IT Landscape Management Using Network Analysis

Daniel Simon[1], Kai Fischbach[2]

[1] Department of Information Systems and Information Management, University of Cologne
simon@wim.uni-koeln.de
[2] Chair in Information Systems and Social Networks, University of Bamberg
kai.fischbach@uni-bamberg.de

Abstract. Dependency analyses have become crucial in today's enterprise architecture practices, which usually face complex IT landscapes with highly interdependent applications. In such environments, a deep understanding of the application's context is essential to determine its qualities and project its further evolution. However, method support for making this context a tangible IT landscape management part and thus facilitating quantitative decision making still seems expandable. Based on the representation as a network of applications and their relations of data exchange, this paper therefore suggests ways to support the IT landscape's examination through network analysis. We develop this approach based on a combination of theoretical explanations, past empirical findings, and experiences taken from the architecture practices of four sample organizations. We illustrate and evaluate our approach with a short case study. Our approach, developed and illustrated in close alignment with insights from actual practice, thus offers ideas and advice for researchers and practitioners alike.

Keywords: IT landscape management, enterprise architecture management, dependency analysis, network analysis, network centrality

1 Introduction

Complex information technology (IT) landscapes of hundreds or even thousands of business and infrastructure applications (cf. [1]) have become commonplace in many organizations. As a result, applications today, more than ever before, cannot be treated individually; rather, they can be fully understood only in the context of their environment – a basic principle of systems thinking [2]. As an interwoven system of applications (cf. [3]), that is, as a specific "collection of components organized to accomplish … a set of functions" [1], the IT landscape is thus subject to a variety of inherent relations and dependencies. Together, these interrelationships form an essential part of the overall architecture of such a system, which represents its "fundamental concepts or properties … in its environment embodied in its elements, relationships, and the principles of its design and evolution" [4].

For a managed evolution of the constituents of the business and IT landscape, including their interdependencies, many organizations have thus implemented enterprise architecture (EA) management practices. Relevant techniques for these

practices include, in particular, regular "neighbourhood analyses" [5] for evaluating applications in their current state, while considering their position in the overall IT environment, and for preparing for architectural impacts across the IT landscape in case of intended application changes, especially of those with high system relevance.

Although the importance of such dependency analyses is now recognized, they seem only partially implemented in EA practices, as Aier et al. [6] discovered in their state-of-the-art survey of EA professionals. In their conclusion, they call for greater methodical support for such analyses. Therefore, the creation of a greater operationalization of systems thinking in EA management, that is, the ability to make more "tangible" the context of relationships and, in particular, the interaction of specific parts with the whole, may need to come to the core of future research.

At the same time, network analysis (NA) comes into more widespread use to explore complex phenomena, for example, at the social, political, economic, and organizational levels. This includes, for example, the interpretation of network structures to discover collaboration patterns within and across organizational boundaries and then correlate these patterns with the performance of individuals or entire groups. In line with the holistic nature of systems thinking, NA is based on the assumption that actors (represented as nodes) are interdependent and that relations (represented as edges) between actors provide access to specific resources [7].

With that in mind, similar considerations can be made at the level of IT applications that support the business tasks of specific users. In fact, one may view the IT landscape as a network of applications (the actors) linked by their inter-dependencies (the edges); the emergent structure reflects dependencies in the work of different stakeholders in the enterprise and manifests their patterns of collaboration.

In this paper, we thus introduce NA into the examination of IT landscapes and explore the research question of how NA measures can support IT landscape management (IT LM). Our findings show that using such metrics from NA may contribute significantly to an increased understanding of applications and their context and substantiate assessments at several points in the IT LM process. Our study thus contributes to the advancement of IT LM method support motivated above.

The remainder of this paper is structured as follows. Section 2 surveys earlier research into the use of NA for IT LM purposes. Section 3 provides the paper's conceptual foundation and introduces relevant NA measures. Section 4 then develops our theoretical concept and discusses the metrics' meaningfulness in IT LM. In Section 5, we put these theoretical thoughts to the test and present the case of an insurance company, the IT landscape of which we studied using NA to gain insights into certain properties both of specific applications and the landscape as a whole. We close the paper with our conclusions and an outlook on future work in Section 6.

2   Related Work

There is little research dealing with the application of NA means to IT LM. The works closest to ours are those of Dreyfus and Iyer [8, 9] and Iyer et al. [10], who also adopt a network perspective on IT architecture. Given limited resources, they argue, it is important to identify a subset of components of the overall architecture (macro

architecture) that are to be controlled or actively managed. They distinguish between intrinsically important components that may, for example, support critical business tasks, represent large investments, or have a large number of users and thus are most often the focus of decision makers, and components that are important because of their positions in the larger network of interdependent components. They call the latter architectural control points, the control of which allows decision makers to influence the evolution of the architecture toward ongoing support of the business goals. These architectural control points, however, are considered moving targets given regular changes in the architecture. So, this is where active management is needed; in their studies, the authors (among others) find that architectural thinking in the form of rules guiding the emergence reduces the decline in control of key nodes.

