Complexity In Enterprise Architectures - Conceptualization And Introduction Of A Measure From A System Theoretic Perspective

Alexander Schuetz
Technische Universität Darmstadt, Darmstadt, Hessen, Germany, schuetz@is.tu-darmstadt.de

Thomas Widjaja
Technische Universität Darmstadt, Darmstadt, Hessen, Germany, widjaja@is.tu-darmstadt.de

Jasmin Kaiser
Technische Universität Darmstadt, Darmstadt, Hessen, Germany, kaiser@is.tu-darmstadt.de

Follow this and additional works at: http://aisel.aisnet.org/ecis2013_cr

Recommended Citation

This material is brought to you by the ECIS 2013 Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2013 Completed Research by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
COMPLEXITY IN ENTERPRISE ARCHITECTURES – CONCEPTUALIZATION AND INTRODUCTION OF A MEASURE FROM A SYSTEM THEORETIC PERSPECTIVE

Schütz, Alexander, Technische Universität Darmstadt, Hochschulstr. 1, 64289 Darmstadt, Germany, schuetz@is.tu-darmstadt.de

Widjaja, Thomas, Technische Universität Darmstadt, Hochschulstr. 1, 64289 Darmstadt, Germany, widjaja@is.tu-darmstadt.de

Kaiser, Jasmin, Technische Universität Darmstadt, Hochschulstr. 1, 64289 Darmstadt, Germany, kaiser@is.tu-darmstadt.de

Abstract

Measurability is the essential basis for management. Although the management of complexity is a central task of CIOs and IT-architects, literature has not proposed a concrete measure to quantify complexity in enterprise architectures so far. Against this background the contribution of this paper is a) the introduction of a system theoretic conceptualization of complexity in enterprise architectures and b) a measure to quantify complexity in enterprise architectures. We define complexity as the number and the heterogeneity of the components and relations of an enterprise architecture. The proposed measure to quantify complexity in an enterprise architecture is based on Entropy, i.e. an established measure in various fields of research such as software engineering or economics. The applicability of the conceptualization and the proposed measure to quantify complexity is demonstrated in cooperation with IT-architects of an IT organization of a leading global universal bank.

Keywords: Complexity, Enterprise Architecture, Systems Theory, Heterogeneity.
1 Introduction

Continuous adaption to changes in business requirements, introduction of new technologies, company growth or mergers often cause enterprise architectures (EA) composed of soft- and hardware solutions provided by many different IT vendors (Mattern et al., 2003). Advantages of a complex IT architecture like an adequate coverage of business requirements are opposed by higher costs (Hinton and Kaye, 1996) and an enormous effort for managing and changing a complex EA (Möller et al., 2011).

A basis for evaluating a desirable degree of an EA’s complexity and for determining actions to be taken is the measurability of an EA’s complexity. Research as well as practice point out the need for a quantification of complexity (e.g. Shafiei-Monfared and Jenab, 2012; Kearney et al., 1986), however IS literature has not mentioned a holistic, i.e. applicable to all dimensions and elements of an EA, approach to quantify complexity in EAs so far. However, there exist various approaches from other fields which might be transferable to an EA. A consistent and holistic measure to quantify complexity is easier to communicate and may lead to a higher acceptance among decision-makers. For a better understanding and a holistic view on complexity in the context of an EA, there is the need for an elementary conceptualization of complexity in EA.

The contribution of this article is two-fold: a) the conceptualization of EA complexity and b) the introduction of a measure to quantify EA complexity. Following a system theoretical perspective we consider EA as a system, consisting of components (or ‘things’) and relations. The application of a system-centric perspective is especially fruitful, since this allows us to transfer the complexity definition established in this domain: we define complexity as the number and the heterogeneity of components and relations. Thus, an increase in number and heterogeneity of components (or relations) leads to a higher EA complexity. Based on this understanding of complexity different alternatives to quantify heterogeneity are analyzed in this paper. The proposed measure to quantify heterogeneity is based on the statistical Entropy Measure which is for example used in the context of anti-monopoly legislation to determine the market power of firms or in the field of software engineering to measure software complexity. Thereby we employ methodically established studies from other fields of research. The application of the conceptualization and the introduced approach to quantify complexity is demonstrated in a real IT organization.

