A Method for Application Evaluations in Context of Enterprise Architecture

Jan Stefan Addicks
OFFIS - Institute for Information Technology
Escherweg 2
Oldenburg, Germany
addicks@offis.de

Hans-Jürgen Appelrath
University of Oldenburg
Ammerländer Heerstr. 114-118
Oldenburg, Germany
appelrath@informatik.uni-oldenburg.de

ABSTRACT

This contribution presents a method to evaluate business applications. The method allows for using artifacts of enterprise architectures. Artifacts like business processes or hardware can exert influence on the application's quality and thus have to be regarded. A central aspect is to modularize the method's basic components, which are key figures and their metrics. The modularization allows for flexible usage and customization to fit the heterogeneity of different organizations. To make the method more intuitive, a key figure is encapsulated by a fuzzy logic based component denoted as criterion. A criterion addresses a certain aspect to evaluate an application and allows for using linguistic terms to represent a key figure.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous

General Terms

Measurement, Management

Keywords

Enterprise Architecture, Application, Evaluation

1. INTRODUCTION

Organizations today have application landscapes that are complex and which continuously grow in complexity, due to the increasing number of heterogeneous interwoven applications. It is difficult for stakeholders, for instance the chief information officer (CIO), to see the big picture of the overall structure (cf. [12]); however, the application landscape (denoted as *landscape* in the following) is only a part of the enterprise architecture (EA) (cf. [27]). Since there is no common definition of EA in literature ([17], [25], and [27]), we present our understanding of EA for clarity:

An EA is a triple $\mathfrak{ea} = (\mathfrak{A}, \mathfrak{R}, \mathfrak{L})$ *in which* $\mathfrak{A}$ *is the set of all artifacts,* $\mathfrak{R}$ *is the set of all relations that connect exactly two artifacts, and* $\mathfrak{L}$ *is a set of layers (cf. [6]) that are used to partition the whole structure.*

Artifacts are for example applications, servers on which an application is installed, or business processes that are supported by an application. Winter et al. presented essential *layers* and artifacts for EA [27]. With this definition we distinguish between the aspect of "architecture of a system" and the "discipline of architecture" (similar to *architecture types* defined in [11]), and further refer to the former with the term EA; however, in literature, several definitions mix both aspects (cf. [17], [21], [24], and [26]). The combination of our EA definition and the aspect of its architectural discipline is referred to as *EA management* (EAM).

Acquisitions and mergers as well as uncoordinated projects often lead to unplanned landscapes with two or more applications having almost similar functionality. To reduce such kind of undesigned redundancy, IT consolidation projects can be performed that include comparisons of alternatives to identify those to be kept (cf. [1]); however, applications cannot be regarded as independent objects but must be seen as part of a whole. There are a lot of artifacts from all layers of the EA (the applications's enterprise context) that exert influence on an application's quality (cf. [1]). These *influence factors* vary from enterprise to enterprise. For instance, influence factors might be regulations, like Basel II, or new strategies from the management like the decision to introduce a service-oriented architecture (SOA) [9].

Due to the high number of influence factors as well as the heterogeneity of organizations, a method to evaluate applications must be flexible in order to be adoptable to each kind of organizational environment. The *Enterprise Architecture based Application Evaluation* (EA|AE) method presented in this contribution is based on key figures and their metrics, which are exchangeable elements of our approach. It is often not sensible to use a certain key figure by itself (cf. Section 3.1), therefore, two or more elementary key figures have to be combined. In order to combine key figures, we use *criteria* to encapsulate the details of coupling elementary key figures (see Section 4.1). Criteria are considered to represent indicators for more concrete questions, like *"How good is an application regarding its availability, its costs per year and its importance"*. One of the results of discussions with our partners from industry was that such questions are good starting point to introduce application evaluations within organizations.

