Enterprise Architecture-Based Service Portfolio Management for Automated Service Catalog Generation

Clemens Landsberg and Markus Esch
Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE,
Fraunhoferstraße 20, 53343 Wachtberg, Germany
itsm@fkie.fraunhofer.de

Keywords: IT Infrastructure Library, ITIL, Service Portfolio Management, Service Catalog, Enterprise Architecture, EA, Nato Architecture Framework, NAF.

Abstract: IT Service Management (ITSM) tackles the problem of the ever rising complexity of IT solutions by defining small manageable fields of duty with clear responsibilities. The de facto standard in this field is the IT Infrastructure Library (ITIL). One of ITIL's important tools is the Service Portfolio Management (SPM) which aligns the IT to the business and serves as an interface to the customer. Another means for handling complex IT systems are Enterprise Architectures (EA). EA are widely used for strategic planning, documentation and analysis of enterprises in general and IT solutions in particular. This paper proposes an approach that connects EA and ITSM by utilizing EA for SPM. The approach described in this paper enables automated generation of user-group-specific Service Catalogs from EA models. By this means the number of information sources is reduced, which increases the efficiency and reduces the error-proneness of information maintenance. Moreover the reuse of the EA models in the context of SPM is facilitated which aligns both processes and improves the organizational efficiency. The paper presents a proof of concept implementation using the Nato Architecture Framework (NAF).

1 INTRODUCTION

The success of an IT service provider is not only based on the product it sells. One of the most common causes for an IT service provider to be unprofitable are wrong decisions in the management. This failures can manifest in a badly controlled infrastructure or wrong business strategies. To address this problem a service strategy is a valuable tool. Forming a well working service strategy is however a very difficult task which requires access to vast IT-, economic- and academic knowledge. The *IT Infrastructure Library (ITIL)* condensates this knowledge in a set of best practices which provide a framework for the management of IT services. This framework divides *IT Service Management (ITSM)* in five phases: *Service Strategy*, *Service Design*, *Service Transition*, *Service Operation* and *Continual Service Improvement (CSI)*. For each phase certain processes are defined that should be implemented by service providers in order to ensure the reliable delivery of their services. As the term 'Service' is highly ambiguous please note that we use the ITIL v3 definition which is: *"A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks."*(ITIL, 2011b). This definition includes both Business Services (a service sold to a customers) and Technical Services (a service that supports a business service).

One of the central processes of Service Strategy is the *Service Portfolio Management (SPM)*. The *Service Portfolio* is the collection of all services of an IT service provider. Each service is described by a service record covering all relevant information about the service such as utility, warranty, management, finances, technology, etc. The Service Portfolio does not only contain the currently available services but also retired and planed services. ITIL uses the Service Portfolio as a tool for strategic decisions on future service development. Each service record needs to include all information needed for such strategic decisions as described in section 3.1. The subset of

Landsberg C. and Esch M.
Enterprise Architecture-Based Service Portfolio Management for Automated Service Catalog Generation.
DOI: 10.5220/0005887502240230
In *Proceedings of the Fifth International Symposium on Business Modeling and Software Design (BMSD 2015)*, pages 224-230
ISBN: 978-989-758-111-3

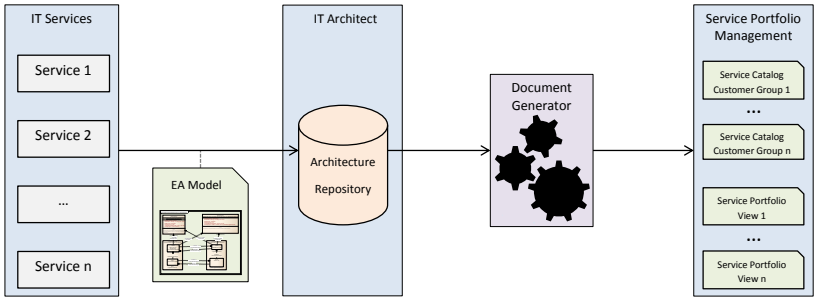

Figure 1: Generating Service Portfolio views and Service Catalogs from EA models.

services which are currently ready for use is the so-called *Service Catalog*. According to ITIL this catalog is maintained by the *Service Catalog Management* process of the *Service Design* phase.

