A Modelling Approach to Support Enterprise Architecture Interoperability

Wided Guédria[1], Khaled Gaaloul[1], Yannick Naudet[1], and Hend*erik* A. Proper[1,2]

[1] Centre de Recherche Public Henri Tudor, Luxembourg
[2] Radboud University Nijmegen, Nijmegen, Netherlands
{wided.guedria,khaled.gaaloul,erik.proper,yannick.naudet}@tudor.lu

Abstract. In this paper, we elaborate on the Enterprise Architecture (EA) and how it can be improved by using Enterprise Interoperability (EI) requirements. We describe how enterprise interoperability is related to EA concepts by especially analysing the definition and the foundation of interoperability, highlighting the relationships with EA. We then propose a conceptual model that defines the enterprise architecture interoperability domain. In doing so, conceptual descriptions of systems, interoperability problems and solutions are identified. The designed model can be inferred for decision-aid related to interoperability. Finally, we illustrate our approach by means of a case in the automotive industry.

1 Introduction

As technology becomes more far-reaching and interconnected, the need of interoperability is becoming increasingly important. To interact with another system or to add new components, a system has to be sufficiently flexible to operate and adapt to other systems. Three main research domains addressing Enterprise Interoperability (EI) issues were identified in [1]: 1) Enterprise modelling dealing with the representation of the inter-networked organisation to establish interoperability requirements 2) Architecture & Platform defining the implementation solution to achieve interoperability 3) Ontologies addressing the semantics necessary to assure interoperability.

In this paper, we focus on the modelling of networked organisation interoperability that need to be taken into account when dealing with Enterprise Architecture (EA). We rely for this on the ArchiMate modelling standard for EA [2]. Amongst other architecture modelling languages, ArchiMate can be used to model an organisation's products and services, how these products and services are realized/delivered by business processes, and how in turn these processes are supported by information systems and their underlying IT infrastructure [3]. However, such techniques and languages do not address interoperability issues and its aspects in a satisfactory way [4,5]. In this paper, we put our effort in the improvement of an architecture language by addressing interoperability core aspects. We restrict ourselves mainly to modelling concepts that are related to

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 189–198, 2013.

EI concerns (i.e. operational concerns) in enterprise networking context, as discussed in [6]. The use of an EA helps to chart the complexity of an organisation [2]. Many organisations have recognized the value of architectures and use them during the development and evolution of their products, processes, and systems [2]. In light of these, the main research questions addressed in this paper can be summarised as follows: How to integrate interoperability concepts into EA models? In particular, how to integrate EI concepts into ArchiMate meta-model and how this can leverage EA principles to deal with interoperability problems? The structure of the paper will be as follows: Section 2 reviews the EI domain and its core concepts. Section 3 reviews EA modelling languages and presents the motivations of choosing ArchiMate. Section 4 presents the integrated meta-model, which includes the main concepts of interoperability into the ArchiMate meta-model. In section 5 a case study is presented to illustrate this integration. Finally section 6 concludes and presents future work.

2 Enterprise Interoperability

Interoperability is ubiquitous but not easy to understand due to its numerous definitions and interpretations. In [7], the authors point out that thirty-four definitions of interoperability were proposed since 1977. The most commonly acknowledged definition is the one provided by IEEE, considering interoperability as *the ability of two or more systems or components to exchange information and to use the information that has been exchanged* [8]. In order to understand the EI domain, we need to study the core concepts and elements of the EI and the operational entities where interoperations take place within an enterprise. These are mainly defined through the Ontology of Enterprise Interoperability (OoEI) and the Framework of Enterprise Interoperability (FEI), that are reviewed in the following sections.

2.1 The Ontology of Enterprise Interoperability (OoEI)

The first attempt to define the interoperability domain was made by [9], where a model for defining interoperability as a heterogeneous problem induced by a communication problem was proposed. On the basis of these research efforts, the Ontology of Enterprise Interoperability (OoEI) [5] as an extension of the Ontology of Interoperability (OoI) [10] was developed using the Ontology Web Language (OWL). This OoEI aims at formally defining Enterprise Interoperability (EI) while providing a framework to describe problems and related solutions pertaining to the interoperability domain. Interoperability exists because there are at least two *Systems* and a *Relation* between them. The *relation* is of primary importance and is the source of interoperability problems [11]. A *System* is defined as a set of interconnected parts, having a *Structure*, a *Function*, an *Objective* and a *Behaviour* [12]. These concepts are necessary to understand a system. The OoEI was defined based on an analysis on the EI frameworks and models [5]. It describes systems as interrelated subsystems: A *System* is composed of *SystemElement*s, which are systems themselves, and *Relation*s. The