The paper has been divided into seven sections: The second section gives a brief overview of definitions of EA and introduces EA from a system theoretic perspective. Section three provides a definition of complexity from a system theoretic perspective and conducts a structured literature review to verify the transferability of this definition to the IS context. In the fourth section, we present a conceptualization of complexity. The following section proposes a measure to quantify complexity in EA applying the Goal Question Metric paradigm. The sixth section shows the application of the approach in a real IT landscape. Finally, we give a summary of our findings, indicate the main limitations of our approach and suggest avenues for further research.

2 Enterprise Architecture as a System

Established EA frameworks such as TOGAF (The Open Group Architecture Framework; The Open Group, 2011), ArchiMate (Lankhorst, 2004), EAP (Enterprise Architecture Planning; Spewak and Hill, 1993) and the FEAF (Federal Enterprise Architecture Framework; The Chief Information Officers Council, 1999) propose a hierarchical separation into four different enterprise architecture dimensions: business process, data/information, application and infrastructure.

Richardson et al. (1990) define EA as follows: “[Enterprise Architecture] defines and interrelates data, hardware, software, and communications resources, as well as the supporting organization required to maintain the overall physical structure required by the architecture.” Thus, Richardson et al. (1990) do not consider the business process domain as part of an EA.
In contrast, Ross (2003) includes business processes, but neglects the aspect of interrelations between the domains. “In some firms the enterprise IT architecture acts as a tool for aligning IT and business strategy. This alignment focuses on the IT components that enable critical business processes. Thus, at the enterprise level, an IT architecture is the organizing logic for applications, data and infrastructure technologies, as captured in a set of policies and technical choices, intended to enable the firm’s business strategy.” This paper adopts the holistic EA definition of Ross (2003), expanded by the relations among the four domains.

We follow the use of the term ‘system’ proposed by Hall and Hagen (1969). According to the IEEE Standard 1471-2000 (IEEE Architecture Working Group, 2000) this paper considers EA as a system, consisting of its components and its relations to each other. This can be mathematically expressed as (Klir, 2001):

\[ S = (T, R) \]  

S represents the considered system, T the set of system components (‘things’) and R the set of relations between the components. An example in the EA context could be the set of applications (T) in an applications landscape as well as the interfaces (R) between the applications.

Considering EA as a system, four subsystems can be distinguished: business, data/information, application and infrastructure architecture. The structure refers to the architecture domains mentioned in TOGAF (The Open Group, 2011). This step allows us to separately consider the complexity of each subsystem or architectural domain. The subsystem ‘business architecture’ (BA) predominantly entails business processes (Pulkkinen, 2006). ‘Data architecture’ (DA) encompasses data models and information for the support of the business architecture. While ‘applications architecture’ consists of applications, ‘infrastructure architecture’ (IA) entails all technology components (hardware and software), which provide services to the subsystems DA and AA (Spewak and Hill, 1993; The Open Group, 2011). To consider the total complexity of an EA, the interrelations between the subsystems should also be included. The relationship between BA and AA results from the support of business processes by applications. Functional requirements result in specific data models leading to a relationship between BA and DA. Examples for the relationship between AA and IA are specifications for the infrastructure (i.e. an application which needs a special operating system). Furthermore, no direct relationship between BA and IA is assumed, since DA and AA act as intermediaries between BA and IA.

3 Complexity in Enterprise Architectures – Literature Review

In order to develop a conceptualization of complexity in EA, it is necessary to clarify exactly what is meant by complexity. Section 3.1 provides a definition of complexity from a system theoretic perspective. To verify the applicability of the definition in the IS context, we conducted a structured literature review shown in section 3.2.

