Empirical Results for Application Landscape Complexity

Alexander W. Schneider
TU München
schneial@in.tum.de

Thomas Reschenhofer
TU München
reschenh@in.tum.de

Alexander Schütz
TU Darmstadt
schuetz@is.tu-darmstadt.de

Florian Matthes
TU München
matthes@in.tum.de

Abstract

The complexity of application landscapes (AL) has been identified as one of the major challenges in enterprise architecture (EA) management for quite some time. Since there is no agreed upon definition of the term complexity in general or in the context of EA management in particular, literature offers a broad variety of concepts and measurements. Therefore, the main purpose of this paper is to (1) provide an overview about metrics to quantify the complexity of ALs proposed in literature, (2) identify metrics currently used in practice to measure AL complexity, and (3) compare empirical results and assess the metrics' applicability with industry experts. By the assessment of four different ALs from the financial sector, we are able to derive various strengths and weaknesses for the different metrics as well as open issues for future research on quantitative EA models.

1. Introduction

Today's information technology (IT) managers do not only face the challenge of steering the increasingly valuable IT assets of their company but are also asked to increase the alignment of business and IT [1], [2]. Therefore, an approach considering the enterprise as a whole including its processes, products and strategies in addition to IT assets like information systems and respective hardware is required. For that reason, scientists and practitioners developed numerous approaches which can be summarized under the term enterprise architecture (EA) management [3], [4]. With business IT alignment as one of their major goals, EA management approaches target, among others, at providing more transparency, ensuring compliance to relevant laws and reducing IT costs [5]. The main constituent of all EA management approaches is a holistic description of the enterprise, called architectural description or enterprise architecture, describing the fundamental components of the system (enterprise), their relations to each other and their environment as well as the principles governing their design and evolution [6]. In previous research, managing complexity has been identified as one major challenge in enterprise architecting [7]. Thereby, complexity can be experienced in various notions including structural, dynamic, qualitative and quantitative complexity [8]. Although no agreed-upon definition of the term complexity exists (see e.g. [9]), the field of EA research already accomplished first steps towards providing tools creating transparency and facilitating complexity management to control both respective risks and costs. The goal of this paper is to (1) identify available metrics to calculate the complexity of application landscapes (AL), (2) analyze the current state-of-the-practice regarding AL complexity measurement and (3) evaluate identified approaches by calculating metrics based on extensive real data and perform group discussions with respective enterprise architects to assess their individual strengths and weaknesses.

2. Related work on EA complexity

Although other fields already applied complexity science successfully, cf. [10], [11], the EA management discipline only achieved first steps. Within the context of electronic government Janssen and Kuk [12] applied the theory of complex adaptive systems to EA and derived architecture design principles for the public sector. The major challenge in this context is to manage diverse independent and local IT-related projects. The derived principles increase inter-organizational jointness and IT implementation success. Although very helpful, the presented design principles can neither explain nor prevent un-necessary application landscape (AL) complexity.

The field of management cybernetics is concerned with the management of organizations in general and was introduced by Beer [13]. Since the management of an EA is a sub-task of management in organizations in general several authors applied cybernetic concepts to design or explain EA management functions. Buckl et al. [14] used Beer's Viable System Model (VSM) to derive different duties of an EA management function. They distinguish, for example, between reactive and proactive EA management as well as EA management governance. By the use of a cybernetic model Buckl et al. were able to define the different tasks an EA management function has to fulfill in concise and well founded manner. Zadeh et al. [15] also used the VSM as well as the Viable Governance Model (VGM) to demonstrate how TOGAF's architecture principles relate to cybernetic concepts. Thereby, they identified shortfalls of TOGAF's currently defined principles: they have no theoretical foundation and provide no means for structuring or classifying. Until today, cybernetic models and principles have successfully been used to identify open issues in the field of EA management but they have not been used to explain or reduce EA complexity although their subject (enterprise) is a complex system.

Based on chaos theory, Saat et al. [16] derived requirements for the design of an EA planning activity. They conclude, for example, that changes of a single element might cause foreseen and unforeseen changes to other elements. Furthermore, with increasing temporal planning scope the predictability for the suitability of to-be models decreases. Thereby, they demonstrate how existing theories can be used to develop well-founded insights in EA management research.

Recently, Kandjani et al. presented a method to reduce the complexity of global software development projects by applying EA cybernetics and axiomatic design theory [17]. By decoupling planning and development activities project tasks become independent and therefore controllable and predictable. This results in a decrease of the project's structural complexity. In order to be useful for reducing the complexity of ALs this approach has to be extended in scope to consider multiple applications and multiple development projects.

Kandjani et al. [18] already used successfully a systemic perspective to explain the evolution of an EA by its adaption to a change in the organization's environment complexity. Based on the finding that a system has to have equal complexity as its environment, they developed a co-evolution path model which explains how organizations have to react to complexity changes while not overreacting. But to explain the complexity of a concrete EA the model is too general. Furthermore, it shows the different paths an organization can evolve its complexity, but it does not provide hints for their reasons.