Relation concept formalizes the existing relationships inside a system, which is the source of the occurrence of interoperability problems. The OoEI makes the distinction between *Structural relation* and *Behavioural relation.* A *structural relation* refers to relations between each couple of sub-systems of the system. It relates to the structure of the related systems and concerns their interfaces, their models or the representation of their models. A *behavioural relation* is a non structural relation. It has a direct influence on systems without being related to a particular subsystem. This kind of relation does not concern the system's structure itself but any relation that influences the system's behaviour without being related to an element of the system's structure.

An enterprise is considered as a complex system in the sense that it has both a large number of parts and the parts are related in ways that make it difficult to understand how the enterprise operates and to predict its behaviour [12].

Dealing with EI requires considering the enterprise from a general perspective, taking into account not only its different components and their interactions but also the environment in which it evolves and the interface through which it communicates with its environment. The *Interface* is a *SystemElement* through which a connection between the *System* and its *Environment* can be established. It also represents the systems boundaries.

The establishment or diagnosis of EI has led to identify the different operational levels that are concerned: *Business*, *Process*, *Service* and *Data* interoperabilies (i.e. the EI concerns as defined by FEI). Interoperability is implemented as a subclass of the *Problem* concept. Problems of interoperability exist when there is a relation, of any kind, between incompatible systems in a super- system they belong to or system they will form. An exhaustive description of the OoEI model can be found in [5].

2.2 Framework for Enterprise Interoperability

The main purpose of an interoperability framework is to provide an organising mechanism so that concepts, problems and knowledge on enterprise interoperability can be represented in a more structured way [13]. The Framework for Enterprise Interoperability (FEI) was developed within the frame of INTEROP Network of Excellence [13] and is published as an international standard (ISO 11354 - 1). It defines a classification scheme for interoperability knowledge according to three dimensions: Interoperability barriers, EI concerns and interoperability approaches. According to FEI, the establishment of interoperability consists in removing all the identified barriers. Three kinds of barriers are identified: *Conceptual* (syntactic and semantic differences of information to be ex-changed), *Technological* (incompatibility of information technologies: architecture & platforms, infrastructure, etc.), and *Organisational* (definition of responsibilities and authorities). *Interoperability Concerns* represent the areas concerned by interoperability in an enterprise. Four concerns are defined, namely *business interoperability* (work in a harmonized way to share and develop business between companies despite the difference of methods, decision making, culture of enterprises, etc.), *process interoperability* (make various processes work together.

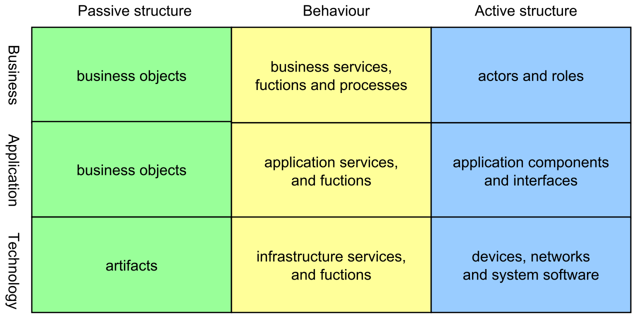
In the interworked enterprise, the aim is to connect internal processes of two companies to create a common process), *service interoperability* (making work together various services or applications by solving the syntactic and semantic differences) and *data interoperability* (make work together different data models with different query languages to share information coming from heterogeneous systems). Finally, three *interoperability approaches*, or ways to establish working interoperations, are considered: *The integrated approach* (characterized by the existence of a common format for all the constituents systems); the *unified approach*, characterized by the existence of a common format but at a meta-level; the *federated approach*, in which no common format is defined. This approach maintains the identity of interoperating systems, nothing is imposed by one party or another and interoperability is managed in an ad-hoc manner.

3 Enterprise Architecture

An architecture is the fundamental organisation of a system embodied in its components, their relationships to each other and to the environment,and the principle guiding its design and evolution [14]. The unambiguous specification and description of components and especially their relationships in architecture require a coherent architecture modelling language [15]. Current languages for modelling in the area of organisations, business processes, applications, and technology share a number of aspects on which they score low [16]. The relations between domains (views) are poorly defined, and the models created in different views are not further integrated. Most languages have a weak formal basis and lack a clearly defined semantics. Moreover, these miss the overall architectural vision and are confined to either the business or the application and technology subdomains.