The Service Catalog is made publicly available to customers in order to announce the provided services. ITIL recommends two kinds of customized Service Catalogs: the *Business Service Catalog*, focusing on the utility for the customer and the *Technical Service Catalog*, focusing on the technical details of the service. Often it is beneficial to distinguish even more catalogs, each specifically tailored for a certain customer group or domain. Such domain specific catalogs can focus on the relevant aspects for the given user by composing the catalog accordingly. Typical domains/customers that can be distinguished are business customers, private individuals, technicians, different departments of a company (finances, HR, etc.) or management. Even event-specific catalogs to provide services for some special event are conceivable. A Mobile Service Provider for example may want to create a fancy catalog for private customers. For business customers a serious design is required and certain irrelevant rates can be excluded. Technicians on the other hand need a spec sheet. IT Managers need to see the performance of the Services and the finance department is interested in the Service costs. Maintaining each Catalog separately implies huge efforts and is error-prone. Our approach solves this problem, based on a consistent Enterprise Architecture (EA).

ITIL recommends the use of the EA approach as blueprint for the strategic development of services and to optimize the IT solutions (ITIL, 2011a). EA is an approach to create an abstract model of a whole enterprise, including the processes, the IT infrastructure, the relation to costumers and other enterprises etc. Usually these abstractions are very coarse-grained to allow high level views. The use of EA provides many benefits, Jung summarizes (Jung, 2009) them as follows: *"An organization believes that an EA can help improve the business/IT alignment gap, business and technology communication, and IT project success rate and provide the benefits such as cost reduction & technology standardization, process improvements, and strategic differentiation"*.

To handle the complexity of EA and to give guidelines for the development of such an architecture, several different *Enterprise Architecture Frameworks (EAF)* have been developed. For example *The Open Group Architecture Framework (TOGAF)* or the *Zachman Framework*. Some have been developed for specific domains such as the *Department of Defense Architecture Framework (DoDAF)* or the *NATO Architecture Framework (NAF)*. ITIL does not specify the use of any particular framework but rather focuses on the integration of the EA approach into ITSM processes. For the approach proposed by this paper it is also not relevant which EAF is used, as long as the framework supports the concept of services. Our proof of concept implementation is based on the NAF.

As the name suggests NAF is an EAF used in the NATO environment. The NAF is divided into subviews which describe enterprises in terms of structure, projects, capabilities, business cases, technology, services, and business relationships. While the framework was developed for the defense sector, it may be used in other domains as well since it does not contain defense specific model elements.

SPM as well as EA captures information about the IT services of a service provider. Although both do not require exactly the same set of information there is a huge overlap. For this reason joining SPM and EA is a reasonable approach that bears the potential to rise efficiency and to reduce inconsistency and error-proneness by avoiding duplicated information maintenance. Another advantage is the automated generation of customized Service Catalogs as well as Service Portfolio excerpts for specific purposes and domains.

We propose to capture all relevant information about an IT service in the architecture. SPM tools can acquire this data automatically from the EA and represent it in an appropriate format for SPM. However the primary data source is the EA. In order to allow automated retrieval of the information from the EA strict modeling guidelines are required. Existing

