Your Landscape in Shape
Scape Enterprise Architecture Office and Consulting

Dr. Christian Schmidt

How to Measure Enterprise Architecture Complexity:
A Generic Approach, Practical Applications, and Lessons Learned

The Open Group Conference – London 2013

London, October 21st, 2013
How to deal with complexity?
Questions to be answered

What does complexity **mean** to the enterprise architect?

How can complexity be **measured**?

Who is **interested** in complexity analyses?

What are the possible **usage scenarios** for complexity measures?

Which **practical experiences** have been made?

What are the **lessons learned**?
Research cooperation
Research projects that we are currently involved in

**Project 1:** (started early 2012)

**CEAR:** *Complexity in Enterprise Architectures*

- how can complexity be defined and quantified in the context of enterprise architectures?
- what impact does complexity have on the enterprise success?
- which principles should be followed in the management of complexity

**Project 2:** (started early 2013)

**CALM3:** *Complexity of Application Landscapes – Models, Measures, Management*

- what does complexity mean in the context of application landscapes?
- what are the drivers for complex application landscapes?
- how can the complexity of application landscapes be measured?
- how can complexity models be used for application landscape planning?
Agenda

1. The Role of Complexity
2. Measuring EA Complexity
3. Case Example: Application Landscape Planning
4. Lessons Learned and Recommendations
Facts on enterprise architecture complexity
Some observations from enterprise reality

- A large industry firm has taken records of over 4,000 business applications
- A large commercial bank employs more than 25 different DBMS versions
- A universal bank has not turned off batches for years due to unknown side effects
- Contrary to the IT strategy, the board of an insurance firm was equipped with iPads
- The capital markets division of a public sector bank developed its own data warehouse
- During an audit, a reinsurance group identified 50 previously unknown EUC applications
- The EA tool of an insurance firm reached three times the expected size after only one year
- During a transformation project, the number of known as-is applications more than doubled
- Two architects of a pharma group, who attended a conference, had never heard of each other

Complexity is everywhere
Complexity is created constantly
Complexity is generally underestimated
The law of rising complexity
Complexity grows over time if no specific action is taken

According to the **Second Law of Software Evolution**, the complexity of a software system in use will increase with time if no explicit action is taken to avoid this [Le97]. It may be argued that the law will also hold on a **macro level** (as an aggregation of the many individual evolution processes) [SB11].
The central role of complexity
Drivers and outcomes of IT complexity

The **TOP 5 drivers and outcomes** of IT complexity according to a survey within the CALM3 working group [Sc13].

1. Local optimization of business departments
2. Rising business complexity
3. Increasing legal regulation
4. Technological progress
5. Short-term optimization of business departments

1. Increase of communication efforts
2. Increase of development costs
3. Increase of maintenance costs
4. Increase of implementation periods
5. Increase of start-up times

**CALM³**
Complexity of Application Landscapes – Models, Measures, Management
The role of the enterprise architect
Active management of the complexity surplus

Due to the negative impacts of complexity, architectures should only be as complex as required. It is a core responsibility of the architect to manage the complexity surplus.

However, non-functional requirements (like vendor dependency, security) and specific context factors (like the business model or sourcing strategy) need to be taken into account as well.
Minimal complexity is often neither required nor appropriate. Instead, the **optimal level of complexity** may vary across **architecture layers and domains**.

<table>
<thead>
<tr>
<th>Low Complexity</th>
<th>High Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>- lower operating costs</td>
<td></td>
</tr>
<tr>
<td>- lower maintenance costs</td>
<td></td>
</tr>
<tr>
<td>- lower procurement costs</td>
<td></td>
</tr>
<tr>
<td>- higher agility / reduced change efforts</td>
<td></td>
</tr>
<tr>
<td>- etc.</td>
<td></td>
</tr>
</tbody>
</table>

For differentiating front-end domains, for instance, a higher level of complexity may be appropriate than for non-differentiating back-end domains.
Agenda

1. The Role of Complexity
2. Measuring EA Complexity
3. Case Example: Application Landscape Planning
4. Lessons Learned and Recommendations
The different faces of complexity

Stakeholders and their view of complexity

Complexity is a fuzzy term. Different stakeholders usually have different views and conceptions.