3.1 Complexity in Systems Science

In the field of systems science, complexity of a system is characterized by the number of components (Klir, 2001; Flood and Carson, 1993; Ashby, 1956) and relations (Flood and Carson, 1993) and the heterogeneity of components and relations (Klir, 2001; Ashby, 1956; Simon, 1962).

On this basis we define complexity \( C \) as the tuple composed of the number \( (N) \) and the heterogeneity \( (H) \) of the components and relations of an EA:

\[ C = (N, H) \]

We emphasize that – considering EA as a system – this definition is applicable to all subsystems of an EA (see section 2) as well as the relations between them. In the next section, we examine whether the understanding of complexity is confirmable in the IS literature.
3.2 Complexity in IS Literature

Based on our understanding of an EA shown in section two we conducted a structured literature review (vom Brocke et al., 2009) to identify definitions of ‘complexity’ related to an EA as a whole or its subsystems in the IS domain. An important precondition to the definition in demand is that it has to comply with the above mentioned subsystems of an EA.

3.2.1 Literature Search Process

Following vom Brocke et al. (2009), the literature review began with a conception of the topic (cf. section 2) and definitions of the key terms. The above mentioned literature on EA was consulted to identify the relevant keywords. After analyzing these sources we discussed adequate terms to gain a common understanding. Applying an iterative process the set of relevant search terms was expanded with synonyms and connected by logical expressions. To define the final search expression we introduced wildcards to consider varying spellings and endings. The following search expression for the EBSCOhost database exemplifies the derived queries: “AB complex* AND AB ((IT OR System OR landscape OR architecture OR infrastructure OR hardware OR platform OR application OR software OR data))”

We only considered contributions published in top ranked journals. Therefore we consulted the AIS Basket of Eight and the top 25 journals of the MIS Journal Rankings (AIS, 2012). After all we searched the following databases by title and abstract: EBSCOhost (Business Source Premier), Science Direct, ISI Web of Knowledge (Web of Science database) and the search engines of the ACM and IEEE. In addition the conference proceedings of ECIS and ICIS were searched through (via the AIS Electronic Library).

From our keyword search (2012-07-19 until 2012-07-25) we retrieved 3963 contributions (3664 journal contributions and 299 conference papers). We screened the title and the abstract and removed publications obviously not related to EA or only using the term ‘complexity’ as filler and not treating ‘complexity’ as a central subject matter (2012-07-25 until 2012-08-10). Afterwards, we conducted an intensive full text screening of the remaining contributions (2012-08-13 until 2012-10-12). We relied on the following exclusion criteria: (a) studies that investigated computational or query complexity (dealing with formal mathematical analysis of algorithm efficiency or use of machine resources), (b) studies that explored psychological complexity by analyzing software characteristics affecting programmer performance (as defined by Curtis (1980)), and (c) studies addressing a dynamic aspect of software complexity such as control flow complexity (e.g. expansions of McCabe’s cyclomatic number (McCabe, 1976)). Our search and review process ended with 39 journal contributions and 8 conference papers which were analyzed in a next step.

3.2.2 Literature Analysis

The identified contributions were analyzed on how the authors deal with or define complexity. According to Webster and Watson (2002), we apply a concept matrix to categorize the papers. As concepts the criteria to define complexity were used which resulted from the analysis of the contributions. Essentially, these have been ‘number of components’, ‘heterogeneity of components’, ‘number of relations’, ‘heterogeneity of relations’, ‘change over time of components’ and ‘change over time of relations’. Table 1 briefly summarizes the result of the analysis:

