Americas Conference on Information Systems (AMCIS)

AMCIS 2009 Proceedings

Association for Information Systems *Year* 2009

A Pattern-based Approach to Quantitative Enterprise Architecture Analysis

Sabine Buckl[*] Ulrik Franke[†] Oliver Holschke[‡]
Florian Matthes[**] Christian M. Schweda[††]
Teodor Sommestad[‡‡] Johan Ullberg[§]

[*]Technische Universitat Munchen, buckls@in.tum.de
[†]Royal Institute of Technology, ulrikf@ics.kth.se
[‡]Technische Universitat Berlin, oliver.holschke@sysedv.tu-berlin.de
[**]Technische Universitat Munchen, matthes@in.tum.de
[††]Technische Universitat Munchen, schweda@in.tum.de
[‡‡]Royal Institute of Technology, teodors@ics.kth.se
[§]Royal Institute of Technology, johanu@ics.kth.se

A Pattern-based Approach to Quantitative Enterprise Architecture Analysis

Sabine Buckl
Technische Universität München
buckls@in.tum.de

Ulrik Franke
Royal Institute of Technology (KTH)
ulrikf@ics.kth.se

Oliver Holschke
Technische Universität Berlin
Oliver.Holschke@sysedv.tu-berlin.de

Florian Matthes
Technische Universität München
matthes@in.tum.de

Christian M. Schweda
Technische Universität München
schweda@in.tum.de

Teodor Sommestad
Royal Institute of Technology (KTH)
teodors@ics.kth.se

Johan Ullberg
Royal Institute of Technology (KTH)
johanu@ics.kth.se

ABSTRACT

Enterprise Architecture (EA) management involves tasks that substantially contribute to the operations of an enterprise, and to its sustainable market presence. One important aspect of this is the availability of services to customers. However, the increasing interconnectedness of systems with other systems and with business processes makes it difficult to get a clear view on change impacts and dependency structures. While management level decision makers need this information to make sound decisions, EA models often do not include quality attributes (such as availability), and very rarely provide quantitative means to assess them. We address these shortcomings by augmenting an information model for EA modeling with concepts from Probabilistic Relational Models, thus enabling quantitative analysis. A sample business case is evaluated as an example of the technique, showing how decision makers can benefit from information on availability impacts on enterprise business services.

Keywords

EA Management, EA Analysis, Dependency Modeling, Quantitative Analysis, Probabilistic Relational Models

INTRODUCTION

Enterprise Architecture (EA) management is increasingly becoming a tool for modern enterprises to keep up with the increasing demand for flexible IT support. In particular, EA *analysis* (Johnson, Johansson, Sommestad, and Ullberg, 2007) is a means of providing decision support throughout the management process. Using EA analysis, system properties (quality attributes) such as availability, performance, flexibility, or operational risk, etc., can be operationalized as metrics. These metrics can then be used to compare different scenarios (Lankes, 2008).

To perform metrics based EA analysis, architecture information is required. Such information constitutes an EA model. The model must reflect the analysis' need – the information relevant to *performance* analysis is most likely different from the one needed for *flexibility* analysis. Thus, the meta model for the EA architecture model – in this article referred to as *information model* – must be *problem adequate* (Stachowiak, 1973), as the analysis addresses specific EA problems. Since different enterprises have different EA priorities, specific EA information models may be developed, based on problem specific information model fragments and thus capable to support specific EA needs. An approach to construct such a model from *EAM patterns* is discussed in Buckl, Ernst, Lankes, Schneider, and Schweda, 2007 and is employed below.

EA analysis may target different system quality attributes. A list is given in Lankes, 2008, where the attributes are prioritized according to their perception by EA practitioners. The most important quality attribute mentioned there is the *operational risk*, which refers to failures in the functionalities offered by the business applications. However, as risk is hard to measure, Lankes, 2008 proposes *availability* as an easily measurable, related property. Availability is understood in accordance to Gertsbakh, 1989 as the ratio between the *uptime* and the *overall operation time*, i.e. as the share of time an application or service is working. Overall operation time is the sum of uptime and downtime. Now, in a complex and interwoven system of systems, a local failure, e.g. of one application, is likely to have serious impact on the overall availability of many other systems. These repercussions of a failure may pose an important threat to *business continuity* – a threat, which is not easy to discover. EA analysis (Johnson et al., 2007) can be helpful in this context, to map and understand the causal dependencies of complex systems. Availability is used as an example in the remainder of this article, even though the method is meant to be more generally applicable.

The remainder of this article is structured as follows. Section 2 outlines the state-of-the-art in EA management literature with an emphasis on EA analysis. Section 3 presents the core contribution of the article – a step-by-step approach to creating an EA analysis model for a given quantitative EA property. This approach uses EAM patterns to present a basic information model and a complementary graphical representation; a viewpoint. This initial model is subsequently refined with information on dependencies, until a formalism for quantitative analysis is reached. Section 4 provides an application of the approach in the availability example. Section 5 concludes the paper with a discussion and an outlook.

STATE-OF-THE-ART IN EA MANAGEMENT AND ANALYSIS

In this section, the state-of-the-art regarding EA management is discussed, focusing on quantitative analysis. In addition, an approach to evaluate availability on EA-level is introduced, which serves as a starting point for Section 3.

