Semantic enterprise architecture management

Willy Chen, Claudia Hess, Melanie Langermeier, Janno Stülpnagel, Philipp Diefenthaler

Angaben zur Veröffentlichung / Publication details:

Chen, Willy, Claudia Hess, Melanie Langermeier, Janno Stülpnagel, and Philipp Diefenthaler. 2013. "Semantic enterprise architecture management." In *15th International Conference on Enterprise Information Systems (ICEIS), July 4-7, 2013, in Angers, France*, edited by Slimane Hammoudi, Leszek Maciaszek, José Cordeiro, and Jan Dietz, 318–25. Setúbal: SciTePress. https://doi.org/https://doi.org/10.5220/0004445003180325.

Semantic Enterprise Architecture Management

Willy Chen[1], Claudia Hess[1], Melanie Langermeier[2], Janno Stuelpnagel[1] and Philipp Diefenthaler[1,2]
[1]*Softplant GmbH, Munich, Germany*
[2]*Department of Computer Science, University of Augsburg, Augsburg, Germany*

Keywords: Semantic Web Technology, Enterprise Architecture Management, Ontology.

Abstract: Enterprise architecture management (EAM) provides information about an enterprise and its information systems landscape for thorough analysis. Semantic technologies allow for integrating data sets from various relevant information sources within enterprises in a so-called EA repository, which serves as a comprehensive formal information base for EAM. We call this approach Semantic Enterprise Architecture Management, SEAM for short. We show how SEAM uses semantic technologies to bridge the heterogeneity between various data sets relevant for EAM, such as The Open Group Architecture Framework (TOGAF) models, Business Process Modeling Notation (BPMN) business process models and Service-oriented Architecture (SOA) models. Query answering and reasoning enable detailed analysis and consistency checks on the SEAM repository in order to discover complex dependencies and implicit knowledge.

1 INTRODUCTION

Enterprise architectures (EA) are formal models of core artifacts of an enterprise and their dependencies (Aier et al., 2008). Within enterprise architecture management (EAM) these models are utilized to design and align IT landscapes of enterprises based on strategic, organizational and technological viewpoints. Therefore, an EA aggregates information about business, especially about the business strategy and business processes, about the application landscape, i.e. the information systems used and planned with their components and interfaces, and last but not least, about the infrastructure and the technology used. The EA provides different types of analysis on all this information for the various stakeholders and their particular concerns.

During the last years, many so called EA frameworks have been developed to facilitate the introduction and the further development of an EAM approach in an organization. As recent analyses have shown, today an EAM approach "is something that is unique to every organization, and will be formed from a combination of frameworks, industry models, and methodologies" (Blowers, 2012). EA frameworks therefore have to be tailored to be used in a particular organization. (Blowers, 2012) found that the majority of the interviewed organizations (66%) are using a customized framework. One-third of the interviewees even developed their framework on the basis of two or more frameworks, resulting in so-called hybrid frameworks.

Customizing a framework means that terminology as well as process and architectural models suggested by the framework are adapted to an organization's needs (Open Group, 2009). The terminology used in the framework description can be replaced where necessary with the organization-specific terms. The Open Group (Open Group, 2009) recommends to link the EAM approach closely to other EAM related activities such as project and service portfolio management processes, project lifecycle, operations handover processes, operational management processes and procurement processes. All these activities create information in their own, specialized information bases (e.g. service architectures, enterprise service bus, configuration management databases). These numerous additional sources could provide valuable input to an EA model. However, when integrating different data sources we have to deal with structural, syntactic and semantic conflicts. In large corporate groups and in cases of mergers and acquisitions it is even more challenging to build a single corporate EA based on multiple EAM approaches and numerous EAM related activities.