According to the authors, these key nodes, which may adversely affect the network as a whole if they fail, can be identified using network centrality measures (also see [11]). In addition, the authors argue that NA can help enterprise architects identify the shared core, that is, the set of components used by most other components, and also the best decomposition of a set of components that minimizes dependencies across architectural clusters while maximizing dependencies within a cluster. Aier and Schönherr [12] focus on the latter in greater detail and present an algorithm-based modeling approach that adopts NA concepts to define service domains and support the design of a service-oriented architecture.

All in all, though, the use of NA in IT LM has not yet been made systematic and has not been detailed in terms of which network centrality measures can help gain what insights into the IT landscape and single applications. That lack is what motivating our research. We ground our study in the work of Simon et al. [13], which offers a framework for IT LM and suggests, among other things, a set of basic dimensions for application analysis, including risk, value, and complexity.

3 Conceptual foundation: Measures of network analysis

Before exploring the use of NA in IT LM, we here set the conceptual foundation of our work, introducing basic concepts of NA and clarifying their meaning in our context. Rooted in graph theory, NA conceptualizes and visualizes structures that emerge from any interaction or connection as networks and allows a quantitative analysis of the network nodes' relationships (cf. [7]).

As indicated, the representation of the IT landscape as a network of nodes and edges is central to our approach. Nodes represent applications, which we consider executable software components that support business functions [1, 9]; edges represent relationships and interdependencies between applications. In general, these relationships can take different forms, that is, there are multiple possible definitions of what constitutes a link (cf. [14]). For example, links may indicate the same vendor or supporting technology of different software components; alternatively, they may also represent physical interfaces between applications. In our interpretation, edges represent data exchange/flow; as such, they can be considered logical interfaces between applications. In this model, two nodes are linked if they exchange data; in other words, one application depends on the other application for data.

In this study, we use different centrality measures to examine both the characteristics of such a network as a whole, that is, the structural patterns of interaction between applications, and the role and distinctive position of individual applications within the overall landscape (exemplary real-life figures are provided in Section 5).

Degree centrality (C_D) represents the number of relations of a given node and thus indicates the degree of "activity" [7, 15, 16] of applications within the IT landscape. Formally, it can be defined as follows:

$$C_D(i) = \sum_j x_{ij} \tag{1}$$

where x_{ij} equals 1 if there is a link between applications i and j, and $x_{ij} = 0$ otherwise. Using directed ties, we can also distinguish in-degree (C_{In-D}) and out-degree centrality (C_{Out-D}) and thus account for the extent to which an application provides or consumes data.

Closeness centrality (C_C) measures the geodesic distance of a given node to all other nodes in the network [7, 15, 16]. The node that can reach all other nodes in the fewest steps is most central. C_C can be formalized as

$$C_C(i) = \frac{1}{\sum_j d_{ij}} \tag{2}$$

where d_{ij} is the number of links in a shortest path from application i to j ($i \neq j$).

Betweenness centrality (C_B) represents the "number of shortest paths that pass through a given node" [7, 15, 16] and therefore indicates whether an application plays some kind of a gatekeeper function, controlling data exchange in the overall network. In mathematical terms, it can be written as

$$C_B(i) = \sum_{j,k} \frac{g_{jik}}{g_{jk}} \tag{3}$$

where g_{jk} denotes the number of shortest paths from application j to k ($j, k \neq i$), and g_{jik} is the number of shortest paths from application j to k passing through application i. Normalization results in values of between 0 and 1. In contrast to C_D, C_B thus also considers indirect relationships in the network, since here the position between other nodes is relevant.

The consideration of indirect relationships is also true for eigenvector centrality (C_E), which quantifies the extent to which nodes are connected to other central nodes in the network [17]. For computing this measure for a given node, the relationships to other nodes are thus weighed based on these nodes' centralities:

$$C_E(i) = \frac{1}{\lambda} \sum_j x_{ij} C_E(j) \tag{4}$$

where $x_{ij} = 1$ if applications i and j are connected, and $x_{ij} = 0$ otherwise, and λ is the largest eigenvalue of the adjacency matrix X.

Eventually, we also consider overall graph density (as the number of edges divided by the maximum number of edges in a full graph), modularity (as the number of edges falling within groups minus the expected number in an equivalent network with edges placed at random) [18], and the clustering coefficient (the average fraction of a node's neighbors that are also neighbors of one another) [19] as global measures for gaining insights into the IT landscape as a whole.

4 The Role of Network Analysis in IT Landscape Management

Based on the representation of applications and their data exchange as a network of nodes and (directed) edges, we now theorize about application scenarios of NA in IT LM and the role of the network metrics just discussed. Two main inputs provided the basis for this exploration, namely a thorough analysis of the literature and information extracted from four architecture practices (see below), both of which we used to first identify and structure activities and variables to be examined in IT LM. To figure out the potential role of NA metrics, we employ a combination of theoretical explanations (i.e., logical reasoning), empirical findings from the literature (e.g., the relationship between application costs and the number of interdependencies), and own empirical evidence about application evaluation (e.g., complexity is typically measured by the number of applications and dependencies) as captured in the personal experiences of the first author (cf. [20]), who has direct knowledge of the architecture practices of a sample of four organizations in the insurance, banking, pharmaceuticals, and public sectors (two operating worldwide, and the others focusing on the German market). In terms of the empirical reasoning, we thus base our discussion on a meaningful combination of convenience sampling (sample selected from this author's practice) and purposive sampling (sample of organizations of different types, e.g., industry sector, size, and geographic scope) [21] to point to opportunities to incorporate means of NA in architectural activities. The four reference practices have an age of at least two years and are part of IT organizations employing between 500 and 1400 people. They also had defined EA processes in which IT LM was part of the scope.

