The Role of Content and Context in Enterprise Repositories

Knut Hinkelmann[1], Emanuela Merelli[2], Barbara Thönssen[1,2]

[1] University of Applied Sciences Northwestern Switzerland FHNW
School of Business
knut.hinkelmann@fhnw.ch; barbara.thoenssen@fhnw.ch

[2] University of Camerino
Dipartmento di Matematica e Informatica
emanuela.merelli@unicam.it; barbara.thoenssen@unicam.it

Abstract.

In this paper we show how to utilize enterprise architectures to model relations between information sources of an enterprise. With our approach the enterprise architecture models are not passive entities anymore but can be operationalized to identify dependencies between the various models and to integrate information sources. It thus is the basis for agile enterprises. We show that the elements of an enterprise architecture and the referenced data can take the role of both context and content depending on the objectives of their use. We present two independent approaches for modeling an enterprise ontology.

Introduction

There are many information resources in an enterprise that serve several purposes, usually residing in different information systems.

- operational data like financial data, bill of material, human resource data are stored in enterprise resource planning (ERP) systems
- customer data can be found in systems for customer relationship management (CRM)
- business process models usually are stored in files of business process management suites
- documents are stored in a document management system (DMS)
- configuration data in configuration management databases (CMD)
- websites for the internet can be found in content management systems (CMS)

These information resources, however, are not independent but depend on each other. An enterprise architecture (EA) - originally developed to support information systems engineering, can also be used to make these interrelations explicit.

An enterprise architecture (EA) consists of a number of models representing information likewise for companies and for public administrations, e.g. Zachmann's[1], FEA[2], TOGAF[3], IAF[4] and many others. It is not the purpose of this paper to go into detail of the different EA frameworks. The most recent overview on enterprise architectures is given by (Aier et al., 2008). In the following we use the Zachman framework (Zachman, 1987) as an example, however the arguments hold for any enterprise architecture.

The structure of enterprise architecture frameworks can be regarded as a scaffolding to organize the models by assigning them to business or IT perspective and classifying them w.r.t. to several aspects. However, hidden in the overall structure of the enterprise architecture frameworks there are additional links between the model contents. Just to mention a few:

- The *information objects* used or created in a business process are described in the *inventory* aspect (called "data" in former versions of the Zachman framework), whereas the *process flow* is decribed in the process aspect.
- By assigning participants of activities via roles a link is made between business *process* and *organization* aspect.
- The motivation and purpose of a business *process* is given by the product or service it produces which itself is determined by the business *motivation*.
- Document types and data models of the *inventory* aspect are related to information infrastructure of the *network* aspects specifying the source where the information is stored.

Although the Enterprise Architecture framework helps to make those relations transparent (Sowa and Zachman, 1992) it falls short when it comes to the concrete models. On operational level, the models of the enterprise architecture are implemented as applications and data:

- For the employees mentioned in the organization structure there exist records stored in the HR module of the ERP system.
- Data about the IT infrastructure can be stored in a Configuration Management Database (Hanschke, 2010).
- Data of Workflow instances is stored in the Workflow Management System.
- Production data is managed with a Production Planning System and so on.

Thus the relations and links defined in the EA models are limited on the concrete model level to APIs and completely lost on the implemented data level.

But, on those two levels changes can take place: a business rule can change due to a revision of a law – in this case, what business processes are affected and what impact does that have on information objects? If the categorization of clients changes due to a new marketing strategy - what business rules have to be adjusted, what information objects adapted? Changing a process model - what does that mean for IT-operation?

[1] John Zachman. URL: http://www.zachmaninternational.com/index.php/the-zachman-framework (18.3.10)

[2] Federal Enterprise Architecture. URL: http://www.whitehouse.gov/omb/e-gov/fea/ (18.3.10)

[3] The Open Group Architecture Framework. URL: http://www.opengroup.org/togaf/ (18.3.10)

[4] Capgemini Enterprise Architecture Framework. URL: http://www.capgemini.com/services-and-solutions/technology/soa/soa-solutions/ent_architecture/iaf/ (18.3.10)