3. Literature on EA complexity metrics

In order to quantify the complexity of application landscapes we started our research endeavor by a detailed literature search following commonly accepted guidelines [19], [20]. We consulted the EBSCOhost database, Science Direct, ISI Web of Knowledge (Web of Science database) and the search engines of the ACM, IEEE as well as Google by using the keywords "enterprise architecture" AND "complex*". Due to the high frequency of use for both terms the search resulted in more than 16,000 articles. Therefore, for each search engine the titles and if necessary the abstracts of the 100 most relevant articles have been analyzed regarding the use of metrics to quantify the complexity of EAs or parts thereof. The remaining articles were then analyzed regarding relevant forward and back references, as recommended by [19]. The structured literature search resulted in three different kinds of complexity metrics for AL complexity which have been proposed by literature and have also been evaluated in practice.

The first was Mocker [21] who identified the age of applications as well as corresponding business requirements as the main drivers of AL complexity. Based on available literature, he identified four different metrics to quantify AL complexity: Interdependency, diversity in technologies, deviation from technology standards and redundancy. In a small case where 273 applications—being part of the AL of a major bank—have been analyzed, correlations of these metrics with costs in terms of maintenance and operating could be found only for the interdependency metric, i.e. number of incoming and outgoing interfaces.

Schuetz et al. [22] introduce a metric to quantify the structural complexity of an IT landscape, which is also applicable to ALs. They mention the number and the heterogeneity of the components and relations of an EA as the major drivers of complexity in EA. This is in accordance with the IEEE Standard 1471-2000 [6], which considers an EA as system composed of components and their relations to each other. The EA as a system is decomposed into four subsystems Business, Data, Application and Infrastructure Architecture [4], [23]. In accordance with the stated conceptualization of complexity in EA, complexity C is rooted in the number N and the heterogeneity H of components T and relations R. Thereby, the term heterogeneity is defined as follows: Heterogeneity in IT landscapes is a statistical property and refers to the diversity of attributes of elements in the IT landscape [24]. The general structure of the metric can be formulated as in (1):

$$C_x = (N_x, H_x) \text{ with } x \in T, R \qquad (1)$$

While the number of components and relations can be determined quite simply by counting the respective elements, Schuetz et al. transfer the concept of concentration measures, especially the Shannon Entropy [25], to quantify heterogeneity. Equation (2) shows how the entropy measure (*EM*) is calculated. n denotes the number of diverse technical flavors (e.g., the number of different operating systems in use) and p_i denotes the relative frequency of a certain flavor i (e.g., the number of instances for operating system type i).

$$H := EM = -\sum_{i=1}^{n} p_i ln(p_i) \quad (2)$$

Note that the entropy measure can be used to quantify the heterogeneity of the components as well as the heterogeneity of the relations of a system. Furthermore, the approach allows for a creation of different complexity metrics taking various perspectives on the problem of EA complexity measurement. To analyze a concrete context, the considered unit of analysis has to be chosen, for example the operating systems in use (as an example of a potential component of the subsystem Infrastructure Architecture) or the different interface implementation types of one or a set of applications (as an example of a potential relation of the subsystem Application Architecture). The dimensions of analysis are limited by the available information of the architecture. In Section 5.3 we present derived metrics based on this approach.

Lagerström et al. [26] proposed to use an approach pervasive in the software architecture discipline—Design Structure Matrix—to visualize the hidden structure of an AL and thereby identify spots of increased complexity. First, based on the topology of the AL, i.e., applications and their dependencies, the type of the AL architecture is determined (core-periphery, multi-core or hierarchical). Second, in case of a core-periphery architecture, applications are classified into core, control, shared and periphery applications. Thereby, core applications are defined as the largest cyclic group of applications. Control applications have even more outgoing dependencies while shared applications have more incoming dependencies. Periphery applications have both less incoming and less outgoing dependencies compared to the core. Especially shared and core applications are expected to require higher cost/effort when they have to be changed due to their amount of transitive dependencies. In addition, the authors also propose to measure the propagation cost. It is defined as the part of the AL which could be impacted when changing a randomly selected application. The plausibility of the metric has been proved in a case where 103 applications, which are part of a more extensive AL, have been analyzed. Lagerström et al. also demonstrated the metrics feasibility for whole EAs [27].

4. Complexity metrics currently used in practice

Since practitioners always contributed to the development of the EA management discipline, we also assessed the current state-of-the-practice regarding the measurement of AL complexity. By conducting a survey we were able to identify six commonly used metrics in practice.

4.1. Survey design and responses

In order to assess the current state-of-the-practice, we invited six companies which have established their EA management functions many years ago to a one day workshop. The headquarters of all companies are located either in Germany or Switzerland. They belong to different industry branches like automotive (1), banking (4) and insurance (1) and employ between 650 and 3500 people in their IT department. Each participating enterprise architect reported on the metrics currently used by the respective company to measure complexity of the company's AL. While we saw smaller approaches consisting of four metrics, the most elaborated approach consisted of 22 metrics. Afterwards, we used the technique of hermeneutics (cf. [28]) to develop a deeper understanding of the metrics as well as their context.

4.2. Identified complexity metrics

By conducting the hermeneutic circle and having individual conversations in order to remove ambiguity, we identified six metrics which have already been used by at least one half of the participating companies:

- **Number of Applications**: 4 out of 6 companies count the total number of applications they have in their AL as well as the number of applications belonging to a specific domain. A higher number is associated with a higher complexity.
- **Number of Information Flows**: 6 out of 6 companies count the number of interfaces (information flows) each application has. A higher number is associated with a higher complexity.
- **Standard Conformity**: 4 out of 6 companies assess the standard conformity of their applications. Therefore, they classify their applications as buy, make or buy and customize.
- **Number of Infrastructure Elements**: 4 out of 6 companies count the number of infrastructure components used to realize an application. A higher number is associated with a higher complexity.

