Harnessing Multimodal Architectural Dependencies for IT-Business Alignment and Portfolio Optimization: A Statistical Approach

Zenon Mathews, Livio Kaeslin, and Bernhard Rytz

Abstract. With the growing importance of architectural considerations over the last decade, larger Enterprise Architecture DataBases (EADBs) have enjoyed accumulation of enterprise data over years. Enterprise data documenting IT-applications, interfaces, data ownerships, business function implementation and usage etc. have been modelled based on various underlying metamodels and using different EA-tools. However, harnessing this data for business benefit is often hard and widely performed manually using expert knowledge. We here propose a statistical method for harnessing the EADB knowledge to infer indirect dependencies between IT-applications and projects. Our approach provides key insights into otherwise unseen multimodal dependencies in the IT landscape and provides a quantifiable methodology for optimizing the IT development plan and portfolio management. We validate our method in the highly integrated IT landscape of the Swiss Federal Railways, one of the world-leading rail transportation providers, and show how business decision making can directly be supported using our approach.

1 Introduction

Over the last decade, Enterprise Architecture (EA) has provided transparence to IT landscapes and made the IT landscape of high-performance enterprises more manageable [1]. Nevertheless, much of the EA data accumulated over the years, documenting different architectural layers including business function, data, application and technology, often remain unused when making architectural decisions with business impacts [1, 2]. The highly complex application architecture with large

Zenon Mathews · Bernhard Rytz
Enterprise Architecture, SBB Informatik AG, Swiss Federal Railways
e-mail: {zenon.mathews,bernhard.rytz}@sbb.ch

Livio Kaeslin
SBB Passenger Traffic
e-mail: livio.kaeslin@sbb.ch

P.-J. Benghozi et al. (eds.), *Digital Enterprise Design & Management,*
Advances in Intelligent Systems and Computing 261,
DOI: 10.1007/978-3-319-04313-5_4, © Springer International Publishing Switzerland 2014

number of interfaces and a rich project portfolio in bigger enterprises prohibit the manual in-depth exploitation of the EA databases for the above use [10]. As a result, many architectural decisions and project portfolio optimizations are based on qualitative assessments and locally optimal expert suggestions. This approach, although very valuable, can subsequently lead to the development of IT solutions to possess undiscovered dependencies to other developments within the organization. In the long-term, this leads to functional and data redundancies, making the enterprise less and less agile with time.

Indirect repercussions of cancellation or postponement of a specific IT-project on other ongoing projects usually remain unknown. We provide a quantifiable measure to assess such indirect impacts and help optimizing the project portfolio from the enterprise perspective. We benchmark our approach in the Swiss Federal Railways (SBB, for the German Schweizerische Bundesbahnen) IT landscape and compare our results with expert assessments. Our results demonstrate the usefulness of the approach in a high-performance transport industry, that provides one of the densest and most punctual public transportation services world-wide.

IT-application portfolios of larger enterprises often have hundreds of applications with thousands of interfaces [10]. Conventional approaches of managing the high complexity in such application portfolios include the division of the landscape into separate functional domains [3, 4]. For example, the SBB application landscape consists of more than 1000 applications divided into several functional domains (figs.1, 2). Each domain covers a subset of applications similar in their (business) function. Each domain is managed by a domain architect, who drives the development plan of the domain and coordinates the different project efforts of the domain, considering both IT architectural principles and business perspectives. Domain expert knowledge is used to optimize the development plan over the coming years. For individual domains that are architectonically complex, domain knowledge itself is not always sufficient for a complete dependency analysis. Moreover, as individual domains are optimized mostly locally, inter-domain dependencies are only addressed in very obvious cases. This approach leads to dependency issues with the growing pace of beyond-domain integration of today's IT-landscapes.

A development plan considering the global view of the enterprise should therefore consider dependencies that go beyond domain boundaries. Furthermore, the so called what-if analysis is considered very helpful in early project stages for recognizing dependencies to other projects both inside and outside the given domain [1]. Portfolio optimization and financially sustainable development plans therefore need to apply methods that enable an enterprise-wide dependency analysis of projects. This is however a challenging task as dependencies in larger IT-application portfolios exist at different levels; e.g. the TOGAF reference metamodel [6] itself has relationships between the business, application, data and technology architectures. With increasing importance of enterprise architecture modelling for sustainable growth of the enterprise-wide IT-systems, the maturity of available information about such dependencies has grown remarkably in the recent years [1]. Nevertheless, effective mathematical techniques are rarely used to harness such multimodal dependency information.