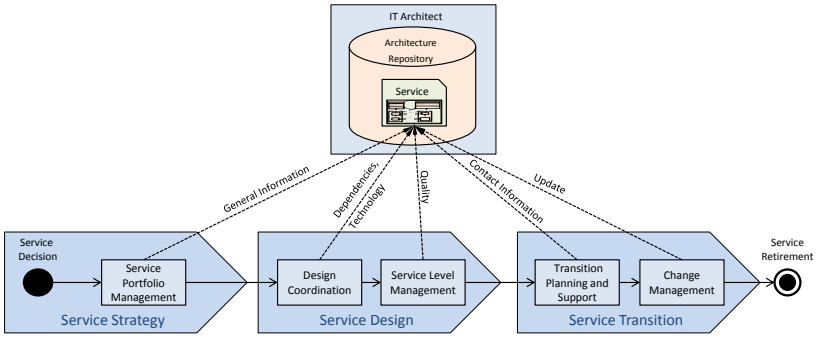

Figure 2: Information are added successively to the EA model during different phases of the *Service Lifecycle*. Changes applied during Service Operation and CSI are induced through the Change Management of the Service Transition phase.

EAF are not specific enough in this point. Hence an extension of the EA meta model is required.

The remainder of this paper is organized as follows: We provide an overview on related work in section 2. Afterward the general approach of *Enterprise Architecture-based Service Portfolio Management* is described in section 3. A prototype realization of the approach is presented in section 4. The paper concludes with a discussion of our main contributions, open issues and future work in section 5.

2 RELATED WORKS

The value of EA is undisputed. Ross and Weill examined these benefits and classify EAs in different categories (Ross and Weill, 2006). In their opinion the value is not primarily given by an EA as an usable product. Instead an enterprise applying EAs improves by other means as for example the learning effect yielded by the modeling of the EA. Later research by Ross and Quaadgras (W.Ross and Quaadgras, 2012) showed that the benefit generated by the use of EA is saturated after a certain amount of effort has been spend for EA. It becomes clear that EA needs to be integrated deeper in the enterprise to derive additional value. Our approach tries to create immediate value by generating a usable product for ITSM.

To produce a usable product becomes even more important when you consider that between 40% (Gartner, 2007) and 66% (Roeleven, 2010) of all EA are predicted to fail and do not deliver any value. Of course preventing general EA failures cannot be guaranteed by our approach. However, the risk of producing an useless EA can be reduced by delivering added value for ITSM.

An EA model is mainly a database for enterprise information. As such it is obvious that a direct value can be produced by making the data available to the stakeholders (IT Managers, Technicians, Customers, etc). This can ether be done by an ad hoc visualization (Roth et al., 2013a)(Roth et al., 2013b) or by preparing a polished document as provided by the approach discussed in this paper.

The Idea to combine EA and SPM is not new. In (Bonham, 2004) architectures are applied for Business-IT alignment. Another approach to combine EA and SPM has been proposed in (Sarno and Herdiyanti, 2010). Sarno and Herdiyanti mapped an *Enterprise Resource Planning* portfolio to EA in order to perform business analysis based on this portfolio. Yet the focus of this work is different since it does not take ITSM and ITIL into account. A more ambitious approach is presented by Lankhorst (Lankhorst et al., 2011). Lankhorst proposes to support Portfolio Lifecycle decisions by predicting the costs based on a suitable EA model and benchmarks data.

SPM is usually done using specialized ITSM software that provides a *Configuration Management System (CMS)* such as *HP Service Manager* (HP, 2015) or *MSM Integrated ITSM Software* (MSM, 2015). These tools are typically independent from the service provider's EA and hence constitute a secondary information source. However first approaches for the integration of EA and CMS are currently developed for commercial ITSM tools as well. Jensen for example describes an approach where *Configuration Management Database (CMDB)* items are synchronized with EA models (Jensen et al., 2009). The approach discussed in this paper represents a viable addition to this development and could be integrated into commercial CMS tools.