<table>
<thead>
<tr>
<th></th>
<th>Number of components</th>
<th>Heterogeneity of components</th>
<th>Number of relations</th>
<th>Heterogeneity of relations</th>
<th>Change over time of components</th>
<th>Change over time of relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occurrence</td>
<td>43</td>
<td>31</td>
<td>24</td>
<td>10</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 1 Statistical Summary of the Concept Matrix
It is apparent from the literature review that very few of the identified contributions cover all parts of the proposed definition. For example, in the dimension ‘data architecture’ a mentioned definition for data complexity is “[...]the number of data elements per unit of functionality” (Banker and Slaughter, 2000). Considering the ‘application architecture’ complexity of e.g. a program is associated with the “number of its components or elements, kind or type of elements [and] structure of the relationship between elements” (Davis and LeBlanc, 1988). Noteworthy exceptions are the contributions of Schneberger and McLean (2003) and Henningsson and Hanseth (2011). On the ‘infrastructure architecture’ level “[c]omplexity can be defined here as the dramatic increase in the number and heterogeneity of included components, relations, and their dynamic and unexpected interactions in IT solutions” (Henningsson and Hanseth, 2011). ‘Computing complexity’ can be defined on the basis of the number and variety of components and interactions plus the rate of change of these (Schneberger and McLean, 2003).

Schneberger and McLean (2003) and Henningsson and Hanseth (2011) focus on the conceptual examination of complexity in their work. We want to go one step further and introduce a holistic conceptualization of complexity in EA and a measure to quantify this complexity. In the current study, we focus on an static examination which can be applied on both an as-is and on a to-be EA (Aier et al., 2011). During the analysis of the identified contributions it turned out that only few methods were presented to quantify complexity and the existing methods merely cover parts of an EA, not the EA as a whole (Booch, 2008; Vardi, 1982; Mocker, 2009). Often the application is so specific that it is not possible to transfer the method to other dimensions of an EA.

4 Conceptualization of Complexity in Enterprise Architectures

Based on prior considerations, this section shows the conceptualization of complexity in EA. For this, we combine the concept of interrelated subsystems of an EA (section 2) with our definition of complexity (section 3.1). From a system theoretic perspective, EA can be expressed as:

$$S_{EA} = (T_{EA}, R_{EA})$$

(3)

$T_{EA}$ represents the components of the EA system ($S_{EA}$) and $R_{EA}$ denotes the relations between the components $T_{EA}$. Similarly, the subsystems ‘business architecture’ (BA), ‘data architecture’ (DA), ‘applications architecture’ (AA) and ‘infrastructure architecture’ (IA) can be expressed as a system:

$$S_{X} = (T_{X}, R_{X}), \text{ with } X \in \{AA, BA, DA, IA\}$$

(4)

As previously introduced in section 3.1, complexity (C) of an EA is defined as tuple of number (N) and heterogeneity (H) of the components or relations of an EA. Since a system covers components as well as their relations, both, number and heterogeneity of components as well as number and heterogeneity of relationships influence the EA’s complexity. A measure of complexity therefore has to include the number N and heterogeneity H of components T and relations R. This can be formulated as follows:

$$C_{Y} = (N_{Y}, H_{Y}), \text{ with } Y \in \{T, R\}$$

(5)

The conceptualization proposed in this section is applicable to various perspectives (system or subsystem). An example of a subsystem of an EA is AA, in which the applications represent the components ($T_{AA}$) and the interfaces represent the relations ($R_{AA}$) within the subsystem ($S_{AA}$).

5 A Measure to Quantify Complexity in Enterprise Architectures

In literature different approaches to define a measure exist (e.g. Fenton and Pfleeger, 1998; Van Solingen and Berghout, 1999). This article adopts the Goal Question Metric paradigm (Basili and

---

1 Due to space constraints the full concept matrix is not included, but the authors would be pleased to provide it on request.
Rombach, 1988) to thoroughly develop a measure to quantify complexity in EA, based on the introduced conceptualization of complexity. The Goal Question Metric (GQM) is a systematic ‘goal-oriented’ approach proposed by Basili and Rombach (1988) to derive metrics, primary used in the field of software engineering. Hereby GQM adopts a three-step approach: (a) GQM defines a goal based on certain criteria, (b) refines the goal into questions, and (c) defines metrics appropriate to provide answers to these questions in a measurable way. We adopt this approach from the field of software engineering to define an abstract measure to quantify complexity in EA (as a system). For this we define the following goal using a template defined by Basili and Rombach (1988): Analyze ‘EA as a system’ in order to ‘characterize it’ with respect to ‘complexity’ from the point of view of ‘IT-architects’. Based on the conceptualization of complexity in EA (cf. section 4) we refine the following questions from the stated goal: “What is the number of components and relations of the system?” (Q1) and “What is the heterogeneity of components and relations of the system?” (Q2).

