Building a Conceptual Skeleton for Enterprise Architecture Specifications

Veikko HALTTUNEN[a], Antti LEHTINEN[a] and Riku NYKÄNEN[b]
[a]*Department of Computer Science and Information Systems, University of Jyväskylä, Finland*
[b]*Yomi Software Oy, Finland*

Abstract. This paper describes a way to build a conceptual model for diversified purposes of modelling Enterprise Architectures (EA). It is commonly known that, due to the complexity, Enterprise Architectures need to be considered from several viewpoints. This provokes an integration problem: how to ensure that parallel EA models are consistent. We believe that the best way to solve this problem is to build a generic conceptual model (or an ontology or a language) that is based on the purpose and needs of EA modelling rather than on the metamodels or modelling techniques of the prevailing (viewpoint-specific) domains of EA modelling. In other words, instead of aggregating existing sub-domains of EA we should try to find the core concepts through a careful analysis of EA modelling as a whole. It is important to know the steps of the process through which the resulting conceptual is produced. Therefore, besides the conceptual skeleton and its utilization we provide an in-depth description of the modelling process we have developed and applied.

Introduction

Enterprise architectures (EA) are seen as an important means to convert ever changing business requirements into well-functioning information systems [1, 2, 3]. Managing enterprise architectures is, however, a difficult and very complex task. The complexity of systems is typically managed by division of problem, i.e. by looking at the system from several viewpoints [4, 5, 6, 7]. Hence, enterprise architectures as descriptions of complex systems also comprise a number of different viewpoints (e.g. [8, 9]). This, in turn, provokes another problem: how to ensure that the diverse viewpoint-specific models form a consistent whole ([10], see also [11]). The problem could be solved or at least eased by an architectural language which conceptually ties up the loose viewpoints ([12, 13], see also [9]). We argue that such a language should be based on careful conceptual analysis resulting in exact definitions for the core concepts and a simple structure between the concepts. The viewpoint-specific languages can then be derived from the simple, generic language. Because the viewpoints are many and they evolve, it is necessary that the solution combines the fixed core with a means to modify and extend the core to emergent situations.

During the past few years some attempts have been made to define concepts necessary for modelling enterprise architectures. The RM-ODP standard [14] and its Enterprise viewpoint in particular, is an important effort in this direction. RM-ODP is, however, very complex as a conceptual framework and even the definitions of the core concepts are difficult to understand [15]. In addition, we see that the enterprise viewpoint of RM-ODP is only a relatively small part of the whole framework, and most of the framework aims at building the architecture of an open, distributed system, not a model of an enterprise. Some applications of RM-ODP can be found in IS research literature [16, 17, 18, 19]. A covering presentation of RM-ODP can be found in [20] where one can also find a metamodel of the

Enterprise viewpoint of RM-ODP (p. 468). In our opinion, however, it is too ambiguous as a basic model for EA modelling purposes.

Recently a promising work has been published by Jonkers et al. [12, 13]. Their aim is to define an enterprise architecture language. We share this target for our research. There are, however, relatively important differences between the ways of developing the language[1]. Whereas Jonkers et al. [12, 13] considers enterprise architectures as a solution for the business-ICT alignment problem, we have adopted a more general approach. Despite seeing information systems as an important part of EA we emphasize that, since utilized by people of very different backgrounds, the generic EA language should not be biased by IT thinking and concepts. We have found this problem real in the ongoing LARKKI project[2] where EA tools have been developed for the purposes of different stakeholders. Our work differs from that of Jonkers et al.'s [12, 13] in another respect, too: although both of us utilize existing frameworks like RM-ODP, Jonkers et al. [12] have anchored their concepts on existing architectural domains (product, organization, process, information, data, application, technical infrastructure), while we have aimed at a conceptual model that is not built *from* the viewpoint-specific domains[3] but *for* existing or future viewpoint-specific domains. In other words, instead of building an EA language by *aggregating* the sub-domains of EA (i.e. by finding generalities from the concepts of the sub-domains), we analyze the purpose of EA domain as a whole. The former approach puts emphasis on which (existing) constructions the EA domain is built from, while the latter one considers the overall target of EA modelling. Therefore, the latter one is also more elastic for new ideas and developments. Consider, for example, a situation where a new viewpoint is added to the EA domain.

In our way of building the conceptual model there are two sources for finding the core concepts. First, we can analyze the concepts of existing standards such as RM-ODP to find some starting point for the further analysis. Too narrow approaches (focusing on a specific viewpoint) should be avoided and only the most generic concepts should be selected into the core concepts. Second, we can discuss with practitioners to understand why to build EAs in practice. The dialogue between researchers and practitioners is an important part of our modeling process.

The paper is organized as follows. In the next chapter we define the basic concepts that are needed to understand our approach. In chapter 2 we describe how the conceptual model has been produced and what the current version of the model looks like. Chapter 3 provides some ideas of utilizing the model, and finally chapter 4 includes a brief discussion and conclusions.

1. Basic Concepts

We consider an enterprise as a complex entity that

1. has an identifiable purpose,
2. can explicitly be separated from other entities, and
3. comprises: (a) real world entities like people, several kinds of artefacts like technologies and documents, and (b) more or less abstract entities like organization structures, processes, and information objects (closely connected with documents).

[1] To language we prefer to use term conceptual model (or ontology).

[2] The LARKKI project is a research project that is funded by the National Technology Agency of Finland (TEKES) and three ICT companies: IBM, TietoEnator, and Yomi.

[3] By viewpoint-specific domain we mean a sub-domain within the enterprise architecture domain.

As mentioned previously, enterprises as complex entities are usually modelled from several parallel perspectives. In our terminology such a viewpoint-specific model is referred to as an enterprise model. As Whitman et al. [10] put it, an enterprise model usually represents, due to the complexity of the modelling object a single perspective of the enterprise. Each model that represents any feature of an enterprise can, thus, be called an enterprise model.

An enterprise architecture (EA) is generally seen as a "blueprint" or a "big picture" that assists in the (re-)design of an enterprise. According to Liles and Presley [7] it should define the following three things: what are the activities that an enterprise performs, how should these activities be performed, and how should the enterprise be constructed. We see that enterprise architecture something more: it holds all the necessary components of an enterprise, although a great majority of them may be hidden at the highest level of abstraction. An EA describes the structures through which the static or dynamic features of an enterprise are integrated into a consistent whole. In the other words, whereas an individual enterprise model can describe either the dynamic or static features of an enterprise, the enterprise architecture combines these two into a structure that tells how the enterprise is built. Furthermore, it is important to include purpose statements for each part of the EA. This is often ignored in individual enterprise models that tell the "how" aspect instead of the "why" and "what" aspects (see [21]).

By an enterprise architecture specification we mean a representation through which the EA can be analyzed and discussed by different stakeholders. The formality of the specifications may vary depending on the purpose of use. A generic conceptual model which is developed in this paper helps to interpret and understand EA specifications that are produced according to the principles of the conceptual model. It also helps to integrate the loose viewpoint-specific models into a whole.

2. The Process and Outcomes of Conceptual Modelling

In this chapter we briefly describe how our conceptual model was produced and what the current version of the model looks like. We emphasize that understanding the conceptual modelling process plays an important role when estimating the acceptability of the resulting model.