Since classifying metrics according to terms like *good* or

SAC'10 March 22-26, 2010, Sierre, Switzerland.

bad is often vague, we integrate a fuzzy logic representation of key figures in our method (cf. Section 4.1). In order to support CIO's the presented method allows for aggregating key figures to an overall indicator of an application's quality. This aggregation is based on fuzzy logic control (cf. [2]).

The remainder of this contribution is structured as follows: In the next section, a brief overview of relevant research work is given. Section 3 presents foundations as well as requirements for the presented method. After that, in Section 4 our method is introduced. Before the last section summarizes this contribution and presents an outline of future research activities, a tool prototype and a case study are presented in section 5.

2. RELATED WORK

In software engineering, metrics are well known and commonly used (cf. [10]). However, those software metrics evaluate single software systems and do not regard the aspects of their environments. In one of our projects with a partner from industry, we will gather and consolidate existing software metrics, evaluate them for use in our coherent evaluation approach and adapt and incorporate them where appropriate.

As opposed to the topic of software metrics, relatively little work has been done in the area of landscape evaluations or evaluations of applications with consideration of the EA context. Lankes [19], for instance, presents an approach for landscape evaluations, which relies on metrics for failure propagation within a landscape. Since applications are often closely coupled, failures propagate quickly within the interwoven system infrastructure. Lankes et al. [20] performed a case study to use failure propagation metrics to evaluate the landscape of a partner from the financial sector. In his dissertation [19] Lankes also highlights the importance of landscape metrics. Nevertheless, the given approach only focuses on one criterion, namely failure propagation, and does not regard the multitude of other aspects which influence business applications.

Johnson et al. (cf. [16] and [23]) have defined an assessment framework as well as a tool for the evaluation and analyses of EAs focusing on business and IT. Their approach - similar to ours - is intended to support IT management in organizations but the methodology is different from ours. In the model described in [16], properties of people in charge of applications are related to their application's properties, for example the administrators experience influences the availability of the administrated application to a certain degree. We do not want to use properties of people in such a manner.

Besides the above-mentioned publications, which address metrics and key figures to assess applications, Gammelgård et al. [13] deal with the reliability of ratings by humans in the given context. Ratings by experts are used frequently, though the ratings' quality is mostly not assured. Gammelgård et al. present a method to estimate the uncertainty of expert knowledge. We will take up these results and use them for our approach, where it becomes inevitable to use expert's ratings.

3. FOUNDATION

In this section, an overview of foundations and as-is situations regarding assessments of applications within organizations is presented. The first subsection deals with the basics of assessments, whereas the second focuses on factors that exert influence on applications. The third subsection presents the state-of-the-art regarding application landscape metrics that are basic components of the EA|AE method.

3.1 Assessments within organizations

As mentioned in the introduction, it is often not sensible to take a certain key figure for itself, for instance the value 100.000 $ for a key figure "cost per year" might be seen as "good" for an "important" application, but "bad" for unimportant ones. Since cost-benefit analysis (CBA) approaches often have fixed hierarchies that only allows for changing the weighting-factors, such an assessment approach is too rigid in practice; therefore, the EA|AE method provides the flexible combinations of key figures.

A very important aspect when dealing with information intensive methods within organizations is having the required data as well as the effort of gathering it. This problem is stated in several publications dealing with EA management (for instance [27]), to which application evaluations belong. Depending on the organizational structure, the CIO can exert more or less influence on the organization's divisions in order to convince their directors to gather the required data.

It may be an advantage if EAM solutions have already been introduced into the organization, since a lot of required information of artifacts and their relations to each other might have been gathered and are maintained through cyclic processes. EAM covers many activities and processes like: documenting, analyzing, planning, and controlling enterprise architectures (EA). Several research projects have been performed to define coherent and holistic models as well as structural and organizational aspects of EA (cf. [21], [7], [12], [4], [27], [8], and [5]). Furthermore, there are several consultancies as well as vendors who offer tools to support EAM (surveys and rankings of current EAM tools can be found in [14] for instance).