According to recent research the documentation of

Chen W., Hess C., Langermeier M., Stuelpnagel J. and Diefenthaler P..
Semantic Enterprise Architecture Management.
DOI: 10.5220/0004445003180325
In *Proceedings of the 15th International Conference on Enterprise Information Systems* (ICEIS-2013), pages 318-325
ISBN: 978-989-8565-61-7

the EA is still mostly done manually today (Buschle et al., 2012; Farwick et al., 2011). This is not only time consuming but keeping the acquired data up-to-date is also difficult. We therefore aim to automate the documentation process of an EA by providing efficient and intelligent means to reuse and interconnect existing data sets. This should not only reduce the amount of effort for creating an initial EA model but also allow to leverage related management processes to keep the information within an EA up-to-date.

To this aim, we analyze different data sources relevant to an EA and their interdependencies. We show, how semantic technologies as being developed for knowledge representation and reasoning on the Web allow for building an EA repository based on open standards and well-studied modeling languages and formalisms. We provide a mapping between several of the data sources identified as relevant for EAM. We then use this mapping to integrate the data sets and show possible analysis on the linked data sets.

The paper is structured as follows. Section 2 briefly gives an overview on EA frameworks and models and discusses possible conflicts that arise when interconnecting different data sources relevant to EA. In Section 3 we present our approach to create a EA repository based on semantic technologies by formalizing and integrating the data sets from various sources. We show the applicability of the approach by integrating EA documentation according to the The Open Group Architecture Framework (TOGAF) content metamodel, related process models in Business Process Model and Notation (BPMN) and information on services according to the Service-Oriented Architecture Ontology. We discuss the results in Section 4 and analyze related works in Section 5. We conclude the paper by discussing further application scenarios based on SEAM repositories and pointing out the future direction of our research.

2 ENTERPRISE ARCHITECTURE REPOSITORIES

2.1 EA Frameworks - A Guideline to EAM Approaches

There are currently more than 50 known frameworks specified by standards setting organizations, companies, consultancies, governmental organizations and military agencies (Matthes, 2011). The Zachman framework (Zachman, 1987) is one of the early approaches. Its development was inspired by the process of constructing a building and the deliverables produced in this process. It provides an analogous set of architectural representations produced for the development of an information system. TOGAF (Open Group, 2009), The Open Group Enterprise Architecture Framework, is an industry standard architecture framework and is nowadays widely accepted. The core of TOGAF is its generic enterprise architecture method, the Architecture Development Method (ADM), which specifies the different steps required for building up and using an EA. The core content metamodel, guidelines, recommended standards and compliant products complete the framework.

The keystone of an EA framework is the architectural model that it provides to represent an organization's enterprise architecture. As EA frameworks differ in purpose and scope, they all come up with their own architectural model. TOGAF, for example, defines the core content metamodel and several extensions. The core content metamodel encompasses a minimum set of artifacts for an EAM endeavor. It partitions the artifacts in five units, first architecture principles, vision and requirements, then the three architecture layers defined by TOGAF (i.e. the business, the information systems and the technology architecture) and last the architecture realization. On this basis, analyses such as 'show me all applications that are implemented on deprecated platforms' can be made.

2.2 Information Sources relevant to EAM

Within large enterprises and complex information system landscapes there are numerous structured and unstructured data sources that hold relevant information for a complete EA documentation. Structured sources include but are not limited to[1]: business process models, software documentation (e.g. UML documentation or Web service documentation), configuration management databases (i.e. configuration items in the IT infrastructure following the ITIL standard), enterprise resource planning (organizational data such as departments and roles, project management and portfolio data), and middleware systems (e.g. enterprise service bus (Buschle et al., 2012)).

Linking those data into an EA repository provides a powerful groundwork for complementing and refining any EA documentation. Today, this information is mostly added manually to the EA repository, see e.g. (Buschle et al., 2012; Farwick et al., 2011). This is very time-consuming and data can hardly be kept

[1]For creating a more comprehensive listing we suggest to utilize electronic product catalogs such as eCl@ss http://www.eclasscontent.com/

up-to-date constantly. For automating this process appropriate interfaces for the data exchange have to be provided and a mapping from the source data to the EA model has to be defined. Conflicts w.r.t. syntactic, structures and semantics have to be faced.