Metrics may play an important role in IT LM; this is in line with De Marco [22], who writes that "you cannot control what you cannot measure." In general, the NA measures introduced represent reasonable instruments in the given context, as they meet the required properties for IT LM metrics identified in [23]: they are basically well understood and have a communicable meaning (see preceding section). This allows for achieving the main goal of using metrics [23]: the communication of actual facts regarding the status quo and future potential. Specifically, the NA measures allow an evaluation that is objective in nature and considers related elements from the overall context rather than relying on subjective ratings by individual experts.

Architectural views and evaluations are of crucial relevance in several EA management tasks, and at different levels of abstraction (cf. [1, 24]). Scenarios of using network measures and views for characterizing the IT landscape and its applications may thus exist across the entire EA lifecycle through which EA elements are driven: documentation; analysis; planning and decision making; implementation; and governance [5, 13]. We explore this in detail in the following subsections.

4.1 IT Landscape Documentation

As for architectural documentation, it is common practice to partition and refine the overall EA into more detailed domain architectures that cover specific segments (e.g., marketing and sales) of the enterprise and allow the distribution of architectural work in large enterprises. Within the limited scope of these domain architectures, dedicated domain architects may also become responsible for further architecture developments [1, 25]. However, the accountability still remains at the level of the enterprise architect; larger changes within the IT landscape of a specific domain may affect the overall enterprise or may be subject to interdependencies with other domains.

Once a strategic initiative at the domain level results in issuance of a request for architecture work that involves a further development of the domain's IT landscape, typically starting with the documentation of the current state, two of our reference practices thus check the extent to which the enterprise architect needs to be involved in that endeavor. The involvement may be restricted to certain governance activities, but may also require close participation in the IT LM process. Criteria being checked to determine the degree of EA involvement include, for example, the domain landscape's complexity, as indicated by the number of applications and dependencies, and the initiative's cross-functional character, as represented by the existence of applications that are used across several business processes and organizational units.

In practice, this check is often completed using rough and subjective estimations; given the above criteria, NA measures may represent more reliable indicators. Assuming that cross-functional applications (e.g., enterprise resource planning) are part of the landscape's shared core and thus are connected to many other applications either directly or at least indirectly, C_C seems a reasonable measure for indicating the initiative's cross-functional character as it can capture the mean distance of an application to all others. Depending on the number of applications with a C_C value that exceeds a pre-defined threshold, the decision about the EA involvement could be made. In addition, one would evaluate the landscape's complexity; there, the average C_D value of the domain's applications, that is, the average number of their relations (cf. [26]), seems an appropriate measure, given the relevance of the number of dependencies for this criterion.

If there is already a documented state of the domain landscape, C_C and C_D can be calculated in a pre-state of actual IT LM activities so that enterprise architects can already be involved in the documentation phase (in this case, restricted to updating activities) if considered necessary due to high centrality values; otherwise, the involvement check would be completed once one has passed through this phase.

With a documented state of the overall landscape, one can also check whether the clusters that may have emerged in the actual network of applications are consistent with the pre-defined structure of domains, which are most often derived based on means of functional decomposition [27] and should thus group close to each other those elements that are cohesive and related [24, 25] (e.g., recruiting and human resource development). As a result, there should be maximal dependencies within a domain, but minimal dependencies across different domains. This can be validated by using network views of the IT landscape and by calculating, for example, the sum of the edges within a domain compared to the sum of the edges to components of other domains. An alternative is to calculate average C_C at the intra-domain level and

compare it to the inter-domain value. In addition, one may use global modularity, which indicates the number of communities within the network, and the clustering coefficient, which considers how nodes are embedded in their neighborhood and thus provides an overall indication of the network clustering.

If the actual clusters represented by the existing relationships and derived from NA are not at all in line with the domain structure, one may not have achieved the best decomposition, that is, one that allows a relatively independent management with a clear scope of responsibility. By applying the approach depicted in [12] and removing edges with highest C_B (as the number of shortest paths between pairs of vertices that run along it) until separate communities emerge, one may find a more suitable structure. However, an inconsistency between the domain structure and any results from NA may not necessarily reflect an improper decomposition, but may also point to a state, in which there are some domains with several redundant applications and thus more relations to other domains if there was only one of these applications in place. This leads us to the next main step in the process – the IT landscape's analysis.

4.2   IT Landscape Analysis

Generally, analyzing the current IT landscape aims at capturing the condition of both the landscape as a whole and each application individually. It has been acknowledged that the overall landscape's condition is determined to a significant degree by its complexity and modifiability [13, 28]. Landscape complexity is related closely to the overall number of interdependencies within the landscape [14, 28]. Similarly, landscape modifiability can be explained by the average coupling between applications, that is, the average number of relationships of service consumption and provision [26]. To quantify these qualities, network density and average C_D thus seem simple but promising instruments that could help in establishing adequate metrics at the landscape level, which apparently have not yet permeated practice [5], as also reflected in our sample.