3.2 Influence factors

Within this section we give an overview of the EA context of our approach. As mentioned in the introduction, there are several factors that must be regarded when evaluations of applications are performed since they are exerting influence on the resulting quality indicator. Although the focus of the EA|AE method lies on applications, the overall context has to be regarded. As described in several publications (cf. [27], [21], [12], and [15]), only a holistic view on an enterprise or organization leads to real improvements because there are lots of physical and logical connections between other artifacts and applications, which can not be ignored.

The influence factors on enterprise applications vary from enterprise to enterprise. By analyzing publications and discussions with project partners from industry we have identified five general groups of influence factors (influencing artifacts): 1) business (business processes), 2) landscape (other applications), 3) hardware infrastructure, 4) externals (organizations, legislation, authorities), and 5) strategy (organizational strategies). Changes in business processes (1) often require changes of applications that provide functionality to support business processes. Properties of applications can be influenced by attributes of other applications (2); for instance the availability can depend on another application's availability. Applications may be influenced by the properties of the underlying hardware infrastructure (3), for example an application's response time can depend on the performance of the server on which it is installed. They are further influenced by rules and regulations, like SOX (Sarbanes-Oxley act) or Solvency II, by external organizations, authorities, or legislation [22] (4). The enterprise board, the chief executive officer (CEO), or the CIO, for instance, can make certain strategic decisions exerting direct or indirect influence on enterprise applications (5). For instance, vendor strategies could be established, which demand that a certain vendor is always preferred in procurement decisions, or a strategic decision towards applying service-oriented architecture [9] could be made.

3.3 Application Landscape Metrics

Metrics are essential components of our method. One kind of metrics with special importance are application landscape metrics (ALM). ALM differ from classical software metrics [10] since these focus on one application and its software architecture. Exemplary metrics of this kind are presented in [19], or [1] - amongst others. Furthermore, Lankes et al. [19] performed a survey to find out whether organizations use metrics to assess or control their landscapes. The results are presented on the left side of Figure 1. Only 11 percent of the surveyed people indicate that the use of metrics does not make sense in their organizations. One reason is the effort of gathering the required data for the metrics' usage.

In order to get the state-of-the-art of ALM, we conducted a literature research (cf. [3]). Our aim was to search for key figures as well as the metrics to calculate their value. All key figures must at least fit one of the following three conditions: (a) it must be used for indications of applications and be based on the application's attributes, (b) it must be an indicator of an application and its value is determined by attributes and relations from other EA artifacts (the applications' enterprise context), and (c) it must indicate a landscape's quality and therefore use all attributes of applications and their enterprise context.

64 key figures and their metrics have been identified as a first result of the literature research. These key figures require 104 different types of data. The distribution of the 64 key figures to the above-mentioned three categories is represented on the right side of Figure 1. For each key figure a template-table (cf. [3]) has been filled out containing necessary information, like the required data, the metric, and the key figure's objective. The objective might be a question that can be answered by the key figure. Such an overview is considered to support CIOs to select relevant key figures taking the requirements and expected results into consideration.

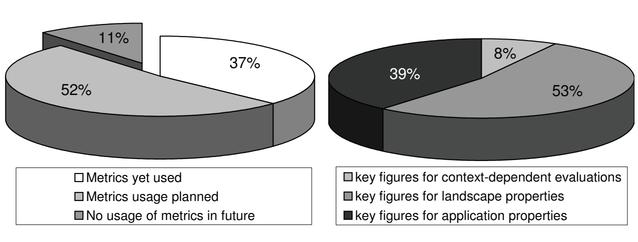

Figure 1: Results of the TUM survey (left) and our metrics investigation (right)

4. ENTERPRISE ARCHITECTURE BASED APPLICATION EVALUATIONS (EA|AE)

The EA|AE method addresses single applications and how to evaluate those in regard to influences and expectations in the organizational context. Before we go into detail, we will first explain some relevant terms to provide a solid foundation for the method. The second subsection explains the organizational method for a stakeholder like the CIO to evaluate a set of applications. The last subsection will present details of the technical process to derive overall indicators for applications out of EA artifacts' attributes.