Syntactic conflicts denote the problem that EAM related data is represented in various languages specialized for the intended use. Within domains, this problem is addressed by standards. Examples are UML for software documentation, BPMN as a standard by the OMG for business process models, ArchiMate for EA documentation or the CMDB Federation (CMDBf) specification for representing items from configuration management databases. Some of these standards are well-accepted and widely used while other standards are only emerging, i.e. most data sources are still provided in their original formats. In order to integrate the various EAM relevant data sources in the EA repository, we have to deal with all these different standards and formats.

Structural conflicts arise when semantically equivalent information is represented in a structurally different way. Attributes of the same entities can have different names or different data types. Such heterogeneity will occur, for example, if for an application component the date of its go-live is indicated. One source might call this attribute initial live date (e.g. according to the TOGAF content metamodel for physical application components), the other source simply go-live date. A more complex structural conflict occurs when a source uses one object to represent certain information, and another source uses two or more objects. This is the case, for example, if the TOGAF content metamodel is used with and without a particular extension. A source using only the core artifacts for the application architecture contains the entity application component while a source using the core with the infrastructure consolidation extension extends the application component to a logical application component and a physical application component.

Semantic conflicts occur where information is assumed to be equivalent, for instance, due to an identical name although being different. An often cited example is here an attribute indicating a price, e.g. the fixed and variable costs of a service, which may be given in different currencies. More complex is the following example. An EA repository might integrate information on organization units, such as the headcount. There is a semantic heterogeneity if a first source counts only the full-time employees (cf. TOGAF content metamodel) while a second source also considers freelancers. Semantic heterogeneities might not only occur with respect to attributes but also with entities. Integrating information from two concurrent EAM approaches in different departments, one might consider as application only the business applications while the other one lists business and infrastructure applications (e.g. as distinguished by the TOGAF Technical Reference Model). The result of a query for all infrastructure applications over the integrated data will be misleading.

3 BUILDING A SEAM REPOSITORY

We call the approach for representing and analyzing enterprise architecture models with semantic Web technologies Semantic Enterprise Architecture Management – SEAM for short. With SEAM we formalize the knowledge about interdependencies of relevant data sets, including EA frameworks. Therewith equalities and inconsistencies are identified. SEAM links existing data within an enterprise instead of duplicating it in yet another tool. It facilitates the documentation of the architecture as it supports the mapping and integration of relevant data sets and deriving the required views for EAM. Semantic technologies, especially ontologies, provide a mean for bridging the structural and semantic heterogeneities between the different data sets. The resulting models for enterprise architecture do not only serve as a static documentation but can also be utilized for architecture analysis and the provisioning of structured data for succeeding processes such as project initiations.

To resolve structural and semantic conflicts, we use the metamodel of one of the data sources as global, neutral model. All other data sources are then mapped to this neutral ontology. Each data source keeps its own data model adapted to its particular application context instead of enforcing a single model to all sources. Through a mapping of the elements and attributes of the local sources to the neutral ontology, implicit assumptions of the local models are made explicit and can be used during query answering. Having mapped local entities and attributes to their counterparts in the neutral ontology and having modeled implicit assumptions, inference mechanisms can be used for consistency checks.

For providing a SEAM repository based on different existing data sources within an enterprise we propose the following approach:

- **Formalize and Analyze Data Sources:** formalize the data sources by identifying and defining the underlying metamodels and analyze their overlap

- **Expose Data Sets:** expose relevant data sets from the underlying data sources as an abstracted view using a simplified metamodel
- **Linking the Data Sets:** define a set of mappings between the data sets within the repository that can be computed automatically
- **Refinement:** manually refine mappings including individual equality assertions

The approach is depicted in Figure 1. In the following sections we describe the technical background of the used representation formalism and give details and examples on every step.