NA metrics may also serve as reasonable indicators for some of the attributes that are relevant for individual application analysis [13]. Applications are typically analyzed in terms of their associated risks. Therefore, one may differentiate between risk cause (i.e., factors that increase the probability of the risk event), risk event (i.e., the risk itself, associated with a certain probability), and risk impact [1, 29, 30]. In simple terms, the risk event describes an application failure in terms of its availability (including, e.g., performance, reliability), confidentiality, and integrity, or, from a long-term perspective, states of insufficient maintainability and modifiability/ adaptability to keep the application performing as expected and required and able to adapt to changing requirements [1, 31, 32, 33]. The risk impact can be manifold; in terms of the effects of non-availability, the insurance reference practice distinguishes productivity, monetary, and immaterial losses (negative impact on, e.g., customer satisfaction, image, or regulatory compliance), together denoted as the criticality of IT applications. For each dimension, applications are rated using a simple ordinal scale, the stages of which represent some sort of risk classes within this dimension.

What is not considered explicitly in these figures, however, is the context of a given application. Obviously, when applications occupy a central position in relation

to their context of related applications, they can influence their environment (maybe even the network as a whole) negatively [32]. To estimate such impacts in terms of the scope of failure, network centrality measures thus seem appropriate instruments. One way of operationalizing possible effects of application failure on other applications is C_{Out-D}, which represents the number of applications that the given application must provide with data; actually, the average value may indicate the extent to which the overall landscape is prone to failure propagation. C_E may offer more specific insights into potential negative effects, taking into account the centrality of the related applications themselves; for example, the failure of an application that provides presumably central applications such as customer relationship management and contract management with data may be more severe than that of one primarily related to rather peripheral applications such as travel expense accounting. Eventually, using C_B allows the identification of applications whose failure could have a negative affect on the coaction of other applications or entire clusters of applications (consider, for example, an inventory management application, which may be required to have a working process from order management to billing); in the worst case, the failure of applications of high C_B could result in clusters falling apart. Network metrics can thus serve to give adequate consideration to an application's context in risk impact analysis and to complement the single-application-centric analysis of criticality with context-related variables represented by network centralities (in the example above, possibly by establishing another dimension with a dedicated network impact factor).

The cause of the risk also seems related closely to the context in which a given application operates. On the one hand, this is due to the influence other applications may exert on an application in terms of the data it consumes, as indicated by C_{In-D}. So, failures of other applications that provide certain data may, in turn, contribute to the risk of a given application, due to its availability or performance that might be negatively affected. On the other hand, the flexibility of application change may also be influenced by connections, whether ingoing or outgoing, to other applications [13, 28]. This has also been observed in [11], where it is shown that increased component coupling, as a measure of the degree of dependency between one software component and all others, is associated with decreased flexibility. For the purpose of measurement, C_D works as a direct indicator, whereas C_C may be used alike, as it indicates a component's distance to all others and thus its extent of being coupled.

Another attribute for which the network position is relevant is the cost of an application. According to [14], application costs are affected significantly by the number of the application's interdependencies (the higher the interdependencies, the higher its operations and maintenance costs). In line with these insights, an application's C_D appears to be a helpful indicator for evaluating costs; while it does not provide cost figures itself, it may be used to provide a proper indication. This seems reasonable, particularly where detailed cost calculations are not part of the scope of EA activities or are difficult to obtain from IT controlling, for example, due to a different level of abstraction used for controlling purposes – a challenge that one reference practice faces in particular. A straightforward alternative could be to group applications into C_D classes and use this classification as a type of cost indicator.

Application value is a particularly important attribute for analysis, because it is only when an application is critical to success that weaknesses in other aspects such

as technical health or operational performance may become particularly relevant [5]. In general, application value can be analyzed along different dimensions such as strategic (e.g., support of business strategy) (cf. [34]), operational (e.g., cycle time reductions) and financial (e.g., contribution to revenue generation) dimensions [13] – an approach basically shared by two of our reference practices. As for the operational dimension, C_B seems an adequate evaluation metric given that applications can be considered especially valuable if they occupy a central position within one (or several) business process(es) and may, for example, somewhat control the data exchange therein. This offers an alternative to the approach of our sample insurance company, for example, where we found a scale used for measuring the operational application value, ranging from "the support of only a small part of process steps" to "the support of all relevant process steps" but without consideration of the application's power of control.

The strategic fit of an application is another attribute, where a current state may be measured by network characteristics. This is because today's strategic directives most often embrace some statements representing a certain pursuit of standardization; this is the case in our four sample practices, where we also find this reflected in their architecture principles (as the general rules and guidelines of architecture work [1]). These may conflict with the existence of several applications with many interrelationships, as represented by their C_D; at least in networks that include representations of elements such as an enterprise service bus (ESB) as nodes, such high C_D values point to a high number of remaining peer-to-peer interfaces (as the use of an ESB would typically suggest an encapsulation of individual interdependencies).