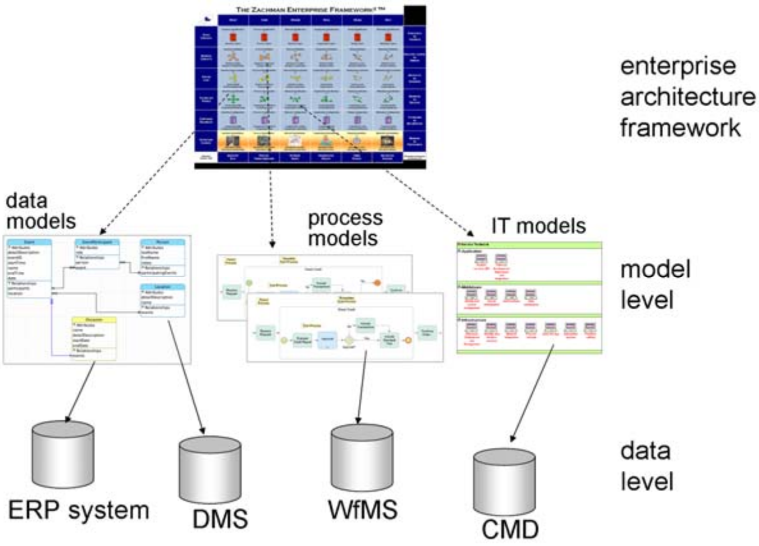

Figure 1: Relation between EA, models and data

Figure 1 depicts the relation between an EA, application models and their data. The dotted lines between the EA and the concrete models indicate the weakly expressed relationship amongst them.

Consider for example a manufacturer of espresso machines: It highly relies on the suppliers for the various parts needed to produce the machines. Thus, supply-chain-management is vital for the company's survival. Information about the supplier are stored in a ERP system whereas the reports are stored in a Document Management System (DMS). Although in the EA relations between suppliers and production are expressed, they are not represented on the operational level. Even if the company has a separate knowledge process for monitoring, identification and validation of changes in place it is separated from daily business work. Changes like increasing late delivery or delivery of damaged parts of that very supplier, would be stored in the respective applications (the ERP system) and identified in the Production Planning System (PPS) but not be linked to the externally gathered information. Having these relations modelled semantically would improve the awareness of changes, therefore is better protected against risks (or earlier in detecting opportunities) and therefore supporting enterprises agility and risk protection - not only with respect to supply chain management.

Reacting accordingly to changes is considered important but difficult, complex and risky according to unintended side effect. Every change has an impact on other parts of the enterprise, which leads to the choice, whether to make a change or abandon the competitive benefits of innovation because of the risk (Mitra and Gupta, 2006). Therefore representing dependencies between enterprise resources in a machine understanding and executable way is necessary if an organization wants to stay competitive.

We propose an ontological representation of the perspectives and aspects of an enterprise architecture using the enterprise architecture as an integration scheme.

How to use Enterprise Architecture

Whereas it is consensus that using semantic technologies to describe an enterprise is an appropriate approach (amongst others (Abecker et al. 1998), (Dietz, 2006), (Fox and Gruninger, 1998), (Leppänen, 2007), (Uschold et al., 1998)) it is at the risk that the created ontologies do not reflect the whole picture and suffer from incompleteness. Therefore (Kang et al., 2010) propose to relate enterprise ontology to an Enterprise Architecture Model. Even though there are many Enterprise Architectures likewise for companies and the public sector available they are designed for humans, to understand the various areas of an enterprise and the relations amongst them but are not 'understandable' for machines. Using ontologies to describe an enterprise helps to

- reach a shared understanding (amongst different stakeholders)
- solving ambiguity
- becoming machine understandable and,
- getting active (using reasoning for access and mining).

Relating the enterprise ontology to an enterprise architecture allows for validating the ontology on the basis of a 'technical neutral' and 'business approved' model to ensure completeness and appropriateness.