4.1 Taxonomy

This section covers the relevant terms for our approach. Figure 2 presents an information model of the terms and their relationships as a UML class diagram. The information model extends the model presented in [1]. Due to the limited space, we will not go into detail for every element but focus on the essential elements. Every evaluation (see Figure 2) consists of several criteria on which certain applications have to be evaluated:

A criterion in our model is a 6-tuple

$$\mathfrak{c} = (\mathfrak{n}_\mathfrak{c}, \mathfrak{q}, \mathfrak{m}, \mathfrak{D}, \mathfrak{k}, \mathfrak{i})$$

in which $\mathfrak{n}_\mathfrak{c}$ *is the criterion's name, a question* $\mathfrak{q}$ *that lies behind the criteria, a metric* $\mathfrak{m}$*, a set of data* $\mathfrak{D}$*, which is the input for the metric to calculate the criterion's key figure* $\mathfrak{k}$*, and an optional indicator* $\mathfrak{i}$ *that is derived from* $\mathfrak{k}$*.*

A criterion is an aspect that can be directly assessed when regarding a concrete application. Criteria can be mapped to one or more attributes of artifacts. Evaluations are based on predefined metrics (cf. Section 3.3) that use attributes of *EA artifacts* (see Section 1) to calculate the metric's key figure. It is important to define the criterion as well as the belonging metric clearly; ambiguous descriptions could cause problems due to misinterpretation by humans, which can be involved in the evaluation process as experts. Whenever key figures are not sufficient to represent an application's value regarding a criterion, a natural language like indicator can be inferred based on the key figure. According to definition of a linguistic variable by Zadeh [29], we define the term *indicator* as follows:

An indicator is characterized by a quintuple

$$\mathfrak{i} = (\mathfrak{n}, \mathfrak{T}(\mathfrak{n}), \mathfrak{c}, \Omega, \mathfrak{R}_{sem})$$

in which $\mathfrak{n}$ *is the indicator's name;* $\mathfrak{T}(\mathfrak{n})$ *is the set of linguistic terms of* $\mathfrak{n}$*;* $\mathfrak{c}$ *is the criterion to which* $\mathfrak{i}$ *belongs;* Ω *is the universe of discourse, which is identical to the one of the metric's key figure that belongs to* $\mathfrak{c}$*;* $\mathfrak{R}_{sem}$ *is a semantic rule that which associates its meaning with each linguistic value* $\mathfrak{v}$*,* $\mathfrak{R}_{sem}(\mathfrak{v})$*, where* $\mathfrak{R}_{sem}(\mathfrak{v})$ *denotes a fuzzy subset of* Ω*.*

In order to use more than one criterion to determine the quality of an application, we specify evaluations. Evaluations are collections of criteria. Each criterion in an evaluation has a particular weighting factor, because the influence of certain criteria may differ from others. An aggregation function is a certain functionality or technology, which is based on indicators of criteria and uses the information to infer an overall indicator for an application. Aggregation functions can vary in the complexity of the underlying function; for instance, a simple kind of aggregation function may only calculate the mean value of indicators for criteria. Aggregation functions could also implement complex

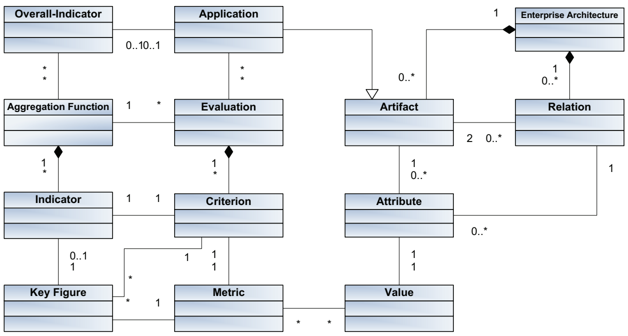

Figure 2: Information model of the approach

data transformations for the input and output values as well as powerful technology for the inference - like for instance Bayesian Nets or artificial neural networks. In our method, the aggregation function is implemented as fuzzy controller (cf. [18]). A fuzzy controller is a rule based system that allows for flexible specification of rules to describe how the overall indicator should be inferred (cf. [2]).

