Using Ontologies to Integrate Multiple Enterprise Architecture Domains

Gonçalo Antunes[1], Artur Caetano[1,2],
Marzieh Bakhshandeh[1], Rudolf Mayer[3], José Borbinha[1,2]

[1] Instituto Superior Técnico, University of Lisbon,
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal,
[2] Information Systems Group, INESC-ID,
Rua Alves Redol 9, 1000-029 Lisboa, Portugal,
[3] SBA Research, Favoritenstraße 16, 1040 Wien, Austria
`{goncalo.antunes,artur.caetano}@ist.utl.pt`

Abstract. A goal of enterprise architecture is to align the business with the underlying support systems. An enterprise architecture description encompasses an heterogeneous spectrum of domains, such as business processes, application components, metrics, people and technological infrastructure. Architectural views express the domain elements and their relationships from the perspective of the system stakeholders. As a result, a view needs to be expressed using a domain language that addresses the specific concerns of its stakeholders. However, enterprise architecture description languages are often based on generic or broad meta-models that cross-cut distinct architectural domains. But describing each domain through a specialized language and then integrating it with the other domains raises challenges at the level of traceability and consistency. This paper proposes using ontologies to specify different enterprise architecture domains and to integrate and analyse these models. This goal is realized through a domain-independent language that is extended by domain-specific languages, each focussing on a set of specific domain concerns. The approach contributes to the alignment of the different domains while ensuring traceability between then concepts. The proposal is demonstrated through an evaluation scenario that uses ArchiMate as the domain-independent language extended with a set of domain-specific languages. The demonstration shows that the architecture domains can be integrated and analysed through the use of ontologies.

Key words: enterprise architecture, ArchiMate, ontology, OWL, alignment, separation of concerns

1 Introduction

Enterprise architecture (EA) is defined by Lankhorst as “a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise’s organizational structure, business processes, information systems, and infrastructure” whose models “focus on alleviating the infamous business–information technology alignment problem" [1]. Alignment results from applying models, methods, patterns and best practices to the specification and governance of the different domains of an organisation [2, 3, 4, 5]. Managing the dependencies between these concepts is fundamental for supporting the communication between the different stakeholders and to maintain the consistency at model and meta-model level [6, 7]. Moreover, EA governance requires the ability to analyse artefacts [1, 8] and is also required to assist business analytics [9].

Despite the efforts for developing comprehensive approaches to enterprise architecture, such as TOGAF [10], a "one language fits all" approach seems to be unable to address specific domains of an organization [11, 12]. The specific needs of different organizations place particular demands on the required EA artefacts. As such, the development of an architecture description language entails ensuring the consistency and traceability between the language concepts [13, 1]. On the other hand, creating a consistent and comprehensive architecture description language that deals with specific domains is a challenge despite existing situational method approaches to enterprise architecture [14].

The ISO 42010 standard suggests describing a system's architecture through multiple views to address the specific interests of the stakeholders on the system. An architecture description should therefore aggregate multiple views, materialized as a set of models, that are formulated according to viewpoints expressing the concerns of the stakeholders of the system-of-interest [15]. In this way, an architecture works as a communication agent between stakeholders, as each is presented with is own view over the system of interest. But creating different viewpoints may actually require using different meta-models, tools, and validation mechanisms. The integration and extension of models and underlying meta-models is common [16, 17], but raises challenges at the level of traceability and consistency because it is difficult to trace concepts between different languages and domains, a problem that is aggravated as the models evolve [18]. Moreover, the integration of different meta-models poses multiple challenges [19].

This paper is concerned with integrating multiple enterprise architecture domains while preserving traceability between the interrelated concepts. The goal is to integrate multiple enterprise architecture description languages as a means to assist alignment. Specifically, we investigate whether ontology technologies (OWL-DL in particular) can be used to specify, integrate and analyse multiple description languages.