We here propose a statistical sampling technique, the Markov-Chain-Monte-Carlo Gibbs sampling [11], to infer dependencies in the IT-landscape and allow project portfolio planners to recognize and align dependent projects. Our approach can harness multimodal dependencies (i.e. dependencies at multiple architectural levels) in the information system landscape and can also cope with the common cyclic dependency issue. We apply and demonstrate the usefulness of the method in the SBB IT-landscape. Our proposal can directly help cut project costs and increase business effectiveness.

1.1 Related Work

EA allows the management of information in enterprises with large information systems. EA is often model-based and diagrams represent systems and their interdependencies. In the recent years, formal analysis of such EA models have been shown to be very powerful in many subdomains such as security, portfolio optimization and ensuring IT agility [2]. Strong proposals for statistical approaches to dependency inference in EA models have already been proposed [5, 9, 8]. Such approaches mostly consider unimodal dependency scenarios in information systems. Moreover, the assumption of acyclic dependencies often cannot be fulfilled in many large integrated information systems as inference in Bayesian networks does not allow cyclic dependencies [5].

Our approach, on the contrary, is more powerful in scenarios with dependencies at multiple levels of the information system (application, business function, data and technology). Moreover, the Gibbs sampling technique for Markov Chain Monte Carlo allows for the existence of cyclic dependencies in the EA model [11].

1.2 The SBB Information Systems Landscape

The SBB is the national railway company of Switzerland, which is a special stock corporation with shares exclusively owned by the Swiss Confederation. The SBB has five major divisions: passenger traffic, freight, real-estate, infrastructure and core services. With almost 1 million passengers per day, one of the densest national railway networks, and almost 90% of arrivals within 3 minutes of scheduled arrival time, the performance index of the SBB is known to be world-leading.

The SBB IT landscape consists of above 1000 applications, thousands of interfaces between applications and about 170 IT projects per year that constantly develop the application landscape (see fig. 1, 2). Owing to this dynamic development scenario and to enable business agility, the team of architecture and quality ensures the design and update of development plans. This process is performed several times per year for each of the 13 IT domains in consultation with the IT finance department.

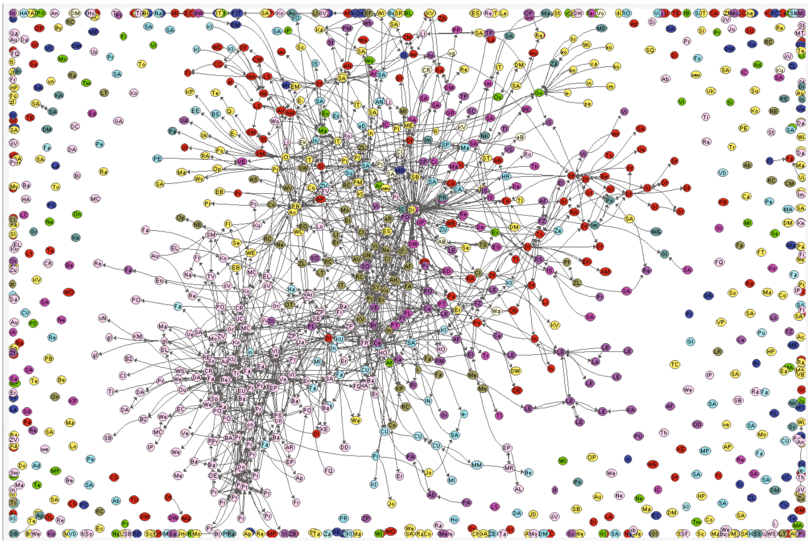

Fig. 1 SBB IT Applications (color coded nodes for IT domains) and corresponding interfaces (directed edges)

