Towards an Expert System for Identifying and Reducing Unnecessary Complexity of IT Architectures

Kenny Wehling[1] Ina Schaefer[2]

Abstract: Over the course of time, IT architectures in business environments grow and often include a range of different architectural variants. This causes various redundancies and a high level of IT complexity leading to higher costs and increased effort for evolving and maintaining the entire IT landscape. In this paper, we propose a concept for a rule based system that analyzes the variability within IT architectures automatically and supports experts in identifying and reducing the architecture's complexity by eliminating unnecessary architectural variants.

Keywords: IT architecture; unnecessary IT complexity; unnecessary variability; expert system

1 Introduction

In many companies, IT architectures grow over years or even decades [Du12]. As developers often refrain from changing productive systems, arising requirements often lead to new solutions although architectural elements with a similar purpose already exist [Ha10]. Moreover, implementation of new technologies and integration of other software systems (i.e., after a company acquisition), as well as uncoordinated change initiatives seldom take into account the entire IT landscape, but only parts of it [Ah12, Ha10]. As a consequence, IT architectures often include a wide range of different architectural variants, which results in a high complexity level accompanied by increasing costs, a reduced adaptability and higher effort for evolving and maintaining the entire IT landscape [Ah12, Sc16]. To alleviate these problems, experts have to reduce unnecessary IT complexity which does not originate from business goals and requirements but f78rom unintended effects of evolving and maintaining IT architectures [Br87]. Although there are different approaches to measure IT complexity (i.e., [SM03, SWK13]), to the best of our knowledge, there is no approach to identify and reduce unnecessary IT complexity that is induced by unnecessary architectural variants. Hence, it is a tedious and time consuming challenge for experts to cope with grown IT architectures.

To solve these challenges, the goal of our approach is to automatically identify variants within IT architectures. Based on this, we propose a concept for an expert system that

[1] Volkswagen AG, Letterbox 011/1802, 38436 Wolfsburg, Germany, kenny.wehling@volkswagen.de
[2] TU Braunschweig, Muehlenpfordtstrasse 23, 38106 Braunschweig, Germany, i.schaefer@tu-braunschweig.de

doi:10.18420/in2017_152

derives recommendations for eliminating unnecessary architectural elements in order to support experts in identifying and reducing unnecessary IT complexity.

2 Background

For a better understanding of our approach, we provide further information on *IT architecture*, *IT complexity* and *expert system*. As we plan to realize an automatic analysis of architectural variants, we give a brief description of *variability mining*. This is a technique from the software product line (SPL) domain, which focuses on development of families of software systems with common and variying parts [PBvdL05].

IT Architecture: IT architectures are the logical organization of applications, data and infrastructure technologies [Ro03]. They can be described in form of abstract models by using architecture frameworks, such as *The Open Group Architecture Framework (TOGAF)*, which introduces different layers. The *application architecture (AA)* layer describes the structure and interaction of applications, the *data architecture (DA)* layer contains major types and sources of data and the *technology architecture (TA)* layer includes the logical and physical components of the infrastructure. These architectures are often modelled by means of different views representing architectural elements and their relations from various perspectives [Th11]. Our approach utilizes such view models and focuses on TAs and AAs.

IT Complexity: To determine the complexity of such architectures, several approaches have been devised. These available methods are categorized by the *complexity cube* [SZM14]. As our appoach focuses on eliminating unnecessary architectural variants, we take structural complexity into account and do not consider behavioral complexity. The following measurement of Schütz et al. fits our concept best and describes *complexity (C)* as the tuple of *numbers (N)* and *heterogeneity (H)* that can be measured for *things (T)* and *relations (R)*: $C_Y = (N_Y, H_Y),\ \text{with } Y \in \{T, R\}$. The number of elements can be determined by counting the considered artifacts within an architecture. To determine heterogeneity, the Entropy Measure (EM) has to be utilized [SWK13].

Variability Mining: In the context of the SPL domain, *variability* can be described as the choice of different artifacts for a specific purpose with a particular relation to other variants [PBvdL05]. Mining techniques can be utilized to identify such variability and different *variability relations* between a set of related artifacts. These relations show *mandatory* parts (i.e., common to all variants), *alternative* parts (i.e., mutually exclusive across variants), and *optional* parts (i.e., only part of certain variants). This variability information can be stored in *150% models* [RC12, Wi16] comprising all analyzed artifacts, their variabiliy relations and which variants they originate from.