In [15], the authors have compared a selection of standards and languages (e.g. RM-ODP, UML and the UML EDOC profil, BPMN and ARIS) to ArchiMate [17], using three criteria for comparison: frameworks, architectural viewpoints and domains that are covered by each language. According to their comparison, ArchiMate distinguishes itself from most other languages by its well defined meta-model, concepts and, most importantly, its relations. The abstraction level of ArchiMate simplifies the construction of integrated models, where most languages appear to persuade architects to detailed modelling. Detailed modelling of most aspects also can be performed in ArchiMate, used as an "umbrella language" [15]. ArchiMate defines three main layers [16]: 1) *The Business Layer* offers products and services to external customers, which are realized in the organisation by business processes (performed by business actors or roles); 2) *The Application Layer* supports the business layer with application services which are realized by (software) application components; 3) *The Technology Layer* offers infrastructure services (e.g., processing, storage, and communication services) needed to run applications, realized by computer and communication devices and system software.

The core concepts that are found in each layer of the language are depicted in Fig 1. A distinction between structural aspect and behavioural one is made [16].



	Passive structure	Behaviour	Active structure
Business	business objects	business services, fuctions and processes	actors and roles
Application	business objects	application services, and fuctions	application components and interfaces
Technology	artifacts	infrastructure services, and fuctions	devices, networks and system software

Fig. 1. The Core Concepts of ArchiMate [16]

4 ArchiMate and Enterprise Interoperability

Besides the core concepts shown in Fig 1, which are mainly operational in nature, there are a number of other important aspects, some of which may cross several (or all) conceptual domains; e.g. Interoperability. In this section we integrate the core concepts of OoEI into the ArchiMate meta-models. In doing so, we propose syntactic and semantic mapping between both meta-models which has been facilitated by applying existing approaches for ontology mapping approaches [18].

4.1 Business Layer and Interoperability

Fig 2 gives an overview of the ArchiMate business layer integrating interoperability aspects. The added concepts, from the OoEI, are presented in dark gray. The structural aspects at the business layer refers to the organisation structure, in terms of the actors that make up the organisation and their relationships. The central concept is the *Business actor* [16]. On the side of EI, the core concepts are the *System* and *Relation*. The whole enterprise, where organisational entities are performing behaviour, is a system. Given that, the *Business actor* is considered as a *SystemElement* within the organisation. The work that an actor performs within an organisation is based on a defined role. In some cases, the work results in a collective effort of more than one business role: this is called Business collaboration. From a behavioural point of view, this is assigned to the concept *Business interaction*. Business interaction is defined as a behaviour element that describes the behaviour of a business collaboration [16]. The ability of an enterprise to interoperate allows business interaction through the collaboration of its business roles. Given that, we define *Business collaboration* as a specialisation of *Structural relation* and *Business interaction* as *Behavioural relation.*

4.2 Application Layer and Interoperability

Fig 3 gives an overview of the ArchiMate application layer integrating main concepts and relations of the EI domain. The main structural concept of the