- **Chief Information Officer (CIO)**
  - Number of Services
  - Number of Vendor Relations

- **Line of Business Manager**
  - Number of Products
  - Number of Process Variants

- **Enterprise Architect**
  - Applications per Capability
  - Number of Platforms
  - Number of Interfaces

- **Domain Architect**
  - Number of Applications (within Domain)
  - Number of Interfaces (within Domain)

- **Head of Solution Development**
  - Number of Applications
  - Number of Development Lines
  - Number of Interfaces

- **Software Developer**
  - Number of Frameworks
  - Lines of Code

- **Head of Operations**
  - Number of Platforms
  - Number of Dependencies

Illustration exemplary.
Traditional ways to measure complexity
Use of specific complexity indicators

Existing approaches to measure EA complexity are mostly based on a set of **specific complexity indicators**. This has a number of shortcomings:

- Number of Applications
- Number of Development Lines
- Number of Interfaces

- Number of Services
- Number of Vendor Relations

- Applications per Capability
- Number of Platforms
- Number of Interfaces

- Number of Platforms
- Number of Dependencies

- Number of Products
- Number of Process Variants

- Number of Applications (within Domain)
- Number of Interfaces (within Domain)
According to a literature review (47 relevant publications out of 4,000 articles in international IS journals and conferences), most authors associate system complexity with the following aspects [SWK13]:

The structural complexity of an Enterprise Architecture (interpreted as a system) may hence be defined as a tuple $C_{EA} = (N_T, H_T, N_R, H_R)$. 

see also [SM03]

---

**Generic approach proposed by CEAR**

Common definition of structural complexity
Quantifying heterogeneity
Heterogeneity as a statistical property

Following the above approach, the problem of quantifying complexity is **reduced** to quantifying heterogeneity.

Heterogeneity can be captured as a statistical property and be described by means of empirical **frequency distributions**.

Based on this, well-studied concentration measures may be adopted from other research domains like economics / monopoly regulation (e.g., Herfindahl-Hirschman-Index) or biology (e.g., Simpson-Index) [Wi12].

**Definition:**

**Heterogeneity** is defined as the diversity of elements or relationships of a system with respect to certain characteristics (attribute values).
Quantifying heterogeneity
Measure requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Scenario A</th>
<th>Scenario B</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional Characteristics</td>
<td><img src="image1.png" alt="Graph" /></td>
<td><img src="image2.png" alt="Graph" /></td>
<td>Het(A) &lt; Het(B)</td>
</tr>
<tr>
<td>Shift in the Classification</td>
<td><img src="image3.png" alt="Graph" /></td>
<td><img src="image4.png" alt="Graph" /></td>
<td>Het(A) &lt; Het(B)</td>
</tr>
<tr>
<td>Impact of Small Shares</td>
<td><img src="image5.png" alt="Graph" /></td>
<td><img src="image6.png" alt="Graph" /></td>
<td>Het(A) &lt; Het(B)</td>
</tr>
<tr>
<td>No Impact of Proportional Changes</td>
<td><img src="image7.png" alt="Graph" /></td>
<td><img src="image8.png" alt="Graph" /></td>
<td>Het(A) = Het(B)</td>
</tr>
</tbody>
</table>

[SWK13]
Quantifying heterogeneity
The entropy measure

The requirements are best met by the **Entropy Measure** as introduced by Shannon [Sh48] [JB79] [SWK13]:

$$EM = \sum_{i=1}^{n} f_i \ln \left( \frac{1}{f_i} \right) \quad \text{with} \quad f_i = \frac{x_i}{\sum_{j=1}^{n} x_j}$$

Interpretation of the Entropy Measure is facilitated by the so-called **Numbers Equivalent Entropy Measure**, which denotes the equivalent number of characteristics at equal distribution:

$$EM_A = e^{EM}$$

$$EM_A = e^{1.12} = 3.06$$
Operationalization
Model application based on EA repository data

In order to apply the model, a further **differentiation** of element and relationship types is necessary. Beyond that, data is required in an appropriate form.

**Architecture repositories** with their data structured along **EA metamodels** provide the ideal basis for data collection and the calculation of complexity measures [SWS13].
Complexity aspects
Types of complexity aspects in metamodels

Type 1: Element Types

- **ET 1**
  - Attr. 1
  - Attr. 2
  - …

  **Number** of element instances of a certain type

  **Heterogeneity** of element instances of a certain type with respect to a defined characteristic (attribute of the type)

Type 2: Relation Types

(***Special Case of Type 3**)

- **ET 1**
  - Attr. 1
  - Attr. 2
  - …

- **ET 2**
  - Attr. 1
  - Attr. 2
  - …

  **Number** of relation instances of a certain type

  **Heterogeneity** of relation instances of a certain type with respect to a defined characteristic of one of the element types involved