4.2 Evaluation of Applications

After covering the foundations for our approach, in this section we introduce the method to evaluate applications in an abbreviated manner. Figure 3 depicts the whole process. The numbers in the following text refer to the according numbers in that visualization.

EA dependent evaluations start with the evaluation's definition. First of all, the applications to be evaluated have to be selected (1). The second step contains the selection of relevant criteria (2). For this purpose, the description and annotations of each criterion (and thus their key figure) can be regarded (cf. Section 3.3). The questions that are to be assigned to each key figure are especially important for our partner from industry to decide whether a key figure is relevant. Based on the annotation of required data, which is also assigned to each key figure, the stakeholder can check whether the data is at hand, or whether it is cost-effective to gather the data (3). In the latter case, the stakeholder may reselect the set of criteria or applications. Following that decision, the evaluation process can be configured (4). For example, the stakeholder may choose whether the results of each criterion of an application should be represented by key figures or indicators or as aggregated overall indicator for an application.

If the data is managed by a tool like our prototypical implementation (cf. Section 5), the tool can check the timeliness of the required data (5). This is an optional feature to assure the result's quality. If the data is not up-to-date or if datasets are missing, the sub process of gathering the required data have to be executed (6). After that task as well as if the data is already there and up-to-date (see 5), the percentage of gathered data should be checked (7). If the degree is insufficient to evaluate the applications, the stakeholder has to verify (8) whether it is in general possible to gather the data. The positive case leads to number 6, the subprocess of gathering the data in order to get the appropriate degree; the negative case leads to aborting the method (9).

If the degree of existing data is sufficient, the metrics can

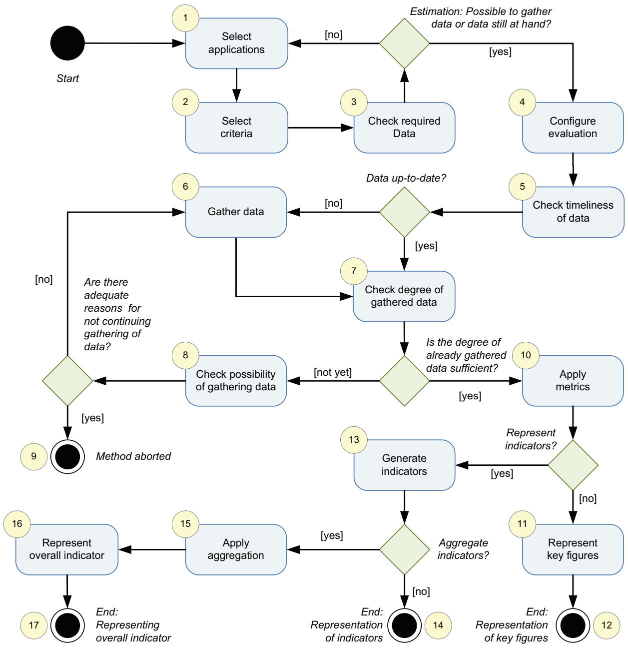

Figure 3: Evaluation method

be applied (10). When the representation of key figures is favored (the first case - see (4)), the representation is performed in the configured way (e.g. as tabular or graphical representation) (11) and the method ends with that result (12). Practitioners note, it is sufficient - if not essential - to stop at this point, because the representation of (numerical) values for each criterion and application is considered to be appropriate. The method provides this possibility and additionally offers the classification (according to indicators) of key figures as well as an aggregation to a single indicator. In the second case, the results are represented by indicators for each criterion and each application (13). Similar to number 12, the method ends with the result's representation (14). When the aggregation of indicators is considered, the aggregation function must be applied after the indicator's generation (15). With the representation of overall indicators for each application (16) the method will end (17).