To be consistent with the above mentioned definition of complexity in EA as a system, the measures have to fulfill the following requirements:

- The greater the number N – while assuming a constant heterogeneity H – of components or relations of an EA, the higher the complexity of an EA. (cf. Q1)
- The greater the heterogeneity H – while assuming a constant number N – of components or relations of an EA, the higher the complexity of an EA. (cf. Q2)

Whilst N is relatively easy to assess (cf. section 5.2) H (i.e. Q2) needs to be considered in detail.

### 5.1 Introduction of a Measure to Quantify Heterogeneity in Enterprise Architectures

The term ‘heterogeneity’ is applied in a variety of contexts of IT. According to an extant literature review conducted in Widjaja et al. (2012, p. 5) we use the following generic definition of heterogeneity in IT landscapes: “Heterogeneity in IT landscapes is a statistical property and refers to the diversity of attributes of elements in the IT landscape” (Widjaja et al., 2012, p. 4).

As an example we examine the heterogeneity of database management systems (DBMS) a company has in use. This heterogeneity can be understood as a frequency distribution and can be expressed graphically as shown in Figure 1. Each of the bars represents a characteristic a certain DBMS-instance can be assigned to.

![Figure 1](image.png)

**Figure 1** Heterogeneity of database management systems expressed as absolute frequencies

A common way to measure heterogeneity is using concentration measures. In the field of anti-trust legislation the ‘Herfindahl-Hirschman Index’ is used by the American authorities to calculate the market concentration (U.S. Department of Justice and the Federal Trade Commission, 2010). Besides the measurement of the market power of firms (Kwoka Jr, 1985), they are used in many different economic sectors, for instance to measure the geographic concentration of industries (Garrison and Paulson, 1973). In the scientific literature there already exists preliminary work regarding concentration measures and even requirements on them. This paper transfers these concepts to the EA context. Therefore these existing requirements are applied to form a list of desirable criteria the potential heterogeneity measures should satisfy:
Table 2  Selected requirements for a measure to quantify heterogeneity

We analyzed three frequently used concentration measures in regard to the requirements listed in table 2. The results of the analysis are listed in table 3. In the following $p_i$ represents the relative frequency of characteristic $i$ of the considered element of the IT landscape, where $x_i$ denotes the absolute number of elements assigned to characteristic $i$: $p_i = \frac{x_i}{\sum_{j=1}^{n} x_j}$

Table 3  Analysis of frequently used measures

Since our analysis showed that the Entropy Measure meets the requirements quite well, we suggest to use this measure to quantify EA heterogeneity. Furthermore, this measure is also well established in the field of software measurement (Berlinger, 1980; Davis and LeBlanc, 1988; Harrison, 1992). Applied to the example shown in Figure 1, $p_i$ denotes the relative frequency of a certain database management system. The resulting Entropy Measure for the example is a value of $EM = - \left( \frac{15}{90} \ln \left( \frac{15}{90} \right) + \frac{30}{90} \ln \left( \frac{30}{90} \right) + \frac{15}{90} \ln \left( \frac{15}{90} \right) + \frac{15}{90} \ln \left( \frac{15}{90} \right) \right) = 1.355$. The Entropy Measure takes the minimum value 0 ($EM = -(1 \cdot \ln(1))$) in the case of all values being associated with exactly one characteristic and reaches its maximum of 1 for an equal distribution of the values on $n$ characteristics (in our example $\ln(4) = 1.386$). To support analysis and interpretations, the numbers-equivalent Entropy Measure can be additionally used:
\[ EM_A = \exp(EM) \]  