Aier, Kurpjuweit, Schmitz, Schulz, Thomas, and Winter, 2008 and Winter and Fischer, 2008 present an engineering approach to EA management. The approach is based on a holistic view, which is organized in different layers – *business, process, integration, software,* and *technology architecture*. The constituents of the layers form the management subject of EA. The approach emphasizes the dependencies between elements from different layers and the impact they have on each other. To conduct impact analysis, Aier et al., 2008 proposes an integrated information model and different graphical model types. While the approach yields a graphical result, it does not provide means to quantify the impacts.

Lankhorst et al., 2005 describes an approach to EA management with an emphasis on visualization aspects. A modeling language called *ArchiMate* (Jonkers, 2006) is introduced and different kinds of visualizations for various usage contexts and stakeholder perspectives are detailed. The ArchiMate languages support modeling of EA artifacts from different domains, and can in addition be used as basis for visualizations and analyses. In order to automate the generation of such visualizations and foster the execution of analyses, Lankhorst et al., 2005 emphasizes the importance of a suitable information model. Nevertheless, only a summary of the information model of ArchiMate is given in Lankhorst et al., 2005, which contains only the core concepts and main relationships and does not provide specialized means for dependency modeling. In particular, no guidelines for actual computations are given.

The approach of *multi-perspective enterprise modeling (MEMO)* (cf. e.g. Frank, 2002), provides a framework consisting of a set of special purpose modeling languages, which can be used to support different perspectives (strategy, organization, and information system) and various aspects (e.g. resource, structure, process) relevant for enterprise modeling. In addition, a meta model is provided, showing how the special purpose languages can be integrated into a single model. The *Score-ML* (Frank, Heise, Kattenstroth, and Schauer, 2008) provides means for modeling indicator systems. A meta model is provided, which relates *indicators,* e.g. monthly revenues, to referred *domain concepts*, e.g. product. Furthermore, a concept named *CustomizedRelationship* is introduced, which exemplifies (causal) dependencies between indicators. Although an application example in the EA performance context is used to explicate the approach, no information is given on how to transform the dependencies described in a qualitative way into quantitative computation prescriptions results.

In Lankes, 2008 metrics for EAs and application landscapes are introduced as decision support techniques based on analysis of structural dependencies. The approach emphasizes on operational risk, failure propagation and availability, based on a practitioner survey. In order to explicate the structural dependencies analyzed in the paper, an information model with derived attributes is used, along with a Bayesian calculation formalism. An EA level application example is also given in Lankes, 2008, visual analysis of ex post information about failure propagation to compare different project proposals for the evolution of the application landscape. Thus, the project portfolio management process is supported.

The method for quantitative analysis provided by Lankes, 2008 constitutes the basis for the present contribution. In the following, we show how it can be linked to the notion of EA management patterns (Buckl et al., 2007), providing an integrated approach for quantitative EA decision support.

APPROACHING QUALITY ATTRIBUTES ON THE EA LEVEL

EA analyses (Johnson et al., 2007) are means to assess quality attributes on EA level. The necessity of considering the dependencies can be exemplified with the availability of a business application, which is related to the availability of the underlying hardware devices. Therefore, an appropriate way to present the failure effects to stakeholders is highly important. This is especially true, as the relationships between the affected EA constituents are usually neither easy to perceive nor to communicate. Complete or partial visualizations of EAs are often alluded to as suitable means to present EA aspects (van der Torre, Lankhorst, ter Doest, Campschroer, and Arbab, 2004). These visualizations are *views* on an underlying architectural description[1] in terms of International Standardization Organization, 2007 and are hence constructed according to *viewpoints*, which select the constituents to be displayed and determine the way to visualize them. This selection of viewpoints can be seen as conforming to an underlying information model, containing the necessary concepts.

The coupling between a viewpoint and an information model is picked up by the idea of EA management patterns as introduced in Buckl et al., 2007. These patterns reflect best practices for addressing typical EA management problems, as identified by EA practitioners from various industries. As of Buckl et al., 2007 three types of patterns can be distinguished:

- *Information model patterns* (I-Patterns), which determine the EA concepts to be modeled as well as their attributes and relationships to each other,
- *Viewpoint patterns* (V-Pattern), which describe the selection of EA concepts to be visualized and the ways to actually perform this visualization, and
- *Methodology patterns* (M-Pattern), which give indications and step-by-step introductions, how I- and V-Pattern can be employed to address the respective EA management concern.

To support EA analysis, the I-Patterns have to be refined and augmented to incorporate the EA quality attributes under consideration. The following process is a step-by-step approach to support the introduction of quality attributes on the EA level:

1) Select the appropriate I-Pattern(s): The concepts and relationships under investigation have to be reflected by classes and associations in the I-Pattern(s), such as *business application* or *business process* in the example. From the integration of the I-Patterns, an appropriate, or problem adequate, information model for the EA analysis under consideration can be created.

Example: An information model for the availability example is presented in Figure 1. It has been derived from two I-Patterns as introduced in the EAM Pattern Catalog (Buckl, Ernst, Lankes, and Matthes, 2008).