- **Functional Scope**: 3 out of 6 companies assess the functional scope for each application. It is determined either by the application's function points or by the number of business functions realized by the application. A higher scope is associated with a higher complexity.
- **Functional Redundancy**: 6 out of 6 companies assess if different applications provide the same functionality. A higher rate of redundancy is associated with a higher complexity since single changes then affect multiple applications.

5. Empirical metric evaluation

In order to evaluate the benefits and drawbacks of each category of identified complexity metrics, i.e.,

- *Heterogeneity-focused* metrics as proposed by Schuetz et al. [22],
- *Topology-based* metrics as proposed by Lagerström et al. [26],
- *Industry metrics* as identified in our practitioner survey,

we induced the information required for calculating each of them and gathered the corresponding data from four of the six companies involved in this research endeavor. Thereafter, we implemented and calculated all metrics—heterogeneity-focused, topology-based, and industry metrics—in order to compare them to each other.

5.1. A core model of relevant EA concepts

To establish a common understanding of the information necessary to calculate the identified metrics, we induced an information model for each metric by modeling the required concepts. For example, the industry metric *Number of Applications* (cf. Section 4) induces a concept *Business application* which is related to a concept *Functional domain*. The concept *Information flow* represents dependencies between two *Business applications* and thus allows the calculation of the topology-based metrics. Since the participating practitioners proposed multiple ways of assessing an application's *Functional Scope*—namely the application's function points as well as the number of business functions realized by the application—we modeled all concepts required for each of the proposed functional scope assessments. By integrating the information models of all metrics, we developed a core model describing the information demand for calculating all identified metrics. Based on this and in close collaboration with the participating practitioners, we consolidated the core model and added additional concepts which are available for delivery by a majority of the companies (e.g., the attribute *programmingLanguages*). Hence, the core model not only reflects the required information for calculating all metrics, but also the availability of information as it nowadays usually exists in enterprises.

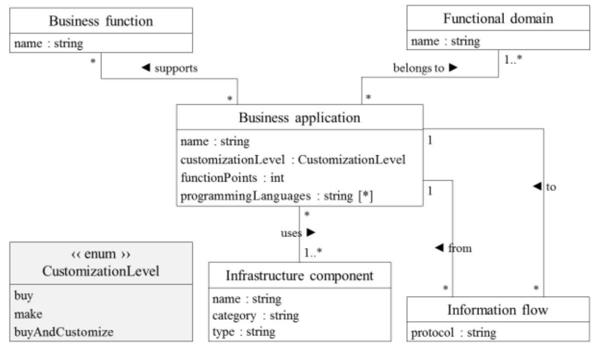

Figure 1. Core information model

5.2. Data base description

Based on the core model, four out of six participating companies provided corresponding anonymized data. Due to the various organization-specific AL documentation approaches, the data provided did not match exactly the core model as depicted in Figure 1. However, by renaming concepts, reverting the direction of relations, or by transforming attributes to relations, and vice versa, we were able to apply a logical mapping from the existing individual information model concepts to the corresponding core model concepts in each of the four cases.

Table 1. Data base

Concept	C1	C2	C3	C4
Business application	432	247	1898	234
- customization level	100 %	89 %	100 %	100 %
- function points	–	99 %	24 %	98 %
- programming languages	–	45 %	–	72 %
- implements (Use Case)	–	–	–	–
- supports (Business function)	–	97 %	–	–
- belongs to (Functional domain)	100 %	99 %	87 %	100 %
- uses (Operating system)	–	67 %	90 %	83 %
- uses (Data base system)	–	48 %	60 %	67 %
Business function	–	580	–	–
Functional domain	13	16	33	41
Infrastructure component (Operating system)	–	9	15	12
- type	–	–	100 %	100 %
Infrastructure component (Date base system)	–	10	8	16
- type	–	–	–	–
Information flow	539	827	8252	1214
- protocol	100 %	–	100 %	100 %
- from (Business application)	100 %	100 %	100 %	100 %
- to (Business application)	100 %	100 %	100 %	100 %

Table 1 shows an overview of the consolidated data basis by giving the number of entities for each type respectively the ratio of maintained values for attributes and relations of the corresponding type for each of the four cases. A hyphen means that the corresponding company does not maintain the corresponding concept and thus could not provide it. The relations *uses (Operating system)* and *uses (Data base system)* of type *Business application* are referring to specific categories of *Infrastructure components*. These are the only specific categories of *Infrastructure components* which are provided by at least two of the four companies. An *Infrastructure component's* type refers to its vendor or component class, e.g., *Windows* could be the type of *Windows XP*.

5.3. Complexity metrics implementation

Based on each company's actual data set, we evaluated the set of actually applicable metrics for each of the four cases.

With regard to the heterogeneity-focused metrics, we derived a set of ten metrics covering different perspectives of the architectures based on the core model as defined in Figure 1. Because this paper has a focus on AL complexity, the metrics presented in Table 2 target the application (A) and the infrastructure (I) layer. We included the infrastructure layer, because applications often rely on special infrastructure components and therefore have an effect on infrastructure complexity. Furthermore, we state whether the metrics refer to the systems components (T) or relations (R). The data quality and quantity varied between the four involved companies. Therefore, Table 2 also contains a mapping from the metrics to the cases each metric is applicable in.