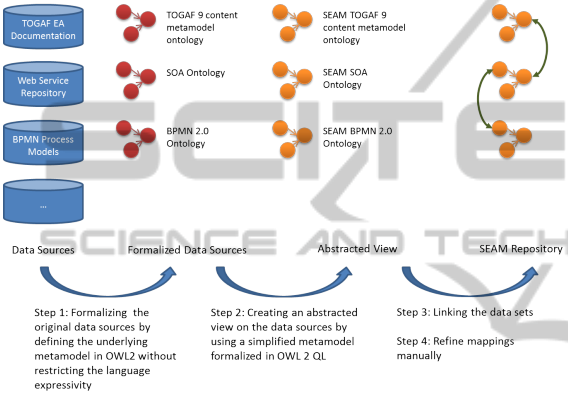

Figure 1: Approach for creating a SEAM repository.

3.1 Technical Foundations

We propose to use ontologies for representing the data sources. While the T-Box covers the metamodel of the data sources, i.e. their structure and semantics, we use the A-Box for representing the actual model instances. We choose to start with the QL language profile from the OWL 2 standard. An OWL 2 profile is a "trimmed down version of OWL 2 that trades expressive power for efficiency of reasoning" (Motik et al., 2012). OWL 2 QL in special is made for "application that use very large volumes of instance data, and where query answering is the most important reasoning task" (Motik et al., 2012). Technically, the query answering can be implemented by rewriting ontology queries, e.g. in SPARQL to SQL queries that can be posed to standard relational database systems. Thus, a SEAM repository can be built on top of mature relational database management systems, which are widely adopted and operated in enterprises.

Clearly, the expressiveness of OWL 2 QL is limited. Nevertheless, it still covers most of the formalisms required for representing mappings between different ontologies. Given two ontologies O and P, class expressions CE, object property expressions OPE and data property assertions DPE we can define a set of mappings M based on:

- EquivalentClasses(CE_O, CE_P)
- EquivalentObjectProperties(OPE_O, OPE_P)
- EquivalentDataProperties(DPE_O, DPE_P)

Additionally individual equalities using the construct SameIndividual(a_1 ... a_n) are defined. Although this construct is not part of the OWL 2 QL language profile, such axioms can be computed in LOGSPCAE w.r.t. the size of data by computing the reflexive-symmetric-transitive closure of the equality relation and using the resulting relation within query answering (Motik et al., 2012).

3.2 Formalize and Analyze Data Sources

As mentioned before, information sources within enterprise as well as the actual EA models are relevant data sets for an EAM initiative. To guarantee for most exact semantic descriptions, we do not need to restrict the language expressiveness of OWL 2 for formalizing the relevant data sources in this first step. Within this paper we consider following existing ontologies, which represent the T-Box of specific data sources relevant for an EA repository:

- TOGAF 9 Content Metamodel Ontology. Gerber et al. (Gerber et al., 2010) have developed an OWL representation, which should represent the information from the TOGAF 9 content metamodel in a more precise and unambiguous manner.
- BPMN 2.0 Ontology. Natschlaeger (Natschlaeger, 2011) provides an ontology for the process modeling standard BPMN. It formalizes the entire BPMN 2.0 specification as well as further textual annotations from the specification.
- SOA Ontology. The service-oriented architecture ontology (Open Group, 2010) describes both business and technical aspects of a service-oriented architecture.

We assume that each data source provides data sets represented within the A-Box of each of the ontologies. Moreover, any additional data source within an enterprise may be formalized using the same approach, either by reusing existing ontologies or by modeling special ontologies. This formalization serves as a starting point for analyzing connections between the different data sources. Possible structural and semantic conflicts are identified in this step.

We did this for the above presented ontologies. All have an overlap within the TOGAF 9 business architecture. In this architecture TOGAF deals with business processes and business services and sets them into context to other organizational elements. Therefore, the TOGAF Ontology will be our neutral ontology whereas the BPMN 2.0 and SOA ontologies are the local ones, which will be mapped into the neutral one.

Based on the formal ontology language following types of mapping can be identified:

- Direct Equivalence: given two class expressions CE_1 and CE_2, we can state $CE_1 \equiv CE_2$ i.e. $EquivalentClasses(CE_1, CE_2)$
- Subset Equivalence: given two class expressions CE_1 and CE_2, we can state $CE_1 \subseteq CE_2$ i.e. $SubClassOf(CE_1, CE_2)$

We identified such equivalences between the selected ontologies. For example, the class Process in TOGAF corresponds to the class Process in the SOA ontology. Here, we have a direct equivalence. Equivalence between object properties and data properties can be defined analogously using object property expressions and data property expressions. While direct equivalences can be applied automatically, subset equivalences only provide a support to find possible mappings in a manual refinement step.