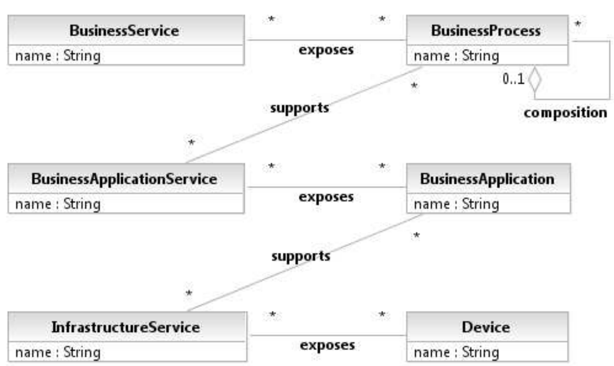

Figure 1 - Combined information model

The classes introduced in the information model have the following meanings:

- ***Device***: A physical computation resource. (Jonkers, Groenewegen, Bonsangue, and van Burren, 2005)
- ***InfrastructureService***: Externally visible unit of infrastructure, which is provided by one or more devices, exposed through well-defined interfaces. (Jonkers et al., 2005)
- ***BusinessApplication***: Software system, which is part of an information system in an organization. A business application refers to the software part of information systems, which are according to Krcmar, 2005 *sociotechnical* systems. The characterization *business* restricts the terms to software systems, which support at least one process of the respective organization.

[1] This description might be augmented with results of an EA analysis.

- ***BusinessApplicationService***: Externally visible unit of a business application, functionality, which is provided by one or more applications, exposed through well-defined interfaces, and meaningful to the environment. (Jonkers et al., 2005)
- ***BusinessProcess***: A sequence (flow) of logical individual *activities* or *functions* with connections between them, which is characterized by clearly defined input and output factors (results) (Jonkers et al., 2005 and Krcmar, 2005). The business process should not be identified with single activities or functions, but with high-level processes as found in enterprise value chains.
- ***BusinessService***: A coherent piece of functionality that offers added value to the environment, independent of the way this functionality is realized internally. (Jonkers et al., 2005)

2) Select the appropriate V-Pattern(s): The selected I-Patterns are complemented with corresponding V-Patterns, which provide adequate architectural viewpoints for the involved stakeholders. The V-Patterns can later be augmented with additional graphical means to provide information on the quality attribute under consideration.

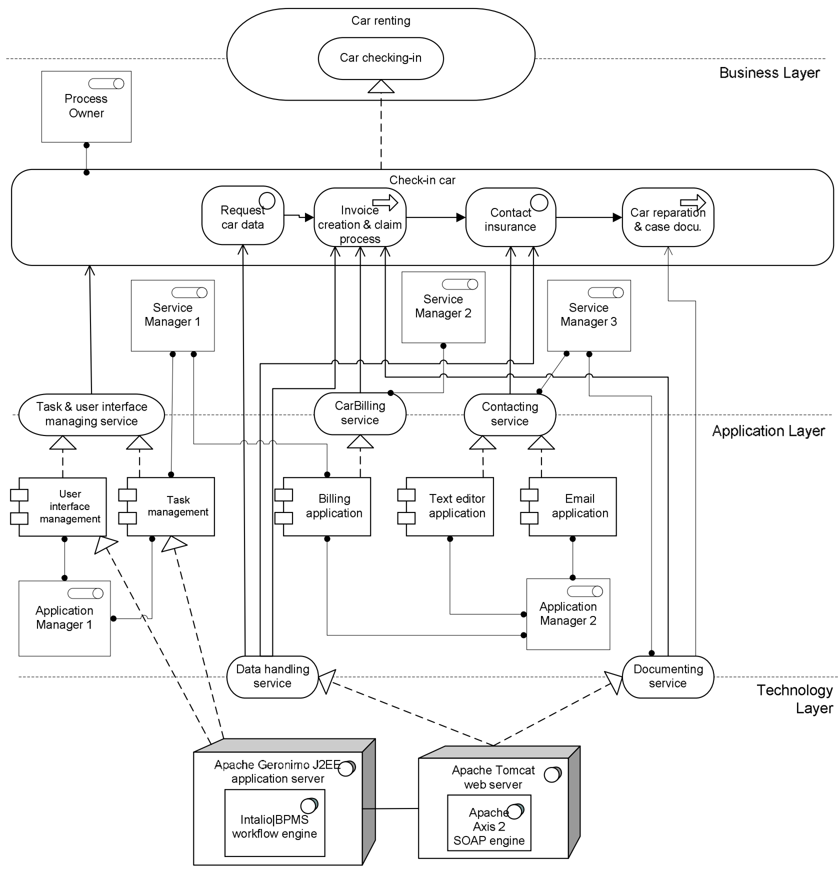

Figure 2 - View on exemplary EA of a car rental agency according to V-Patterns V-18 and V-56 (Holschke et al., 2008)

Example: V-Patterns suitable for the availability example are presented in Holschke, Närman, Rocha-Flores, Eriksson, and Schönherr, 2008 and Jonkers et al., 2005. Figure 2 contain a view conforming to the V-Patterns described in Buckl et al., 2008 with the *Application Layer* line marking the border between the Viewpoints V-18 in the upper and V-56 in the lower half.