In order to apply the topology-based metrics to the data sets, we are interpreting the *Information flows* between *Business applications* as dependencies between these applications. For example, if there is an *Information flow* from a *Business application* A to a *Business application* B, then B depends on A. Due to the availability *of Business applications* as well as corresponding *Information flows* in each of the four cases, the topology based metrics are applicable in all of them.

Since the core model is primarily induced by the industry metrics as described in Section 4, the determination of an applicable set of industry metrics for each of the four cases based on the actual data set is straightforward. For the *Functional scope* metric there might be multiple instances, since an application's scope can be determined by multiple ways, as already described previously in this section.

While numeric application-level metrics can be aggregated to domain-level metrics by summing up or averaging the values of each application, the aggregation of nominal scaled values like the *customizationLevel* requires a transformation from a nominal scale to an interval scale by applying an order and subsequently assigning numbers to the nominal values. However, both the applied order and the assigned numeric values hugely affect the actual value of the *Standard conformity* metric. In collaboration with the participating practitioners, we agreed on calculating the *Standard conformity* metric for a specific domain by transforming each application's customization level to 1 (*buy*), 3 (*make*), or 5 (*buyAndCustomize*) and taking the average according to their daily practice.

5.4. Complexity metrics results

We calculated the complexity metrics based on the approach proposed by Schuetz et al. [22] as shown in Table 2. In the following, we focus on a presentation of the results of the metric *Coupled Domain Complexity*, because it provided interesting insights into the architectures of the companies and because this metric was realizable in all cases. First, we want to explain more precisely how this metric was calculated and what the enterprise architects can learn from this metric.

As can be seen from Table 2, the metric *Coupled Domain Complexity* has a focus on the application layer and on the architecture element's relations. The metric calculates the number of interfaces of applications assigned to a functional domain D and the heterogeneity of the functional domains, a given domain D is coupled with. Coupling between domains is determined by the information flows of the applications assigned to the domains. An example: Imagine an application APP1 assigned to domain D1 having three interfaces to three applications, each assigned to a distinct domain. In this case, the interfaces of APP1 would be equally distributed among three domains. Now imagine an application APP2 assigned to domain D1 having three interfaces to three applications, all assigned to domain D2. Because of the concentration of the interfaces on one domain, the functional heterogeneity of the interfaces of APP2 is lower than the functional heterogeneity of the interfaces of APP1.

Because the enterprise architects were very interested in an evaluation of their domain models from a complexity perspective, we discussed the characteristics of a common target architecture. It became apparent that interfaces of applications assigned to value generating domains (e.g., loans, trade or sales) should, in principle, have a low functional heterogeneity. This should be the case, because the corresponding applications share their information and services with just a few systems (e.g., with a data-warehouse) outside of their domain. However, interfaces of the applications in domains providing information and services to numerous domains (e.g., data-warehouse) are assumed to have higher functional heterogeneity. Interfaces of applications in domains concerned with corporate management tasks (e.g., HR, accounting or risk management) should have an average functional heterogeneity.

Table 2. Derived heterogeneity metrics

Metric	Perspective A	Perspective I	Perspective T	Perspective R	Cases	Interpretation
Application Type Complexity	✓		✓		C1 C2 C3 C4	Number and heterogeneity of the Customization Levels (make, buyAndCustomize or buy) of a domain's applications
Business Function Complexity	✓			✓	C2	Number and heterogeneity of business functions supported by a domain's applications
Component Category Complexity		✓	✓		C1 C2 C3 C4	Number and heterogeneity of infrastructure components of a given component category (e.g., operating systems, data bases, etc.)
Coupled Domain Complexity	✓			✓	C1 C2 C3 C4	Number and heterogeneity of the functional domains a given domain is coupled with, whereas the coupling between domains is determined by their applications and information flows between them
Data Base Complexity		✓		✓	C2 C3 C4	Number and heterogeneity of data bases used by a domain's applications
Interface Implementation Complexity (Application)	✓			✓	C1 C3 C4	Number and heterogeneity of the technical implementations of an application's information flows
Interface Implementation Complexity (Domain)	✓			✓	C1 C3 C4	Number and heterogeneity of the technical implementations of the information flows of a domain's applications
Operating System Complexity		✓		✓	C2 C3 C4	Number and heterogeneity of the operating systems used by a domain's applications
Operating System Type Complexity		✓	✓		C3 C4	Number and heterogeneity of the operating system types (e.g., vendor and version) used by an application
Programming Language Complexity	✓		✓		C2 C4	Number and heterogeneity of programming languages the applications of a domain are based on

A = Application layer I = Infrastructure layer T = Components R = Relations

Table 3. Results for heterogeneity metrics

	C1		C2		C3	
	EM_A	NC / NE	EM_A	NC / NE	EM_A	NC / NE
Loans	7.9	13 / 153	5.15	9 / 213	18.69	28 / 588
Sales	7.96	10 / 71	7.86	11 / 129	16.16	27 / 285
Legal Audit	5.59	11 / 112	3.2	5 / 34	13.52	28 / 203
HR	3.39	4 / 10	4.95	6 / 17	10.44	24 / 174
DWH	6.84	10 / 90	9.23	14 / 249	15.75	29 / 946
Controlling	7.8	12 / 124	8.73	12 / 67	12.99	30 / 1069
Risk	4.52	6 / 40	5.81	8 / 62	5.77	27 / 1124
Number of domains	13		16		33	

EM_A = Numbers-Equivalent Entropy Measure
NC = Number of characteristics NE = Number of interfaces

The results of the metric *Coupled Domain Complexity* are shown in Table 3. Note that we mapped similar domains to the domains listed in the table and that these domains are just excerpts of the companies actual domain models. Furthermore, we focus on the results of cases 1 to 3, because these companies belong to the same industry branch (banking) and hence are assumed to have similar organizational structures.