3.3 Expose Data Sets

As the different data sources cover very specific details about a single aspect of an EA, a full integration of all ontologies into a single ontology covering the complete semantic and representation is a highly complex task. To reduce the complexity we limit the used OWL 2 fragment as explained in chapter 3.1. Complex OWL constructs can be materialized in OWL 2 QL by applying forward chaining. Additionally highly specific model elements may be neglected. This is necessary since we want to enable efficient querying in the combined ontology although we have to deal with a very high amount of data.

We have done this task with the three ontologies introduced before. The following examples illustrate these results.

SEAM TOGAF 9 Content Metamodel Ontology. Figure 2 shows a TOGAF model excerpt, which covers organizational and functional aspects of a business and connects them with business processes. Furthermore it links business services with the supported processes. We use a simple UML class notation to visualize this ontology excerpt. UML stereotypes denote the TOGAF content metamodel classes.

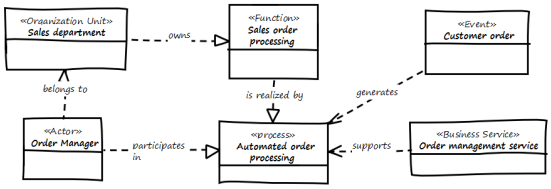

Figure 2: Excerpt of SEAM TOGAF 9 content metamodel Ontology.

Figure 2 can be represented in OWL 2 functional-style syntax (FSS). The following excerpt includes the relevant part of T- and A-Box, but omits the definition of object properties and object property assertions.

```
Class(T:OrganizationUnit)  Class(T:Function)
Class(T:Process)   Class(T:Actor)
Class(T:BusinessService) Class(T:Event)
Individual(T:''Sales department'')
ClassAssertion(T:OrganizationUnit T:''Sales
department'')
Individual(T:''Sales order processing'')
ClassAssertion(T:Function T:''Sales order
processing'')
Individual(T:''Order manager'')
ClassAssertion(T:Actor T:''Order manager'')
Individual(T:''Automated order processing'')
ClassAssertion(T:Process T:''Automated order
processing'')
Individual(T:''Order management service'')
ClassAssertion(T:BusinessService T:''Order
management service'')
Individual(T:''Customer order'')
ClassAssertion(T:Event T:''Customer order'')
```

SEAM BPMN 2.0 Ontology. Using BPMN the business process "automated order processing" is modeled in more details (see Figure 3). As typical for BPMN models, the lane holds a business role (Order Manager) that is the performer of the tasks associated with the lane. However, the usage of lane elements within BPMN is not defined.

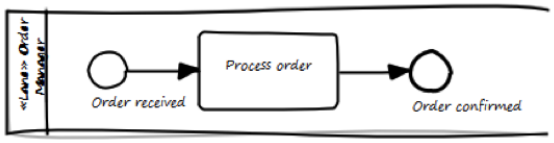

Figure 3: Excerpt of SEAM BPMN 2.0 Ontology.

The representation in OWL 2 FSS omitting object properties would be (note that the Events and the FlowNodes are connected via the SequenceFlow class):

```
Class(B:Task) Class(B:FlowNode)
Class(B:SequenceFlow)
Class(B: Event) SubClassOf(B:Event B:FlowNode)
Class(B:StartEvent) Class(B:CatchEvent)
SubClassOf(B:CatchEvent B:Event)
SubClassOf(B:StartEvent B:CatchEvent)
```

```
Class(B:EndEvent) Class(B:ThrowEvent)
SubClassOf(B:ThrowEvent B:Event)
SubClassOf(B:EndEvent B:ThrowEvent)
Individual(B:''Process order'')
ClassAssertion(B:Task B:''Process order'')
Individual(B:''Order received'')
ClassAssertion(B:StartEvent
   B:''Order received'')
Individual(B:''Order confirmed'')
ClassAssertion(B:EndEvent
   B:''Order confirmed'')
```

SEAM SOA Ontology. The SOA ontology allows for modeling the example from an additional point of view. Figure 4 represents an excerpt of this SEAM SOA ontology. A process is thereby drilled down into tasks. Tasks use services to fulfill a service contract. Note that the service contract is part of the SOA ontology but not of this example.