3) Augment information model with quality attributes: The classes of the information model are augmented with properties reflecting the considered quality attribute. It should be distinguished, whether the quality attribute is actually measurable at the instances of the associated class, i.e. forms an *observable* attribute, or can only be *derived* from attribute values of related instances. This distinction lays the basis for the subsequent step, which is applied to explicate dependencies between attributes.

Example: In the availability example, all classes are assigned an *availability* attribute. Nevertheless, the availability of a business process or business service might per se not be observable, but can be computed from the availabilities of the supporting business applications.

4) Augment information model with causal dependencies: The augmented information model from step 3 does not account for the impact of a local failure in one EA element on other EA elements. To do so, the model has to be enriched with further information, indicating the *causal dependencies* underlying the propagation of failures between the different elements. These dependencies can be explicated as relationships between the attributes of classes, more precisely between the attributes' values. Such relationships, called *attribute relationships*, are discussed in Johnson et al., 2007. The graphical notation for these relationships as introduced in Buckl, Ernst, and Schweda, 2008 is used for explicating the attribute relationships in Figure 3.

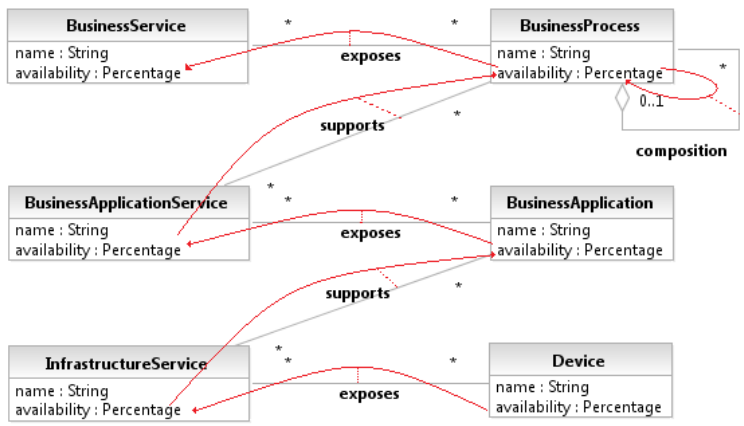

Figure 3 - Augmented information model with attribute relationships.

Example: The *attribute relationship* between the availability attribute of a business application service and that of an associated business processes, as graphically indicated above, explicates that the availability of the business application service *somehow* influences the availability of its supported business processes. This influence is thus only specified *qualitatively* and no indications of the type (positive, negative) or amount of influence are made, which would have to be formalized in order to support availability computations.

5) Provide a formalization for the augmented information model: The notion of availability in an EA context is non-trivial. From the perspective of an IT decision maker, a lot of uncertainty is involved. These uncertainties can be categorized into three main classes (Johnson, Lagerström, Närman, and Simonsson, 2007):

- **Definitional uncertainty**. There is no general agreement on the meaning of all concepts. In a technical context, availability can be defined in terms of mean time to failure (MTTF) and mean time to recovery (MTTR), etc. In a management context, however, availability might be defined in terms of its impact on business processes, and the requirements put in monetary terms.
- **Theoretical heterogeneity**. There is no general agreement on a single framework within which all EA knowledge can be expressed. While there are attempts to rectify this such as Franke, Höök, König, Lagerström, Närman, Ullberg, Gustafsson, and Ekstedt, 2009 and International Standardization Organization, 2007, this is nevertheless likely to remain the case for the foreseeable future.
- **Causal uncertainty**. It is rare that empirical phenomena are fully understood. For example, it is unlikely that an IT decision maker *knows for certain* to what extent the addition of redundant servers increases availability.

Now, these uncertainties pose requirements on the languages used for EA descriptions and analysis. Following the evaluation in (Johnson et al., 2007), a fully deterministic language for availability calculations is not sufficient. Instead, we opt for the probabilistic framework offered by Probabilistic Relations Models (PRM), a technique related to Bayesian Belief Networks (BBN). As it is succinctly put in Getoor, Friedman, Koller, Pfeffer, and Taskar, 2008, PRMs "are to Bayesian networks as relational logic is to propositional logic".

A PRM model may be instantiated as a *relational skeleton*, σ_r, containing just objects, object relationships, and attributes. Furthermore, a *qualitative dependency structure* S defines the details of the attribute relationships, i.e. the sets of probabilistic parents influencing each attribute. Finally, the PRM is completed by the set of *parameters* θ_S specifying the full conditional probabilistic dependencies between attributes in the form of numbers in Conditional Probability Tables (CPT). The following expression thus defines the conditional probability of an instance I, given σ_r, S, and θ_S:

$$P(\mathcal{I}|\sigma_r, \mathcal{S}, \theta_{\mathcal{S}}) = \prod_{x \in \sigma_r} \prod_{A \in \mathcal{A}(x)} P(\mathcal{I}_{x.A}|\mathcal{I}_{\mathrm{Pa}(x.A)}) = \prod_{X_i} \prod_{A \in \mathcal{A}(X_i)} \prod_{x \in \sigma_r(X_i)} P(\mathcal{I}_{x.A}|\mathcal{I}_{\mathrm{Pa}(x.A)})$$

Compared to the standard chain rule for BBN, this equation is different in three ways: (i) the random variables are the attributes of a set of objects, (ii) the parents of a random variable depend on the model context of the object, and (iii) the parameters are shared between the attributes of objects in the same class. In other words, the variables in the dependency structure are the properties of the objects in the instantiated information model, and their causal relations are expressed by the CPT. (Getoor et al., 2007) We thus have a framework for probabilistic analysis of EAs such as Figure 3.