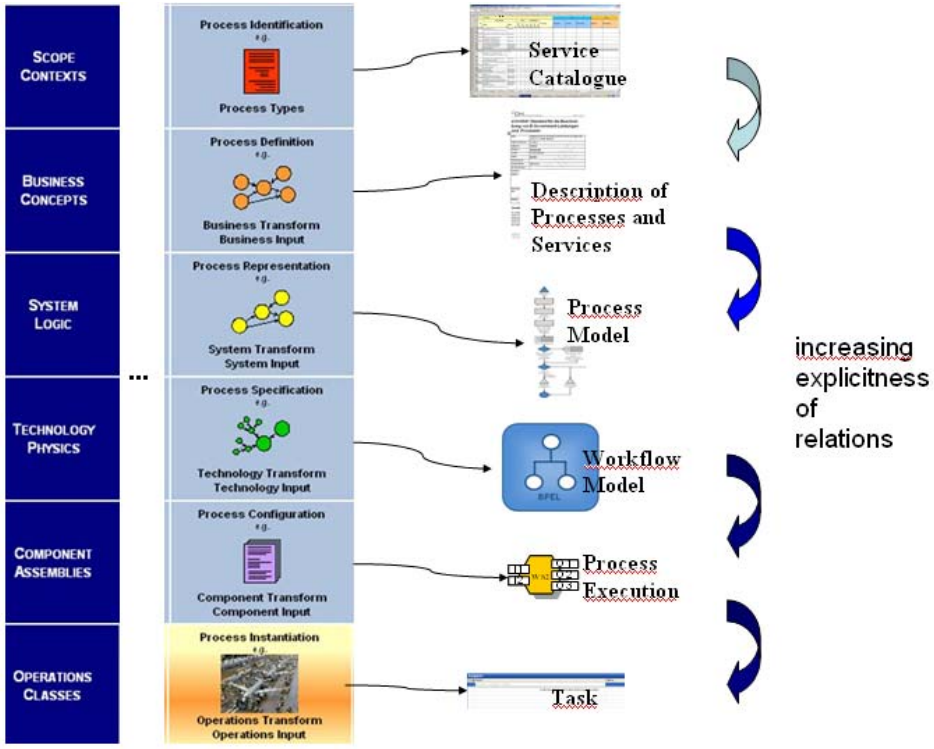

Figure 2: Vertical Relations between EA perspectives and Business Objects

Figure 2 gives an example of how the various perspectives of an Enterprise Model (here for example the function aspect of Zachman's enterprise architecture framework) are related to business objects. The relation, however, is an implicit one as well as the relations between the upper level business objects. Thus, the number of a process, defined in a service catalogue on the Scope Concepts level may occur in the process description on the Business Concepts level but that relation is not formalized and therefore hard to trace. The same holds true for the relation between a process model on the System Logic Layer and the process description. The closer the relation to implementation the more explicit the relations become, e.g. from a BPEL process on the Technology Physics layer and the concrete web-service on the Component Assemblies layer. It is obvious, that changes on the lower level will not be automatically identified on lower levels and vice versa.

In addition to the 'vertical relations', the 'horizontal relations' between Business Objects have to be considered. For the sake of better reading only relations between models on the System Logic layer are regarded in the following.

Figure 3 depicts some of the relations between business objects on the same EA layer: The relation between entities of a database model to the process model, between a records management model and the process model, between the process model and the organizational model and between the database model and the records management model to show, that within one aspect there can be more than one model also related to each other.

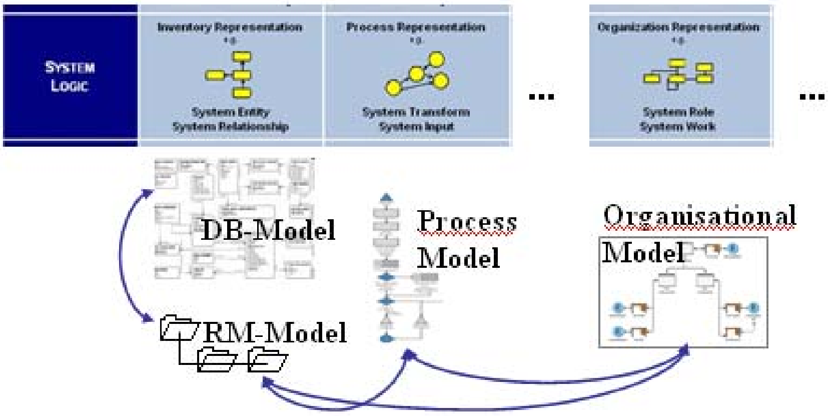

Figure 3: Horizontal Relations between EA aspects and Business Objects

In case the models on the various layers (horizontal and vertical) are managed by IT-systems (e.g. a Business Process Management System like ARIS[5] or ADONIS[6] or a Document Mangement System like Filenet[7]) some of the relations are handled. However none of them comprises all of an EA's aspects and they do not support semantically enriched descriptions.

[5] http://www.ids-scheer.com; http://www.ariscommunity.com/aris-express

[6] http://www.boc-eu.com; http://www.adonis-community.com

[7] http://www-01.ibm.com/software/data/content-management/

In addition to the already high complexity of dependencies between the various business objects on the various layers another dimension has to be added. As the several stakeholders, contributing to a product, often do not have one but several Enterprise Architecture frameworks, an approach is needed to deal with that, too.