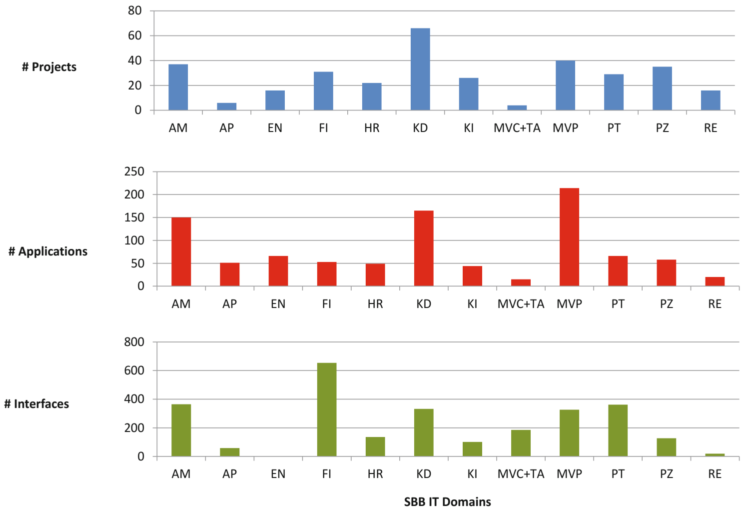

Fig. 2 SBB IT domains at a glance

2 Methods

2.1 From Metamodel to Dependency Graphs

SBB employs the metamodel shown in figure 3. We use the following entities and relationships from the metamodel (translated from German):

- "Anwendung": Application: a high level logical representation of a collection of software applications, usually implementing a specific business functionality. We consider interfaces between applications as dependencies (interfaces are modelled as messages "Nachricht")
- "Projekt": IT project: an approved IT project from the IT portfolio that has a direct impact on one or more applications, which we consider as dependencies.
- "Geschäftsfunktion": business function: an application either implements a business function or uses it. Implementation and usage of business functions are modeled as dependencies in our analysis.
- "Plattform": platform: an application is run on a platform. We consider an application as dependent on the platform it runs.
- "Geschäftsdatentyp": business data type: an application can own or use one or several business data type. Also this data ownership or usage are used to model dependencies in our analysis.

We therefore consider multimodal dependencies at the data level, business functionality level and at the platform level. By considering these multimodal dependencies in the enterprise database, we form the so called *dependency graph*, as shown in figure 4. Such dependency graphs are used by the Markov Chain Monte Carlo algorithm to deduce indirect dependencies between applications and IT-projects. Such indirect dependencies are useful for coordinating the development plan and making use of synergies. Indirect dependencies are often hidden in early project phases and cause much effort and cost in later phases as the projects have already advanced further.

2.2 The Markov Chain Monte Carlo (MCMC) Gibbs Sampler

Let S be a finite set of states $S = s_1, s_2, \dots, s_n$. A finite, discrete time, first order Markov Chain is a stochastic process, which is a sequence of random variables $X = ((X_t)_{t=0,1,\dots}$ with values in S and with the property:

$$\begin{aligned} P(X_{t+1} = s_{j_{t+1}} \mid X_t = s_{j_t}, X_{t-1} = s_{j_{t-1}}, \dots, X_0 = s_{j_0}) \\ = P(X_{t+1} = s_{j_{t+1}} \mid X_t = s_{j_t}) \end{aligned} \tag{1}$$

The *transition probability* is the probability that the random variable with time-index t changes from state s_i to state s_j. This property means that the probability to reach a certain state in the next step only depends on the actual state and is unaffected by the anterior states. This probability is p_{ij} and is defined as