Table 1. Application analysis and the role of network metrics

Application attributes	Role of network metrics for application analysis
Risks	• Out-C_D quantifies the number of applications possibly affected by application failure (i.e., the scope of direct failure propagation). • In-C_D quantifies the number of applications, which in case of failure may affect a given application. • C_D and C_C indicate the state of modifiability of a given application. • C_B quantifies, given the failure of an application, the threat of other applications or entire clusters impeded in their coaction. • C_E quantifies the threat of central applications affected by application failure.
Costs	• C_D indicates the cost associated with a given application.
Value	• C_B indicates applications that control data flow within one or across several business processes.
Strategic fit	• C_D may indicate applications that are not compliant with strategic directives that stipulate standardization and homogeneity.
Technical complexity	• C_D may indicate the number of physical interfaces of an application.

For the same reason, C_D may also support analysis of an application's technical health, because one of the factors that determine the technical health is complexity [13]. It can (among others) be explained by the application's number of (physical) interfaces (cf. [28]), which is also one of the main criteria our sample insurance provider considers to determine the future prospects of an application from a technical perspective. As this number may be related to the number of links in the representing network (see above), C_D may also be considered a reasonable health indicator.

Table 1 summarizes the potential role of network metrics in IT landscape analysis.

4.3 IT Landscape Planning and Decision Making

To approach any issues discovered in the analysis stage, the succeeding planning of the future IT landscape typically involves defining alternative scenarios, which are weighed against one another to determine the desired option [5, 13]. This is also how our sample organizations proceed. Evaluation of this sort includes the scenario costs, risks, and time constraints [5]. Therefore, it is also necessary to consider the effects of changes captured within a scenario across the IT landscape [1, 13]. If there are plans to take action in terms of the application lifecycle, for example, a "simple" upgrade to a new application release may require others to be upgraded before (cf. [35, 36]).

Our reference practices basically aim to keep the scenario evaluation rather simple and mostly use ordinal scales for their ratings; again, this is where NA measures may provide added value, since they are straightforward to use and meaningful alike. Risks can come from potential negative impacts of the planned change of an IT landscape element on others. So, modifications of positionally important applications could expose the organization to significant risks. Such risks can be indicated by C_D, given the number of related applications it represents. C_E may take this even further, since it allows for a focus on impacts on applications that are themselves central. In addition, C_B allows a more differentiated approach to scenario risk evaluation; it may pinpoint applications that could adversely affect the coaction of other applications or entire clusters that could fall apart, given their position between others. Eventually, it is C_D that can apparently also help evaluate scenarios in terms of time and costs. It can indicate necessary efforts of adapting other applications or their interfaces due to changes of a particular application; this is because it explicates the number of related applications that might also be modified in result (though it does not say anything about the quality of these related changes).

Once the evaluation has led to the selection of a favored scenario, this target landscape is typically compared to the current one to identify gaps between these states. According to common practice, these gaps are then consolidated and assigned to projects aimed at closing them [1, 34]. Other demands or project ideas that may have been brought from other business units into project portfolio management need to be integrated in this planning of projects [5]. Enterprise architects typically help refine the scope of potential projects, check their effects on the architectural landscape, and bring the insights gained by surveying their architectural content (e.g., applications, their dependencies, and affected business functions) into the assessment of project proposals in terms of criteria such as costs, risks, and value and into the synchronization of projects changing the same or related elements [1, 24, 27, 34, 35].

Using network centrality measures, the enterprise architect may bring new qualities into these project evaluations; in fact, consideration may also be given not only to architectural objects affected directly by a project (typically depicted in a project context diagram), but also those affected indirectly because they are related to an object in project scope. In this way, NA metrics help operationalize a project impact check, considering the network position of the architectural content of a project.

4.4 Implementation of Target IT Landscape

Once projects for implementing the target landscape are defined and initiated, the EA function needs to assure that adequate and conformant architectural solutions are designed, for example, by sending off solution architects to accompany the projects. However, not every project may need to be monitored in such detail, and resources may also be limited. To determine which architectural project support is most appropriate (e.g., regular compliance reviews, ongoing participation) [1], three of our sample organizations conduct a check of architectural relevance. Among other criteria, they check the complexity of the project architecture, which our sample bank gauges by the number of interfaces of the components to be implemented and the resulting number of affected components.

Again, for this measurement of topologic complexity, C_D of the applications in scope of the project seems a reasonable indicator, since it quantifies the number of affected components; as such, it can facilitate the architectural relevance check (cf. [37]). It can also be visualized in system context diagrams. However, it could well be that one project affects several rather small applications while another affects only a few, but large, applications. That is why the use of C_E should be considered as well.

The subsequent processes of solution architecture design, that is, the component design at the micro-architectural level that occurs within the designated projects for implementing the target landscape, can also be informed by NA metrics (cf. [36]). For example, they may indicate whether design principles such as coupling and cohesion are sufficiently realized and help ensure that resources are allocated primarily to critical modules (cf. [38]); further details are beyond the scope of this paper though.

4.5 Governance of IT Landscape Management

Across the above fields of activity within IT LM (i.e., from documentation to implementation), it is crucial to assure ongoing architectural governance [1]. This includes the measurement of results, which is of particular importance since it provides the basis for justifying and selling the value of an EA function in the long run – a challenge that EA practitioners increasingly face today [39].