\( EM_A \) is equivalent to the number of values that would lead to the same value of \( E \) for an equal distribution of \( EM \) values. \( EM_A = 5 \) means, for example, that the same value of Entropy Measure would be reached by a distribution with five values, each with a share of 20\%. Note that this property can be used to facilitate the interpretation of the index values but further conclusions have to be drawn with care: The distributions \{5\%, 5\%, 10\%, 80\%\} and \{50\%, 50\%\} lead for example to nearly the same value of the numbers-equivalent Entropy Measure (2.03 respective 2.0) but could be related to completely different benefits and costs. This effect could be mitigated by stating the number of characteristics that were considered in addition to the \( EM \) respectively the \( EM_A \).

5.2 Quantification of Complexity in Enterprise Architectures

To stay consistent with our system theoretic perspective, we define heterogeneity of components \( T \) (\( H_T \)) respectively of relations \( R \) (\( H_R \)) as follows:

\[ H_T := (EM_T, |T|) \quad (7) \]
\[ H_R := (EM_R, |R|) \quad (8) \]

\( EM_T \) and \( EM_R \) represent the Entropy Measures of components \( T \), respectively of relations \( R \). \( |T| \) and \( |R| \) are the particular cardinal numbers of the sets \( T \) and \( R \) and therefore represent the number of characteristics which occur at least once. They serve as reference values for the numbers-equivalent Entropy Measures of components \( T \) (\( EM_{A,T} \)) and relations \( R \) (\( EM_{A,R} \)). We define equations 7 and 8 as the metrics M2 to answer question Q2 as stated in section 5.

The number of components \( (N_T) \) and relations \( (N_R) \) result from counting the particular elements of the considered set \( T \) or \( R \). Contrary to the cardinal numbers of the sets, identical elements get counted a bunch of times. As an example we have a look at a system ‘Database systems’: The components \( T \) are the elements of the set of database instances in use \{IMS, IMS, DB2, DB2, IMS, IMS, DB2\}. The number of components \( (N_T) \) is equal to seven. The cardinal number of the set \( |T| \) depicts the number of different technologies in use (two). We define this calculation of the number of components \( (N_T) \) and relations \( (N_R) \) as the metric M1 to answer question Q1 as stated in section 5.

Combining M1 and M2 with equation (5) leads to the following complexity measures \( C_T \) and \( C_R \) which also represent the abstract measures derived from the GQM process:

\[ C_T = (N_T, (EM_T, |T|)) \quad (9) \]
\[ C_R = (N_R, (EM_R, |R|)) \quad (10) \]

We emphasize that we did not integrate a specific context when defining the goal in the GQM process. The derived measure is expected to work for each sub-system of an EA. Hence, IT-architects – as the main stakeholders – are enabled to apply the measure to various ‘application fields’ by (a) defining a specific system (e.g. IA) and (b) the components or relations (e.g. the databases) and (c) by choosing the characteristics on which the components \( T \) or relations \( R \) of a (sub-) system \( S \) can be distinguished (e.g. the vendors). Ideally the measurement data can be extracted from a central data repository, an EA management tool or has to be collected from peripheral organized sources by the IT-architects.