3 ENTERPRISE ARCHITECTURE-BASED SERVICE PORTFOLIO MANAGEMENT

This section presents the details of our Enterprise Architecture-based Service Portfolio Management

approach. The basic idea is to align SPM and EAs in a business in order to raise the efficiency. We are using the EA as exclusive information source to avoid duplicated data management. For this reason, the approach is well applicable by enterprises that have established both SPM as well as EA. The general idea is depicted in Figure 1. All services are described by means of an EA model. The models are aligned by detailed modeling guidelines. These guidelines ensure, that all relevant information is captured by the service models and that the information is modeled equally for all services. All service models are kept in a central repository. At any time current Service Catalogs tailored for specific domains and customer groups can be created using a document generator. Moreover different views on the entire Service Portfolio can be generated for different use cases. Please note, that the remainder of this paper focuses on generating tailored Service Catalogs. However, generating Service Portfolio views can be done in an equivalent way.

A detailed description of our approach is provided in the subsequent sections. Subsection 3.1 describes the service record which is the set of all relevant information for a service. Afterwards subsection 3.2 presents an service architecture management process that coordinates the development of the service model during the realization of the IT service. In order to be able to model all the required information on a service the EA meta model needs to be extended as described in subsection 3.3.

3.1 Service Record

A service record is the set of all relevant information on a given service. In order to allow informed strategic decisions on the one hand and to give customers detailed information on the provided services on the other hand, a service record needs to contain various information. This includes aspects of finances, user satisfaction, technology, quality, service levels, service dependencies etc. For the purpose of automatically generating a Service Catalog, the minimum set of information contained in a service record is the information provided to customers. To explain our approach we are using an excerpt of this information displayed in Table 1 throughout this paper.

A name or identifier is needed to refer to the service. A service also needs a description. Sometimes it is advantageous to have more then one description, one for each Service Portfolio view and Service Catalog. A classification is often used for a better understanding and to improve searchability. A service record usually contains various contact information. This contact information serves as a reference to all related ITSM functions. Furthermore a service record has to refer to the *Service Level Agreement (SLA)*. As a service can have different variations or different implementations, the SLA specifies the actual service quality as provided by the enterprise.

Table 1: An excerpt of the information contained in a typical service record of a Service Catalog.

Entry	Explanation
Name	Name / identifier of the service.
Description	A description of the service.
Classification	A classification of the service. For example if it is internal or external facing. Often done by means of a taxonomy.
Contact	Reference to important organizational units. E.g. Service Owner, User Help Desk...
Dependencies	The composition of the service: Which services are using the service and which services are used.
SLA	References to Service Level Agreement (SLA) or Operational Level Agreement (OLA). The SLA is a detailed contract describing the utility and warranty of the service.

3.2 Process Description

In order to implement our approach efficiently it is crucial to define a process for the management of the IT service model. This process needs to interweave the creation of the service model with the processes of the ITSM. Although ITIL defines the IT architect as an organizational unit responsible for the design, creation and maintenance of EA models, it does not specify how EA models are handled.

The process for the creation of the service model is depicted in Figure 2. Responsible for the service model is the *IT Architect*. The life cycle of a new service begins with a strategic decision in the Service Strategy phase. At this point a new, empty service model will be created. During the *Design and Transition* process the service is successively enhanced by additional information. At the first stage in the SPM general information about the service such as name and classification is already available and can be added to the model. The remaining information is not yet available and will be added to the model later on. During the *Service Design* phase the service is planed and designed in detail including technological specifications and dependencies to other services. This information is also added to the service model.

The *Service Level Management* is responsible for the determination of service quality parameters and the definition of service levels. These quality information has to be added to the service model as well. Implementation and operation specific information such as contact information is added to the service model during the *Service Transition* phase through the *Planning and Support* process. Finally the service model needs to be update continuously in order to provide up-to-date information about the service. All service changes are controlled by the *Change Management* process. This process is hence responsible for updating the service models during *Service Operation* and *Continuous Service Improvement*. Since the service models are used as primary information source, continuously updating the service models along with quality assurance by the IT Architect is crucial for the success of our approach.