The EA charter [1] is a fundamental basis for any governance activities; it defines the mission, vision, goals, objectives, and key performance indicators (KPIs). So, this is where the measurement of success finds its variables and instruments. Two of our sample organizations have established such a charter. In the pharmaceutical organization, two specific objectives defined in the charter are a minimal number of dependencies between applications and between technology components. The number

of individual peer-to-peer application interfaces and the average number of dependencies to other technology components are used as the corresponding KPIs.

With that in mind, (average) C_D apparently manifests as a suitable metric for governance purposes as well (note that edges in the network should represent physical relationships to measure the number of peer-to-peer interfaces; at the logical level, relationships of data exchange remain the relevant aspect to be represented by edges).

In the next section, we present the case of an insurance company and our use of NA to study its IT landscape to provide some evaluation of our approach (the case description was reviewed and validated by the organization).

5   Case Study: An Application of the Approach to Practice

The enterprise we studied is a German insurance company that is a key player in the markets for household, liability, and car insurance. An EA practice was established in 2009, embedded predominantly in the IT department (of about 250 employees) with five enterprise IT architects, about twenty solution architects, and the architecture board and architecture review board as the decision-making bodies for strategic and solution architecture issues, respectively. EA processes were defined, including for architecture documentation and analyses, target architecture planning, project portfolio evaluations, and compliance assessments. In 2010, a toolbox of templates and checklists for solution architecture design was developed. Likewise, a standards management framework was introduced. At the same time, considerable efforts were undertaken to capture an initial state of the IT landscape, resulting in a map of business and infrastructure applications and their relationships of data exchange. What still remained to be substantiated are methods and techniques to support the regular IT LM process. To help evaluate our concept and create new insights for its IT LM practices, the organization thus provided us with the above landscape map. We imported the corresponding file into Gephi, an open-source software tool for graph and network analysis [40]. The resulting network of applications showed numerous interrelationships. In total, the network comprised 338 nodes and 859 edges, accounting for a density of 0.008 and modularity of 0.49; average C_D was 5.08.

On that basis, we computed C_D, C_B, and C_E of the network's applications (as the organization asked for a focus on only a few metrics). Armed with these metrics, we went into discussions with the lead enterprise IT architect, one of his fellow enterprise IT architects, and the IT architect who had created the draft IT landscape map. We compared the computation results with intuitive notions about the applications that are most critical, valuable, prone to failure, complex, costly, and difficult to adapt. We asked the architects to suggest a few applications to which they would ascribe these properties, and later presented the top-ten applications according to each metric.

This stimulated considerable discussion among the architects. Some figures were obvious to them, that is, some of the applications indicated to be most central by the metrics were also those they had assumed to represent key applications. In fact, all metrics indicated the same three applications to be most central: a premium collection, claims (e.g., $C_{Out-D}=39$, $C_{In-D}=35$, $C_B=0.22$, $C_E=0.81$), and policy application (with a much higher C_{In-D} (=38) than C_{Out-D} (=13)). It was also interesting

to see that the participants had assigned at least one application to each of the above properties that also showed up among those with highest C_D. This seems in line with our theory about the supporting role of C_D in application analysis.

In contrast, other insights were less expected. C_B indicated, for example, a sales support application to be significant (C_B=0.12) that did not have a high C_D at all. Upon further discussion, the architects were able to justify that this application was important because it allowed the automatic forwarding of application data, which plays a crucial role in achieving high quantities of new policies per day. This fosters our suggestion about the potential role of C_B in criticality and value assessments of applications, as it may allow the incorporation of new facets into such analyses.

The C_B data also identified one financial accounting application as central (C_B=0.1) that had been considered especially valuable by the architects themselves; this means they can also substantiate initially subjective valuations. Notably, this application also had a high C_{In-D}, but a rather low C_{Out-D}. This let us to conclude that a considerable C_{In-D} may also point to a certain value; examples may be business intelligence and premium collection applications that collect, aggregate, and process data from many different sources to support activities of great strategic or monetary relevance.

Using C_E also led us to further insights beyond those gained by using C_D; for example, it identified a family insurance application as positionally important (C_E of 0.38) that also did not have a high C_D. So, while each metric was considered helpful, C_B and C_E brought unexpected insights and helped us capture non-intuitive findings. C_{In-D} turned out to add another dimension into the assessment of application value.

When checking the network values for selected applications that did not appear in the top-ten lists, we also identified an application scenario of NA in IT LM that we had not previously considered in our approach. This is because the architects found applications that showed unexpectedly low centrality values (or even seemed to be disconnected), which raised the question of whether this actually represents reality or whether information about some links in the network was still missing. Network metrics thus also help resolve questions of data quality and aid in reviewing architectural documentation in terms of its correctness and completeness.

All in all, the nature of discussion changed quickly from the simple examination of raw counts to broader concerns of analysis and usage-oriented issues. It was acknowledged that the network metrics could support different aspects of IT LM, as represented by the EA lifecycle introduced above. Particular value was seen how they supported checking new demands or projects in terms of their architectural relevance.