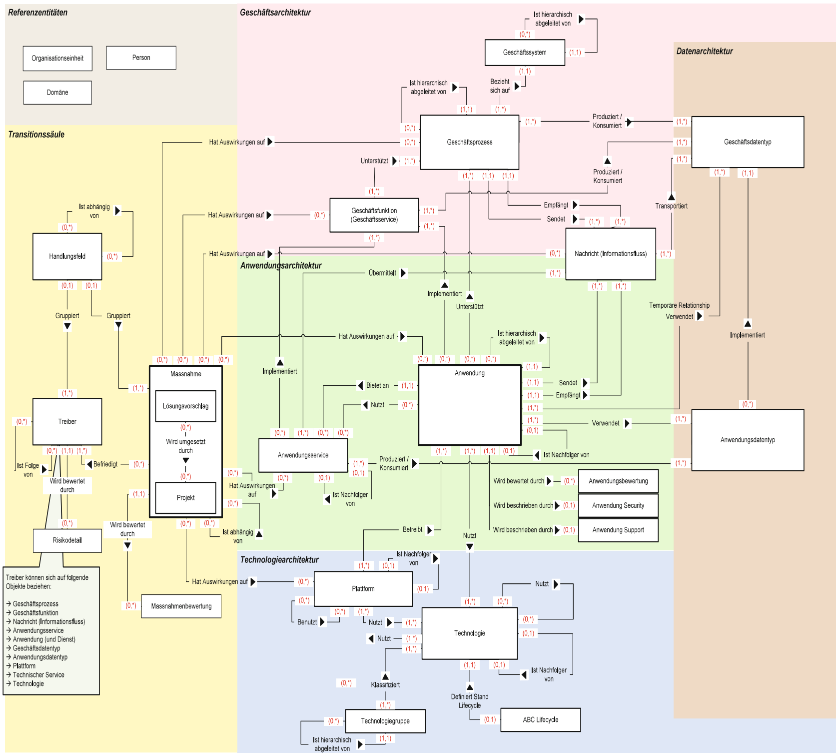

Fig. 3 SBB IT Metamodel: We use the relationships between applications (Anwendung), Projects (Projekt), platforms (Plattform), business data type (Geschäftsdatentyp) and business functions (Geschäftsfunktion) in the MCMC algorithm.

$$p_{ij}(t) := P(X_t = s_j \mid X_{t-1} = s_i) \tag{2}$$

A Markov chain is *homogeneous* if all transition probabilities are independent of time. Often a *transition matrix* is used to indicate the transitions probabilities from every state to every other.

The probabilities of the initial states are given by the initial state probability vector. The stationary distribution is a probability vector that describes the state probability vector of the sought solution of the problem. Convergence to the stationary distribution is achieved if the following conditions are fulfilled:

• *irreducibility*: a Markov Chain is irreducible when each state can be reached from any other state in a finite number of steps

• *aperiodicity*: a Markov Chain is aperiodic if no state is reached only periodically

Convergence of a homogeneous, irreducible and aperiodic Markov Chain is mathematically guaranteed and there is just one stationary distribution to which the chain converges.

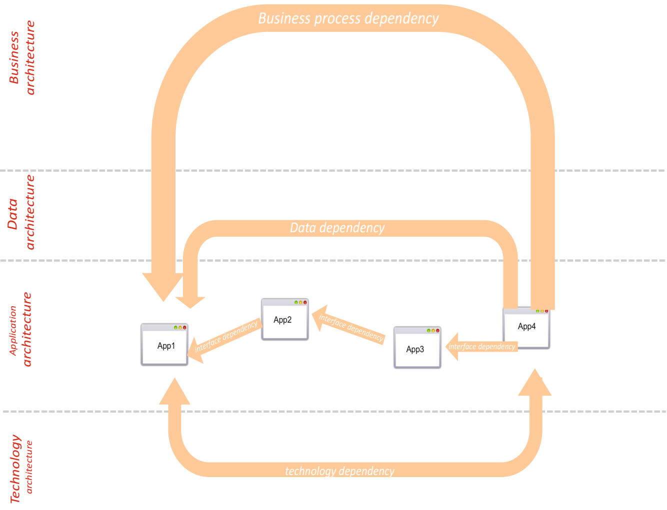

Fig. 4 Schematic of the used multimodal dependency graph constructed from the SBB EA-model

In our case, we consider the state of the dependency graph of the IT-landscape in each Markov chain step. The state of the dependency graph is simply the joint states of each application in the landscape. Each application can thereby have one of the two following states:

- A, meaning that the application undergoes a change
- B, meaning that the application does not undergo a change

Given the dependency graph G, we define the probability that an application will have the state A in the next step as follows:

$$P(State = A \mid G) := \frac{\#altered\ in-ports}{\#in-ports} * c \tag{3}$$

where, a constant $c(0 < c < 1)$ ensures the irreducibility of the chain. An *altered* $in-port$ thereby means an interface/port leading from an application that is in state A to the current application. Further, it can also be shown that the above defined Markov Chain is aperiodic. We use the Gibbs Sampling, a so called Monte Carlo method [11], to find the stationary distribution of the above Markov Chain.