Figure 4 depicts a relation between two different Enterprise Architectures (here: the Zachman's, for example used in a company and the eGovernment Architecture, provided by the Swiss Government). Making such kinds of relations transparent is crucial as cross-organizational processes become more an more relevant: in the public sector to provide 'one-stop-shop-sevices' incorporating the various public administrations as well as companies - either as contributor or consumer of a product - as well as in the industrial sector as a chance for example for SME to form virtual enterprises.

Related models can be regarded as the *context* of the model in the focus. Following (Winograd, 2001) defining context as 'an operational term' we consider *context* as 'everything that is not '*text*', that is not *content* of the focused model. To the Process Model, shown in Figure 3, the DB-Model, the RM-Model and the Organizational Model build its context, whereas for the RM-Model, the DB-Model and the *Process Model* are its context. In general: what content is and what context is, is determined by its use.

In this sense, context plays a major role as "if knowledge is extracted and formalized [in an application independent context model] it can then be used in as many different [applications] as necessary, whenever and wherever it is needed" (Mitra and Gupta, 2005).

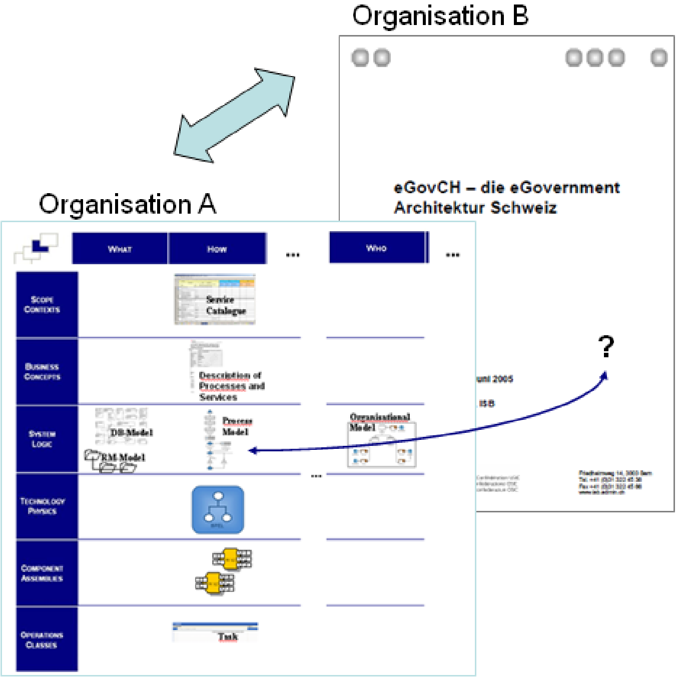

Figure 4: Transversal Relations between EA aspects and Business Objects

In our approach we focus on use of an Enterprise Architecture model as an integration scheme for knowledge structuring to support

- the data model for data integration and
- metadata for non structured documents (information resources)

in the various dimensions of relations.

How to represent Enterprise Architectures

Often EAs are regarded as 'Enterprise Blueprint', aiming to model the relationships between the various components constituting an enterprise (e.g. business processes, organisational structure, resources and technology) in a way that dependencies become visible and can be used for decisions (Bailey, 2006) but they are not build to be executable. Although well structured, EAs are described in natural language fostering ambiguity and lacking of formalization.

(Kang et al., 2010) criticize EAs for the lack of detailed models for the components, of modelling the relationship between the components and the lack of a model for implementation. To overcome that drawbacks (Kang et al., 2010) remodelled the EA with ontologies. They took the Federal Enterprise Architecture, that is based on Zachmann's, used the structure of WordNet to describe terms and SBVR to structure the relationships. However, the model is very detailed and it seems to be hard to extend it into a full blown ontological representation of the EA model (in addition there is no description of how the ontologies are technically implemented).

(Allemang et a., 2005) built the FEA-Reference Model Ontology (FEA-RMO). Even though they provide a lot of insights about dealing with problems, modelling an EA as an ontology it is limited with respect to general use and content. (Kang et al., 2010) state "Although FEA-Reference Model Ontology (FEA-RMO) (Allemang et a., 2005) 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".