Despite this value, we found some limitations and needs for extensions during the discussions. One interesting point concerned the possibility of weighting the edges to increase the meaningfulness of the analysis. On the one hand, one could quantify the edges in terms of frequency of data exchange. An impact of failure of an application with several connections might be less relevant if, for example, data are transferred only once a year over these connections. On the other hand, a qualitative weighting of edges also seems reasonable to account for their relevance to any critical business processes. Concerns were also raised regarding the meaningfulness of C_B if the relevant network path is subject to several media disruptions. More generally, in- and outgoing links of a node with a high C_B could refer to separate business processes; in such a case, any failure may have no immediate impact. The architects concluded that any indications that the NA measures provide should be analyzed in further detail.

Final words of attention came out of the discussions regarding the semantics of the network's elements. In particular, this relates to the question of what actually constitutes an IT landscape's node. A differentiation between business and infra-structure applications and their representation in separate networks was considered reasonable. As such, the former network would represent the real data sources and sinks, which are the ones that are more likely meaningful to the business. Brokering applications such as integration buses are of minor relevance to the business and their inclusion in one and the same network may thus complicate communication.

Another question was the level of abstraction at which a network should be spanned: this could be either the level of aggregated building blocks (cf. [1]) or of the individual parts. In the latter case, larger applications would be represented by several nodes in the network, which seems reasonable if one knows about the dependencies at that level of application modules. One further issue then arose in the discussions, which was the question of what actually defines a module such that it would be represented by a node in the network. It was opined that for a module to be considered a separate component, one should generally be able to procure, build, install, and operate that module independently from any others (this is also in line with [9]).

Altogether, we found qualitative support for our approach with this case study, and could even extend it at some points on that basis. We also encountered some area where further development is needed to advance the approach.

6 Conclusion and Outlook

With this study, we have uncovered the potential role of NA in IT LM (as requested by this paper's research question). There are several benefits of using NA metrics across the EA lifecycle. As for documentation, they can support the check for EA involvement in domain initiatives, and also the review of documentation in terms of its quality; in the analysis, network metrics can then help gain specific insights into the risks, costs, value, strategic fit, and technical complexity of applications. In other words, the metrics may alert enterprise architects of weaknesses and of applications in which one should invest to mitigate risks and ensure value. When it comes to IT landscape planning, the use of network metrics can support the evaluation of transformation scenarios and related projects in terms of any impacts. The support of architectural relevance checks also makes network metrics relevant in the implementation phase. Finally, governance may also be facilitated by the use of such metrics; in this regard, they help measure the IT LM performance.

In line with systems thinking, the NA metrics presented thus enable a simplified and objective evaluation of applications in consideration of their context of related applications. One of the main use cases are application risk assessments, in which C_C, C_B, and C_E complement C_D such that they allow a more differentiated examination of dependencies. As such, these metrics uncover specific insights and should be used in combination with other methods to verify potential areas of action. Our empirical findings obtained through the case study support and, to some degree, extend our concept; however, they also point to aspects that should be the subject of further research, such as the weighting of edges.

Our results have value despite the study's limitations. For the development of our approach, we could not make use of a fully representative sample, from which we could infer universal statements about the entire organizational population, although we could take advantage of our use of purposive sampling. Likewise, the case study used for testing our approach in practice is limited to one sample organization, in which IT LM has yet to be completely implemented using the network metrics.

Our study could thus be extended in various ways in the future. First, the extension of our sample seems a promising way to move the research process along. An essential future step thus would be to apply the approach iteratively in a number of real-life cases to verify its applicability and uncover further areas for improvement. These cases should also incorporate the requirements already gathered in our case study. Moreover, the set of NA metrics could be widened to include other instruments that facilitate identifying an entire group of critical applications (cf. [10]), which would collectively have the greatest (and most disruptive) reach in the network. With respect to decision making in IT LM, human aspects could be given greater consideration to include the cost of changing what is in the heads of users.

All in all, we believe this study offers ideas and advice for academics and practitioners alike. Aside from the academic value that lies in the understanding of our concept, this study also provides a foundation on which to draw in the future in terms of the further development of method support for IT LM; as indicated, further research and empirical findings is required to validate and extend the findings. For practitioners, our approach should help them scrutinize dependencies in the IT landscape and serve as a conceptual guideline for applying NA metrics as a quantitative basis for activities across the EA lifecycle. This is supported by this study's use of cases, which illustrate IT LM peculiarities in practice and helped build and test our approach. We can thus conclude with a recommendation for EA tool vendors to make their tools able to compute NA metrics (beyond C_D) so that they actually become integral in IT LM in practice.

References

1. The Open Group: TOGAF Version 9. Van Haren Publishing, Netherlands (2009)
2. Gharajedaghi, J.: Systems Thinking: Managing Chaos and Complexity, A Platform for Designing Business Architecture. Morgan Kaufmann, USA (2011)
3. Bernus, P., Schmidt, G.: Architectures of information systems. In: Bernus, P., Mertins, K., Schmidt, G. (eds.) Handbook on Architectures of Information Systems. Springer (2006)
4. International Organization for Standardization, ISO/IEC/IEEE 42010:2011, Systems and software engineering – Architecture description, http://www.iso-architecture.org/42010/
5. Niemann, K.D.: From enterprise architecture to IT governance. Vieweg, Wiesbaden (2006)
6. Aier, S., Riege, C., Winter, R.: Unternehmensarchitektur – Literaturüberblick und Stand der Praxis. Wirtschaftsinformatik 50(4), 292-304 (2008)
7. Otte, E., Rousseau, R.: Social network analysis: a powerful strategy, also for the information sciences. J Inf Sci 28(6), 441-453 (2002)
8. Dreyfus, D., Iyer, B.: Architectural control and emergent architecture: a network perspective. Boston University Working Paper (2005)
9. Dreyfus, D., Iyer, B.: Enterprise Architecture: A Social Network Perspective. In: 39th HICSS (2006)