Let A_k^i be the status of the $k-th$ application in the $i-th$ iteration step. Given the initial state of all m applications and their states in the $i-th$ iteration step:

$$A^i = (A_1^i, A_2^i, \ldots A_m^i) \tag{4}$$

the Gibbs method uses the state probability as in equation 3, and samples the next state of each application for the next step as

$$A^{i+1} \sim P(A_j \mid A_1^i, A_2^i, \ldots, A_{j-1}^i, A_{j+1}^i, \ldots, A_m^i) \tag{5}$$

until the stop criterion is reached. The stop criterion is defined by ε, which is a small enough euclidean value, that indicates the almost steady average state of the dependency graph. It is important to note that the computed probabilities here represent the probability of dependence between entities (applications and projects), and not for example how dependent an entity is dependent of another one.

2.3 Tools

The ©MEGA Suite is a widely recognized multiuser tool for enterprise architecture modelling, governance, risk and compliance tasks. The SBB-IT employs the ©MEGA tool for EA modelling. All domain architects model the attributes of the applications, projects, interfaces (ports) between applications of their corresponding domains respecting the SBB-wide metamodel and the ©MEGA tool. We here use the ©MEGA repository for the dependency analysis. We use the prefuse visualization toolkit [7] to visualize the dependency graph of the IT-landscape and the indirect dependencies between several IT projects.

3 Results

3.1 Project Indirect Impacts in the Application Landscape

We consider a single IT undertaking of the SBB IT, removing a legacy applications from the SBB IT landscape. The direct impacts are currently not yet documented in the EA-DB but are widely known by the domain experts. Nevertheless, the indirect impacts of the project are crucial yet unknown. We use the proposed MCMC algorithm to compute the indirectly impacted applications of this undertaking. The results of this "what-if" analysis are shown in figure 5. (the considered project is 'SYFA-Ablösung'). This result is invaluable for the project and the domain architect, and can be used as a guide when coordinating the development plan of the domain. Analysis with such pseudo-projects (i.e. projects that in reality do not exist in the portfolio) can be easily performed using our tool, to analyze potential impacts in the application landscape of a change to a particular application.

Currently, following an strategic IT decision, the operation of the Mainframe host platform is being relocated from a local provider to foreign provider. A major concern during this relocation project was that iterative function calls from non-host applications to host applications could cause prohibitively high round trip times, which subsequently can be unacceptable to critical applications. As there are numerous applications running on the host platform, with large number of data interfaces to non-host applications, a manual inspection of the potentially impacted

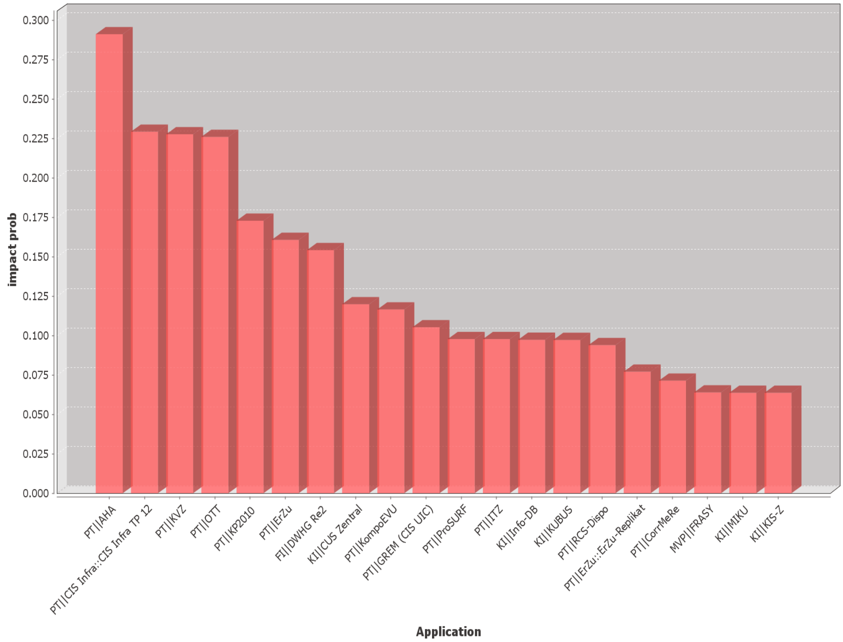

Fig. 5 What-if analysis: indirectly impacted IT-applications of a potential IT project

applications was not possible (thousands of potential candidates). We applied our tool to aid solving this problem. The direct project impacts were applied to the host platform, that was to be relocated. After this we analysed the indirect impacts in the application landscape. The 30 most probable indirectly impacted applications are shown in fig. 6. Our results could help platform architects and application managers to look into specific interfaces and perform tests to check the round trip times. It turns out that interfaces were found using our tool, which were previously not even considered and potentially could be critical after the relocation of the host.