It is difficult to perform these steps manually, thus an appropriate tool support is essential. We are working on a prototype of such a tool that implements the methodology and provides additional representations, views, and data input forms as well as data import interfaces (see Section 5).

4.3 Process to infer the overall indicators

The main process for evaluating an application from a stakeholder view is presented in Section 4.2. Within this section the technical process to infer an overall indicator for an application is explained, but with respect to the limited space in abbreviated form. This technical process is visualized in Figure 4.

Let a_1 be the application under investigation; further let there be two criteria $\mathfrak{c}_1$ and $\mathfrak{c}_2$. Each metric ($\mathfrak{m}_1$ and $\mathfrak{m}_2$) of those criteria are connected with either attributes of a_1 as well as attributes of connected (via $\mathfrak{r} \in \mathfrak{R}$ - cf. Section 1) artifacts $\mathfrak{a} \in \mathfrak{A}$, if it is an ALM (cf. Section 3.3), and only connected to attributes of a_1 otherwise. Applying the metrics leads to key figures - one for each metric and thus one for each criterion.

The CIO can decide (cf. Section 4.2 (4)) whether to use natural-language based indicators or key figures to represent the quality of an application. In the latter case, the result of the evaluation has to have exactly one key figure for each criteria applied on each application. This can be reported within a tabular representation containing the values.

In the former case, each key figure within a criterion is assigned to an indicator that is comparable to a linguistic variable (cf. [29]). All linguistic terms of an indicator are interpretable as classification of key figures. Fuzzy sets (cf. [28]) are used to represent the linguistic terms. In contrast to classical set theory (crisp sets), the fuzzy sets allows elements to have a partial membership to a set. It is often hard for stakeholders in an organizational context to defined exact intervals for classifications. If an interval border is defined in a wrong way so that a lot of elements are marginal out of the interval, the results may be wrong. Fuzzy sets allow for defining 'flowing' borders so that each element's degree of membership is falling the higher the distance of the element to the set border is. Nevertheless, it is possible to define membership functions (see [18]) such that the corresponding (fuzzy) set is equal to a crisp set. Indicators like such allow for using natural-language like terms like *"good"* or *"average"* when regarding the criteria of an application. This is considered more intuitive for the stakeholder.

The last step of the process is an aggregation of each application's indicators to overall-indicators. The aggregation is implemented as a fuzzy controller for the EA|AE method (cf. [2]). A fuzzy controller is based on linguistic variables in order to infer a target linguistic variable based on the definition of fuzzy rules. This is well suited for our method since indicators as well as the overall indicator are linguistic variables.

Fuzzy rules have a premise which is a propositional logic concatenation of assertions that in turn consist of linguistic variables and their linguistic terms; the conclusion is an assignment of an linguistic term to the target linguistic variable (cf. [18]). A brief example of a fuzzy rule set is depicted in Figure 4. Due to the fact that the membership degrees of the linguistic variables are included within the rules' inference, there is no loss of information in this phase.

The aggregation's result is represented by a single (overall) indicator for each application that is considered; however, with a defuzzification method (cf. [18]) a numeric output value can be obtained from an indicator.

5. TOOL-PROTOTYPE AND CASE STUDY

There are no appropriate tools that evaluate business applications and which provide the above-mentioned required flexibility. Some EA tools have integrated metrics and support evaluations of application in general; however, most of the metrics are hard-coded and do not allow for adding new ones or for combining them with other artifacts' attributes. An overview and a categorization of common EA tools is presented in [14].