(Goudos et al., 2006) base on the Governance Enterprise Architecture (CEA) to address the problem of matching a citizen’s needs with available public services. Based on GEA they created an ontology represented in OWL-DL and created with Protégé (with the OWL plug-in). The approach lacks to important points: the GEA is not implemented by a PA[8] and it does not consider various knowledge levels. Thus we consider that a crucial point. An EA is not necessarily used throughout a company as a whole. That holds true all the more regarding virtual enterprises. Therefore enterprise knowledge has to be modelled accordingly in the ontology to allow for specific enhancements and refinements.

Figure 5 depicts three levels of abstraction for building an enterprise ontology. Organizing knowledge on multi levels has been introduced by Guarino (1998) and

[8] According to (Liimatainen et al., 2007) the adoption of xGEA (an extension of GEA, a model for cross-Government Enterprise Architecture) is only starting in the UK.

was followed by Cacciagrano et al. (2009) for the Resourceome Knowledge Model as well as for the ATHENE approach, as detailed in the Implementation section below.

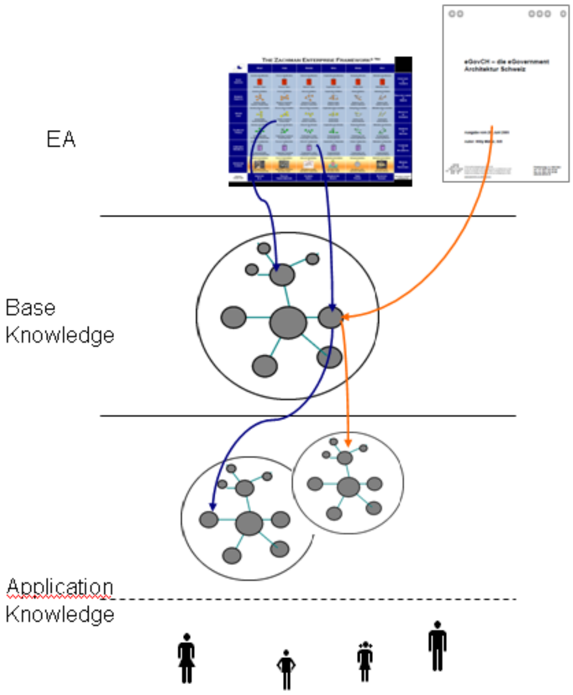

Figure 5: 3 Levels of Abstraction

Implementation

We represented knowledge of an enterprise architecture using two independent systems emphasizing different foci of the architecture model:

- ATHENE has a graphical modeling environment with a common concept repository which allows business users to model the enterprise architecture using a well-known graphical modeling language and to represent the relations between concepts of various architecture aspects
- The Resourceome is a tool based on a multilevel model for structuring the concept repository by separating general (resources and tasks) from domain-specific concepts that provides a suitable structure for representing the different EA aspects.

The integration of ATHENE, the graphical environment with Resourceome the modeling tool, seems to be promising framework for managing an enterprise ontologies.

ATHENE

ATHENE is a graphical modeling environment with a semantic meta-model, i.e. modeling language and models are represented as ontologies (see Figure 6). ATHENE consists of two components: a meta-modeling component to defing modeling languages and the modeling component to build semantic models using a graphical interface.