6 Application of the Model

We evaluated the applicability of the proposed approach in the IT organization of a leading global universal bank by instantiating the proposed abstract measure in two contexts. The model was introduced to the IT-architects of the bank in joint workshops. The bank relies on an IT landscape that is structured by a domain model. The domains consist of subdomains which are furthermore subdivided into functional units. Each application of the IT landscape is assigned to exactly one functional unit. All analyses presented in this section are exclusively based on information extracted from an ARIS repository used in the company. In order to evaluate the presented measure to quantify
complexity in different dimensions of the EA, the application architecture (subsystem AA) as well as the infrastructure architecture (subsystem IA) was analyzed. On the AA level we calculated the complexity of interfaces between the applications regarding their implementation and functional complexity. For the IA, the complexity of the databases and the operating systems were analyzed on functional unit level. Due to space constraints we focus on the presentation of the analysis of the components in the infrastructure architecture. The IT-architects chose the vendors and the version as characteristics to distinguish the elements. In collaboration with the IT-architects we defined the following goal (cf. the description of the GQM process in section 5): Analyse the ‘infrastructure architecture’ in order to ‘characterize it’ with respect to ‘the component induced complexity of the databases and operating systems on functional unit level’ from the point of view of ‘IT-architects’. As measures we used instances of the proposed complexity measure (equation 9): $C_{T,OS} = \left( N_{T,OS}, (EM_{T,OS}, |T_{OS}|) \right)$ and $C_{T,D} = \left( N_{T,D}, (EM_{T,D}, |T_D|) \right)$.

We examined 136 functional units of 32 subdomains regarding the database management systems (DBMS) and the operating systems (OS) in use. For this we identified the applications assigned to each of the functional units. Then we determined the corresponding databases and the required OS. Based on this information we calculated the complexity measures for all functional units regarding the productive DBMS and the OS. The analysis revealed the distributions shown in tables 4 and 5.

Table 4 Distribution of productive database management systems

| DBMS 1 | DBMS 2 | DBMS 3 | DBMS 4 | DBMS 5 | DBMS 6 | DBMS 7 | DBMS 8 | $N_{T,D}$ | $EM_{T,D}$ | $|T_D|$ | $EM_{A,T,D}$ |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|--------|-------|-----------|
| 316    | 72    | 94    | 25    | 326   | 29    | 117   | 405   | 1384   | 1.74   | 8       | 5.68      |

Table 5 Distribution of productive operating systems

| OS 1  | OS 2  | OS 3  | OS 4  | OS 5  | OS 6  | OS 7  | OS 8  | OS 9  | OS 10 | OS 11 | OS 12 | OS 13 | OS 14 | OS 15 | OS 16 | $N_{T,OS}$ | $EM_{T,OS}$ | $|T_{OS}|$ | $EM_{A,T,OS}$ |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|------------|---------|-------------|
| 11    | 74    | 4     | 63    | 725   | 115   | 39    | 379   | 484   | 31    | 556   | 1     | 53    | 18    | 27    | 2     | 2582     | 1.93      | 16      | 6.92       |

In cooperation with the bank’s IT-architect and based on the proposed approach to quantify complexity, we analysed the seven domains and 32 subdomains. In this context we identified several needs for action on IA level which confirmed the IT-architect’s – subjective – expectation. For one of the domains this paper goes into detail. The considered domain consists of four subdomains which on their part consist of 13 functional units. Table 6 summarizes the calculated measures for the complexity of the database management systems and the operating systems. The distance between $|T_X|$ and $EM_{A,T,X}$ is an indication for the cause of the heterogeneity and therefore complexity. A relatively high distance implies that a large proportion of applications distribute to a relatively small number of DBMS or OS. The heterogeneity in this case is less caused by an equal distribution but rather by the number of DBMS- or OS-types ($|T_X|$). However, a small distance suggests an equal distribution to be the reason for heterogeneity. Regarding the OS, a relatively high distance between $|T_{OS}|$ (12) and
\( EM_{A,T,OS} \) (5.29) can be noticed on domain-level. A closer look showed that there exist four OS-types which are only required by one application in each case. The IT-architect considered these to be candidates for a migration to one of the other OS-types.