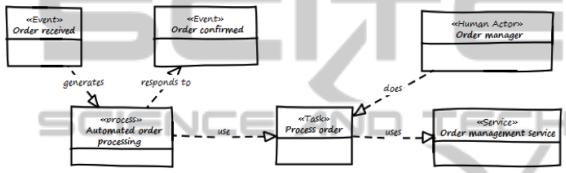

Figure 4: Excerpt of SEAM SOA Ontology.

An equivalent OWL 2 FSS representation, again omitting the object properties, would be:

```
Class(S:Event) Class(S:Process)
Class(S:Task) Class(S:Service)
Class(S:HumanActor)
Individual(S:''Process order'')
ClassAssertion(S:Task S:''Process order'')
Individual(S:''Order received'')
ClassAssertion(S:Event S:''Order received'')
Individual(S:''Order confirmed'')
ClassAssertion(S:Event S:''Order confirmed'')
Individual(S:''Automated order processing'')
ClassAssertion(S:Process S:''Automated order
processing'')
Individual(S:''Order management service'')
ClassAssertion(S:Service S:''Order management
service'')
Individual(S:''Order manager'')
ClassAssertion(S:HumanActor
   S:''Order manager'')
```

The examples above only show a small range of possible contents. In practice the models are far more detailed and complex. If required, data aggregation operations can be applied within this step to reduce the complexity of the next mapping step.

3.4 Linking the Data Sets

The equivalences that were identified in step 1 (Formalizing and Analyzing Data Sources, see section 3.2) are now formalized. In our example, we can now add following OWL mappings:

```
EquivalentClasses(T:Event, B:Event)
EquivalentClasses(T:Event, S:Event)
EquivalentClasses(T:BusinessService, S:Service)
EquivalentClasses(T:Process, S:Process)
EquivalentClasses(T:Actor, S:HumanActor)
```

By applying the mappings on concrete data sets within the SEAM repository, equality assertions from the different data sets can be derived. In our example this is quite obvious by matching the labels. However, more intelligent means, e.g. partial matchmaking (Stuckenschmidt, 2007), are required in real world scenarios. Also, it is crucial in this step to consider possible semantic conflicts. For our example we can derive:

```
SameIndividual(B:''Order received''
   S:''Order received'')
SameIndividual(B:''Order confirmed''
   S:''Order confirmed'')
SameIndividual(T:''Automated order processing''
   S:''Automated order processing'')
SameIndividual(T:''Order management service''
   S:''Order management service'')
SameIndividual(T:''Order manager''
   S:''Order manager'')
```

3.5 Refinement

As mentioned in section 3.2 there are subset mappings that only provide hints on possible equivalences. In a final step this knowledge is utilized to provide suggestions to the user to refine the SEAM repository. The user can then add further mappings with the *SameIndividual* construct.

For the considered ontologies following subset mappings can be analyzed in more detail:

$$\begin{aligned}&\text{T:BusinessService} \cup \text{T:PlatformService} \cup \\ &\quad \text{T:InformationSystemService} \equiv \\ &\quad\quad \text{S:Service} \cup \text{S:ServiceComposition} \\ &\text{T:Actor} \cup \text{T:Role} \cup \\ &\quad \text{T:LogicalApplicationComponent} \cup \\ &\quad\quad \text{T:LogicalTechnologyComponent} \subseteq \text{S:Element}\end{aligned}$$

The latter mapping for example states that instances of the class Element in the SOA ontology may show equivalences within individuals of the TOGAF classes Actor, Role, LogicalApplicationComponent and LogicalTechnologyComponent.