However, it remains to be shown more precisely how to connect the model made up from I- and V-Patterns with the quantitative aspect of the PRM analysis framework. In the following, we describe two method steps, adhering to those proposed in (Franke, Flores, and Johnson, 2009), demonstrating how this is achieved. Each step sets different parts of the qualitative dependency structure S, and the corresponding parameters θ_S:

1. **Identify deterministic attribute relations.** Starting with the attributes relevant to the concern, the first step is to identify the *necessary* conditions for the goals to be reached. Often, these can be encoded in simple logical relations such as AND or OR relations.

   Example. In the availability context, many simple logical relations that can be identified, e.g. both routers AND servers must be working if the network is to work, either one OR the other of two duplicate hard drives must be working if a certain computer is to work, etc. It is crucial to realize that these relations only provide the foundations for a working architecture, no *guarantee* of functioning. This more complex systems behavior is what we aim to describe in Bayesian attribute relations.

2. **Identify probabilistic attribute relations.** The second step is to identify *sufficient* conditions for the modeled goal of high availability. This is where the probabilistic framework is necessary. Statistical studies, professional expertise, literature sources, or rule-of-thumb reasoning can provide modelers with clues as how to model these relations, setting the CPT values. Even when the basis for these relations is only the tacit knowledge of professionals, it is still a tangible benefit to explicitly model this knowledge, thus subjecting it to objective analysis, rather than letting it remain informally hidden, defying critical scrutiny.

   Example. In the case of availability, there is often statistics available on properties such as mean time to failure (MTTF), mean time to repair (MTTR), etc. Such knowledge can then relatively easily be encoded in the architecture models. In the absence of hard data, it is possible to use a set of standard relations for modeling of probabilistic dependencies. One set of such relations is *noisy* versions of the AND and OR logical relations (Johnson and Ekstedt, 2007), another set, tailored specifically for architecture analysis, is presented in Gustafsson, Franke, Höök & Johnson, 2008.

Having thus described this five-step-process in the abstract, we now proceed to give a concrete example of its use.

APPLICATION EXAMPLE

This section will provide an example of step 5 in the process as well as how the information model for availability analysis can be applied. The example shows involves the large power distribution enterprise LIAM Energy, who's CIO has initiated the implementation of an automatic meter reading (AMR) system. This project is initiated mainly due to legislative requirements on billing based on actual consumption, but also to improve customer relations by providing more accurate invoices. Since this new systems adds to the complexity of LIAM Energy's application architecture the CIO is concerned about its effects on the *business service* for *billing*, which is known to impact customer relations. More particularly the CIO is concerned with the AMR project's impact on the *availability* of this *business service*. Having previously performed manual meter readings and partly manual energy consumption calculations the CIO hopes that this new AMR system will increase the availability of the business service. However, in order to facilitate a comparison both scenarios (with and without AMR functionality) are detailed.

With step 1-4 of the step-by-step approach completed (cf. Figure 4) the fifth step that formalizes the augmented information model is detailed here. This requires formalizing the characteristics of the attribute relationships in the information model. This can either be done for the information model (i.e. on the meta level), or for an instantiated information model. LIAM Energy formalizes the generic characteristics of the attribute relationships in the information model.

LIAM Energy uses a *deterministic AND* to represent the dependency between the *availability* of the *business process* and the *business service*, the compositions of *business processes,* and the dependencies between *business application* and *business application service*. This means that the services are fully dependent of the availability of the processes and the applications respectively.

All other dependencies are represented by a *noisy AND*, with varying noise. Between the *availability* of the *business process* and *the business application service* the noise is assessed to 0.2, indicating that the processes on average be able to run in 20% of the cases when the business application service is unavailable. Similarly, the noise in the *infrastructure service availability* to *business application availability* is set to 0.1. Finally the influence of the *device availability* on the *infrastructure service availability* is found to be 0.05. Table 1 show an example of the noisy AND dependency with the noise parameter set to 0.2 for the attribute availability of Calculate Energy Consumption in Figure 4.

Table 1 - A noisy AND conditional probability distribution with noise 0.2 for how the availability of *Calculate Energy Consumption* depends on the availability of *Get Meter Reading* and *Get Previous Customer Consumption*.