Ontologies describe a domain model by associating meaning to its terms and relations. A more formal and widely used definition is that of Grüber who defines an ontology as a "formal specification of a conceptualisation" [20]. The importance of this technology is evidenced by the growing use of ontologies in a variety of application areas [21, 22] and, especially, by their role on the Semantic Web initiative [23, 24]. Ontology technologies are also used in the field of enterprise architecture to formalize organizational artefacts and to assist with model analysis [25, 26, 27, 28, 29, 30]. In fact, there is a wide body of knowledge that may improve the practice of EA, including ontology matching [31], and model extension and validation [32]. Ontologies facilitate the construction of complex models and can assist model analysis by depicting the consequences of a model. Formal ontology technologies also contribute to viewing and understanding the implicit consequences of explicit statements and can help ensuring that a model is consistent [33].

This paper posits that modelling the different enterprise architecture domains with a set of integrated description languages contributes to their alignment because consistency and traceability become ascertained. The approach entails using ontologies to represent and integrate the multiple architecture description languages and to analyse the resulting models. We argue for the integration of ontologies and associated technologies as mechanisms for developing consistent enterprise architecture models. The combination of formally specified models with their analysis via automatic mechanisms contributes to aligning the heterogeneous domains of an EA. One example is the impact analysis of changes from the business on the IT infrastructure and vice-versa. The main contribution of this paper is thus proposing an architecture based on the use of ontologies with the purpose of enhancing the extensibility with domain-specific aspects while enforcing consistency. We demonstrate the applicability of the proposal through the application of formal ontologies to model a set of different EA domains and through the consistent integration of these domains. In particular, we develop an ontology to specify the ArchiMate 2.0 meta-model and then create traceable maps to it from a set of domain-specific languages. We also describe an example that maps the sensor technology domain to ArchiMate in the context of a real-world scenario. This demonstration shows that the application of ontologies to enterprise architecture modelling effectively assists consistently aligning and analysing different domains.

The rest of this paper is organized as follows: section 2 describes a ontology-based proposal to integrate and analyse enterprise architecture models; section 3.1 describes the realization of the proposal; section 3 evaluates the solution using a scenario; finally, section 4 concludes the paper.

2 Enterprise Architecture Domain Integration

This paper proposes an ontology-based framework to formalize and integrate different domains of an enterprise architecture. The design of this artefact adheres to the following architectural principles:

- **Concern orientation**. The architecture represents the concepts that address an explicit set of concerns as a meta-model. The meta-model does not support any concepts that are not derived from the stakeholders' concerns.
- **Viewpoint-orientation**. The architecture supports defining views over subsets of its concepts. This facilitates communication because viewpoints act as a separation of concerns mechanism. Viewpoints facilitate addressing multiple concerns and can improve decision-making by isolating certain aspects of the architecture according to the needs of stakeholders.

- **Expressiveness**. The architecture represents a set of unambiguous domain concepts. This entails defining the minimum set of types and relationships to describe the domain.
- **Extensibility**. The architecture supports the integration of multiple domain-specific and domain-independent meta-models while minimizing coupling.
- **Modularity**. The architecture observes high-cohesion and low-coupling. Observing these qualities contributes to the expressiveness and extensibility of the architecture with the goal of minimizing the impact of adding new domain-specific concepts.

The ontology-based framework uses a meta-model to formalize the upper-level or core concepts. This meta-model is formalized as an upper-level ontology and is designated as *domain-independent ontology* (DIO). The design goal of the DIO is to represent the set of concepts pertaining to the central modelling domain. The DIO concepts are extended by defining a variable number of domain-specific meta-models, each depending on a particular system concern. Each domain-specific meta-model is formalized as a *domain-specific ontology* (DSO). Thus, a DSO represents a domain-specific language that addresses a particular set of concerns, and should also have the minimum set of concepts required to describe the domain. Therefore, separation of concerns, low-coupling and high-cohesion are the primary qualities that affect DIO and DSO design.

