug. 2011.
- [10] DSCI, "ISO/IEC/IEEE 42010 Website," 2012. .
- [11] P. Johnson, R. Lagerström, P. Närman, and M. Simonsson, "Enterprise architecture analysis with extended influence diagrams," *Information Systems Frontiers*, vol. 9, no. 2–3, pp. 163–180, May 2007.
- [12] D. Kang, J. Lee, S. Choi, and K. Kim, "An ontology-based Enterprise Architecture," *Expert Systems With Applications*, vol. 37, no. 2, pp. 1456–1464, 2010.
- [13] M. Hepp and D. Roman, "An Ontology Framework for Semantic Business Process Management," in *Proceedings of Wirtschaftsinformatik 2007*, 2007, pp. 1–18.
- [14] M. S. Fox, M. Barbuceanu, and M. Grüninger, "An organisation ontology for enterprise modeling: Preliminary concepts for linking structure and behaviour," *Computers in Industry*, vol. 29, no. 1–2, pp. 123–134, Jul. 1996.
- [15] M. de Vries, "A Process Reuse Identification Framework Using an Alignment Model," University of Pretoria, 2012.
- [16] D. Allemang, R. Hodgson, and I. Polikoff, "FEA Reference Model Ontologies (FEA RMO)," *Development*. pp. 1–43, 2005.
- [17] S. K. Goudos, V. Peristeras, and K. Tarabanis, "Mapping Citizen Profiles to Public Administration Using Ontology Implementations of the Governance Enterprise Architecture (GEA) models," in *Proceedings of the Workshop on Semantic Web for eGovernment 2006*, 2006, no. June.
- [18] J. Euzenat, A. Polleres, and F. Scharffe, "Processing ontology alignments with SPARQL," in *IEEE international workshop on Ontology alignment and visualization (OAaV)*, 2008, pp. 913–917.
- [19] T. Kilpeläinen and M. Nurminen, "Applying Genre-Based Ontologies to Enterprise Architecture," in *Proceedings of the 18th Australasian Conference on Information Systems*, 2007, p. pp. 10.
- [20] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," 2008. [Online]. Available: <http://www.w3.org/TR/rdf-sparql-query/>. [Accessed: 11-Jan-2013].
- [21] The Open Group, "ArchiMate 2.0 Specification," 2012. [Online]. Available: <http://pubs.opengroup.org/architecture/archimate2-doc/>. [Accessed: 29-May-2013].
- [22] The Open Group, "ArchiMate," 2012. [Online]. Available: <http://www.opengroup.org/subjectareas/enterprise/archimate>. [Accessed: 21-Jun-2012].
- [23] M. Lankhorst, "ArchiMate Language Primer," 2004. [Online]. Available: https://doc.telin.nl/dsweb/Get/Document-43839/ArchiMate_Language_Primer.pdf. [Accessed: 01-Apr-2012].
- [24] R. Winter and R. Fischer, "Essential Layers, Artifacts, and Dependencies of Enterprise Architecture," *Journal of Enterprise Architecture*, vol. 3.2, no. May, pp. 7–18, 2007.
- [25] S. Aier, C. Riege, and U. R. Winter, "Unternehmensarchitektur – Literaturüberblick und Stand der Praxis," *Wirtschaftsinformatik*, vol. 4, pp. 292–304, 2008.
- [26] L. Agosta, "Mastering Master Data Management," *Information Management*, 2006. .
- [27] V. Vaishnavi and B. Kuechler, "Design Science Research in Information Systems," *January 20, 2004*, 2004. [Online]. Available: <http://desrist.org/design-research-in-information-systems/>. [Accessed: 25-Jun-2012].
- [28] M. Grüninger and M. S. Fox, "Methodology for the Design and Evaluation of Ontologies," *Industrial Engineering (1995)*, vol. 95, pp. 1–10, 1995.
- [29] B. Thönssen, "Automatic, Format-independent Generation of Metadata for Documents Based on Semantically Enriched Context Information," University of Camerino, 2013.
- [30] D. C. Tongco, "Purposive Sampling as a Tool for Informant Selection," *Ethnobotany Research & Applications*, vol. 5, pp. 147–158, 2007.
- [31] P. Sieber, "Enterprise-Search Katalysator für den internen Informations- und Wissensfluss," Bern, 2009.
- [32] J. B. Spira, "Information Overload: Now \$900 Billion – What is Your Organization's Exposure?," 2008. .
- [33] DCMI, "Dublin Core Metadata Initiative," 2010. .
- [34] S. Rühle, T. Baker, and P. Johnston, "User Guide. DCMI," 2011. [Online]. Available: http://wiki.dublincore.org/index.php/User_Guide. [Accessed: 15-Jul-2012].
- [35] TopQuadrant, "Towards Executable Enterprise Models: Building Semantic Enterprise Architecture Solutions with TopBraid Suite™," *Representations*. pp. 1–13.
- [36] W3C, "RDF Vocabulary Description Language 1.0: RDF Schema," *W3C Recommendation 10 February 2004*, 2004. .
- [37] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein, "OWL Web Ontology Language Reference," 2004. [Online]. Available: <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>. [Accessed: 05-Jan-2013].
- [38] V. Kashyap and M. Flanagan, "From Web 1.0 --> 3.0: Is RDF access to RDB enough?," in *W3C Workshop on RDF Access to Relational Databases*, 2007.
- [39] W. Hu and Y. Qu, "Discovering Simple Mappings Between Relational Database Schemas and Ontologies," in *Proceedings of ISWC/ASWC 2007, Busan, South Korea*, 2007, no. DI, pp. 225–238.