3.3 Meta Model

The basic idea of our approach is to retrieve all required information for the creation of a Service Catalog from the EA models of the services. This requires coherent modeling of the services. EAFs such as the NAF were created to provide modeling guidelines and to ensure uniform models. However EAFs are typically not strict enough in order to enforce biunique models. Practical experience shows, that EA models created by different architects can differ significantly although the same EAF is used. Hence additional modeling guidelines and conventions are required for our approach. Since services are logical constructs that abstract from the internal, technical structure such strict modeling guidelines can easily be defined by specifying for each information item exactly how it has to be represented in the model. Additionally naming conventions help to avoid ambiguous naming.

An EAF is based on a meta model which defines the rules and guidelines for the creation of EA models. However EAFs typically do not cover all aspects of ITSM. Hence a biunique mapping of the information required in SPM to elements of the service model is often not possible. For this reason the meta model needs to be extended by specific entries required in the context of SPM. Extensions are basically additional tagged values and associations between stereotyped classes. For our prototype, we extended the NAF accordingly. Our extensions will be discussed in section 4.1. Based on the extended meta model we were able to define strict modeling guidelines on how to model all the relevant information of a service.

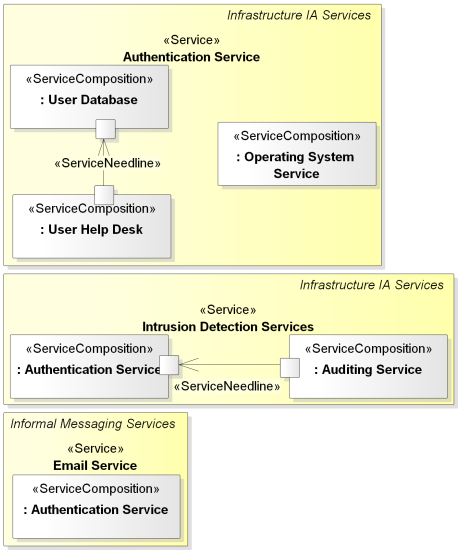

Figure 3: This diagram shows a NOV-6. NAF uses a directed composition relationship with parts to represent dependencies between services. The figure shows on the left hand side on which services the Authentication Service depends. On the right hand side two services are shown that require the Authentication Service.

4 PROTOTYPE IMPLEMENTATION

We have realized the approach presented in section 3 prototypically in order to show the approach's viability. For this purpose we have extended the NAF meta model and modeled an example IT service (see subsection 4.1). Additionally a document generator has been implemented that can be used to automatically generate a customer facing Service Catalog from this model (see subsection 4.2). The prototype implementation as well as the modeling has been done using the *Enterprise Architect* of *Sparx Systems* (SparxSystems, 2014).

4.1 Example Model

Automatically generating a Service Catalog for the EA model requires that all relevant information is contained in the model. As described in section 3.3 this requires an extension of the meta model by tagged values and associations. We have extended the meta model of the NAF and modeled an *Authentication Service* as an example.

The main element for services in NAF is the stereotype *Service*. Each service is supposed to be represented by a class with this stereotype. As shown in Figure 4 we have extended the Service stereotype