Ontology integration is required to link concepts from the DIO to each DSO. Integration combines different ontologies while ensuring consistency and maximum coverage of the domain being addressed. The simplest case is that of integrating the DSO concepts with the core concepts represented in the DIO. Cross-DSO integration occurs whenever more expressiveness is required to model a specific domain. Ontology integration makes use of model transformation, which involves defining a mapping strategy from a source to a target model [34, 35]. Figure 1 depicts the types of transformation mapping strategies between the DIO and the DSOs. Ideally, a map defines a one-to-one correspondence between each pair of concepts from a source and destination. But three types of mapping deficiencies may occur [36]: a source concept may map to more than one destination concept resulting in *overload*, a source concept may not be mappable to any destination resulting in *deficit*, or several source concepts may map to the same destination concept leading to *redundancy*. Deficiencies can be addressed by revising the DSO so that a one-to-one correspondence is achieved. If not possible, the deficiencies are addressed when querying or reasoning with the ontologies.

One of the goals of integrating the multiple ontologies is to analyse the resulting model instances. The analysis is performed by querying the models or by using a reasoner to infer the model's properties and relations. Four types of reasoning are possible with this architecture:*DIO reasoning* when inference is limited to the DIO concepts, *Single DSO reasoning*, when inference is limited to the concepts of a single DSO, *Cross-DSO reasoning*, when inference use concepts from more than one DSO, and *DIO-DSO reasoning*, when inference uses concepts from the DIO and one or more DSOs.

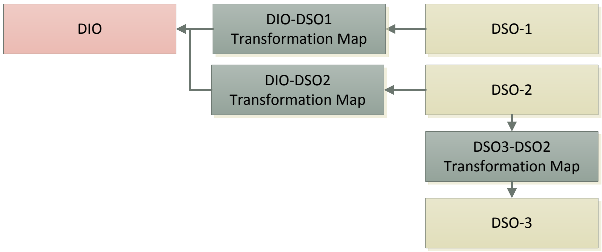


Fig. 1: Mappings between domain-independent and domain-specific ontologies

3 Application to Enterprise Architecture

This section describes the specification of a DIO, a DSO, the transformation maps and an application of the integrated meta-models. The evaluation scenario concerns a civil engineering safety authority that monitors structures such as hydroelectric power dams, reservoirs and bridges. Each structure has different sensors that measure physical phenomena and produce data that is analysed to assess its dynamics. The business process that deals with structure assessment includes activities to acquire data and to analyse data. Instance of this process are long-running as they may be active for decades, from the early construction phases until structure disposition. Part of this process is supported and automated by an information system that provides the following functions:

- Instrumentation: manages sensor installation, configuration and deployment.
- Transformation: manages the algorithms that transform sensor raw data into information.
- Observation: manages geodetic data, visual inspections data, and the data acquired from monitoring systems.
- Analysis: manages data analysis, visualization and reporting.
- Synchronization: synchronizes data between multiple geographic locations and logical systems.

The authority is required to acquire and preserve the monitoring data during the structure's life cycle. Therefore, capturing and preserving the information about the acquisition processes and supporting technological infrastructure is fundamental to attest the provenance and authenticity of the monitoring data. Moreover, historic data can be used to analyse and predict the behaviour of the structure. In this setting, enterprise architecture plays a valuable role to assist with the specification, evolution and the alignment of these processes with the supporting technology.

This scenario was modelled with the ArchiMate 2.0 enterprise architecture modelling language [37]. Although ArchiMate is able to specify the different domains of the scenario at a high-level of abstraction, it has not the expressiveness to model the domain-specific concerns pertaining to sensors and data acquisition. The first step was to create an ArchiMate model of the scenario. The model was produced with the Archi tool [1]). Figure 2 depicts an overview of the acquisition

---

[1] http://archi.cetis.ac.uk/

process and the services supporting it. Figure 3 depicts an overview of the application components and underlying technological infrastructure. The ArchiMate model was then exported from the Archi tool and automatically converted to OWL with a tool developed for that purpose.