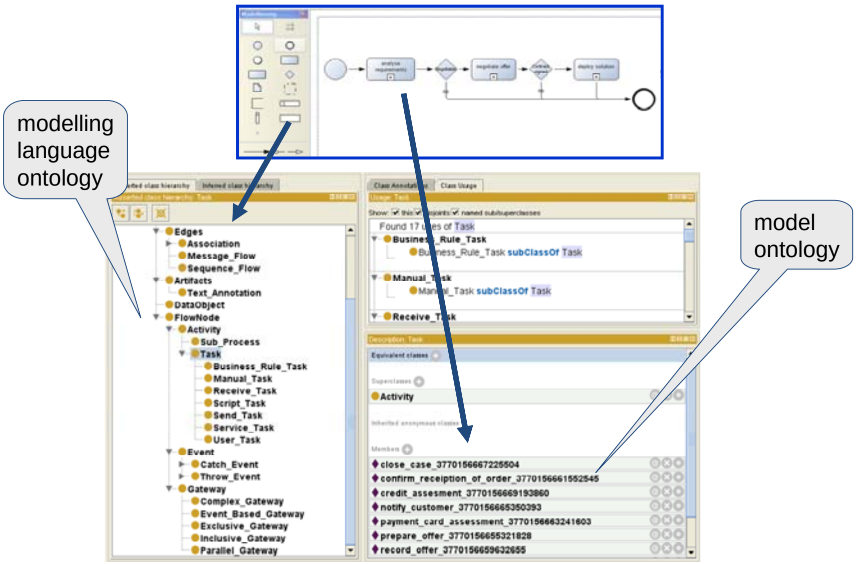

Figure 6: Ontologies for model and modeling language

A modeling language consists of a *concrete syntax* representing the appearance of the modeling elements and an *abstract syntax* representing the semantics of the modeling elements. For the abstract syntax, all the modeling elements of all modeling languages are represented as concepts within a common ontology which consequently also represents the relations between concepts. More precisely, the repository specifies the meta-meta model and allows to define and (re-)use the concepts and properties used by any modeling language as subclasses of the given elements.

For example, the business process model contains modeling elements for process activities with attributes for specifying the participant for the activity (i.e. who has to execute the activity) and which IT system is to be used in order to execute the activity. As already mentioned in the Introduction, the role of the participants and the IT systems are defined in models for the organization and infrastructure aspect, respectively. By defining the meta-model concepts for business processes, organizational structures, and IT infrastructure using in the common ontology, the linking between different models is made explicit.

As a further advantage, when a new modelling language is defined, instead of modelling the elements from scratch already existing concepts can be reused and adapted. In the example of Figure 7 the elements *Start Event*, *Intermediate Event* and *End Event* are reused while the element *Activity* is specialized to *Manual Activity* and *Auto Activity*. A part of their semantics is already described as they are defined as subclasses of *Activity*. Additionally, the model allows reference to instances of the concepts *Person* and *Web Service*, which are modeled modeling languages for Organization and Infrastructure, respectively.

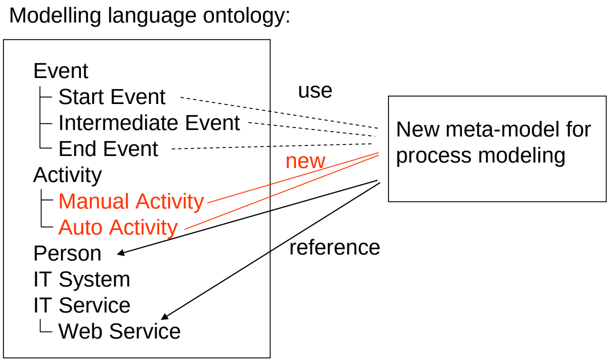

Figure 7: Defining a new modeling language

Resourcesome

Resourceome is a multilevel model and a semantic web tool for managing declarative and procedural knowledge (Cacciagrano et al., 2009). It distinguishes three knowledge levels: top level, base level and application level as shown in Figure 8. Top level ontology is based on the Upper Ontology, describing very general and domain-independent concepts shared across a large number of ontologies. The choice of what Upper Ontology concepts depends on what and how the knowledge is going to be described. The base level ontology describes a specific vocabulary by specializing the terms introduced in the Upper Ontology w.r.t. a particular domain of interest. The knowledge space is partitioned in three different ontologies - Domain, Resource and Task Ontologies - each of which captures and models respectively domain, resource and operational aspects. In particular, Domain and Resource Ontologies follow by an “orthogonal” splitting of the Guarino approach’s Domain Ontology. This “orthogonality” property is realized by a “concerns” relation, which permits to customize, in a very flexible way, the Resourceome knowledge space w.r.t. any specific domain corresponding to the linked Domain Ontology. The application level ontology describes specific concepts, depending on the particular domain, resources and activities.

To implement the Resourceome model we exploit two different languages OWL-DL (Patel-Schneider et al. 2004) and SKOS (Miles and Bechhofer, 2009). We represent concepts as individuals of SKOS concept class. For the representation, visualisation, integration, storing and querying a Resourceome we have developed a semantic web-based tool whose functionalities are organized into three tiers: Front End, Business Logic and Back End. Front End of the system is characterized by a user friendly interface allowing the navigation of the Resourceome model through web-based interfaces or stand-alone applications. The communication with the business logic is based on SOAP messages. The knowledge visualisation is provided by using Grappa Java library (Ganser and North, 2000). The Business Logic provides both management of resources and execution of goals. At this level the reasoner supports the querying of ontologies and guarantees their integrity. Back End encodes the ontologies of Resourceome and stores them in files. The tool provides a guide to add and maintain resources, mapping them on the domain by means of a user-friendly web interface[9].