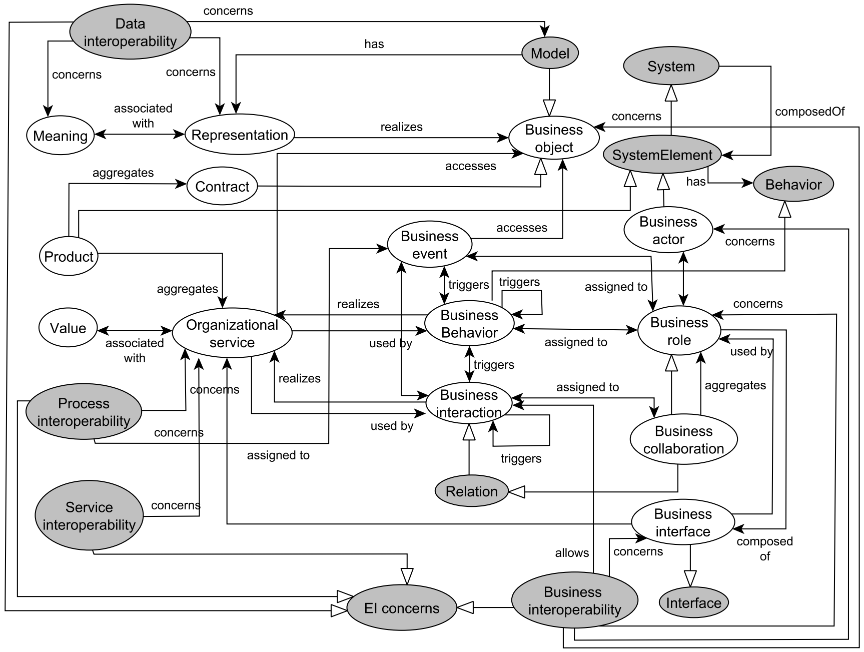


Fig. 2. Business layer meta-model with interoperability concepts

application layer is the *Application component* concept. It is a modular, deployable, and replaceable part of a system that encapsulates its contents and exposes its functionality through a set of interfaces. We define an *Application component* as a specialisation of a *SystemElement*. The interrelationships between components are an essential ingredient in the application of EA techniques. Subsequently we find that the *Application collaboration* concept is assigned to the *Application interaction*. The *application collaboration* is defined as an aggregation of two or more application components that work together to perform collective behaviour [16]. The compatibility and subsequently interoperability between the application components allows application collaboration and application interaction. Given that, we define *Application Collaboration* as a specialisation of *Structural Relation* and *Application Interaction* as a specialisation of *Behavioural Relation*.

4.3 Technology Layer and Interoperability

Fig 4 gives an overview of the ArchiMate technology layer and the main relationships with the integrated interoperability concepts. The main structural concept for the technology layer is the *Node*. It is a computational resource upon which artifacts may be stored or deployed for execution [15]. We define a *Node* as a specialisation of a *SystemElement*. Artifacts are used to model the representation, in the form of, e.g. a file of a data object or an application component,

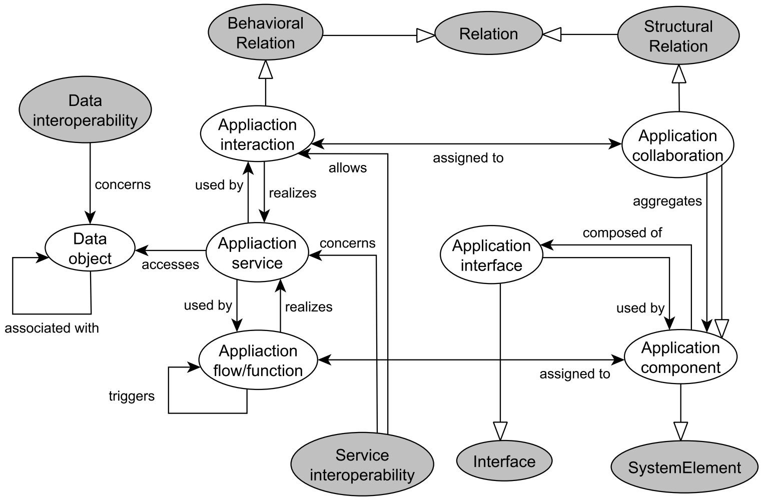


Fig. 3. Application layer meta-model with interoperability concepts

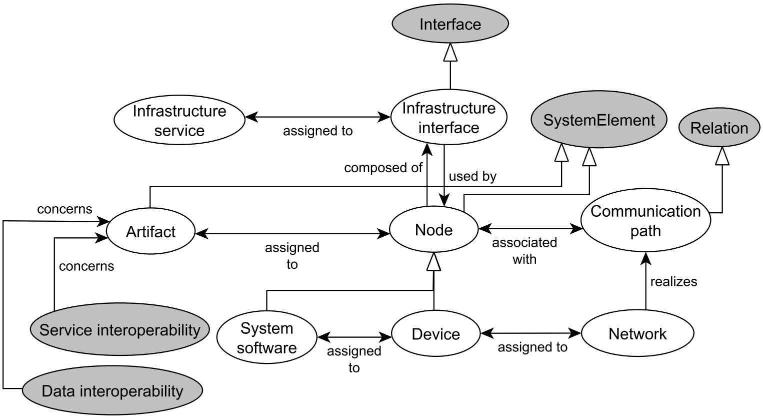


Fig. 4. Technology layer meta-model with interoperability concepts

and can be assigned to (i.e., deployed on) a *node*. Given that, artifacts can be source of interoperability problems at data level as well as service level. The technology layer provides *Infrastructure services* to be used by applications [15]. In this sense, service interoperability concerns *Infrastructure services*, as it can be source of interoperability problems between applications. An *Infrastructure interface* is a specialisation of the concept *Interface*. It specifies how the infrastructure services of a node can be accessed by other nodes (provided interface), or which functionality the node requires from the environment.