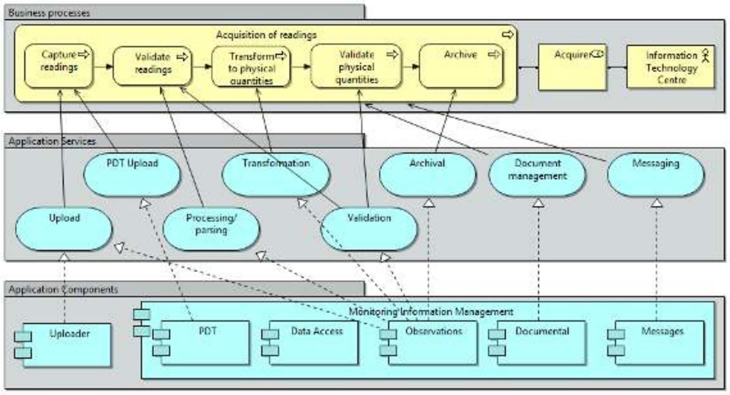


Fig. 2: Business processes and application infrastructure

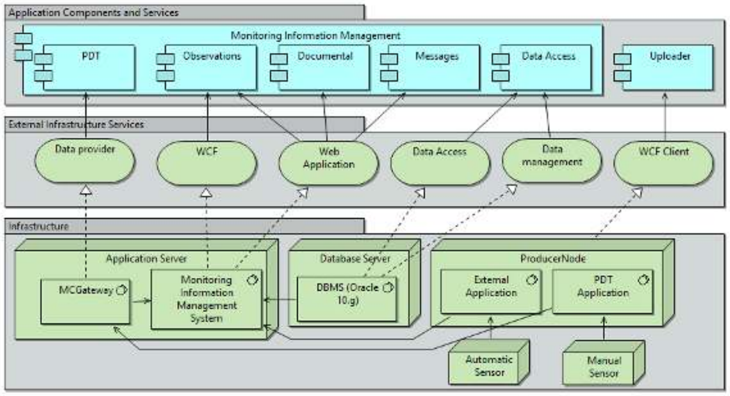


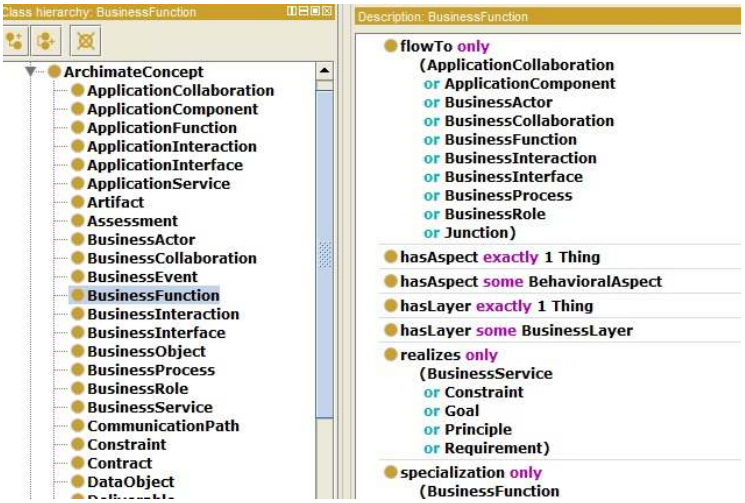
Fig. 3: Technological infrastructure

3.1 The ArchiMate Domain-Independent Ontology

ArchiMate describes the core concepts pertaining to enterprise architecture. The DIO is therefore a specification of the ArchiMate meta-model using OWL-DL. OWL-DL enables taking advantage of existing inference and querying mechanisms to analyse the models and assessing their consistency. The ArchiMate ontology was mainly developed according to the ontology engineering methodology defined by Horridge [38]. The steps include:

1. Identification of the concepts and concept hierarchy.
2. Identification of the disjoint concepts.
3. Modelling composition.
4. Addition of all the relationships between concepts.
5. Identification of definitions.
6. Addition of annotations.
7. Refinement of the ontology through various iterations of the above steps.