Expert System: As part of artificial intelligence in computer science, expert systems simulate the reasoning and conclusions of human experts based on their domain knowledge and experience within a specific problem domain. Major components of an expert system are the *knowledge base*, which includes domain knowledge in form of rules, the *working memory*, which contains problem specific facts, and the *inference engine*, which derives new information from the problem-specific facts by applying particular rules [Du02, SK93].

3 Concept

To identify and reduce complexity that is induced by unnecessary variability within related IT architectures, we propose the concept illustrated in Figure 1. It describes an analyzing and restructuring process by means of an expert system. This system utilizes manageable decision rules for general restructuring strategies and various analyzing techniques to provide users with reasonable suggestions for decision making in order to reduce structural complexity.

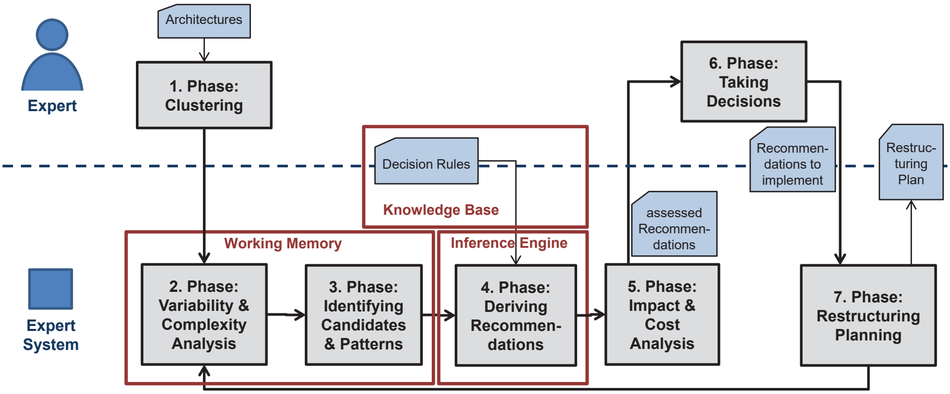

Fig. 1: Concept for Reducing Unnecessary Complexity of IT Architectures

The *first phase* focuses on clustering of the input data to build sets of related TAs or AAs with a similar purpose of use. This step has to be done manually as the purpose of use has to be determined by an expert for a business relevant view on the architectures, which requires business and domain knowledge. The *second phase* analyzes the variability and complexity within a given cluster of architectures. For this purpose, we adapt the variability mining technique from the SPL domain. Based on the resulting variability information, the *third phase* compares elements on the basis of their functional types and identifies all elements as candidates that might be part of unnecessary variability. Furthermore, it analyzes combinations of architectural elements to find frequently used patterns. The first three phases provide required data for the expert system in the working memory. In the *fourth phase*, the inference engine utilizes a decision method to derive recommendations for each candidate by applying decision rules from the knowledge base. These recommendations provide a first overview of necessary and unnecessary architectural elements within the regarded cluster. To use these recommendations for decision making, they have to be assessed in the *fifth phase*, which analyzes possible impacts of each recommendation on the variability and complexity of the given cluster as well as cost benefits. In the *sixth phase*, experts decide which of the assessed recommendations will be implemented based on their domain knowledge. Finally, the *seventh phase* analyzes required restructuring operations for each architecture, such as *removing* or *replacing* of an artifact, to eliminate unnecessary elements in accordance with the expert decisions. A list of all required restructuring operations per architecture in appropriate order is given to the experts in form of a restructuring plan.