In order to facilitate interpretation, we report the so-called Numbers-Equivalent Entropy Measure EM_A instead of the Entropy Measure (cf. [21]). The EM_A is calculated as shown in equation 3 and is equivalent to the number of characteristics (in this context the domains) that would lead to the same Entropy value, in the case of an equal distribution of the elements (in this context the interfaces of the applications). The number of characteristics (NC) represents the number of domains the respective domain is coupled with. Therefore, NC is the maximum value of EM_A. NE denotes the number of interfaces of all applications assigned to the respective domain.

$$EM_A = exp(EM) \qquad (3)$$

To validate the state of the current domain structure, we regard a coupling up to 33% of the existing domains as low coupling, from 34% to 66% as medium coupling and from 67% on as high coupling.

A glance at the calculated measures for the domains loan and sales reveals that the current situation differs substantially from the target architecture in all three cases. Especially for domain loans, we observe the almost highest values of EM_A in cases 1 and 3, instead of a relatively low functional heterogeneity of the interfaces. Not quite as noticeable, yet higher than expected is the calculated value in Case 2. In addition to the high heterogeneity, Cases 1 and 2 have a comparatively high number of interfaces, indicating a high complexity of the underlying application structure. The domains legal audit, human resources (HR) and data-warehouse (DWH) are essentially in line with the characteristics of the target architecture. However, the interfaces of domain controlling show a comparatively high functional heterogeneity in Cases 1 and 2. Although the functional heterogeneity of the interfaces in Case 3 is relatively low, the number of interfaces in this domain indicates a high complexity of the underlying application structure. The functional heterogeneity of domain risks' interfaces are in accordance with the target architecture, however, the number of interfaces in Case 3 is even higher than the number of interfaces of domain DWH. This observation was obviously surprising.

Based on the approach described by Lagerström et al. [26], we clustered the applications in all four cases into core, control, shared and periphery applications.

The results are depicted in Table 4. On the first glance, we can see that the different clusters show some similarity in size. The core is mostly the largest group followed by the periphery. This outcome makes the metric plausible. On the second glance, we see some fundamental differences in the presented cases. For example, the largest core consists of 55% (Case 2) while the smallest core consists only of 37% (Case 4). Therefore, on a relative scale the largest core is about 50% larger than the smallest. An even greater difference can be found in the comparison of the different propagation costs. The propagation cost in Case 2 is about two times the propagation cost observed in Case 4. Therefore, in Case 2 changes to applications are much more likely to create unforeseen changes compared to Case 4.

Table 4. Results for topology metrics

Class	C1	C2	C3	C4
Core	38%	55%	45%	37%
Control	14%	7%	7%	17%
Shared	8%	11%	13%	3%
Periphery	40%	27%	35%	43%
Propagation cost	24%	41%	30%	21%

Table 5. Results for industry metrics

Metric	C1	C2	C3	C4
Application-level industry metrics				
Number of Information Flows	[0;108], 2.5	[0;70], 6.7	[0;319], 8.7	[0;122], 10.4
Number of Infrastructure Components	–	[0;20], 4.7	[0;22], 0.4	[0;13], 3.9
Functional Scope (Function points)	–	[1;34], 6.7	[1;56], 7.3	[10;400], 178.9
Functional Scope (Number of business functions)	–	[0;34], 6.7	–	–
Domain-level industry metrics				
Number of Applications	[6;28], 14	[1;53], 16	[1;194], 50	[6;88], 24.8
Total Number of Information flows	[10;195], 83	[1;390], 110	[0;1316], 491	[23;580], 281.8
Average Number of Information flows	[0.2;15.4], 5.7	[0;19.154], 12	[0;77.2], 13.3	[3.1;27.8], 14.2
Standard conformity	[1.4;3.9], 3	[1.4;5], 2.7	[1;3.5], 2.5	[2;3.3], 2.5
Total Number of Infrastructure Components	–	[0;103], 7.6	[0;44], 11.5	[9;34], 19.5
Average Number of Infrastructure Components	–	[0;8], 1.0	[0;3], 0.6	[2.2;8], 4.5
Total Functional Scope (Function points)	–	[0;559], 22.1	[0;295], 67	[440;13810], 4655
Average Functional Scope (Function points)	–	[0;4.9], 0.4	[0;22.3], 5.7	[88;301.7], 192.4
Total Functional Scope (Number of business functions)	–	[0;543], 22.5	–	–
Average Functional Scope (Number of business functions)	–	[0;4.9], 0.4	–	–

(range, average)