The resulting DIO represents the ArchiMate concepts as OWL *Classes* and relations as OWL *ObjectProperties*. Restrictions were added to the properties, such as *InverseObjectProperties* and *SuperObjectProperties*, so that ArchiMate's derived relationships are correctly inferred by the reasoner. Figure 4 depicts a partial OWL-DL specification of ArchiMate's *Business Function* as displayed in Protégé 4.3.



- ArchimateConcept
  - ApplicationCollaboration
  - ApplicationComponent
  - ApplicationFunction
  - ApplicationInteraction
  - ApplicationInterface
  - ApplicationService
  - Artifact
  - Assessment
  - BusinessActor
  - BusinessCollaboration
  - BusinessEvent
  - BusinessFunction
  - BusinessInteraction
  - BusinessInterface
  - BusinessObject
  - BusinessProcess
  - BusinessRole
  - BusinessService
  - CommunicationPath
  - Constraint
  - Contract
  - DataObject

```
flowTo only
    (ApplicationCollaboration
     or ApplicationComponent
     or BusinessActor
     or BusinessCollaboration
     or BusinessFunction
     or BusinessInteraction
     or BusinessInterface
     or BusinessProcess
     or BusinessRole
     or Junction)
hasAspect exactly 1 Thing
hasAspect some BehavioralAspect
hasLayer exactly 1 Thing
hasLayer some BusinessLayer
realizes only
    (BusinessService
     or Constraint
     or Goal
     or Principle
     or Requirement)
specialization only
    (BusinessFunction
```

Fig. 4: Partial OWL-DL specification of the ArchiMate Business Function

3.2 The Domain-Specific Ontology

In this scenario, the organizational stakeholders required modelling and analysing specific information about sensors. However, it is out of scope of the ArchiMate language providing the expressiveness to capture the specifics of this particular domain. Sensors measure values that are processed to perform structural analysis. There are sensors for making different types os measurements, which have specific transformation algorithms and calibration parameters. Some sensors are georeferenced and others capture data according to dynamic acquisition rates.

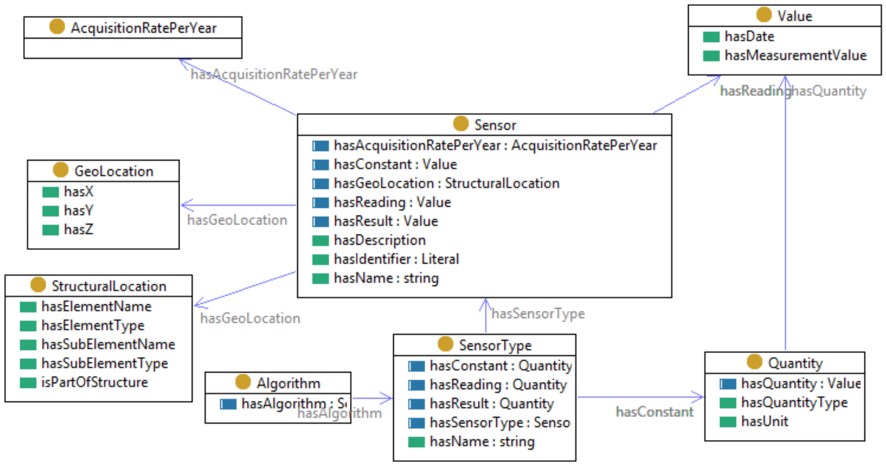


Fig. 5: Structure of the sensor DSO

As such, the particularities of this domain imply defining a domain-specific language. The organization evaluated different sensor modelling languages, such as SensorML [2] and TransducerML[3], but suggested a language based on SensorML to address its specific concerns. As a result, we developed the sensor DSO using the ontology engineering methodology described earlier. The core concepts of the sensor DSO are depicted in figure 5. The transformation map between the sensor DSO and the ArchiMate DIO contains the relations described on Table 1.