During the refinement, the SEAM repository can be constantly validated using reasoning and query answering in order to detect inconsistencies within the formal model.

3.6 Architecture Analysis

The resulting SEAM repository can now be utilized for architecture analysis. For example, having identified equality between a process within TOGAF and a business process model in BPMN would allow an enterprise architect to directly navigate from an EA model to the actual process definition. Additionally, such an integrated approach provides further means for consistency checks. The consistency of the EA model cannot only be verified based on the metamodel but also w.r.t. to the data quality. Documented information systems services within the EA can be compared with other sources for service definition.

Clearly, enterprise architecture aims at a holistic model of an enterprise whereas the other data sets contain detailed information about specific aspects thereof. As these data is not manually documented within an EAM initiative but automatically linked from the actual sources, they can serve as additional information that can be considered for decisions within the EA. Also data redundancies can be reduced if required elements are looked up within these data sets and referenced using natural or unique identifiers (e.g. department names, cost centers, project numbers, Web service name, host name, business process name). Detailed and up-to-date information can be accessed later on using these identifiers instead of working on an outdated copy.

4 DISCUSSION

In the previous section we presented a methodology for building a SEAM repository. We evaluated the approach by creating a EA repository based on business process descriptions, service specifications of a service-oriented architecture and TOGAF-based EAM documentations. For each of the data sources, we utilized an ontology-based metamodel to formalize them and to analyze their overlap. Focusing on efficient query answering and analysis in the SEAM repository, we derived simplified data sets based on a formal language with reduced language complexity, namely OWL 2 QL. Step by step we added mappings between the data sets to capture all relations and allow for analyses within the linked data. Parts of the resulting SEAM repository has already been shown as a running example in the last section (the authors may be contacted to obtain the ontologies).

We have shown that semantic technologies provide adequate means to overcome syntactic and structural conflicts. Data sources relevant for an EA can be transformed into data sets that are linked via the explicit definition of equivalent classes, properties and individuals. Still, semantic conflicts (cf. section 2.2) have to be considered separately. When mapping the data sets, we have to ensure that the semantics of the mapped classes and individuals correspond to each other. Otherwise, there is still the possibility to alter the step of exposing the data sets.

The SEAM approach allows creating an EA repository based on linking existing data sources. This supports in comprising different data formats, enhances the reusability of existing results in research and industry and allows for consistency checks and analysis also between the different data sources. That means, through an integration of the local ontologies in the neutral EA ontology inconsistencies between these local ontologies can be detected, since we have one big mapped ontology. For example if an activity in a business process requires a specific data input, but does not call the corresponding service, this error can be detected if there is an integrated SOA, BPMN and EA repository. Through traversing the ontology the mismatch between the service and the required data is detected.

5 RELATED WORK

The Essential Meta-Model (Essential Project, 2012) is an EA model that has been developed in the scope of the Essential Project. It provides on the basis of the ontology editor Protege an open source EA tool, the Essential Architecture Manager. They predefine the T-Box of the architectural model in OWL. It incorporates best practices and can be extended to meet particular requirements. Also, they claim that it can be easily mapped to industry standard frameworks. In SEAM we aim at automating the acquisition and maintenance of the data required for an EAM initiative while relying on existing research and best practices on EA modeling.

Using semantic technologies for formalizing information sources yields a number of advantages, starting with having a formal, unambiguous model to the possibilities of reasoning and consistency checking. TopQuadrant, for example, describes in its whitepaper (TopQuadrant, 2012) the vision of executable EA models based on semantic technologies. Executable means that the knowledge base can be consulted at run time by humans as well as by applications. In the last years, architectural models have been published as ontologies (such as for the TOGAF content metamodel (Gerber et al., 2010)). As described in section 3, the SEAM approach uses these models.

6 CONCLUSIONS

Enterprise architecture management (EAM) aims to optimize the alignment of business and IT. EA repositories provide the required information base for decisions on the information systems landscape, e.g. for the development of a new application or new services. EA frameworks such as TOGAF provide metamodels as guideline for creating EA repositories. They define which information is required and how it is linked. This information, however, is only the core. A more comprehensive information base can be provided by integrating data from other EAM related information sources. Connecting such already existing information sources to the EA documentation avoids redundant time and effort for its creation and maintenance. However, integrating the data from these heterogeneous sources, each coming with its own data model in a particular language, is a complex task.