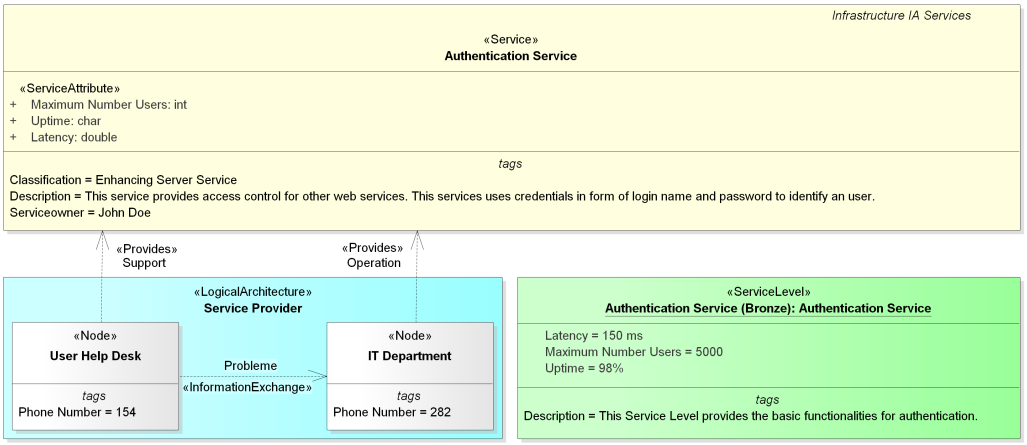

Figure 4: This diagram shows a NOV-2. This view focus on relationships between logical entities. Note that the *ServiceLevel* is not part of a NOV-2 diagram but has been added for illustration reasons.

by additional tagged values. These contain the information required for the Service Catalog. For example the information items *Description* and *Classification* from Table 1 are represented in additional tagged values. Another classification is done by a class hierarchy where each service extents a class from a taxonomy. As shown in Figure 4 the Authentication Service extends the *Infrastructure IA Service* from the *NATO C3 Classification Taxonomy*. The contact information is represented by a *Provides* dependency of the *NATO Operational View 2 (NOV-2)* as also shown in Figure 4. By this means the service is connected to organizational resources.

Using a *NATO Service Oriented View 6 (NSOV-6)*, as shown in Figure 3, dependencies to other services can be modeled using the composition mechanism. The *Utility* and *Warranty* parameters of a service, which are a fundamental parts of the SLA, are defined within the *ServiceAttribute* of the *NATO Service Oriented View 1 (NSOV-1)*. For illustration reasons these attributes are depicted in the NOV-2, shown in Figure 4 as well. The actual service level is expressed by the *ServiceLevel* stereotype and is linked to a particular service provider via tagged value of the *«Provides»Operation* dependency, also depicted in Figure 4.

4.2 Document Generator

The purpose of the document generator is to create a human readable document describing the service. The document has to contain the important information of the service in a mixture of text and tables. EA tools typically have features for document generation. They are sufficient for internal use, but reach there limits when a highly polished customer facing Service Catalog or a non sequential processing is required. This requires a sophisticated document generator.

We have implemented an document generator for the Sparx Systems Enterprise Architect. As most EA software this tool provides a build-in scripting runtime environment. It allows us to use internal mechanisms for extracting values from the model, graph traversal and SQL queries on the underlying database. The document generator is a script that reads the information from the EA model and successively creates the output document. For this purpose it uses an *Component Object Model (COM)* API to control an external word processing tool (*MS Office*). Whenever a data item from the model is needed, a subroutine is called that retrieves the desired information. This information will then be written in a document via the COM-API. This process is depicted in Figure 5.

To realize the idea of a customizable Service Catalogs for different domains it was necessary that each subroutine retrieves one atomic unit of information and embeds it into the output document. Document generator scripts for customized Service Catalogs can now easily be created using these subroutines and arranging them as desired. While our prototype implementation still requires writing a document generator script, the process can easily be facilitated by a wizard software. Such a software allows selecting and arranging the information items via a GUI and automatically produces the desired catalog.

5 CONCLUSION

The approach presented in this paper combines ITSM and EA. The aim is to improve the benefit of EA for an IT service provider. Moreover organizational efficiency can be improved and error-proneness can be reduced by maintaining just one primary information