| Subdomain | NT,OS | EM,OS | \(|T,OS|\) | EM,\(A,T,OS\) | NT,D | EM,D | \(|T,D|\) | EM,\(A,T,D\) |
|-----------|-------|-------|-----------|--------------|-------|-------|-----------|--------------|
| Subdomain 1 | 178 | 1.41 | 9 | 4.08 | 82 | 1.07 | 6 | 2.92 |
| Subdomain 2 | 48 | 1.52 | 6 | 4.56 | 37 | 1.42 | 6 | 4.14 |
| Subdomain 3 | 89 | 1.50 | 6 | 4.48 | 76 | 1.36 | 5 | 3.89 |
| Subdomain 4 | 24 | 1.29 | 7 | 3.63 | 19 | 0.88 | 3 | 2.41 |
| Domain | 339 | 1.67 | 12 | 5.29 | 214 | 1.54 | 7 | 4.69 |

Table 6  Aggregated presentation of the results

Regarding the DBMS the analysis showed a similar case. The distance of \( |T,D| \) and \( EM_{A,T,D} \) of the subdomains 1, 2 and 3 indicated a potential to degrease heterogeneity and therefore complexity. Again the IT-architect confirmed a need for action. In each of these subdomains the complexity can be reduced by a migration of one, respectively two DBMS to an alternative DBMS. Furthermore this would reduce the complexity (\( |T,D| \) potentially degreased to a value of 6) on domain-level, because one DBSM would be removed.

Based on the measures the IT-architects were able to derive various needs for action (i.e. investment decisions, projects to clarify the underlying causes of the observed complexity and the need for a companywide measurement tool of EA complexity). Beside the identified needs for action the IT-architects emphasized the advantage of the holistic approach. Because of consistent usage of the measure, the results of the analysis are easier to communicate to other stakeholders and especially the business organization.

7 Conclusion, Limitations and Further Research

By adopting a system theoretic perspective to the EA context we derived the – to the knowledge of the authors – first holistic conceptualization of complexity of an EA: we define complexity as the number and the heterogeneity of the components and relations of an EA. Based on the idea that the prerequisite of any management activities is the measurability of the respective artifact’s characteristic, we proposed a measure to quantify complexity in EA which is applicable to all dimensions (or subsystems) of an EA. Using the Entropy Measure we employed a methodically established concentration measure from other disciplines. To demonstrate the applicability and the practical benefits of the conceptualization and the proposed measure we applied the model in one case company from the financial sector.

From a practical point of view we developed a basis for an adaptable measurement system (according to company or industry-sector-specific requirements). IT-architects and CIOs can freely form the subsystems of an EA whose complexity is to be determined. On that basis a perceived complexity of an EA can be quantified and compared. Moreover the results can be used as a basis for decision-making in view of refining the EA. This may include decisions with respect to the further development of the architecture to prevent the EA from an “uncontrolled growth”, to allow for a better usage of the existent architecture which leads to a higher efficiency and to achieve a higher flexibility regarding the implementation of new IT requirements. Another benefit of the introduced measure is the facilitated interpretation of the instantiated measures for a specified context.

A limitation of the presented paper is the static perspective on EA. Dynamic effects like the rate of change mentioned by Schneberger and McLean (2003) were not subject of our analysis and could be a fruitful extension of the proposed model. Furthermore, the proposed measure is not suitable for dynamic processes (e.g. the examination of the control flow complexity or non-deterministic business processes). Avenues for further research are especially the development of a taxonomy of components.
and relations, for which the complexity is typically to be analyzed. So far, a generally accepted opinion on which subsystems are the most relevant regarding complexity management is not existent. The objective of our future work is to combine the proposed conceptualization and the measure with this taxonomy in order to derive a best practice approach to complexity management in EA. In a next step this approach would facilitate a benchmarking of different EAs (e.g., in a specific industry sector). In a related project we are currently implementing a dashboard for the measurement of EA-complexity (on the basis of the proposed measures) in collaboration with the IT-architects of the bank mentioned in section 6. Further research might also explore the effect of complexity on IT flexibility and IT efficiency. In sum, we hope that the proposed conceptualization will serve as a springboard for future research studies on the business impact of EA complexity and also aid practitioners in better understanding their EA complexity management.

References