Table 1: Transformation map between the Sensor DSO and the ArchiMate DIO

Sensor DSO	**ArchiMate DIO**
Sensor	Node
GeoLocation	Location
StructuralLocation	Location
Algorithm	ApplicationComponent
Value	Data Object
AcquisitionRatePerYear	Data Object

3.3 Model Analysis

One of the stakeholder concerns relates to the technological infrastructure elements that support the acquisition process. This concern can be addressed through DIO reasoning, i.e. via the ArchiMate meta-model. Figure 6 depicts the

---

[2] http://www.ogcnetwork.net/SensorML

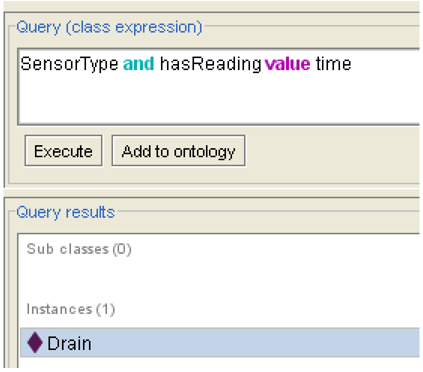
[3] http://www.ogcnetwork.net/infomodels/tml

question formalized as an OWL-DL query along with the *ObjectProperty* chains that identify the 19 instances that support the acquisition process. An example of intra-DSO reasoning is depicted in figure 7 that depicts the sensors that are able to make temporal readings. Finally, figure 8 shows the result of DIO-DSO reasoning where the integrated models are queried about which ArchiMate *Application Components* rely on the reading of the sensors of type *Drain*. The reasoner uses the mappings between the sensor DSO and the ArchiMate DIO to infer the reasoning chains and thus to answer the query.

Thing and hasLayer some TechnologyLayer and hasAspect some BehavioralAspect
or hasAspect some ActiveStructuralAspect and dependsDown value Acquisition_of_readings

◆ IIS
◆ Red_Hat_Linux
◆ Oracle_Client
◆ WCF_Client
◆ PDT_Application
◆ Data_management
◆ Producer
◆ MCGateway
◆ Data_provider
◆ Data_Access_
◆ Windows_Server_2008
◆ Application_Server
◆ Database_Server
◆ GestBarragens
◆ External_Application
◆ WCF
◆ .NET_Framework
◆ DBMS_-Oracle_10.g-
◆ Web_Application

Fig. 6: Intra-DIO query results



SensorType and hasReading value time

◆ Drain

Fig. 7: Intra-DSO query results

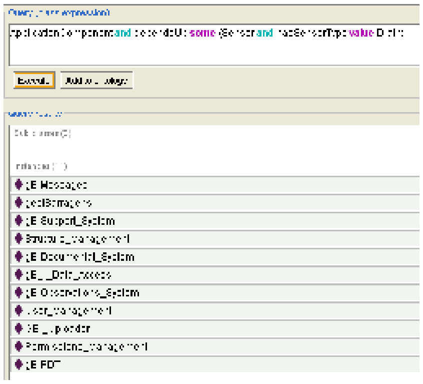


Fig. 8: Cross DIO-DSO query results

4 Conclusions

This paper proposes using ontologies to integrate different enterprise architecture domains and to analyse the resulting model instances. This goal is realized through the specification of a core domain-independent language that is extended by multiple domain-specific languages, each focussing on a set of specific concerns. The approach contributes to the alignment of the different domains while ensuring traceability, consistency and extensibility. As observed from the case study, ontologies can enhance the quality of meta-modelling due to their automated analysis capability that can be used to assess meta-model consistency as well as model conformance. Moreover, ontologies positively contribute to enterprise architecture alignment because multiple meta-models can be integrated and represented in such a way that its information can be traced and analysed while the reasoning consequences are exposed. The proposal was evaluated using ArchiMate as the DIO. To do that, we converted the ArchiMate meta-model to OWL-DL. A scenario was modelled using the ArchiMate DIO and its domain-specific aspects were modelled using a set of integrated DSOs. This paper partially described one of the DSOs, the sensor DSO, and exemplified different analysis types that can be accomplished using this approach. This demonstration shows that the application of ontologies to enterprise architecture modelling effectively assists aligning and analysis different domains.