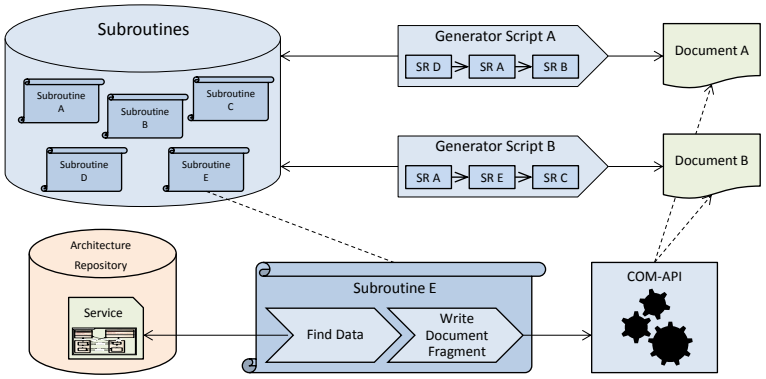

Figure 5: Schematic overview of the document generation.

source. Our approach allows automated generation of up-to-date Service Catalogs and Service Portfolio views tailored for different customer groups, domains and purposes. Moreover strict modeling guidelines for the creation of service models leads to clearly structured models and improves comparability, reproducibility and quality. Finally this entails a speedup of the modeling process since architects can stick to the predefined guidelines.

To show the viability of our approach we presented a proof of concept implementation based on the NAF and Sparx Systems Enterprise Architect. This implementation covers the entire process chain from the architecture model through the document generator to the domain specific Service Catalog.

While we have implemented our approach just for the automated generation of a Service Catalog, an adapter that automatically synchronizes the information from the EA model with a sophisticated SPM tool can easily be realized (Hauder et al., 2013). The implementation of such an adapter is part of our future work in order to show the usability of our approach for the entire Service Portfolio process. Moreover an even deeper integration of EA into ITSM is conceivable by automatically generating up-to-date self-service web pages from the EA model and enabling customers to order the provided services immediately via the web interface.

Another aspect for future work is the improvement of the usability. We are considering a tool that allows non-architects to enter most of the required information about a service in a template or wizard dialog. This information can then be used to generate a service model fragment that constitutes the basis for further modeling by an experienced architect.

REFERENCES

Bonham, S. (2004). *IT Project Portfolio Management*. Artech House, Norwood, 1st edition.

Gartner (2007). Gartner enterprise architecture summit.

Hauder, M., Matthes, F., and Roth, S. (2013). Challenges for automated enterprise architecture documentation. 7th International Workshop on Trends in Enterprise Architecture Research.

HP (2007-2015). Service manager. http://www8.hp.com/us/en/software-solutions/service-manager-service-desk.

ITIL (2011a). *Service Design*. IT infrastructure library. Stationery Office.

ITIL (2011b). *Service strategy*. IT infrastructure library. Stationery Office.

Jensen, A., Knowles, J., and McBride, S. (2009). The dynamic landscape of enterprise architecture.

Jung, C. (2009). Actionable enterprise architecture. In *SNPD*. IEEE Computer Society.

Lankhorst, M., Boekhoudt, P., and Lagerström, R. (2011). A lifecycle approach to portfolio management, journal of enterprise architecture. *Journal of Enterprise Architecture, ISSN–2166-6768*.

MSM (2000-2015). Msm integrated itsm software. http://www.marval.co.uk/itsm-software.aspx.

Roeleven, S. (2010). Why two thirds of enterprise architecture projects fail.

Ross, J. W. and Weill, P. (2006). Understanding the benefits of enterprise architecture.

Roth, S., Hauder, M., and Matthes, F. (2013a). Collaborative evolution of enterprise architecture models. In *Models@run.time 2013*. 8th International Workshop on Models at Runtime.

Roth, S., Hauder, M., Zec, M., Utz, A., and Matthes, F. (2013b). Empowering business users to analyze enterprise architectures: Structural model matching to configure visualizations. In *TEAR 2013*. 7th Workshop on Trends in Enterprise Architecture Research.

Sarno, R. and Herdiyanti, A. (2010). A service portfolio for an enterprise resource planning. *International Journal of Computer Science and Network Security, ISSN–1738–7906*.

SparxSystems (1998-2014). Enterprise architect. http://www.sparxsystems.de.

W.Ross, J. and Quaadgras, A. (2012). Enterprise architecture is not just for architects.