2.1. The Modelling Context

Our modelling effort was a result of the needs that came up during the first year of the ongoing LARKKI project. The aim of the project is to develop methods and other tools for the diversified purposes of EA modelling and management. In one of the three company parties, Yomi Solutions, there was a need to develop a method and/or a framework that complements their software development method with an EA modelling tool. Quite soon, it was realized that - since the tool would be an important part of communication among different stakeholders - it would be necessary to build the work upon a sounder conceptual basis. Hence, we started the conceptual modelling that will be presented in the subsequent sections. There were two objectives for the process: (1) to keep the conceptual structure as simple and generic as possible, and (2) to use a participatory approach (the utilizers of the results i.e. representatives of the company were involved throughout the process).

2.2. The Modelling Process and its Outcomes in short

The main principle we followed in our modelling process was that - despite using existing terminologies, ontologies, metamodels etc. – the prevailing conceptual structures have been used as little as possible. In the other words, we have attempted to distil the plain concepts from complicated structures. The primary argument for this procedure was simple: in addition to a limited number of concepts we aimed at limited number of relationships. Most of the existing conceptual models seemed to include unnecessary relationships. In our model we have included only those relationships that are necessary and excluded those ones that are possible but not necessary. In our opinion this helps utilizing the conceptual model for the purposes of different domains, even those that occur in the future.

Concerning our conceptual modelling process we owe much to various contributors of the ontology engineering field. Just to mention some, these contributions include the ontology learning process [22], ideas on ontology competence [23], steps for obtaining a global ontology [24], evaluation of ontologies [25], and heuristics based ontology creation methodology [26].

In producing the conceptual skeleton we have combined the top-down and bottom-up approaches. We agree with van der Vet et al. [27] who state that the bottom-up approach of ontology building complements the top-down approach.

A good ontology consists of two mutually supplementary parts: a natural-language for explanatory purposes and a formal part for ambiguity reduction (see [28]). In our process this is aimed at by building a graphical model and by developing a complementary dictionary.

The following tasks are the generic constituent of the modelling process:

1. Find concepts;
2. Define concepts;
3. Analyze and discuss concepts and relationships between them, then, if necessary, go back to (1) or (2).

The actual process can be composed of different combinations of these tasks. The basic idea is to combine analytical/theoretical concept defining with validating discussions with practitioners who will use the modelling language in practice. In each discussion round the concepts are weighted to find the most relevant ones. To put the same thing more formally, assume that Task 1 produces three concepts, say X, Y, and Z. Then Task 2 attaches definitions x, y, and z to the concepts respectively. In Task 3 the concepts X, Y, and Z are analyzed by analyzing the definitions x, y, and z. If definition x, for example, refers to some already identified concept, say Y, that concept must be analyzed further in the same way, and so on. If x refers to concepts not yet identified, say A, B, and C, there is a need to estimate whether or not these concepts should be added to the "dictionary". Furthermore, in discussions some concepts (I, J, K etc.) may appear that are not referred to by any of the already defined concepts. This "emergent" stuff of brainstorming is an important part of building the conceptual model. Common concepts are not included in the conceptual base, only those that are seen as specific for the area of interest (which in this case is EA). By this simple method the coverage and integrity of the conceptual model can be evaluated.

In Figure 1 we depict an instance process that is build upon the generic tasks. This actual process was followed by in the LARKKI project as we built the conceptual model. The process is presented as a linear one, but the cyclic nature of the process can be seen as repetition of similar phases (e.g. analysis).

Step I was carried out by researchers as a literature survey. Based on the survey, steps II and III were organized as group work between the researchers and the representatives of the company. As a result, an extended list of EA concepts was identified. These concepts were seen as candidates for the conceptual model and they were discussed in another group session (step IV). On the basis of this discussion the researchers analyzed the concepts and their relationships (step V). The outcome of this step was the preliminary version of the conceptual skeleton. This version was intentionally left small: only the very basic concepts were included in it. After that, the researchers analyzed the conceptual model further and extended the model by some additional concepts (step VI). This version was discussed with the representatives of the company (step VII), and some improvements were made (step VIII).

In the following we describe the steps in more detail.

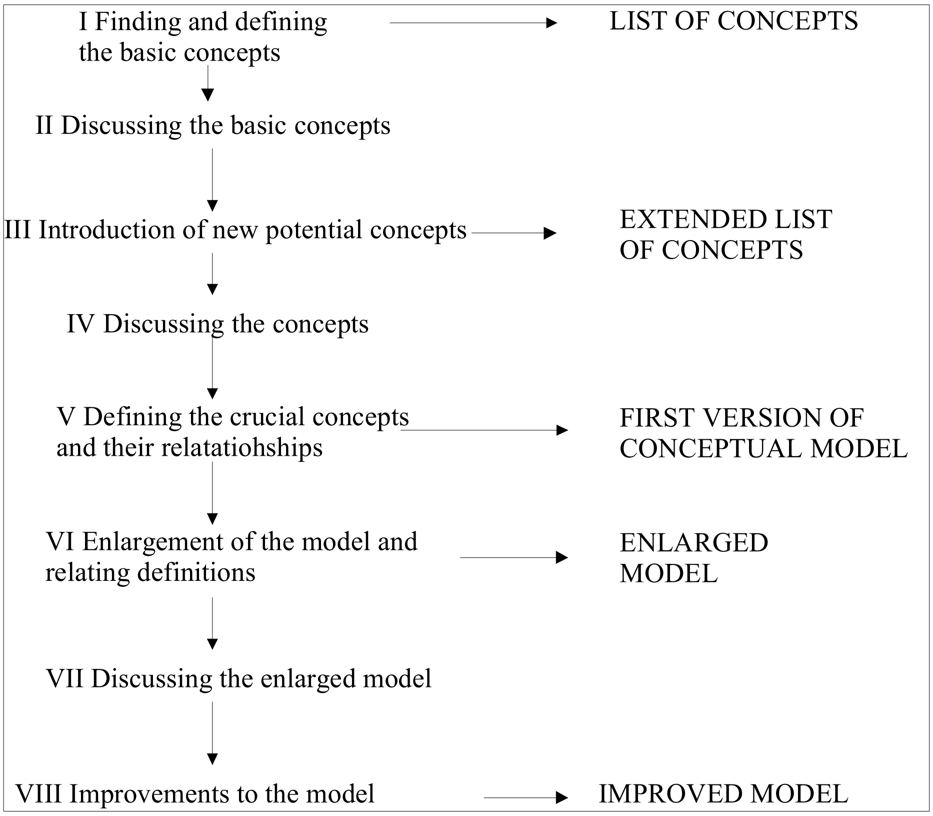


Figure 1. The development process

2.3. A Detailed Description of the Steps

Step I: Analysis of the basic concepts

To agree on a single top-level ontology is a convenient way to integrate ontologies [29]. This principle was the starting point for our modelling efforts. In the first step, we attempted to find the very basic EA concepts that could form the backbone of the model. The selection of the concepts was based on a survey (reported in [30]) on the existing RM-ODP standard and ongoing development work by OMG (MDA and the efforts on Enterprise Distributed Object Computing). One strength of the RM-ODP standard is that it includes very generic concepts that could be utilized in a work like ours. On the basis of our survey we suggested that the following concepts should be included in the conceptual model:

- entity
- action
- purpose
- scope
- policy

The concepts were preliminarily defined as follows:

Entity: This concept is similar to the basic modelling concept ‘object’ of RM-ODP [31]. An entity can be anything that exists. Regarding information systems, entities are users, producers or carriers of information. Entities have their internal properties and particular behaviour. At a given point of time an entity is in a state possible to that entity. An entity can be, for instance, a person, a system or a part of it, an order, an invoice etc.