For all metrics we observed in industry the results are depicted in Table 5. Therein, for each metric the ranges as well as the average value are included. For example, the N*umber of Information Flows* per application provides insights about an AL's connectedness. Although the AL of Case 3 has an application with 319 information flows—which is nearly three times as much information flows as the most connected application in Case 4 (122) has—the average number of information flows is much smaller (8.7 in Case 3 compared to 10.4 in Case 4). The standard conformity metric shows how different the analyzed ALs are regarding to the respective sourcing strategy. For example, in Case 2 the range values reveal that there exists a domain which includes only customized standard software, which is mostly an unwanted situation. Likewise, in Case 3 we can see that there exists a domain which consists only of unmodified standard software products. Nevertheless, the function point metric per application shows how the industry metric results might depend on the underlying modeling approach which complicates their comparability. Furthermore, the metric results need to be analyzed on a more fine-grained (domain-specific) level to reveal more insights. It is unlikely, for example, that a company has a general sourcing strategy for all domains. Instead, different strategies are used for different domains and therefore the metric analysis needs to be done in more detail than is possible here.

5.5. Complexity metrics correlations

Especially when metrics need to be reported to higher management or included in some kind of dashboard, it is desirable to have a minimum amount of indicators. In order to reduce the number of metrics required to achieve a holistic view on the complexity of ALs, correlations between the results of two metrics can indicate a dependence of their outcomes. Therefore, we determined the correlation coefficients for all metric combinations in all four cases. Due to page constraints, we will only report on the identified metrics highly correlating with each other.

In order to determine correlations between heterogeneity-focused metrics and industry metrics we calculated the Pearson correlation coefficient for each combination. This coefficient takes values between -1 and 1 and describes the degree of a linear dependence between two variables. In this case, we assume a high dependence if the correlation coefficient is greater than 0.4 for two complexity metrics. Table 6 summarizes the relevant coefficients for the first identified cluster of heterogeneity-focused metrics. We see strong evidence for a correlation between the *Operating System Complexity*, the *Data Base Complexity* and *the Application Type Complexity*. Therefore, we can conclude that if the variety of application types within a domain is very high, it is likely that the operating systems and databases in use are also heterogeneous. Table 7 summarizes the relevant coefficients for the industry metrics. Therein, we see strong evidence for a correlation between the *Number of Applications*, the *Total Number of Information Flows* and the *Total Functional Scope* (Function points) on the domain level. Therefore, we can conclude that the more applications are grouped in one domain, the more information flows and the more function points this domain has. While this result is not very surprising, it contributes to the plausibility of these metrics. If the amount of metrics should be reduced for reporting, then one metric per cluster could be sufficient.

In order to determine the correlation between heterogeneity-focused as well as industry metrics and topology-based metrics we used logistic regression models since the topology-based metric is not interval scaled. Although we found some significant correlations, e.g. a p-value < 0.001 for the *Functional Scope* in terms of *Business functions* in Case 3, we found no empirical evidence for a general correlation between heterogeneity-focused or industry metrics and topology-based metrics.

Table 6. Heterogeneity metrics' correlation

Metric	Operating System Complexity	Data Base Complexity
Application Type Complexity	C2: 0.67 C3: 0.49 C4: 0.30	C2: 0.61 C3: 0.43 C4: 0.62
Data Base Complexity	C2: 0.89 C3: 0.60 C4: 0.31	-

Table 7. Industry metrics' correlation

Metric	Total Number of Information Flows	Total Functional Scope (Function points)
Number of Applications	C2: 0.71 C3: 0.75 C4: 0.68	C2: 0.82 C3: 0.63 C4: 0.97
Total Functional Scope (Function points)	C2: 0.61 C3: 0.63 C4: 0.73	-

6. Interpretation of measurement results

Basically, there are two different possibilities to evaluate the metric results given in the previous section: quantitatively and qualitatively. A quantitative evaluation could, for example, measure some of the consequences of high AL complexity and then calculate the correlation coefficient for both metrics. In case of a strong correlation the respective metrics can be considered to measure complexity. But, we see some issues in such an evaluation approach. First, the cause-and-effect relationships between complexity and other variables, e.g., costs, might be non-linear, delayed in time or simply unknown. Furthermore, the cause-and-effect relations might be multi-causal instead of mono-causal meaning that there might be other causes having the same effect as well as other effects having the same cause. Instead, a qualitative evaluation could, for example, compare metric outcomes with the mental models of responsible enterprise architects. This approach would not be as sharp as a correlation based on concrete numbers but it allows access to the tacit knowledge of enterprise architects about complexity implications.

Based upon these considerations, we decided to do a qualitative evaluation in order to determine the relative value of each complexity metric under investigation. Enterprise architects of each participating company were provided with their individual metric results. We used no graphical representation, but provided the numerical metric results for all domains/applications in descending order. Thereafter, we conducted individual interviews as well as a full-day workshop to discuss the benefits and drawbacks of each type of complexity metric. The results are discussed in the following.

In general, all calculated metrics provided the expected results to the architects. That means that domains/applications considered to have a high complexity have been ranked higher by all metrics while those considered having low complexity have been ranked lower. This holds true also for the topology-based classification metrics (cf. [26]) since applications considered to be complex have been classified as core or shared while others have been classified as control or periphery. Asking for surprises we recognized that the responsible architect of Case 1 found some high ranked applications which he did not expect to appear there. After investigating the data, we found out that all those applications belonged to the IT domain. That revealed that the architect obviously considered only business applications unconsciously. Therefore, we can conclude that complexity metrics can overcome such limitations by omitting such selective perception.