We are implementing a tool-prototype that provides the flexibility as well as the above-mentioned aggregation method. The prototype is based on the Eclipse Rich Client Platform, which supports module-based software applications. Each software module, which is denoted as "bundle", can be easily added or removed from the main application. In our case

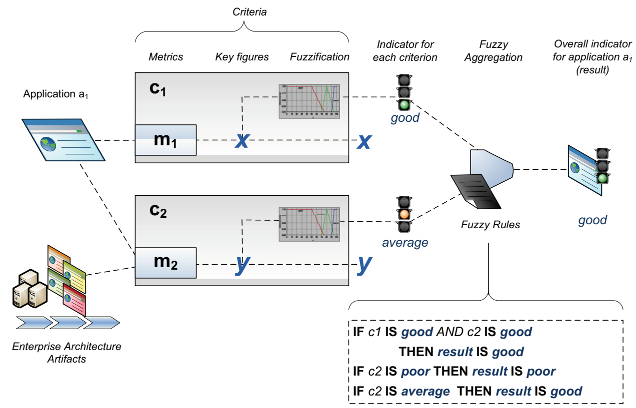

Figure 4: Technical process to infer overall indicators based on attributes of EA artifacts and using a fuzzy controller as aggregation function

each criterion is represented by a bundle. The CIO can flexibly add or remove criteria to the method, furthermore visualizations such like tabular reports or graphical visualization can be added to the prototype to represent the results.

Our prototypical application evaluation by one of our partners from industry is intended to be followed by an extensive project to introduce an holistic landscape evaluation approach. This project should encompass every of the few hundred applications and vary from cyclic evaluations to real-time monitoring. With the results of the first evaluation, the directors of the divisions should be convinced to gather further required data to extend the criteria. Well-defined key figures (see Section 3.3) are therefore essential for the directors to estimate the effort as well as to understand the key figure's objective.

For the results we consider two different kinds of representation: a tabular report and a software map. The latter is intended to visualize the landscape with its associated applications by using graphical figures and symbols. On such a map the indicator values can be represented by small icons like traffic lights, pie charts, or other adequate symbols.

6. CONCLUSION AND FUTURE WORK

This contribution introduces our EA based method to evaluate applications. First of all, we provided a motivation of the importance of using the EA context. To have a solid starting point, we gave foundations for the approach before we presented our method briefly.

We are currently improving our tool prototype to support application evaluations. This tool has a modular architecture, so that criteria (and key figures and metrics) as well as visualizations can be implemented as modules in order to provide extensibility. The method is to be applied within an extensive scenario with a few hundred applications. To this end, the missing but required data has to be gathered for all applications.

The method is intended to be able to cope with the uncertainty of expert knowledge in assessments in future. Here, fuzzy logic promises to be an appropriate technique. We intent to continue with this aspect, having the objective of improving the results of expert ratings.

7. REFERENCES

[1] J. S. Addicks. Enterprise architecture dependent application evaluations. In *Proceedings of the 3rd IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2009): Cyber Engineering and Creating Value by Making Connections*, pages 628–633, Istanbul, 2009.

[2] J. S. Addicks. A fuzzy logic based approach for enterprise application evaluations. In G. I. Doukidis and C. Avgerou, editors, *Proceedings of the 4th Mediterranean Conference on Information Systems (MCIS)*. 2009.

[3] J. S. Addicks and P. Gringel. Application landscape metrics: Overview, classification, and practical usage. In W. Esswein, J. Mendling, and S. Rinderle-Ma, editors, *Proceedings of the 3rd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA)*, 2009.

[4] S. A. Bernard. *An Introduction to Enterprise Architecture*. AuthorHouse, Bloomington Ind., 2nd ed. edition, 2005.

[5] C. Braun and R. Winter. A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. *Enterprise Modelling and Information Systems Architectures, Proc. of the Workshop in Klagenfurt, GI–Edition Lecture Notes (LNI)*, P–75:64–79, 2005.

[6] C. Braun and R. Winter. Integration of it service management into enterprise architecture. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pages 1215–1219, New York, NY, USA, 2007. ACM.

[7] S. Buckl, A. M. Ernst, J. Lankes, K. Schneider, and C. M. Schweda. A pattern based approach for constructing enterprise architecture management information models. *Wirtschaftsinformatik*, (2):145–162, 2007.