Action: This concept is similar to the basic modelling concept ‘action’ of RM-ODP. Action is something which happens. Action is associated with at least one entity. Actions can be partitioned into internal actions and interactions.

Purpose: This concept is partially similar to the concept ‘purpose’ described in an Enterprise Viewpoint document [32]. In that document purpose is defined as “the practical advantage or intended effect of the system”. We remark that a system should be understood in a system theoretical sense, i.e. the whole organization can be seen as a system. In our framework the term ‘purpose’ is defined as the reason why an (business) entity, which in this context usually means a business organization, exists. Because the reason tells the final objective of a business entity, we see that ‘purpose’ includes the concept ‘objective’ (which is presented as a separate concept in the previously mentioned ISO/IEC JTC7 document). Hence, ‘objective’ is seen as a sub-concept to ‘purpose’.

Scope: By this concept we mean the distinction between (business) entity and its environment. Thus, in terms of enterprise modelling the scope deals with the boundaries of an enterprise or other business entity (see [20], p. 471). In [32] the scope is defined as “the behaviour that system is expected to exhibit”. This is in line with our definition keeping in mind that behaviour is always associated with entities and a system can be any system like an enterprise organization.

Policy: This concept is in accordance with a set of the RM-ODP policy concepts ([31], pp. 10-11). Policies deal with the behaviour of an entity. They can be declared by specifying obligations, permissions and prohibitions. The concepts of obligations, permissions and prohibitions are often referred to as deontic statements about a system (see [18]).

Step II Discussing the basic concepts and step III Introduction new potential concepts (these steps were intertwined)

At the next two steps the concepts selected in the previous step were discussed, part of them were re-named and/or re-defined and new concepts introduced. Step II was accomplished in two phases: first researchers and practitioners discussed the concepts in own groups and next the concepts were discussed together. Immediately after the group discussion step III was carried out. Actually, the latter part of step II and step III were organized as one group session. Ideas from step III were collected by the researchers. As a result the following list of concepts was formed for further analysis:

- Business Action
- Business Entity
- Domain
- Environment
- Frequency
- Interface
- Objective
- Obligation
- Permission
- Policy
- Process
- Prohibition
- Purpose
- Relationship
- Role
- Scope
- Strategy

It is necessary to remind the reader that the concepts above can be at different level of ontological hierarchy. They were included in the list because they were potentially significant for modelling enterprise architectures.

Step IV: Discussing the concepts

In this step we aimed to attach preliminary definitions to the above concepts in order to work with the hierarchical structures and other relationships among the concepts. Some of the concepts seemed to be more relevant than the others and gained thus more attention by the development group. For some of the concepts it was difficult to find an exact purpose and meaning. Those concepts were ignored at this phase, but they could be added in the conceptual model later on. This step and its outcomes were very intuitive. This was, however, an intended way to work with the concepts, since the next steps were more formal and aimed to reveal inconsistencies and missing parts.

Step V: Defining the crucial concepts and their relationships

Next, on the basis of the group discussions, the researchers defined the selected concepts (Table 1) and developed the first version of the conceptual model (Figure 2). It was produced by using the GOPRR metamodelling language (GOPRR comes from Graph, Object, Property, Role, and Relationship, see e.g. [33]). The GOPRR language seemed very useful since we did not like to make any difference between entities and attributes at this phase. The GOPRR language feature of property link made it easy to modify the model. For example, the modelling concept Purpose was at first modelled as a similar modelling concept type as the modelling concept Business Entity (both modelled as GOPRR object type, represented as square boxes). As the relationship between Business Entity and Purpose was analyzed to be of attribute type, the symbols of the modelling concept could be untouched, the relationship was just modelled as property link.

Table 1. Definitions of EA concepts at the first step of development

EA Modelling Concept	Definition
Business Action	A functional whole that is important to run the business
Business Entity	An entity that is important to the object system. The object system of the highest level is an enterprise as a whole.
Objective	Defined state of things that is to be reached. Described as a goal that is both concrete and measurable. Can be attached as an attribute to Business Entity or Business Action.
Policy	Constrains the behaviour attached to a role. Policies are categorized into obligations, permissions and prohibitions
Purpose	An attribute for Business Entity. Defines why a Business Entity exists.
Role	Is a “container” for behaviour. A Business Entity behaves through roles. The same entity may play several roles in the object system.
Scope	Defines the boundary between an object system and its environment.
Strategy	Defines the means by which the purpose and objectives are to be reached. An attribute for a Business Entity.

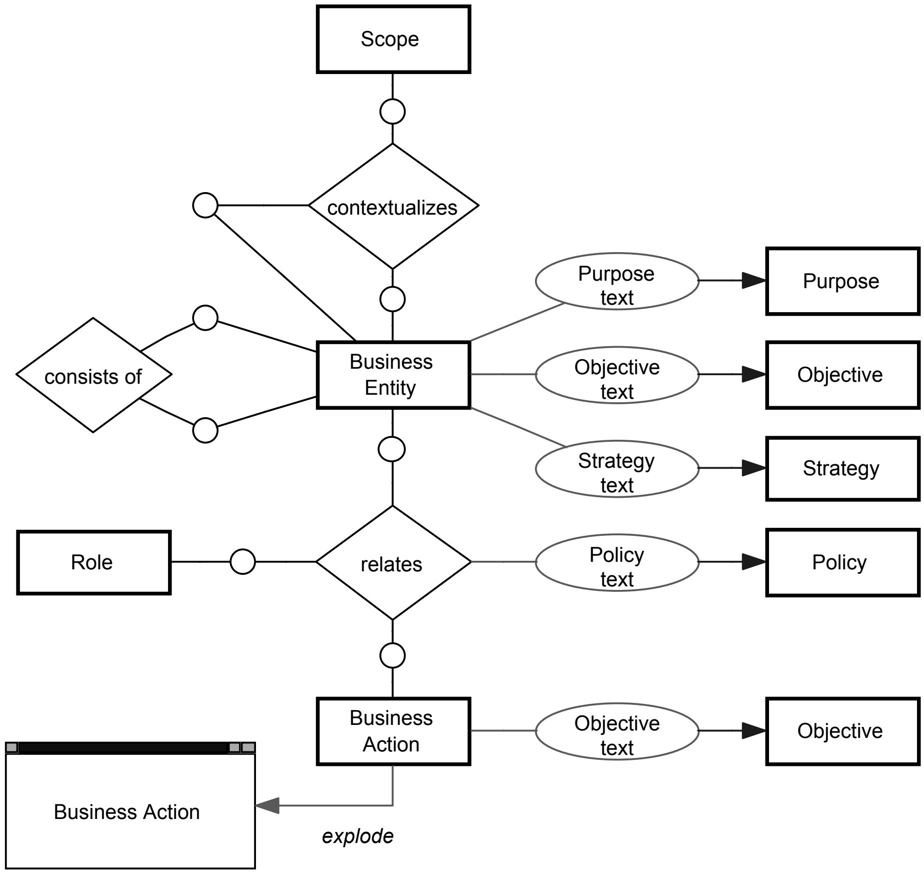


Figure 2. The first version of the conceptual model

As one can see, only a subset of the previous list of concepts was included in this version of the conceptual model. The aim was to find out the core concepts upon which the new versions could be built.