According to the enterprise architects we talked to, the metrics we observed in practice are able to explain a concrete EA sometimes better than common visualizations. For example, aggregated numbers about the amount of applications per domain or their functional overlap provide deep insights for architects unfamiliar with the architecture or parts thereof. That means the metrics are able to reduce the subjective complexity of an enterprise architect. Furthermore, after in-house calculations practitioners reported on a good correlation coefficient between their individual industry metrics and occurring incidents and therefore costs. Nevertheless, some metrics are difficult to handle. For example, the customization level is usually described by a nominal scale. In most of our observed cases we found the following manifestations: *buy*, *make* and *buyAndCustomize*. But, since one manifestation is usually preferred over the others, in each case this nominal scale has been transformed into an ordinal or even interval scale. This can be a particular problem if the transformation is not based on a theoretical foundation. In Case 2, the order was *buy*, *make*, *buyAndCustomize* while it was *buy*, *buyAndCustomize*, *make* in Case 3. In fact, one group of architects considered custom built software (make) to be more complex than a customized standard product (customized) while the other group of architects considered the complexity vice versa. This example clearly demonstrates the limits of complexity metrics currently used in practice.

In our four cases, the heterogeneity-focused complexity metric (cf. [22]) was best suited to the infrastructure elements, i.e., operating systems and databases. In this area, the architects all concluded that high heterogeneity directly impacts associated cost, e.g., due to the fact that you need to have different skills and a stand-by person available all time. Furthermore, due to the generic definition, the metric can be applied on every layer of the architecture which provides a consistent way of calculating EA complexity. This allowed us, for example, to identify one metric which seems to be able to identify parts of the AL not conforming to the target architecture of information flows. The *Coupled Domain Complexity* metric provides a clearer, if not necessarily more detailed representation of the target information flow architecture. Nevertheless, in general it remains unclear which entities/attributes of the AL are best suited for heterogeneity considerations while fundamental critics have to be regarded [29].

The unanimous opinion of the five enterprise architects participating in our evaluation regarding the topology-based classification metric (cf. [26]) was that the classification indeed provides valuable insights in the structure of the AL. They praised the simplicity of the required data collection process since applications and information flows are typically gathered in every company having an EA initiative. The results can be very useful in order to determine the criticality of IT projects and decide on the architectural guidance needed. Nevertheless, it remains unclear to what extend the metric can be used to steer the evolution of the architecture.

In addition to the individual benefits and drawbacks for each investigated complexity metric we gained two additional insights during the group discussions with experts. Gathering data to document EAs is related to high effort required by data suppliers, e.g. application owners or release managers, who often do not benefit directly from providing this data. The experts we talked to confirmed that all of the investigated metrics, if used as decision support, can oblige data suppliers to provide such EA data. Furthermore, one expert decided to replace a currently used metric by the heterogeneity-based metric proposed by Schuetz et al. [22]. Instead of just counting the number of databases or programming languages currently in use, a heterogeneity-based metric seems to better satisfy his expectations.

7. Conclusion

The aim of the presented research endeavor was to identify and assess metrics to quantify application landscape complexity to be found in scientific literature and in practice. Therefore, we interviewed enterprise architects from six different companies from Germany and Switzerland and identified recurring complexity metrics extending the results of a structured literature review. As an intermediary result, we identified three groups of complexity metrics: heterogeneity-focused metrics, topology-based metrics and common industry metrics. In order to analyze respective benefits and drawbacks for each type of metric we implemented and calculated the metrics using data provided by four participating companies. The four companies all operate in the financial sector which allows a comparison of the calculated metrics. The detailed outcomes have been described in Section 5. With the results at hand, we evaluated the metrics' benefits and drawbacks by individual interviews as well as group discussions. It was apparent that the heterogeneity-focused metrics were best suited to the infrastructure layer but we also identified a use case in which the metric can be used to indicate the adherence to a given to-be architecture. The topology-based metrics were best suited to assess the criticality of change projects while their actual steering capability remained unclear. Finally, the metrics observed in industry were best suited to predict costs and increase transparency but lack a theoretical foundation. Therefore, future research should identify suitable variables usable for correlations to underpin the validity of the analyzed AL complexity metrics. This requires a time-referenced analysis of ALs, because most often the actual value of a complexity metric is not as valuable as its trend. Furthermore, future research should concentrate on how target values of complexity–from a business perspective–can be determined.

8. References

[1] J. W. Ross, P. Weill, and D. C. Robertson, Enterprise Architecture as Strategy: Creating a Foundation for Business Execution. Harvard Business Press, 2006.