[8] M. Ekstedt, P. Johnson, Å. Lindström, M. Gammelgaard, E. Johansson, L. Plazaola, E. Silva, and J. Lilieskold. Consistent enterprise software system architecture for the cio: A utility-cost based approach. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pages 225–234, 2004.

[9] T. Erl. *Service-oriented architecture: Concepts, technology, and design*. Prentice-Hall, Upper Saddle River, NJ, 6th print. edition, 2006.

[10] N. E. Fenton and S. L. Pfleeger. *Software metrics: A rigorous and practical approach*. PWS Publ., Boston, 2 edition, 1998.

[11] I. O. for Standardization (ISO). Industrial automation systems – requirements for enterprise-reference architectures and methodologies, 2000.

[12] U. Frank. Multi–perspective enterprise modeling (memo) conceptual framework and modeling languages. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002. HICSS*, pages 1258–1267, 2002.

[13] M. Gammelgaard. *Business Value Assessment of IT Investments: An Evaluation Method Applied to the Electrical Power Industry*. PhD thesis, Royal Institute of Technology (KTH), Stockholm, 2007.

[14] G. A. James. Magic quadrant for enterprise architecture tools, Gartner Research, 2008.

[15] P. Johnson, M. Ekstedt, E. Silva, and L. Plazaola. Using enterprise architecture for cio decision-making: On the importance of theory. *Proceedings of the 2nd Annual Conference on Systems Engineering Research (CSER)*, 2004.

[16] P. Johnson and M. M. Lankhorst, editors. *The 2nd Workshop on Trends in Enterprise Architecture Research (TEAR 2007)*. Trends in Enterprise Architecture Research. Via Nova Architectura, 2007.

[17] L. Kappelmann, T. McGinnis, A. Pettite, and A. Sidorova. Enterprise architecture: Charting the territory for academic research. In *Proceedings of the Fourteenth Americas Conference on Information Systems (AMCIS 2008)*. 2008.

[18] V. Kecman. *Learning and soft computing: Support vector machines, neural networks, and fuzzy logic models*. MIT Press, Cambridge, Mass., 2001.

[19] J. Lankes. *Metrics for Application Landscapes: Status Quo, Development, and a Case Study: Dissertation*. PhD thesis, Technische Universität München, München, 2008.

[20] J. Lankes and C. M. Schweda. Using metrics to evaluate failure propagation and failure impacts in application landscapes. In M. Bichler, T. Hess, H. Krcmar, U. Lechner, F. Matthes, A. Picot, B. Speitkamp, and P. Wolf, editors, *Multikonferenz Wirtschaftsinformatik, MKWI 2008, Proceedings: [26. - 28. Februar 2008, TU München in Garching]*, pages 1827–1838. GITO–Verlag, Berlin, Berlin, 2008.

[21] M. M. Lankhorst. *Enterprise Architecture at Work: Modelling, communication, and analysis*. Springer, Berlin, 2005.

[22] J. N. Luftman. Key issues for it executives 2004. *MIS Quarterly Executive*, 4:269–285, 2005.

[23] P. Närman, M. Schönherr, P. Johnson, M. Ekstedt, and M. Chenine. Using enterprise architecture models for system quality analysis. In *Proceedings of the 12th IEEE International EDOC Conference (EDOC)*, 2008.

[24] M. Rohloff. An integrated view on business– and it–architecture. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 561–565, New York, NY, USA, 2008. ACM Press New York.

[25] M. Schönherr. Towards a common terminology in the discipline of enterprise architecture. *Lecture notes in computer science*, pages No. 5472 (2009),400–414, 2009.

[26] The Open Group. *TOGAF Version 9*. van Haren Publishing, 2009.

[27] R. Winter and R. Fischer. Essential layers, artifacts, and dependencies of enterprise architecture. *EDOC Workshop on Trends in Enterprise Architecture Research (TEAR 2006), October 17 2006, Hong Kong*, 2006.

[28] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[29] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, (8):199–249, 1975.