3.2 Inter-Project Dependencies

Going beyond indirectly impacted applications, we are also interested in dependencies between projects. To infer such dependencies, we run the MCMC algorithm pairwise for each project-project pair. This allows to infer the highly probable dependencies as discussed in section 2.2. As a result of such pairwise comparison of projects and subsequent thresholding of dependency probabilities, we can compute the probabilistically highest dependencies between projects. Note that such highly probable dependencies can occur beyond domain boundaries, making the conventional domain oriented development plans insufficient.

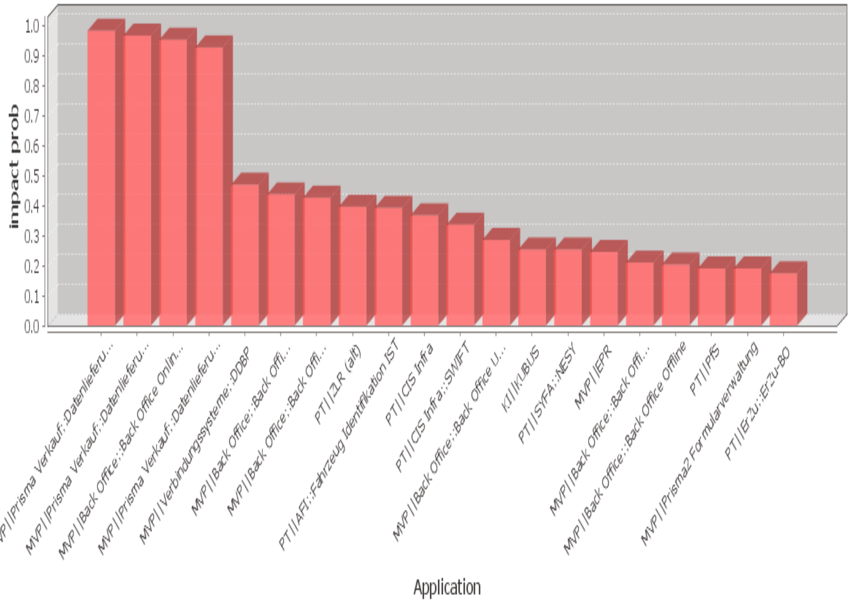

Fig. 6 Indirectly impacted applications of the host relocation project

Figure 7 illustrates the probable dependencies of the pseudo-project, which would remove a legacy application from the SBB-IT landscape. Such what-if analysis are highly valid not only for planning the removal of highly critical and integrated applications but also when introducing new projects into a rich portfolio. In figure 7, the pseudo-project is the filled circle in the center, the concentric circles indicate probability bounds and the colored filled circles indicate other real IT projects. The closer the projects to the pseudo project, the more probable their dependencies. Note that projects from a wide range of domains might be affected by the removal of the above mentioned legacy system. The shaded triangles indicate the common application to all projects inside the triangle, with indirect project impacts on that application.

4 Discussion

Our results serve as an entry point for further investigation of project dependencies and indirect impacts of projects in the IT-landscape. In our experience, the added value has been received very well by the domain experts. We have also compared the results of our approach with expert knowledge to evaluate the quality of the dependency analysis. Our approach could thereby not only confirm known dependencies between projects, but also discover unseen dependencies and indirect impacts.