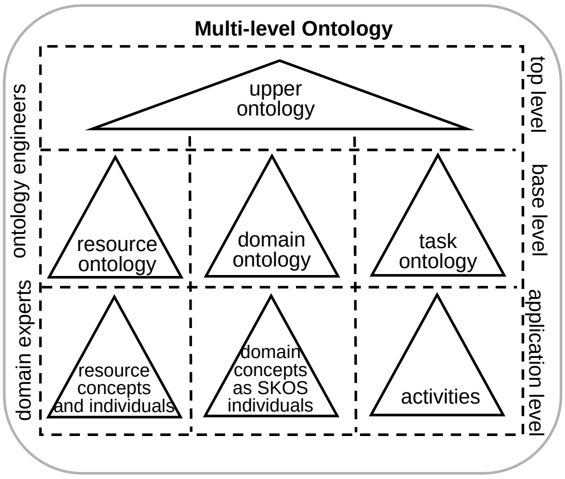

Figure 8: Resourceome ontology model

In the context of this paper,

- the top level ontology describes the common concepts of an enterprise like organization, people, role, process etc. that can be found in all enterprise architectures;
- the base level ontology describes all those composed concepts that are suitable to represent the various aspects of the architecture of a specific enterprise and correspond to the business objects models. Figures 2 and 3 show the relations that can exist among EA aspects and business objects.
- The application level ontology describes all those specific concepts and individuals of a concrete enterprise; they can be regarded as “context” or “content” depending on the view.

To this end, Figure 8 depicts a possible fitting of the Resourceome model to the specific EA framework domain: the Resource Ontology is refined (by its splitting and expanding) into 'IT Systems Infrastructure Ontology', 'People', and 'Information Objects'; the Domain Ontology is refined into 'Organizational Structure and Roles', 'Business Rules' and 'Business Motivation Ontology'; and finally the Task Ontology is refined into 'Process Model Ontology' and 'Business Activity Ontology'.

[9] http://resourceome.cs.unicam.it

The resulting Resourceome Ontology then represents a multi-level integration scheme between an enterprise architecture framework and its concrete data models (cf. Figure 9).

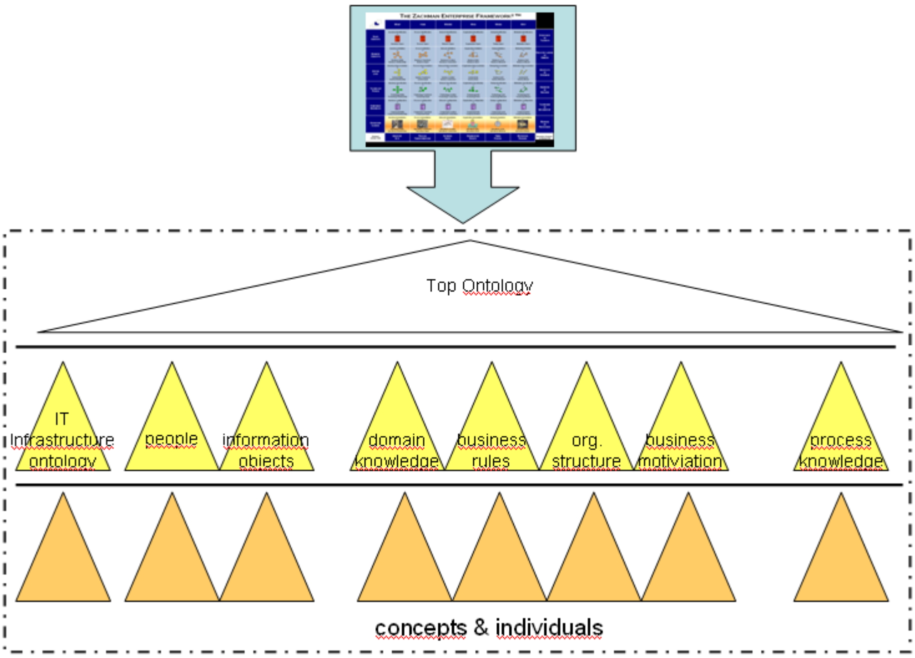

Figure 9: Extension of Resourceome for EA representation

Therefore, the use of ATHENE and Resourceome allow from the one side to offer a user friendly interface for modeling business processes of every enterprise and for any application context from the other hand to semantically represent and manage the enterprise concepts, by maintaining their physical separation.