Get_Meter_Reading.Availability	Available		Unavailable	
Get_Previous_Custuomer_Consumption.Availability	Available	Unavailable	Available	Unavailable
Calculate_Energy_Consumption.Available	1	0,2	0,2	0,2
Calculate_Energy_Consumption.Unavailable	0	0,8	0,8	0,8

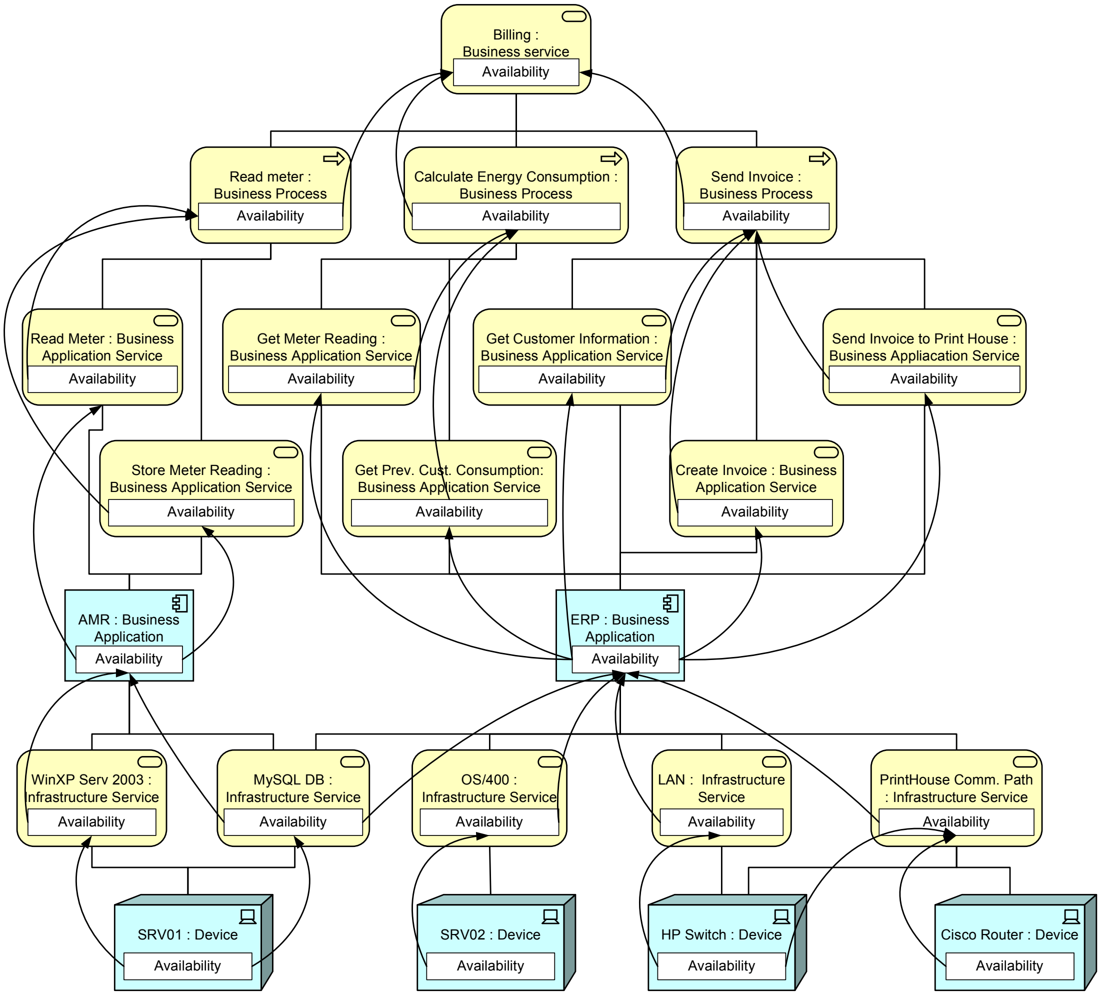

Figure 4 - The instantiated information model for one of the scenarios, containing AMR functionality.

With the dependencies formalized in the information model layer the information necessary to quantify attributes of instantiated information models is provided. Figure 4 shows the instantiation of the information model in Figure 3 for the scenario with an AMR system. In order to perform the availability analysis LIAM Energy also need to collect data on the availability of devices in their architecture. These are found in various SLAs as well as through interviews with maintenance staff and vendors.

The instantiated information model in Figure 4 constitutes a BBN through the attributes and their relationships. This BBN is shown in Figure 5. Based on the availability figures collected for the scenario it is possible to derive a value for the availability of the business service. As seen in Figure 5 the availability of the billing business service was assessed to a availability of 95 percent.