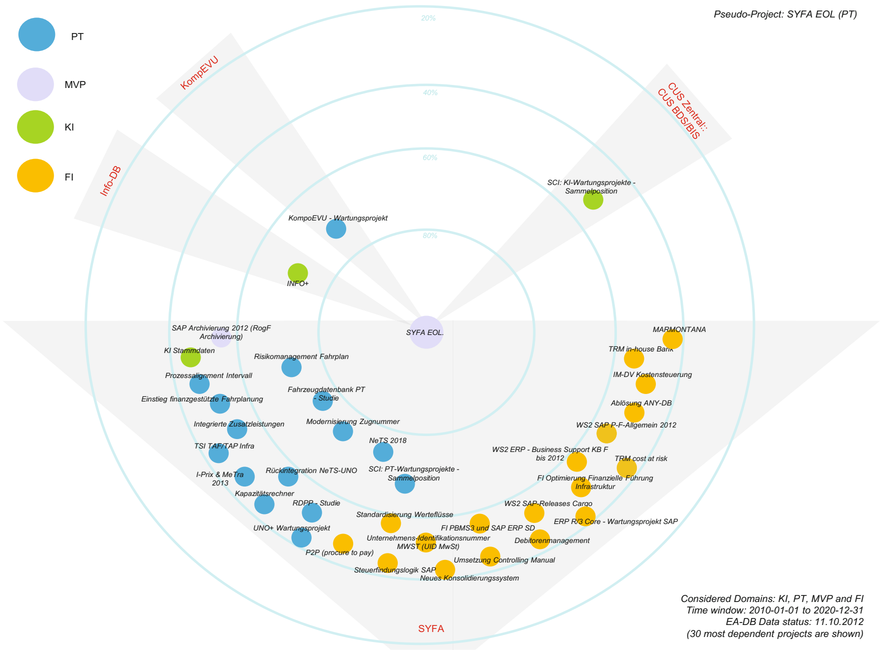

Fig. 7 Using a pseudo-project to estimate the impacts of an application EOL on ongoing IT-projects

Our approach is currently being tested at SBB-IT as an analysis tool to aid domain experts and portfolio planners with the development plan. Our proposal overcomes the restrictions of earlier approaches based on Bayesian inference, where only uni-modal dependencies in acyclic dependency graphs could be used. Inclusion of the approach in formal development planning procedure is currently being considered. Furthermore, we are currently investigating the application of the approach in the software change management process, where changes to the current IT-landscape can have indirect impacts. Today, the dependencies between changes are assessed merely using expert knowledge.

Acknowledgments. The authors would like to thank Hans-Jakob Gfeller, Peter Moosmann, Jakob Erber, Stefan Meichtry, Yannis Baillet, Christoph Trutmann, Ugur Asuroglu, Peter Hug, Rudolf Gysi, Oliver Hofer and Hans-Peter Weiss (all at SBB Informatik and SBB Passenger Traffic) for valuable suggestions and discussions.

References

1. Ahlemann, F., Stettiner, E., Messerschmidt, M., Legner, C.: Strategic Enterprise Architecture Management: Challenges, Best Practices, and Future Developments. In: Management for Professionals. Springer (2012)
2. Beyer, M., Lovelock, J.D., Sommer, D., Adrian, M.: Big data drives rapid changes in infrastructure and 232 billion in it spending through 2016. Gartner Research (2012)
3. Evans, E.: Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley (2004)
4. Fowler, M.: Patterns of Enterprise Application Architecture. Addison-Wesley Longman Publishing Co., Inc., Boston (2002)
5. Franke, U., Flores, W.R., Johnson, P.: Enterprise architecture dependency analysis using fault trees and bayesian networks. In: Proceedings of the 2009 Spring Simulation Multiconference, SpringSim 2009, pp. 55:1–55:8. Society for Computer Simulation International, San Diego (2009)
6. Haren, V.: Togaf Version 9.1. Van Haren Publishing Series. Bernan Assoc. (2011)
7. Heer, J., Card, S.K., Landay, J.A.: Prefuse: a toolkit for interactive information visualization. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2005, pp. 421–430. ACM, New York (2005)
8. Johnson, P., Lagerström, R., Närman, P., Simonsson, M.: Enterprise architecture analysis with extended influence diagrams. Information Systems Frontiers 9(2-3), 163–180 (2007)
9. Lagerstrom, R., Johnson, P.: Using architectural models to predict the maintainability of enterprise systems. In: Proceedings of the 2008 12th European Conference on Software Maintenance and Reengineering, CSMR 2008, pp. 248–252. IEEE Computer Society, Washington, DC (2008)
10. Sessions, R.: Simple Architectures for Complex Enterprises. Best Practices. Microsoft Press (2008)
11. Walsh, B.: Markov Chain Monte Carlo and Gibbs Sampling. Lecture Notes for EEB 581 (2004)