Our current work focuses on extending the analysis capabilities to support the validation of models and the assessment of models and meta-models. We are also working on a set of automated and semi-automated extractor and process mining tools to instantiate the domain-specific and domain-independent ontologies with operational data to test the conformance of the "should-be" models towards the actual "as-is" models.

Acknowledgements

This project is partially supported by the European Commission under the 7th Framework Programme (FP7/2007-2013) under grant agreement 269940, TIMBUS project (`http://timbusproject.net`) and by COMET K1, FFG, Austrian Research Promotion Agency. Gonçalo Antunes is partially supported by FCT, Fundação para a Ciência e a Tecnologia, through project PEst-OE/EEI/LA0021/2011 and grant SFRH/BD/69121/2010.

References

1. Lankhorst, M.: Enterprise Architecture at Work: Modeling, Communication, and Analysis. 3 edn. Springer (2013)
2. Pereira, C.M., Sousa, P.M.A.: Enterprise architecture: business and it alignment. In Haddad, H., Liebrock, L.M., Omicini, A., Wainwright, R.L., eds.: SAC, ACM (2005) 1344–1345

3. Winter, R., Fischer, R.: Essential layers, artifacts, and dependencies of enterprise architecture. In: Enterprise Distributed Object Computing Conference Workshops, 2006. EDOCW'06. 10th IEEE International, IEEE (2006) 30–30
4. Wegmann, A., Balabko, P., Lê, L.S., Regev, G., Rychkova, I.: A method and tool for business-it alignment in enterprise architecture. In: Proceedings of the CAiSE. Volume 5., Citeseer (2005) 113–118
5. Caetano, A., Silva, A.R., Tribolet, J.M.: Business process decomposition - an approach based on the principle of separation of concerns. Enterprise Modelling and Information Systems Architectures **5**(1) (2010) 44–57
6. Galster, M.: Dependencies, traceability and consistency in software architecture: towards a view-based perspective. In: ECSA '11 Proceedings of the 5th European Conference on Software Architecture: Companion Volume. (2011)
7. Romero, J., Jaen, J.I., Vallencillo, A.: Realizing correspondences in multi-viewpoint specifications. In: Proceedings of the 2009 IEEE International Enterprise Distributed Object Computing Conference. (2009)
8. Buschle, M., Ullberg, J., Franke, U., Lagerstrom, R., Sommestad, T.: A tool for enterprise architecture analysis using the prm formalism. In: CAiSE Forum 2010, Hammamet, Tunisia, June 7-9, 2010, Selected Extended Papers. (2010)
9. Binz, T., Leymann, F., Nowak, A., Schumm, D.: Improving the manageability of enterprise topologies through segmentation, graph transformation, and analysis strategies. In: 2012 16th IEEE International Enterprise Distributed Object Computing Conference (EDOC). (2012)
10. The Open Group: TOGAF 9 - the open group architecture framework version 9 (2009)
11. Saat, J., Franke, U., Lagerstrom, R., Ekstedt, M.: Enterprise architecture meta models for it/business alignment situations. In: 2010 14th IEEE International Enterprise Distributed Object Computing Conference. (2010)
12. Buckl, S., Matthes, F., Schweda, C.M.: Conceptual models for cross-cutting aspects in enterprise architecture modeling. In: 2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops. (2010)
13. Braun, C., Winter, R.: A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. Enterprise Modelling and Information Systems Architectures (2005) 64–79
14. Buckl, S., Schweda, C.M., Matthes, F.: A design theory nexus for situational enterprise architecture management. In: 2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops. (2010)
15. ISO: ISO/IEC/IEEE 42010:2011 - systems and software engineering - architecture description (2011)
16. Burmester, S., Giese, H., Niere, J., Tichy, M., Wadsack, J.P., Wagner, R., Wendehals, L., Zundorf, A.: Tool integration at the meta-model level: the fujaba approach. Int J Softw Tools Technol Transfer **6** (2004) 203–218
17. Kapsammer, E., Reiter, T., Schwinger, W.: Model-based tool integration - state of the art and future perspectives. In: Proceedings of the 3rd International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2006). (2006)
18. France, R., Rumpe, B.: Model-driven development of complex software: A research roadmap. In: Future of Software Engineering, 2007. FOSE '07. (2007)
19. Almeida, J.P.A., Iacob, M.E., van Eck, P.: Requirements traceability in model-driven development: Applying model and transformation conformance. Inf Syst Front **9** (2007) 327–342