[2] R. Winter and R. Fischer, “Essential Layers, Artifacts, and Dependencies of Enterprise Erchitecture,” Journal of Enterprise Architecture, 2007.
[3] J. A. Zachman, “Enterprise Architecture: The Issue of the Century,” Database Programming and Design, vol. 10, no. 3, pp. 44–53, 1997.
[4] The Open Group, “TOGAF Version 9.1,” 2011.
[5] S. Buckl, T. Dierl, F. Matthes, and C. M. Schweda, “Building Blocks for Enterprise Architecture Management Solutions,” in Practice-driven research on enterprise transformation. Springer, 2010, pp. 17–46.
[6] IEEE and ISO/IEC, “Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems (ISO/IEC 42010 IEEE Std 1471-2000),” 2007. [Online]. Available: http://dx.doi.org/10.1109/ieeestd.2007.386501
[7] C. Lucke, S. Krell, and U. Lechner, “Critical Issues in Enterprise Architecting – A Literature Review,” in Proceedings of the 16th Americas Conference on Information Systems (AMCIS), 2010.
[8] A. W. Schneider, M. Zec, F. Matthes, “Adopting Notions of Complexity for Enterprise Architecture Management,” in Proceedings of the 20th Americas Conference on Information Systems (AMCIS), 2014.
[9] A. Backlund, “The concept of complexity in organisations and information systems,” Kybernetes, vol. 31, no. 1, pp. 30-43, 2002.
[10] E. D. Beinhocker, The origin of wealth: Evolution, Complexity, and the Radical Remaking of Economics. Boston and Mass: Harvard Business School Press, 2006.
[11] N. F. Johnson, Two’s Company, Three is Complexity: A simple guide to the science of all sciences. Oxford: Oneworld, 2007.
[12] M. Janssen and G. Kuk, “A Complex Adaptive System Perspective of Enterprise Architecture in Electronic Government,” in 39th Hawaii International Conference on System Sciences (HICSS), 2006.
[13] S. Beer, Cybernetics and Management, 2nd ed. London: English Universities Press, 1967.
[14] S. Buckl, F. Matthes, and C. M. Schweda, “A Viable System Perspective on Enterprise Architecture Management,” in Proceedings of the International Conference on Systems, Man and Cybernetics. IEEE, October 2009.
[15] M. E. Zadeh, G. Millar, and E. Lewis, “Mapping the Enterprise Architecture Principles in TOGAF to the Cybernetic Concepts–An Exploratory Study,” in 45th Hawaii International Conference on System Science (HICSS), 2012, pp. 4270–4276.
[16] J. Saat, S. Aier, and B. Gleichauf, “Assessing the Complexity of Dynamics in Enterprise Architecture Planning-Lessons from Chaos Theory,” in Proceedings of the 15th Americas Conference on Information Systems, AMCIS 2009, San Francisco, California, USA, August 6-9, 2009, R. C. Nickerson and R. Sharda, Eds. Association for Information Systems, 2009.
[17] H. Kandjani, P. Bernus, and L. Wen, “Enterprise Architecture Cybernetics for Complex Global Software Development: Reducing the Complexity of Global Software Development Using Extended Axiomatic Design Theory,” in 7th IEEE International Conference on Global Software Engineering (ICGSE), 2012, pp. 169–173.
[18] H. Kandjani, P. Bernus, and S. Nielsen, “Enterprise Architecture Cybernetics and the Edge of Chaos: Sustaining Enterprises as Complex Systems in Complex Business Environments,” in 46th Hawaii International Conference on System Sciences (HICSS), 2013, pp. 3858–3867.
[19] J. Webster and R. T. Watson, “Analyzing the Past to Prepare for the Future: Writing a Literature Review,” MIS Quarterly, vol. 26, no. 2, pp. 13–23, 2002.
[20] J. Vom Brocke, A. Simons, B. Niehaves, K. Riemer, R. Plattfaut, and A. Cleven, “Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process,” in Proceedings of the 17th European Conference on Information Systems (ECIS), S. Newell, E. Whitley, N. Pouloudi, J. Wareham, and L. Mathiassen, Eds., 2009.
[21] M. Mocker, “What is Complex about 273 Applications? Untangling Application Architecture Complexity in a Case of European Investment Banking,” in 42nd Hawaii International Conference on System Sciences (HICSS). IEEE, 2009.
[22] A. Schuetz, T. Widjaja, and J. Kaiser, “Complexity in Enterprise Architectures - Conceptualization and Introduction of a Measure from a System Theoretic Perspective,” in 21st European Conference on Information Systems (ECIS), June 5-8, 2013.
[23] J. W. Ross, “Creating a Strategic IT Architecture Competency: Learning in Stages,” MIS Quarterly Executive, no. 2, pp. 31–43, 2003.
[24] T. Widjaja, J. Kaiser, D. Tepel, and P. Buxmann, “Heterogeneity in IT Landscapes and Monopoly Power of Firms: A Model to Quantify Heterogeneity,” International Conference on Information Systems (ICIS), 2012.
[25] C. Shannon, “A Mathematical Theory of Communication,” Bell System Technical Journal, no. 27, pp. 379–423, 1948.
[26] R. Lagerström, C. Y. Baldwin, A. D. Maccormack, and S. Aier, “Visualizing and Measuring Enterprise Application Architecture: An Exploratory Telecom Case,” Harvard Business School Working Paper, no. 13-103, 2013.
[27] R. Lagerström, C. Y. Baldwin, A. D. Maccormack, and D. Dreyfus, “Visualizing and Measuring Enterprise Architecture: An Exploratory BioPharma Case,” Harvard Business School Working Paper, no. 13-105, 2013.
[28] H.-G. Gadamer, “Hermeneutics and Social Science,” Philosophy & Social Criticism, vol. 2, no. 4, pp. 307–316, 1975.
[29] L. Jost, “Entropy and diversity,” Oikos, no. 113, pp. 363–375, 2006.