Conclusion

Cooperation and agility are two requirements enterprises and public administrations alike have to deal with. The benefit of an Enterprise Architecture is widely accepted for building transparency of an enterprise's objects and their relations - representing objects' *context* by linking objects represented in models for different asptects. However, it is not designed for and used to improve daily business operations, because it does not contain the objects themselves but a model of them. The concrete *content*, i.e. the concrete data, process implementations, configuration of IT systems, are - if at all - stored in separate information sources and only loosely linked to the EA. On the other hand side, ontologies are increasingly used to formalize business objects. Representing Enterprise Architectures in an ontology allows for structuring business objects and for quality insurance, thus bringing the potential of Enterprise Architectures to full blown.

Experimental implementations with the ATHENE and Resourceome tools showed that a combination of ATHENE's graphical modeling and Resourceome's ontology structures are preferable to fully support EA maintenance, exploitation and use.

References

Abecker, A. et al. (1998). Toward a Well-Founded Technology for Organizational Memories. IEEE Intelligent Systems and their Applications, Vol. 13, No. 3, 40-48.

Aier, S., Riege, C. and Winter, U.R. (2008). Unternehmensarchitektur – Literaturüberblick und Stand der Praxis. Wirtschaftsinformatik, 4, 292–304.

Allemang, D., Hodgson, R. and Polikoff, I. (2005). FEA Reference Model Ontologies (FEA RMO). Development, 1-43.

Bailey, I. (2006). A Simple Guide to Enterprise Architecture. Futures.

Cacciagrano, D. et al., (2009). Resourceome: a Multilevel Model and a Semantic Web Tool for Managing Domain and Operational Knowledge, Sliema, Malta.

Dietz, J.L. (2006). Enterprise Ontology, Berlin Heidelberg: Springer-Verlag.

Fox, M.S. and Gruninger, M.. (1998). Enterprise Modelling. AI Magazine, Fall 1998, American Association for Artificial Intelligence, 109-121.

Ganser, E.R. and North S.C. (2000). An open graph visualization system and its applications to software engineering. Software: Practice and Experience, Vol. 30, No. 11, pp. 1203-1233.

Guarino, N. (1998). Formal Ontology in Information Systems. IOS Press.

Goudos, S.K., Peristeras, V. and Tarabanis, K. (2006). Mapping Citizen Profiles to Public Administration Services Using Ontology Implementations of the Governance Enterprise Architecture (GEA) models. In: Abecker, A., Mentzas, G. and Stojanovic, L. (Ed.): Proceedings of the Workshop on Semantic Web for eGovernment, at 3rd ESWC, Budva

Hanschke, I. (2010). Strategic IT Management, Berlin Heidelberg: Springer-Verlag.

Kang, D. et al. (2010). Expert Systems with Applications. An ontology-based Enterprise Architecture. Expert Systems With Applications, 37(2), 1456-1464. Available at: http://dx.doi.org/10.1016/j.eswa.2009.06.073.

Leppänen, M. (2007). A Context-Based Enterprise Ontology, Poznan, Poland: Springer Berlin / Heidelberg.

Liimatainen, K., Hoffman, M. and Jukka, H. (2007). Overview of Enterprise Architecture work in 15 countries. Finnish Enterprise Architecture Research Project. Finance.

Miles, A. and Bechhofer, S. (Eds.) (2009). SKOS Simple Knowledge Organization System Reference. W3C Recommendation, Available at: http://www.w3.org/TR/2009/REC-skos-reference-20090818/ (28 April 2010).

Mitra, A. and Gupta, A. (2005). Agile Systems with reusable patterns of Business Knowledge: A component-based approach, Norwood, MA: Artech House.

Mitra, A. and Gupta, A. (2006). Creating Agile Business Systems with Reusable Knowledge, Cambridge: Cambridge University Press.

Patel-Schneider, P.F., Hayes, P. and Horrock, I. (2004). OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation, Available at: http://www.w3.org/TR/2004/REC-owl-semantics-20040210/ (28 April 2010).

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

Uschold, M., King, M., Moralee, S., and Zorgios, Y. (1998). The Enterprise Ontology. The Knowledge Engineering Review, 13 (Special Issue on Putting Ontologies to Use).

Winograd, T. (2001). Architectures for Context. Human Computer Interaction, 16.

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