Step VI: Enlargement of the model and relating definitions

In this step the researchers analyzed the conceptual model further, especially concerning the modelling object 'Business Action' which was divided into three special cases: Behaviour, Business Process and Function. The result of this step is depicted in Figure 3.

Step VII: Discussing the enlarged conceptual model

In the next step the enlarged model was discussed with a representative of the company and some improvements were made to the model resulting in the current version of the conceptual model (Figure 4, Table 2). It is thoroughly elaborated on in the next paragraphs.

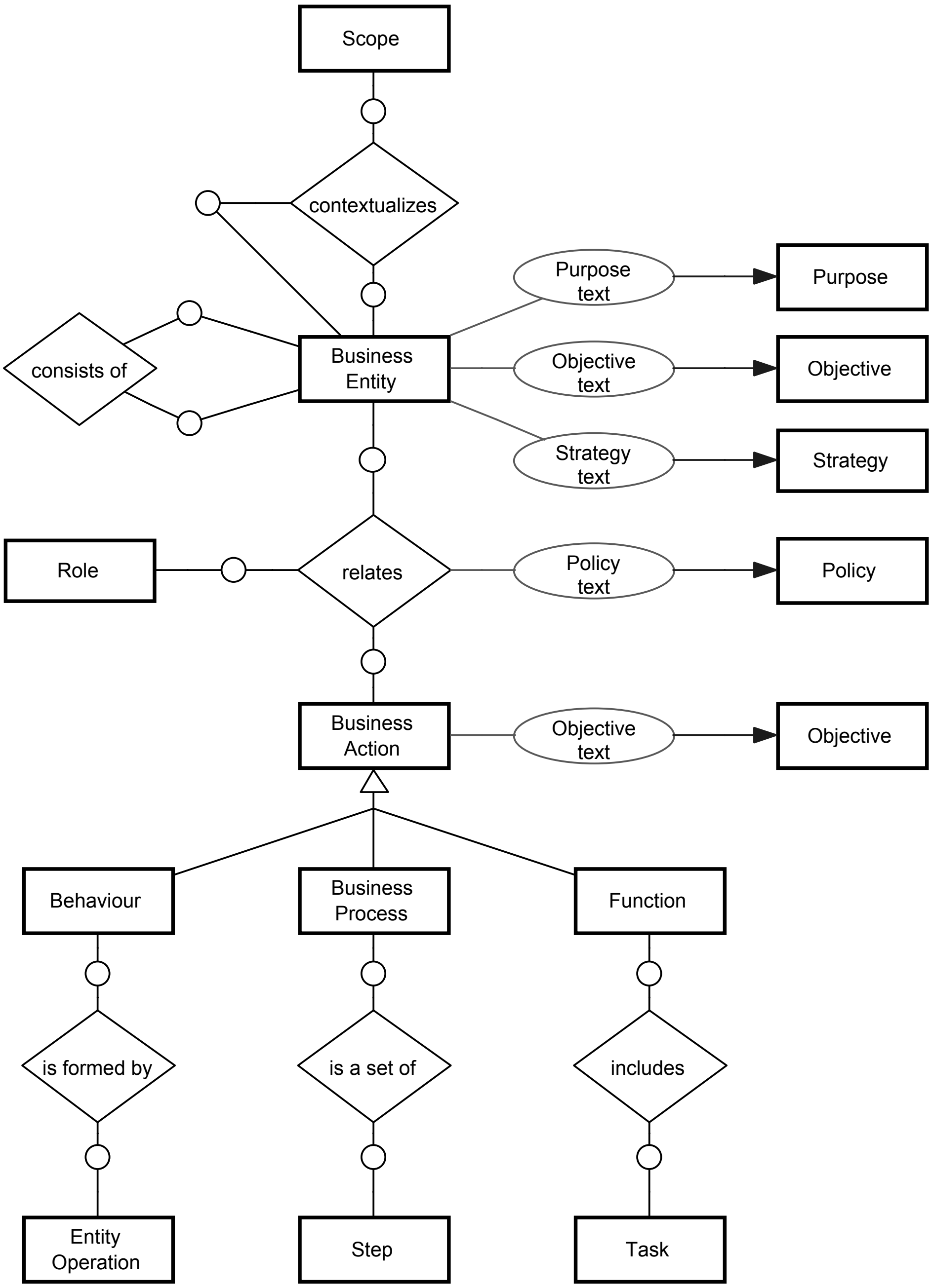


Figure 3. The enlarged conceptual skeleton

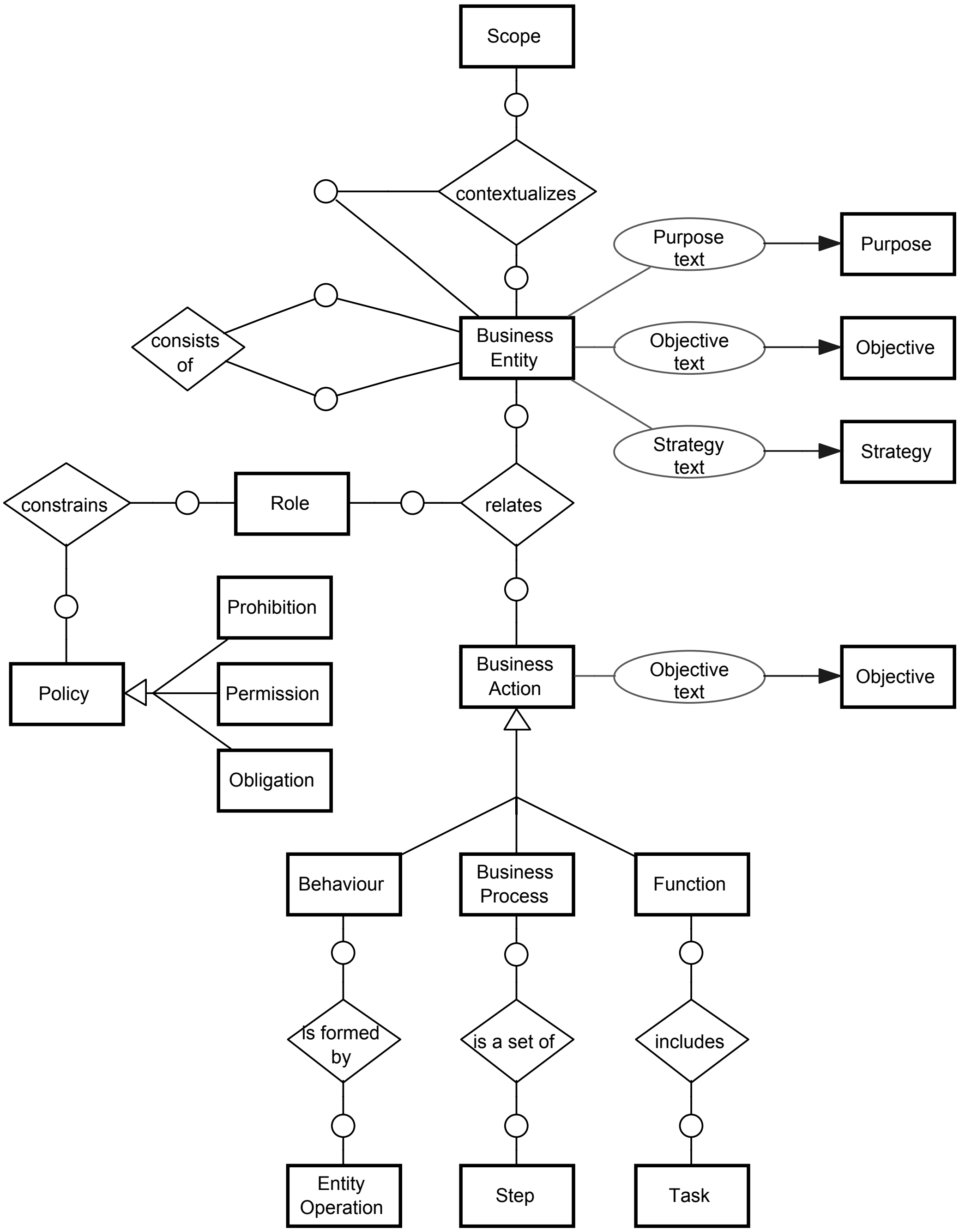


Figure 4. The current version of the conceptual skeleton

Table 2. Refined and new definitions of EA concepts

EA Modelling Concept	Definition
Behaviour	Business Action can be considered from the viewpoint of the *behaviour of a Business Entity*.
Business Action	A functional whole that is important to run the business. Compare with RM ODP concept Action: "Something which happens"
Business Entity	An entity that is important to the object system. The object system is the highest level business entity.
Business Process	Business Action can be considered from the process viewpoint, where a Business Process is a set of consecutive operations or steps that contributes to the achievement or realisation of a business objective/purpose (or of a well defined business activity).
Entity Operation	The operations of a business entity form the behaviour of that entity.
Function	Business Action can be considered from a viewpoint, where functions are ordered according to the hierarchy of the organisation.
Objective	Defined state of things that is to be reached. Described as a goal that is both concrete and measurable. Can be attached as an attribute to Business Entity or Business Action.
Obligation	A category of policies that tells what are the obligatory features of a role.
Permission	A category of policies that tells what is permitted to do by a role.
Policy	Constrains the behaviour attached to a role. Policies are categorized into obligations, permissions and prohibitions
Prohibition	A category of policies which tells what is prohibited to do by a role.
Purpose	An attribute of a Business Entity, which defines why a business entity exists.
Role	Is a "container" for behaviour. A Business Entity performs Business Actions in certain Roles. A Business Entity may be in several roles in the object system.
Scope	Defines the boundary between an object system and its environment.
Step	A constituent of Business Process
Strategy	An attribute of a Business Entity that defines the means by which the purpose is pursued and objectives are to be achieved.
Task	A constituent of a function.
Objective	Defined state of things that is to be achieved. Described as a concrete and measurable statement. Can be attached as an attribute to Business Entity or Business Action.
consists of	A relationship: A Business Entity consists of other business entities.
contextualizes	A relationship: A Scope puts a Business Entity in a context. For an object system there should be one scope definition. The object system is one entity. The other entities of the scope description are environmental entities (belong to the environment of the object system)
constrains	A relationship: A Policy sets constraints for a Role.
includes	A relationship: A Function includes tasks.
is a set of	A relationship: A Business Process is an ordered set of tasks i.e. steps.
is formed by	A relationship: Behaviour is formed by Entity Operations.
relates	A relationship: A Business Entity relates to Business Action and a Role).

The conceptual skeleton described above is based on the idea that an enterprise architecture specification should express the purpose, scope and policies of the object system. This is in accordance with what is stated in the RM-ODP standard [34]. In the following we try to explain how this target is pursued in our model.