4 Progress and Evaluation

In [We16], we proposed a *first concept* for reducing IT complexity of related architectures by using variability information to identify standard variants. In [We17], we introduced a *manual approach* to support experts in identifying and reducing unnecessary IT complexity caused by architectural variants. It provides a manual variability analyzing technique for IT architectures and consists of a five step decision process, which covers the 1st, 2nd, 3rd, 6th, and 7th phase of our concept (cf. section 3). The priority goal was to analyze the feasibility of our approach for reducing IT complexity by identifying and removing unnecessary variability. By means of a case study, we showed that our approach is feasable for clusters of architectures (A, B, C) with different variability levels (low, medium, high). As seen in Table 1, we were able to reduce the complexity of regarded clusters by eliminating unnecessary architectural variants. We reduced the number of different artifacts (N) within a cluster by at least -20% as well as their heterogeneity (H) by at least -23%. On an avarage, we were able to reduce N by -24% and H by -30%. The case study also revealed that our approach requires significant manual effort and only scales by means of an automated solution. Thus, our major goal is to automate our approach as illustrated in section 3.

	N_1	H_1	N_2	H_2	ΔN	ΔH
A) Low Variability	5	0.398	4	0.231	-1 (-20%)	-0.167 (-42%)
B) Medium Variability	7	0.778	5	0.455	-2 (-29%)	-0.323 (-42%)
C) High Variability	13	1.292	10	1.031	-3 (-23%)	-0.261 (-23%)
Total	**25**	**0.823**	**19**	**0.572**	**-6 (-24%)**	**-0.251 (-30%)**

N = number *H* = heterogeneity *1/2* = before/after restructuring

Tab. 1: Case Study Results [We17]

Currently, we are working on the automation of the 2nd, 3rd and 4th phase of our concept (cf. section 3). We design and develop appropriate algorithms for the automatic variability analysis, the identification of candidates, the pattern analysis and the derivation of recommendations. For this purpose, we utilize and adapt concepts of variability mining and rule-based expert systems. In collaboration with domain experts, we devise suitable decision rules. To evaluate correctness of our automatic approach, we will conduct expert interviews. We also plan to automate the 5th and the 7th phase of our concept. The focus of the 5th phase will be the assessment of each recommendation by a metric that involves the results of an impact analysis and a cost-benefit-analysis. To identify the particular impact of a recommendation, we have to consider the variability and complexity of the regarded cluster after its implementation. For the cost-benefit-analysis we take into account restructuring costs as well as cost benefits after restructuring. The emphasis of the 7th phase is the restructuring of the architectures in regarded cluster. We have to devise algorithm to identify required operations and to put them in a reasonable order. We plan to realize both phases by utilizing and adapting delta analysis techniques from the SPL domain [Sc10]. To evaluate correctness, scalability and runtime of our approach, we will conduct a case study on real-world scenarios from our industry partner with participation of experts.

5 Related Work

Schütz et al. [SWG13] propose a set of seven design principles for IT architectures which consider complexity and other aspects, such as end-user-acceptance and data quality. Although their definition of design principles is interesting for proactive development of standard artifacts, it does not consider variability within architectures. Schneider [Sc16] introduce a diversity framework for managing variability of TAs. It describes the variety, disparity, and balance of concepts as well as the variation of individuals. To determine these different aspects of diversity, the framework provides several definitions and single methods. Aleatrati Khosroshahi et al. [Al16] describe causes and consequences of application portfolio complexity. To reduce such complexity of AAs, they propose solution strategies driven by domain experts. While these approaches allow reasoning about different variability and complexity aspects of IT architectures, the authors do not provide methods or algorithms to analyze the variability of given architectures automatically. In contrast, our approach realizes an automated technique to identify variability of related architectures and to derive specific recommendations for restructuring to reduce the architecture's complexity.

6 Conclusion

Unnecessary variants within IT architectures lead to a high level of IT complexity, which results in increased costs and higher effort for evolving and maintaining the entire IT landscape. In this paper, we propose a concept for a rule-based expert system that is able to analyze the variability of related architectures automatically and to derive appropriate recommendations for experts to reduce IT complexity by eliminating unnecessary variants. The results from a case study show the feasibility of our manual approach and outline the need for an automatization. In future work, we plan to automate and to further evaluate our approach.