5 Illustrative Example

To illustrate and validate the proposed meta-model, we present a case study of a multinational company. Some information are intentionally skipped or not detailed due to the enterprise privacy policies. The company is part of a German group, which is specialized in automobile manufactures with modern wiring harness systems, exclusive interiors and electrical components. The company is 100% export oriented. The entire production is directed to the headquarters in Germany, which are then responsible for the distribution to the clients or other production sites. The "normal business process" starts when the company receives an order of production from the headquarters in Germany. If the order concerns a new product, then a prototyping is needed and a sample is produced. After a decision is reached, the production process can be launched. There are five main stakeholders for the company: 1) The headquarters in Germany, from where the company receives orders; 2) The production site in Poland to whom the company exports the semi-final products; 3) The production sites, from where the company receives semi-final products to finalize; 4) The suppliers of the raw materials and accessories; 5) Customs for the export.

As analyzing relations are the first requirement for identifying interoperability problems, a formal representation of the Company and the main relations that may be source of incompatibility are provided, using the model presented in previous sections. The instantiated concepts are represented by rectangles as shown in Fig 5.

The company is represented by the *Enterprise_GR* concept. As an instance of the *System* concept, it inherits all its properties and constituents. Hence it has its own structure and behaviour, represented respectively by *Structure_GR* and *Behaviour_GR*. The company produces wire harnesses for the cars and has two main objectives: continuous reduction of the costs of its production and to be the leader within its market. This is represented by the concept *Harness_production*, instance of the *Function* concept and two instances of the *objective* concept: *Market leader* and *Reduce costs*. As any multinational enterprise, the company evolves in its environment and has many partners. This is represented by *GR_env* concept, instance of *Environment*. Within this environment, the customs, the supplier of the accessories, the transporter, the headquarters, the supplier of raw material and the provider of all other services are found. This is respectively represented by the concepts: *Customs*, *Ac_supplier*, *Transporter*, *GR_group*, *Rm_supplier*, *Service_provider*. As business event we have added the specific concept *Order receipt* which influences the production of the enterprise. The instantiation of the integrated model provides an overview of the enterprise structure and the main relations that exist.

Based on that model, we can have a clear idea about the actual situation of the enterprise. Moreover, a more complete model integrating interoperability problems will allow us to point out exactly the element that is responsible of a potential problem (be it active or passive) and to fix it by proposing adequate solutions.