The core of the conceptual model is the concept of Business Entity. A business entity is any real or abstract part of the object system (enterprise) and which due to its importance must be taken into account when trying to understand and model the system. Such an entity can be, for example, an organization, an actor of a business action, a user or a producer of products or information, an information system or part of it, a database, or a document. The enterprise itself is the highest level business entity. Business Entities are named things of the "universe of discourse" and together they form collections of things that exist in that context. Business Entities can be categorized in different ways and they can be set on different levels of abstraction. A Business Entity may have several attributes. At the moment we have included only those attributes that have been seen as the most relevant ones for enterprise modelling, namely those that tell the purpose, objective, and strategy of a business entity (see the definitions in Table 2).

In our conceptual model two kinds of relationships between Business Entities are defined. The 'contextualizes' relationship consists of one Business Entity (called object system) that is related to one or more other Business Entities (called environmental entities). When modelling an enterprise the object system is the whole enterprise. That is the highest level entity we are interested in. The other entities (environmental entities) are entities that are (1) at the same conceptual level with the enterprise, (2) important to the enterprise, and (3) not parts of the enterprise. The scope can be seen as an abstract entity that holds both the enterprise and the environmental entities and tells what the enterprise is as well as its environment. Briefly, the scope puts the enterprise into its context. As an entity the scope can have attributes of its own, like description or name. There is a one-to-one relationship between the enterprise and the scope, i.e. one scope defines one enterprise.

The other relationship between Business Entities is named 'consists of'. That relationship tells that there are hierarchical structures between business entities. A business entity can be decomposed into its parts which, in turn, can be further decomposed into their parts, and so on. It is a matter of practice how far this decomposition is extended. To put it simply, decomposition can be done until further decomposition gives no benefit. Another practical point which helps to manage hierarchies is that only one decomposition level is considered at a time. This means that each decomposition picture or other specification should usually cover only one hierarchical level.

Besides Business Entity the second core concept in our model is Business Action. In our conceptual model Business Entity is related to Business Action by a simple 'relates' relationship. Since the concept Role takes part in this relationship this relationship is not a binary relationship. The 'relates' relationship should be read as follows: Business Entities participate Business Actions through Roles. It is necessary to keep the concepts Business Entity and Role apart from each other although they can sometimes look very much alike.

The other concepts include Policy, Behaviour, Business Process, Function, Obligation, Permission, Prohibition, Entity Operation, Step, and Task (see definitions in Table 2).

A Policy is a constraint for a business action, and it is attached to a certain role. So, since the intended behaviour of an enterprise is formed by roles, the policies for the whole enterprise can be aggregated from the roles included in the enterprise. Policies are divided in three categories: Obligations, Permissions and Prohibitions. They are statements that tell what the object system must do, can do, and must not do.