10. Iyer, B., Dreyfus, D., Gyllstrom, P.: A Network-based View of Enterprise Architecture. In: Saha, P. (ed.) Handbook of Enterprise Systems Architecture in Practice. IGI Global (2007)
11. Dreyfus, D., Wyner, G.M.: Digital Cement: Software Portfolio Architecture, Complexity, and Flexibility. In: 17th AMCIS (2011)
12. Aier, S., Schönherr, M.: Integrating an enterprise architecture using domain clustering. In: Trends in Enterprise Architecture Research 2007, pp. 23-30 (2007)
13. Simon, D., Fischbach, K., Schoder, D.: Application Portfolio Management – An Integrated Framework and a Software Tool Evaluation Approach. CAIS 26, 35-56 (2010)
14. Mocker, M.: What is complex about 273 applications? Untangling application architecture complexity in a case of European Investment Banking. In: 42nd HICSS (2009)
15. Wasserman, S., Faust, K.: Social Network Analysis: Methods & Applications. CUP (1994)
16. Freeman, L.C.: Centrality in Social Networks. Soc Netw 1, 215-239 (1978/1979)
17. Bonacich, P: Some unique properties of eigenvector centrality. Soc Netw 29, 555-64 (2007)
18. Newman, M.E.J.: Modularity and community structure in networks. In: Natl Academy Sci 103(23), 8577-8582 (2006)
19. Uzzi, B., Spiro, J.: Collaboration and Creativity: The Small World Problem. American J Sociology 111(2), 447-504 (2005)
20. Webster, J., Watson, R.T.: Analyzing the Past to Prepare for the Future. MIS Q 26 (2002)
21. Teddlie, C., Yu, F.: Mixed Methods Sampling: A Typology with Examples. J Mixed Methods Res 1, 77-100 (2007)
22. De Marco, T.: Controlling software projects. Yourdon Press, New York (1982)
23. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F., Schweda, C.M.: State of the Art in Enterprise Architecture Management. TU Munich, Chair for Informatics 19, Germany (2009)
24. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F.: Enterprise Architecture Management Pattern Catalog. TU Munich, Chair for Informatics 19, Germany (2008)
25. Bruls, W.A.G., van Steenbergen, M., Foorthuis, R.M., Bos, R., Brinkkemper, S.: Domain Architectures as an Instrument to Refine Enterprise Architecture. CAIS 27, 517-540 (2010)
26. Vasconcelos, A., Sousa, P., Tribolet J.: Information System Architecture Evaluation: From Software to Enterprise Level Approaches. In: 12th Eur. Conf. on IT Evaluation (2005)
27. Hanschke, I.: Strategic IT Management. Springer, Berlin (2010)
28. Vasconcelos, A., Sousa, P., Tribolet, J.: Information System Architecture Metrics: an Enterprise Engineering Evaluation approach. Electron J Inf Syst Eval 10(1), 91-122 (2007)
29. OGC: Managing Successful Projects with PRINCE 2. The Stationery Office Ltd. (2009)
30. Sherer, S.A., Alter, S.: Information Systems Risks and Risk Factors: Are They Mostly about Information Systems? CAIS 14, 29–64 (2004)
31. BSI, Bundesamt für Sicherheit in der Informationstechnik, https://www.bsi.bund.de
32. Jordan, E., Silcock, L.: Beating IT Risks. Wiley, Hoboken (2005)
33. Maizlish, B., Handler, R.: IT Portfolio Management Step-by-Step. Wiley, Hoboken (2005)
34. Quartel, D., Steen, M.W.A., Lankhorst, M.: Application and project portfolio valuation using EA and business requirements modelling. Enterp Inf Syst 6(2), 189-213 (2012)
35. Saat, J.: Zeitbezogene Abhängigkeitsanalysen der Unternehmensarchitektur. In: MKWI 2010, pp. 119-130 (2010)
36. Jönsson, P.: The Anatomy – An Instrument for Managing Software Evolution and Evolvability. In: 2nd Int. Workshop on Softw. Evolvability, pp. 31-37 (2006)
37. Buckl, S., Ernst, A.M., Matthes, F., Schulz, C., Schweda, C.M.: Constructing an Enterprise-specific Radar System for Assisted Project Surveillance. In: MSI 2009, pp. 33-47 (2009)
38. Zimmermann, T., Nagappan, N.: Predicting defects using network analysis on dependency graphs. In: 30th Int. Conf. on Softw. Engineering, pp. 531-540 (2008)
39. Schelp, J., Stutz, M.: A Balanced Scorecard Approach to Measure the Value of Enterprise Architecture. In: Trends in Enterprise Architecture Research 2007, pp. 5-11 (2007)
40. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An Open Source Software for Exploring and Manipulating Networks. In: Int. Conf. on Weblogs & Social Media, pp. 361-362 (2009)