References

[Ah12] Ahlemann, Frederik; Stettiner, Eric; Messerschmidt, Marcus; Legner, Christine: Strategic Enterprise Architecture Management – Challenges, Best Practices, and Future Developments. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Al16] Aleatrati Khosroshahi, Pouya; Beese, Jannis; Matthes, Florian; Winter, Robert: Causes and Consequences of Application Portfolio Complexity – An Exploratory Study. In (Horkoff, Jennifer; Jeusfeld, Manfred A.; Persson, Anne, eds): (PoEM 2016) The Practice of Enterprise Modeling. volume 267, Springer International Publishing, Cham, pp. 11–25, 2016.

[Br87] Brooks: No Silver Bullet Essence and Accidents of Software Engineering. Computer, 20(4):10–19, 1987.

[Du02] Durkin, John: Tools and Applications. In (Leondes, Cornelius T., ed.): Expert systems, pp. 23–53. Academic Press, San Diego, 2002.

[Du12] Durdik, Z.; Klatt, B.; Koziolek, H.; Krogmann, K.; Stammel, J.; Weiss, R.: Sustainability Guidelines for Long-Living Software Systems. In: Proc. of the Intl. Conference on Software Maintenance (ICSM). pp. 517–526, 2012.

[Ha10] Hanschke, Inge: Strategic IT Management – A Toolkit for Enterprise Architecture Management. Springer, 2010.

[PBvdL05] Pohl, Klaus; Böckle, Günter; van der Linden, Frank J.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer, 2005.

[RC12] Rubin, Julia; Chechik, Marsha: Combining Related Products into Product Lines. In: Proc. of the Intl. Conference on Fundamental Approaches to Software Engineering (FASE). volume 7212 of Lecture Notes in Computer Science. Springer, pp. 285–300, 2012.

[Ro03] Ross, Jeanne W.: , Creating a Strategic IT Architecture Competency: Learning in Stages, 2003.

[Sc10] Schaefer, Ina; Bettini, Lorenzo; Bono, Viviana; Damiani, Ferruccio; Tanzarella, Nico: Delta-Oriented Programming of Software Product Lines. In: Software Product Lines: Going Beyond. Lecture Notes in Computer Science. Springer, pp. 77–91, 2010.

[Sc16] Schneider, Alexander W.: , Decision Support for Application Landscape Diversity Management, 2016.

[SK93] Smith, Suzanne; Kandel, Abraham: Verification and validation of rule-based expert systems. CRC Press, Boca Raton, Fla, 1993.

[SM03] Schneberger, Scott L.; McLean, Ephraim R.: The complexity cross. Communications of the ACM, 46(9):216–225, 2003.

[SWG13] Schütz, Alexander; Widjaja, Thomas; Gregory, Robert W.: Escape from Winchester Mansion - Toward a Set of Design Principles to Master Complexity in IT Architectures. In: (ICIS 2013) International Conference on Information Systems. pp. 1–19, 2013.

[SWK13] Schütz, Alexander; Widjaja, Thomas; Kaiser, Jasmin: Complexity In Enterprise Architectures – Conceptualization and Introduction of a Measure From a System Theoretic Perspective. In: (ECIS 2013) European Conference on Information Systems. pp. 1–12, 2013.

[SZM14] Schneider, Alexander W.; Zec, Martin; Matthes, Florian: Adpting Notions of Complexity for Enterprise Architecture Management. In: (AMCIS 2014) Americas Conference on Information Systems. pp. 1–10, 2014.

[Th11] The Open Group: TOGAF Version 9.1. Van Haren Publishing, 2011.

[We16] Wehling, Kenny; Wille, David; Pluchator, Martin; Schaefer, Ina: Towards Reducing the Complexity of Enterprise Architectures by Identifying Standard Variants Using Variability Mining. In: 1. Automobil Symposium Wildau. pp. 37 – 43, 2016.

[We17] Wehling, Kenny; Wille, David; Seidl, Christoph; Schaefer, Ina: Decision Support for Reducing Unnecessary IT Complexity of Application Architectures. In: 2nd International Workshop on decision Making in Software ARCHitecture (MARCH). 2017.

[Wi16] Wille, D.; Schulze, S.; Seidl, C.; Schaefer, I.: Custom-Tailored Variability Mining for Block-Based Languages. In: Proc. of the Intl. Conference on Software Analysis, Evolution, and Reengineering (SANER). volume 1. IEEE, pp. 271–282, 2016.