In our model business actions can be considered from three different viewpoints:

- the behaviour of business entities
- business processes
- organizational structures of tasks

The categorization is influenced by practice. It is accomplished by analyzing the different (practical) purposes of modelling business actions. The first viewpoint (the behaviour of business entities) is based on the idea of object-oriented system modelling. The viewpoint is useful when an enterprise is seen as a system, especially when trying to integrate information systems at enterprise level (compare with component thinking). The second viewpoint (business processes) emphasizes the importance of business processes in business modelling. Business processes have been on focus during the last 10-15 years and many modelling practices have been developed around business process re-engineering. These practices have proved to be valuable and applicable. The third viewpoint relies on the fact that throughout the history business organizations have been organized into units which can be seen as groupings of business task reflecting the managerial and organisatory purposes. Sometimes, it is the most convenient and easiest way to model business actions. For example, directors may look at the business actions through organizational structure (hierarchy) which is the mechanism to divide power and responsibility.

3. Utilization of the Conceptual Skeleton

As described before, we have set as our target the building of a conceptual model that is simple and easy to use in different situations and for different purposes of EA modelling. In our conceptual modelling process we followed three principles: (1) the core concepts should be as general as possible (not biased by any existing (viewpoint-specific) domain), and (2) the number of core concepts and the number of relationships between the concepts should be minimized, and (3) there should not be any "circular" relationship between the core concepts (e.g. A refers to B that refers to A; instead, a relationship of a concept to itself is allowed); this helps "read" the model and find the core of the model (which in this case includes Business Entity and Business Action). As a result, we had a conceptual skeleton that, in our opinion, is flexible and comprehensive enough to be easily modified for the purposes of different EA modelling domains. The skeleton has two functions: (1) it serves as a basic model that guides in developing the viewpoint-specific metamodels and, furthermore, enterprise models, and (2) it provides a means for integrating diversified enterprise models, i.e. the skeleton remains the same although the muscles around can be different.

We provide here an example of how to use the skeleton model for building enterprise models that are being integrated into a whole. There are some basic guidelines that are necessary to be followed when utilizing the skeleton model:

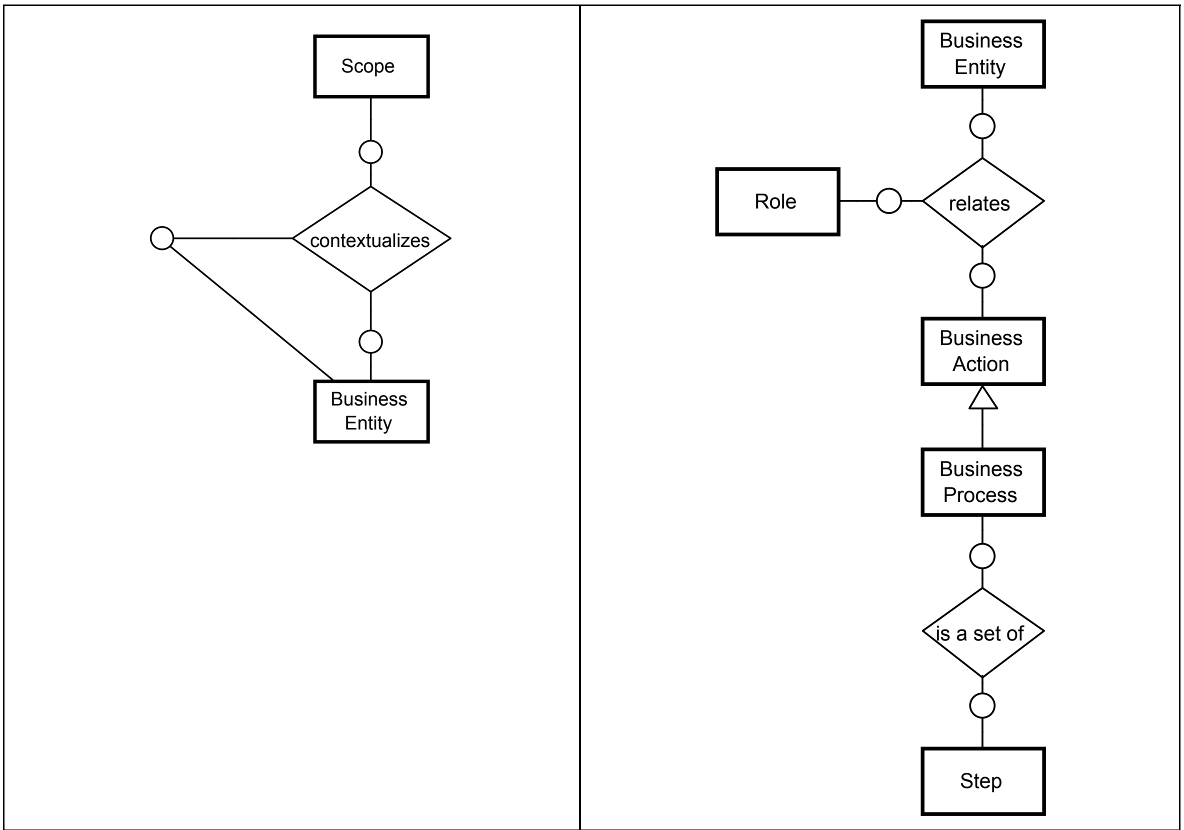
1. Since the concepts of the skeleton model are connected with each other in a very simple way ("the backbone plus the ribs"), integration of separate enterprise models is a natural outcome, if the partial (viewpoint-specific) conceptual models use or explicitly refer to the concepts of the skeleton model.
2. Selection of the concepts of the skeleton model should be based on the purpose of the viewpoint-specific model. By analyzing the purpose against the skeleton model and the complementary dictionary the following issues should be addressed:

- Which concepts of the skeleton model are necessary/useful for the viewpoint
- Which additional concepts are needed for the viewpoint-specific model (specialization of the skeleton concepts)

3. Relationships between the concepts of the skeleton model can be specialized to make the relationships more explicit.
4. Attributes can be attached to the concepts of the skeleton model.
5. The integration mechanism should be decided. Basically, the integration is based on including the same concepts of the skeleton model in the viewpoint-specific conceptual models. If this is not the case, it must be decided on a reference system which is built upon the concepts of the skeleton model.

The above mentioned principles are briefly considered through a simple example in the following sections.

Let us assume that our aim is to model two features of an organization: the environment (context) and the basic processes of the organization. After a careful analysis we decided to select those parts of the skeleton model that are represented in Figure 5.