20. Grüber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**(2) (1993) 199–220
21. Happel, H.J., Seedorf, S.: Applications of ontologies in software engineering. In: In 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006), held at the 5th International Semantic Web Conference (ISWC 2006). (2006)
22. Burger, T., Simperl, E.: Measuring the benefits of ontologies. In: On the Move to Meaningful Internet Systems: OTM 2008 Workshops. (2008)
23. Shadbolt, N., Hall, W., Berners-Lee, T.: The semantic web revisited. IEEE Intelligent Systems Journal **4** (2006) 96–101
24. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American (May 2001) 29–37
25. Uschold, M., Uschold, M., King, M., King, M., House, S.B.R., Moralee, S., Moralee, S., Zorgios, Y., Zorgios, Y.: The enterprise ontology. The Knowledge Engineering Review **13** (1995) 31–89
26. Geerts, G.L., McCarthy, W.E.: An ontological analysis of the economic primitives of the extended-rea enterprise information architecture. International Journal of Accounting Information Systems **3**(1) (2002) 1 – 16
27. Kang, D. ande Lee, J., Choi, S., Kwangsoo, K.: An ontology-based enterprise architecture. Expert Systems with Applications **37** (2010) 1456–1464
28. Wagner, G.: Ontologies and rules for enterprise modeling and simulation. In: 2011 15th IEEE International Enterprise Distributed Object Computing Conference Workshops. (2011)
29. Azevedo, C.L.B., Almeida, J.P.A., van Sinderen, M., Quartel, D., Guizzardi, G.: An ontology-based semantics for the motivation extension to archimate. In: 2011 15th IEEE International Enterprise Distributed Object Computing Conference. (2011)
30. Almeida, J.P.A., Guizzardi, G.: An ontological analysis of the notion of community in the rm-odp enterprise language. Computer Standards & Interfaces **35**(3) (2013) 257 – 268
31. Schvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal on Data Semantics IV, Lecture Notes in Computer Science **3730** (2005) 146–171
32. Santos Jr., P.S., Almeida, J.P.A., Guizzardi, G.: An ontology-based semantic foundation for organizational structure modeling in the aris method. In: 2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops. (2010)
33. Pidcock, W.: What are the differences between a vocabulary, a taxonomy, a thesaurus, an ontology, and a meta-model? available online `http://infogrid.org/wiki/Reference/PidcockArticle` (accessed April 2013) (2002)
34. Guizzardi, G.: Ontological foundations for structural conceptual models. PhD thesis, University of Twente, Enschede, The Netherlands (2005)
35. Rosemann, M., Green, P., Indulska, M.: A reference methodology for conducting ontological analyses. In: Proceedings of the 23rd International Conference on Conceptual Modelling (ER 2004). (2004)
36. Bunge, M.A.: Treatise on Basic Philosophy Volume 3: Ontology I - The Furniture of the World. Kluwer Academic Publishers (1977)
37. The Open Group: ArchiMate 2.0 specification (2012)
38. Horridge, M., Knublauch, H., Rector, A., Stevens, R., Wroe, C.: A practical guide to building owl ontologies using the protégé-owl plugin and co-ode tools edition 1.2. The University Of Manchester (2009)