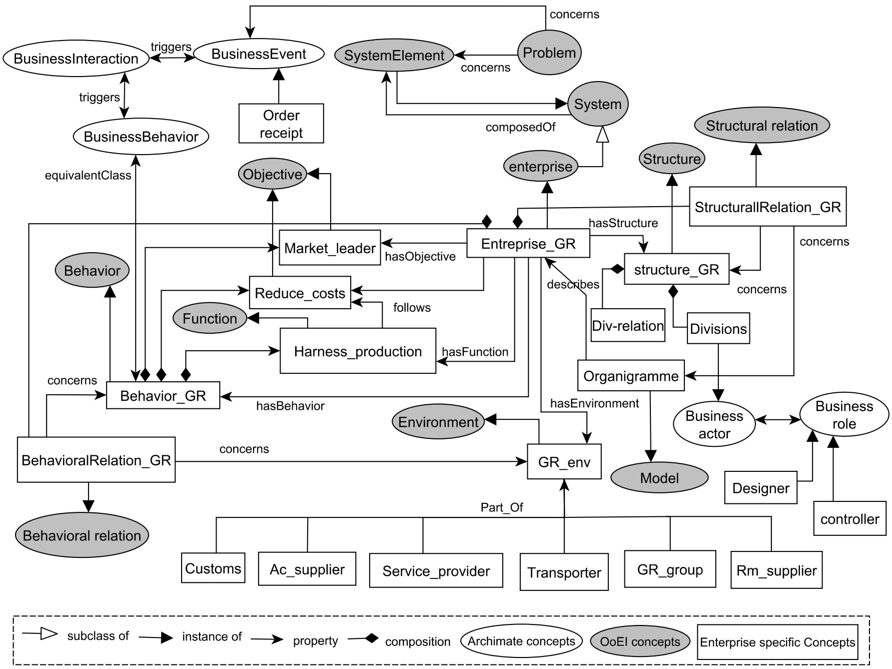


Fig. 5. Main concepts of the integrated model applied to the use case

6 Conclusion and Future work

In this paper, we have proposed an integrated model integrating interoperability and enterprise architecture concepts. At a first stage, the core concepts of the two domains were identified; The main relations between concepts were analysed in order to construct the integrated conceptual model. The evaluation stage was done through a case study in the automative industrial context. The resulting integrated model is based on the OoEI and the ArchiMate model at the three layers which are respectively Business layer, Application layer and Technology layer.

Future work is planned to assess the integration of interoperability concepts within ArchiMate. The idea is to assess the maturity of the meta-model and its future extension, as a generic model, to different modelling techniques and languages. We envisage also the deployment of the meta-model within EA framewoks such as TOGAF (the framework for ArchiMate) [2]. The idea is to support the dynamic definition of the enterprise principles including interoperability as part of the readiness factors for enterprise innovation capacities.

References

1. Chen, D., Doumeingts, G.: European initiatives to develop interoperability of enterprise applications-basic concepts, framework and roadmap. Annual Reviews in Control 27(2), 153–162 (2003)
2. Jonkers, H., Proper, H.A.: TOGAF and ArchiMate: A Future Together. White Paper W192. The Open Group (November 2009)
3. Lankhorst, M.M., Proper, H.A., Jonkers, H.: The Architecture of the ArchiMate Language. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) BPMDS 2009 and EMMSAD 2009. LNBIP, vol. 29, pp. 367–380. Springer, Heidelberg (2009)
4. Lankhorst, M., et al.: ArchiMate Language Primer. Telematica institute (2004)
5. Guedria, W.: A Contribution to Enterprise Interoperability Maturity Assessment. Ph.D. thesis, University of Bordeaux1, Bordeaux, France (2012)
6. Chen, D.: Enterprise Interoperability Framework. In: Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP) (January 2006)
7. Ford, T., et al.: Measuring System Interoperability: An i-Score Improvement. In: Proceedings of the 6th Annual Conference on Systems Engineering Research (2008)
8. Institute of Electrical and Electronics Engineers. IEEE standard computer dictionary: A compilation of ieee standard computer glossaries (1990)
9. Rosener, V., Latour, T.: A model-based ontology of the software interoperability problems: preliminary results. In: CAiSE 1998, vol. 3, pp. 241–252 (2004)
10. Naudet, Y., Latour, T., Haussmann, K., Abels, S., Hahn, A., Johannesonn, P.: Describing Interoperability: the OoI Ontology. In: Workshop on Enterprise Modelling and Ontologies for Interoperability (EMOI-INTEROP) (January 2006)
11. Rosener, V., Naudet, Y.: Latour. T. A model proposal of the interoperability problem. In: EMOI-INTEROP (2005)
12. Giachetti, R.E.: Design of Enterprise Systems: Theory, Architecture, and Methods, 1st edn. CRC Press, Inc., Boca Raton (2010)
13. Chen, D., Dassisti, M., Elvester, B.: Interop noe deliverable di.3: Enterprise interoperability framework and knowledge corpus - final report. Technical report, Interoperability Research for Networked Enterprises Applications and Software (INTEROP Network of Excellence), IST - Contract no.: IST-508 011 (2007)
14. IEEE Computer Society. IEEE Std 1471-2000: IEEE Recommended Practice for Architectural Description of Software- Intensive Systems (2000)
15. Jonkers, H., Lankhorst, M., Van Buuren, R., Bonsangue, M., Van Der Torre, L.: Concepts for modeling enterprise architectures. International Journal of Cooperative Information Systems 13, 257–287 (2004)
16. Lankhorst, M.M.: Enterprise Architecture at Work - Modelling, Communication and Analysis, 4th edn. The Enterprise Engineering Series. Springer (2013)
17. Engelsman, W., Jonkers, H.: Quartel. D. ArchiMate Extention for Modeling and Managing Motivation, Principles and Requirements in TOGAF. White paper. The Open Group (August. 2010)
18. Noy, N.F., Musen, M.A.: The prompt suite: interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies 59(6), 983–1024 (2003)