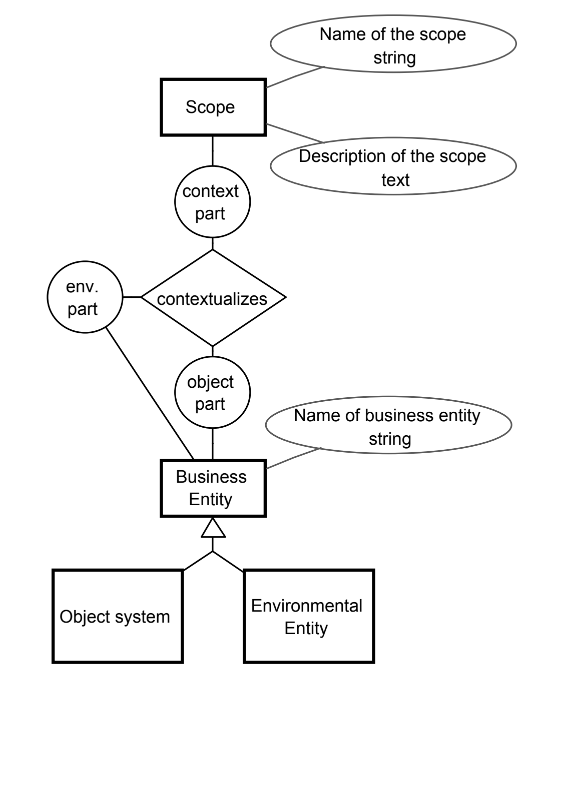


(a) the basic modelling concepts of organization's environment

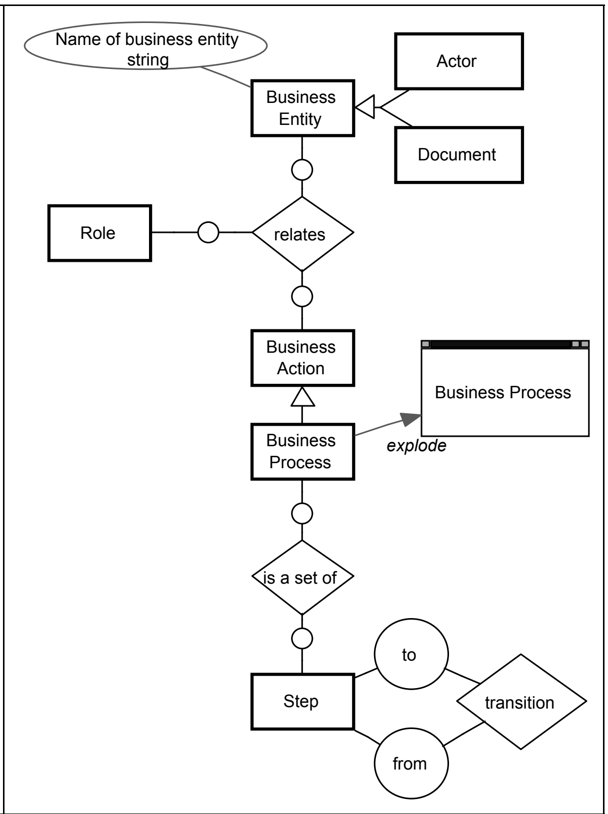
(b) the basic modelling concepts of organizational processes

Figure 5. Selected parts of the skeleton model

Next, we considered what additional concepts or modifications were needed. First, we noticed that the concept 'Business Entity' is too general in terms of both context modelling and process modelling. Regarding context modelling Business Entity was divided into two subtypes, (1) Enterprise and (2) Environmental Business Entity. In the process modelling case, two subtypes were also been seen as necessary, they were (1) Actor and (2) Information Object. The relationships were clarified so that in terms of context modelling the Roles (the circle symbol; This refers to the GOPRR concept Role not to the skeleton model concept Role) were defined for the relationship contextualizes. On the process modelling side an additional relationship, transition, was identified. The attribute name was attached to some of the modelling concepts. The result of all these modifications can be found in Figure 6.



(a) the modified conceptual model of modelling organization's environment



(b) the modified conceptual model of modelling organization's processes

Figure 6. The modified conceptual models

Finally, the integration mechanism(s) needed to be decided. In this simple example it is not problematic since there is a common concept, Business Entity, in both conceptual models. This should be a means to integrate the models into a whole. In other words, referencing from one model to another happens through Business Entities. In practice, it is a slightly more problematic task, because identifying and naming the business entities (instances of Business Entities) vary in different models. The solution would be to decide on an unambiguous identifying system. This is, however, a problem that is always present in modelling efforts. If there were not any concept shared by the partial models the integration would be less straightforward. In this case the skeleton model should be used for finding a "connecting concept" which would build a bridge between the partial models.

Since the two models of our example deal with Business Entities at different levels, it is obvious that additional models would be needed for full integration of the two types of enterprise models. Such a model could be a decomposition model the coverage of which is depicted in Figure 7. The decomposition model would describe how the context level entities would be decomposed into smaller entities that would be relevant in process modelling aspects.

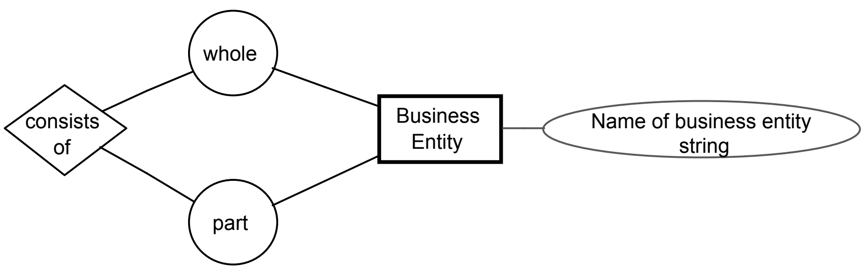


Figure 7. A conceptual model for a decomposition type enterprise model

4. Discussion and Conclusions

Fox et al. [22] argue that the key criterion for evaluating ontologies is that what they call competence. Briefly, competence tells how well an ontology supports the task in which it is utilized. Every task sets different requirements for the ontology. We apply this idea for evaluating our conceptual model.

The primary aim of our work is to provide a simple conceptual basis for diversified EA modelling purposes so that the resulting enterprise models could form a consistent whole. The competence questions can be put as follows: Is the model easy to understand and use? How does the model support in integrating partial enterprise models?

Is the model easy to understand and use? Our model is based on a semantic analysis on the key concepts of EA modelling and relationships between them (compare the approach with the ideas by Huhns et al. [11]). It has been knowingly developed as a skeleton model rather than a big picture of loosely coupled individual models. We have intentionally avoided representing relationships that could be biased by a viewpoint-specific domain. All relationships are seen as generic and, thus, necessary in all circumstances where the concepts that are part of the relationship are used. We have also avoided representing circular relationships between modelling concepts. In reality, circular relationships are common, but at a more abstract level it is useful to avoid them in order to realize what the core concepts are, where to start to read the model. As compared with the existing frameworks, such as the extended Zachman framework [9] and RM-ODP (e.g. [20]), our model has far less concepts and much easier conceptual structure. Furthermore, our model as a generic conceptual model for EA is not biased/charged by IT-related concepts (compare with the EA language by Jonkers et al. [12] the generic concepts of which include concepts like data object, message, and document).

How does the model support in integrating partial enterprise models? Since the partial conceptual models are instantiations and modifications of the skeleton model, integration will be a natural outcome. To obtain a good result requires, however, that (1) the purpose of the modelling effort is carefully analyzed against the skeleton model and the complementary dictionary, (2) the concepts of the skeleton model is used as such, or they are explicitly referred to, (3) practical decisions are made on the integration mechanisms through which integration is implemented in practice (for example, how modelling elements in different models are named). If these principles are followed, the skeleton model provides an unambiguous way to integrate parallel enterprise models into a consistent EA specification. In other words, our aim was not to aggregate existing viewpoints (compare with [9]) or enterprise model types (compare with [35]) but to find a conceptual basis (the most relevant concepts and relationships between them) that helps building situational viewpoint-specific architectural languages having a common core. This is somewhat different from the majority of current approaches.

To sum up, EA modelling is such a large field that the applicability and usefulness of the skeleton model cannot be fully assessed until it has been applied for several purposes of EA modelling. It is possible, even obvious, that the skeleton evolves in time. We, however, are convinced that our way of creating the conceptual skeleton is durable and can be applied to many other situations where one is trying to catch the conceptual basis of a complex, multi-view task similar to EA modelling.