We propose the use of semantic technologies for integrating heterogeneous EAM relevant information sources allowing for a Semantic Enterprise Architecture Management - SEAM. We defined a process for building a SEAM repository that uses linked data within an enterprise. Thereby, existing enterprise data sets are exposed as ontologies and linked into the enterprise architecture. The EAM initiative can thus focus on modeling the interdependencies between business, information systems and IT infrastructure. Mismatches between the documentation within the information sources and reality can be easily discovered and made transparent. We demonstrated the approach by linking EA documentation (according to TOGAF), business process models (in BPMN) and aspects of services (according to the SOA ontology).

Depending on the existence of relevant data sets the documentation task within an EAM initiative can thus be reduced. Moreover, the resulting EA model does not only statically document an EA, but is available in a formal and machine-readable representation formalism. Semantic technologies provides a strong foundation for architecture analysis and consistency checks as well as additional supporting tasks for EAM. These will include visualization but also novel applications of AI methodologies such as automated planning in the field of EA.

Future works include the refinement of the defined mapping axioms and the inclusion of additional data sources based on well-known standards. As the chosen OWL language fragment is limited, complex mapping operations such as aggregations or conversions may be applied outside the repository within a data provisioning step.

REFERENCES

Aier, S., Riege, C., and Winter, R. (2008). Unternehmensarchitektur literaturüberblick und stand der praxis. *Wirtschaftsinformatik*, 4(50):292–304.

Blowers, M. (2012). Hybrid enterprise architecture frameworks are in the majority. http://ovum.com/2012/03/22/hybrid-enterprise-architecture-frameworks-are-in-the-the-majority/, accessed Feb. 12 2013.

Buschle, M., Grunow, S., Hauder, M., Matthes, F., and Roth, S. (2012). Automating enterprise architecture documenation using models of an enterprise service bus. In *AMCIS 2012 Proceedings*, Seattle, Washington.

Essential Project (2012). Essential meta-model. http://www.enterprise-architecture.org/component/content/article/11-project-components/35-essential-meta-model, accessed Feb. 12 2013.

Farwick, M., Agreiter, B., Breu, R., Ryll, S., Voges, K., and Hanschke, I. (2011). Automation processes for enterprise architecture management. In *Proceedings of the 2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops (EDOCW '11)*, pages 340–349. IEEE Computer Society.

Gerber, A., Kotz, P., and van der Merwe, A. (2010). Towards the formalisation of the TOGAF content metamodel using ontologies. In *Proceedings of the 12th International Conference on Enterprise Information Systems (ICEIS 2010)*, pages 54–64, Funchal, Madeira, Portugal.

Matthes, F. (2011). *Enterprise Architecture Frameworks Kompendium: Über 50 Rahmenwerke für das IT-Management*. Springer, 1 edition.

Motik, B., Grau, B., Horrocks, I., Wu, Z., Fokoue, A., and Lutz, C. (2012). OWL 2 web ontology language: Profiles. http://www.w3.org/TR/owl2-profiles/, accessed Feb. 12 2013.

Natschlaeger, C. (2011). Towards a BPMN 2.0 ontology. In *Business Process Model and Notation*, Lecture Notes in Business Information Processing, page 115. Springer.

Open Group (2009). TOGAF version 9. Technical report, Van Haren Publishing.

Open Group (2010). Service-oriented architecture ontology. ISBN 1931624887.

Stuckenschmidt, H. (2007). Partial matchmaking using approximate subsumption. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI 07)*, pages 1459–1464.

TopQuadrant (2012). Towards executable enterprise models: Building semantic enterprise architecture solutions with topbraid suite. http://www.topquadrant.com/docs/whitepapers/WP-BuildingSemanticEASolutions-withTopBraid.pdf, accessed Feb. 12 2013.

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