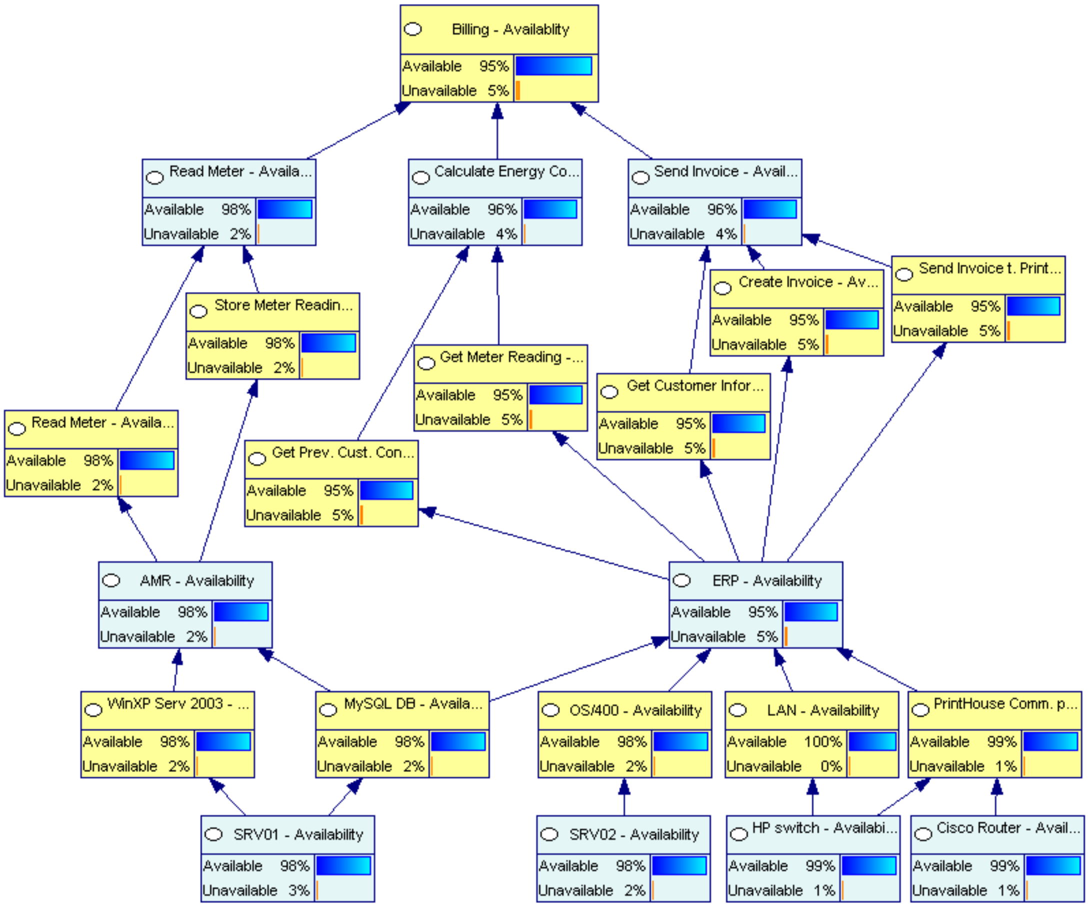

Figure 5 - The corresponding BBN showing the inferred values

CONCLUSION

In this paper, it has been presented an approach for performing EA analysis based on information models augmented with attribute relationships, and how these models can be constructed step-by-step and utilized for EA analysis. It is also shown how PRMs can be used to formalize these augmented information models so that quantitative analysis can be performed. This approach is capable of represent the availability model presented in Lankes, 2008 and to perform quantitative analysis of instantiations of such a model.

Currently, tool support for performing EA analysis based on augmented information models complemented by BBNs is still at an early stage – a corresponding prototypic tool is presented in Ekstedt, Franke, Johnson, Lagerström, Sommestad, Ullberg, and Buschle, 2009 and Johnson et al., 2007. This tool currently supports visualizations of models and analysis results similar to the graph in Figure 4, but it is not capable to generate visualizations conforming to other viewpoints, like those in the Enterprise Architecture Management Pattern Catalog (Buckl et al., 2008). As the generic visualization approach presented in Buckl, Ernst, Lankes, Matthes, Schweda, and Wittenburg, 2007 has been implemented in another prototypic tool this points an interesting research direction. The tools could be integrated to complement EA analysis capabilities with techniques to create stakeholder specific visualizations of models and results.

Furthermore, BBNs while being powerful means of formalizing dependencies between properties, they should not be considered the single appropriate formalism for EA analysis. Other techniques exist, of which some have already been discussed in Johnson et al., 2007. BBNs have shown up as suitable means for a large number of computations, they are limited concerning the formalization of dynamics in an enterprise, where for example feedback loops want to be represented. For such relationships a different formalism would be needed, e.g. (stochastic) difference equations or System Dynamics models. This would require adaptation of the steps four and five in the model presented herein.