References

[1] Armour F.J., Kaisler S.H., Liu S.Y. (1999a) A Big-Picture Look At Enterprise Architectures. IT Professional. Volume: 1, Issue: 1, Jan/Feb 1999, pp 35-42.

[2] Armour F.J., Kaisler S.H., Liu S.Y. (1999b) Building an Enterprise Architecture Step by Step, IT Professional, Volume: 1, Issue: 4, Year: Jul/Aug 1999.

[3] Chalmeta R., Campos C., Grangel R. (2001). Reference Architectures for Enterprise Integration. The Journal of Systems and Software. Num. 57. pp. 175-191.

[4] Deubler H-H. (2001) Employing multiple views to separate large-scale software systems. The Journal of Systems and Software. Volume 56, Number 2, 1 March 2001.

[5] Groznik A., Kovacic A. (2002) Business renovation: from business process modelling to information system modelling. Proceedings of the 24th International Conference on Information Technology Interfaces ITI 2002, 24-27 June 2002, pp. 405 -409 vol.1.

[6] Kateel, G. P. Kamath, M. Pratt, D.B. (1996) An Overview of CIM Enterprise Modeling Methodologies. Winter Simulation Conference, pp. 1000-1007.

[7] Liles, D. and A. Presley (1996). Enterprise Modeling Within an Enterprise Engineering Framework. In Proceedings of the 28th conference on Winter simulation. Coronado, California, United States.

[8] Zachman, J.A. (1987) A Framework for Information Systems Architecture. IBM Systems Journal, vol. 26, no. 3, pp. 276-292.

[9] Sowa J. F., Zachman J. A., (1992) Extending and formalizing the framework for information systems architecture, IBM Systems Journal, Vol. 31, No. 3, pp. 590-616.

[10] Whitman, L., Ramachandran, R., Ketkar, V. (2001) A taxonomy of a living model of the enterprise. Winter Simulation Conference 2001: 848-855.

[11] Huhns M., Jacobs N., Ksiezyk T., Singh M., Shen W.M., Cannata P. (1993) Integrating enterprise information models in Carnot. Proceedings of International Conference on Intelligent and Cooperative Information Systems, Rotterdam, Holland.

[12] Jonkers H., van Buuren R., Arbab F., de Boer F., Bonsangue M., Bosma H., ter Doest H., Goroenewegen L., Guillen Scholten J., Hoppenbrouwers S., Iacob M.-E., Janssen W., Lankhorst M., van Leeuwen D., Proper E., Stam A., van der Torre L., Veldhuijzen van Zalten G., (2003) Towards a Language for Coherent Enterprise Architecture Descriptions, Proceedings of the Seventh IEEE International Enterprise Distributed Object Computing Conference (EDOC'03).

[13] Jonkers H., Lankhorst M., van Buuren R., Hoppenbrouwers S., Bonsangue M., van der Torre L., (2004) Concepts for Modelling Enterprise Architectures, International Journal of Cooperative Information Systems, vol. 13, no. 3, Sept. 2004, pp. 257-287.

[14] ISO/IEC 10746 Information technology – Open Distributed Processing - Reference Model.

[15] Genilloud G., Wegmann A. (2000) A foundation for the concept of role in object modelling. Enterprise Distributed Object Computing Conference, 2000. EDOC 2000. Proceedings. Fourth International, pp. 76-85.

[16] Blanc X., Gervais M.P., Le-Delliou R. (1999) Using the UML language to express the ODP enterprise concepts. In Proceedings of the Third International Enterprise Distributed Object Computing Conference, pp. 50-59.

[17] Tanaka, A., Nagase, Y., Kiryu, Y., Nakai, K. (2001) Applying ODP Enterprise Viewpoint Language to hospital information systems. Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International , 4-7 Sept. 2001, pp. 188 -192.

[18] Linington, P. Milosevic, Z. Raymond, K. (1998) Policies in communities: extending the ODP enterprise viewpoint. Proceedings of the 2nd International Conference on Enterprise Distributed Object Computing, Workshop. pp. 14-24.

[19] Wegmann, A., Naumenko, A. (2001) Conceptual Modeling of Complex Systems Using an RM-ODP Based Ontology Proceedings of the 5th IEEE International Enterprise Distributed Object Computing Conference - EDOC 2001, Seattle, USA, September 2001, pp. 200-211.

[20] Putman, J. (2001) Architecting with RM-ODP. Upper Saddle River, N.J. Prentice Hall PTR , 2001.

[21] Gordijn J., Akkermans H., van Vliet H., Business Modelling is not Process Modelling. Conceptual Modeling for e-Business and the Web, ECOMO 2000.

[22] Maedche A., Staab S. (2001) Ontology Learning for the Semantic Web, IEEE Intelligent Systems, 16(2): 72-79, March/April 2001.

[23] Fox M.S., Barbuceanu M., Gruninger M., (1995), An Organisation Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour, Computers in Industry, Vol. 29, pp. 123-134.

[24] Lopez de Vergara, J.E., Villagra, V.A., Berrocal, J., Asensio, J.I., Pignaton, R. (2003) Semantic Management: Application of Ontologies for the Integration of Management Information Models, Proceedings of the Eighth IFIP/IEEE International Symposium on Integrated Network Management, Colorado Springs, Colorado, U.S.A. 24-28 March 2003.

[25] Gómez-Pérez A. (1995) Some ideas and examples to evaluate ontologies. Proceedings of the 11th Conference on Artificial Intelligence for Applications, 20-23 Feb. 1995 Page(s): 299 -305.
[26] Sugumaran, V., Storey, V. C. (2002) Ontologies for conceptual modeling: their creation, use and management. Data & Knowledge Engineering 42, pp. 251-271.
[27] van der Vet, P.E., Mars, N.J.I. (1998) Bottom-up construction of ontologies. IEEE Transactions on Knowledge and Data Engineering, Volume 10 Issue: 4, July-Aug. 1998, pp. 513 -526.
[28] Gruber T. R. (1991) The role of common ontology in achieving sharable, reusable knowledge bases. In J. A. Allen, R. Fikes, and E. Sandewall, eds, Proceedings of the Second International Conference. San Mateo, CA. Morgan Kaufman.
[29] Guarino N. (1998) Formal Ontology and Information Systems. In N. Guarino, editor, Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy. IOS Press, June, pp. 3-15.
[30] Halttunen V. (2002), Conceptual Framework for Enterprise Architecture Modelling, Internal Project Report of the LARKKI-project, University of Jyväskylä.
[31] ISO/IEC 10746-2 (1996) Information technology - Open Distributed Processing - Reference Model. Part 2: Foundations.
[32] ISO/IEC JTC7 N2359 FCD 15414: Information Technology - Open Distributed Processing – Reference Model - Enterprise Viewpoint (A Draft).
[33] Tolvanen J.-P., (1998) Incremental Method Engineering with Modeling Tools, PhD Thesis, University of Jyväskylä, Jyväskylä.
[34] ISO/IEC 10746-3 (1996) Information technology - Open Distributed Processing - Reference Model. Part 3: Architecture.
[35] Bubenko J., Persson A., Stirna J., (2001) EKD User Guide, IST-2000-28401.