REFERENCES

1. Aier, S., Kurpjuweit, S., Schmitz, O., Schulz, J., Thomas, A. and Winter, R. (2008) An Engineering Approach to Enterprise Architecture Design and its application at a Financial Service Provider, in Peter Loos, Markus Nüttgens, Klaus Turowski, and Dirk Werth (Ed.) *Modellierung betrieblicher Informationssysteme (MobIS 2008)*, November, Saarbrücken, Germany, 115-130.
2. Buckl, S., Ernst, A., Lankes, J., Schneider, K., and Schweda, C. (2007) A Pattern based Approach for constructing Enterprise Architecture Management Information Models, in Andreas Oberweis, Christof Weinhardt, Henner Gimpel, Agnes Koschmider, Victor Pankratius, and Björn Schnitzler (Ed.) *8. Internationale Tagung Wirtschaftsinformatik*, Karlsruhe, Germany, 145-162.
3. Buckl, S., Ernst, A., Lankes, J., Matthes, F., Schweda, C., and Wittenburg, A. (2007) Generating Visualizations of Enterprise Architectures using Model Transformation (extended version), in: *Enterprise Modelling and Information Systems Architectures – An International Journal*, 2, 2, 3-13.
4. Buckl, S., Ernst, A., Lankes, J., and Matthes, F. (2008) *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008)*, Technische Universität München, Chair for Informatics 19, Technical Report TB 0801, Munich, Germany.
5. Buckl, S., Ernst, A., and Schweda, C.M. (2008) *An Extension to the Essential Meta-Object Facility (EMOF) for Specifying and Indicating Dependencies between Properties,* Technische Universität München, Chair for Informatics 19, Technical Report TB0802, Munich, Germany.
6. Ekstedt, M.; Franke, U.; Johnson, P.; Lagerström, R.; Sommestad, T; Ullberg, J; and Buschle, M. (2009) A Tool for Enterprise Architecture Analysis of Maintainability. to appear in: *Proceedings of the 13th European Conference on Software Maintenance and Reengineering*, Kaiserslautern, Germany.
7. Frank, U. (2002) Multi-Perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages, in *Proceedings of the 35th Annual Hawaii International Conference on System Science 35*, 1258-1267.
8. Frank, U., Heise, D., Kattenstroth, H., and Schauer, H. (2008) Designing and Utilising Business Indicator Systems within Enterprise Models – Outline of a Method, in: Peter Loos, Markus Nüttgens, Klaus Turowski, and Dirk Werth (Ed.) *Modellierung betrieblicher Informationssysteme (MobIS 2008)*, November, Saarbrücken, Germany, 89-105.
9. Frank, U. (2008) *MEMO Metamodelling Language (MML) and Language Architecture (revised version).* ICB Research Report, Universität Duisburg-Essen, Germany 2008.
10. Franke. U., Flores W. R., Johnson, P. (2009) Enterprise Architecture Dependency Analysis using Fault Trees and Bayesian Networks, in: *Proc. 42nd Annual Simulation Symposium (ANSS 2009)*, March, San Diego, USA, 209-216.
11. Franke, U., Höök, D., König, J., Lagerström, R., Närman, P., Ullberg, J., Gustafsson, P., and Ekstedt, M. (2009) EAF2 - A Framework for Categorizing Enterprise Architecture Frameworks, to appear in: *Proceedings of the 1st International Workshop on Enterprise Architecture Challenges and Responses*, Daegu, Korea, to appear.
12. Gertsbakh, I. (1989) *Statistical Reliability Theory*. Marcel Dekker, Inc, New York, USA.
13. Getoor, L., Friedman, N., Koller, D., Pfeffer, A., and Taskar, B. (2008) Probabilistic Relational Models, in Lise Getoor, and Ben Taskar (Ed.) *An Introduction to Statistical Relational Learning*. MIT Press.
14. Gustafsson, P., Franke, U., Höök, D., and Johnson, P. (2008) Quantifying IT impacts on Organizational Structure and Business Value with Extended Influence Diagrams, in *Lecture Notes in Business Information Processing*, 15, 138-152.
15. Holschke, O., Närman, P., Rocha-Flores, W., Eriksson, E., and Schönherr, M. (2008), Using Enterprise Architecture Models and Bayesian Belief Networks for Failure Impact Analysis, in *Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research (TEAR 2008)*, Sydney, Australia,
16. International Standardization Organization/International Electrotechnical Committee (2007), *ISO/IEC 42010:2007 – Systems and software engineering – Recommended practice for architectural description of software-intensive systems*.
17. Jonkers, H., Groenewegen, L., Bonsangue, M., and van Buuren, R. (2005), A language for Enterprise Modelling, in Marc Lankhorst (Ed.) *Enterprise Architecture at Work*, Springer, Berlin.
18. Jonkers, H. (2006) *Architecture Language Reference Manual (04-04-2006)*, ArchiMate D2.2.2b v4.1, Telematica Instituut (TI), Radbound Universiteit Nijmenge, Netherlands.
19. Johnson, P., Johansson, E., Sommestad, T., and Ullberg, J. (2007) A Tool for Enterprise Architecture Analysis, in *11th IEEE International Distributed Object Computing Conference (EDOC 2007)*, Annapolis, Maryland, USA, 142-156.

20. Johnson, P., Lagerström, R., Närman, P., and Simonsson, M. (2007) Enterprise Architecture Analysis with Extended Influence Diagrams, in *Information Systems Frontiers*, 9, 2, 163-180
21. Johnson, P. and Ekstedt, M. (2007) *Enterprise Architecture – Models and Analysis for Information Systems Decision Making*. Studentlitteratur, Stockholm, Sweden.
22. Krcmar, H. (2005) *Informationsmanagement*. Springer, Berlin.
23. Lankhorst, M. et al. (2005) *Enterprise Architecture at Work – Modelling, Communication, and Analysis.* Springer, Berlin, Heidelberg.
24. Lankes, J. (2008) *Metrics for Application Landscapes – Status Quo, Development, and a Case Study.* PhD thesis, Chair for Informatics 19, Technische Universität München, Munich, Germany.
25. Stachowiak, H. (1973) *Allgemeine Modelltheorie*, Springer, Wien.
26. van der Torre, L., Lankhorst, M., ter Doest, H., Campschroer, J., and Arbab. F. (2004) *Landscape Maps for Enterprise Architectures*. Technical report, Information Centre of Telematica Instituut, Enschede, Netherlands.
27. Winter, R. and Fischer, R. (2007) Essential Layers, Artifacts, and Dependencies of Enterprise Architecture, in *Journal of Enterprise Architecture*, 3, 2, 7-18.