Type 3: Path Types

- **ET 1**
  - Attr. 1
  - Attr. 2
  - …

- **ET ...**
  - Attr. 1
  - Attr. 2
  - …

- **ET n**
  - Attr. 1
  - Attr. 2
  - …

  **Number** of path instances of a certain type

  **Heterogeneity** of path instances of a certain type with respect to a defined characteristic of one of the element types involved

[SW13]
Example: Complexity aspect of type 1
Applications by vendor

\[ N = 4 \quad H = 1.39 \]

\[ N = 4 \quad H = 1.04 \]
Example: Complexity aspect of type 1
Applications by vendor

\[
N = 4 \quad H = 1.39
\]

\[
N = 3 \quad H = 1.10
\]
Example: Complexity aspect of type 2/3
Application usage by applications

\begin{align*}
N = 4 & \quad H = 0.69 \\
N = 2 & \quad H = 0.69
\end{align*}
Example: Complexity aspect of type 2/3
Application usage by applications

Consolidation

SAP FI

UK Branch

XIA

DK Branch

N = 2       H = 0.69

SAP FI

XIA

N = 2       H = 0
Agenda

1. The Role of Complexity
2. Measuring EA Complexity
3. Case Example: Application Landscape Planning
4. Lessons Learned and Recommendations
Case example
Application landscape planning

Project Outline

Mission: Develop Target Application Landscape
Industry: Special Insurance
Period: May – Dec 2012
Drivers: high redundancy, poor acceptance of core systems, extensive use of EUC components (Access, Excel, etc.)
Objectives: elimination of EUC applications, regional consolidation, technological standardization and alignment with group standards, introduction of a centralized integration platform
Methodology: TOGAF ADM / ADD [TOG13]

[SWS13]
Case example
Metamodel of the organization
Case example
Metamodel and defined complexity aspects

Dr. Christian Schmidt  |  How to Measure Enterprise Architecture Complexity   |  Open Group Conference  |  October 21st, 2013
### Defined complexity aspects

Interpretation of number and heterogeneity (1)

<table>
<thead>
<tr>
<th>Complexity Aspect</th>
<th>Interpretation N</th>
<th>Interpretation H</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Application System</strong></td>
<td>Number of all existing (productive) applications</td>
<td>Concentration of applications by vendors</td>
</tr>
<tr>
<td>by Application Vendor (Type 1)</td>
<td>Decreases with the decommissioning of applications</td>
<td>Decreases with a shift from special to standard / mainstream vendors</td>
</tr>
<tr>
<td><strong>2 Application Interface</strong></td>
<td>Number of all existing application interfaces</td>
<td>Concentration of interfaces by interface technologies</td>
</tr>
<tr>
<td>by Interface Type (Type 1)</td>
<td>Decreases with the decommissioning of application interfaces</td>
<td>Decreases with a shift from special to standard technologies</td>
</tr>
<tr>
<td><strong>3 Function Implementation</strong></td>
<td>Number of function implementations through applications</td>
<td>Concentration of function implementations by applications</td>
</tr>
<tr>
<td>by Application Name (Type 2)</td>
<td>Decreases with the removal of function implementations</td>
<td>Decreases with a shift of functions from peripheral to core applications</td>
</tr>
<tr>
<td><strong>4 Application Usage</strong></td>
<td>Number of application uses by organization units</td>
<td>Concentration of application usage by applications</td>
</tr>
<tr>
<td>by Application Name (Type 2)</td>
<td>Decreases with the reduction of application uses by organization units</td>
<td>Decreases with a usage shift from ‘local’ to ‘global’ applications</td>
</tr>
</tbody>
</table>
## Defined complexity aspects

### Interpretation of number and heterogeneity (2)

<table>
<thead>
<tr>
<th>Complexity Aspect</th>
<th>Interpretation N</th>
<th>Interpretation H</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5 LoB Support</strong></td>
<td>Number of business line supports by applications</td>
<td>Concentration of LoB support by applications</td>
</tr>
<tr>
<td><strong>by Application Name</strong></td>
<td>Decreases with a reduction of business line support by applications</td>
<td>Decreases with a shift from LoB specific to ‘global’ applications</td>
</tr>
<tr>
<td><strong>(Type 2)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>6 Information Usage</strong></td>
<td>Number of information object usages through applications</td>
<td>Concentration of information usage by applications</td>
</tr>
<tr>
<td><strong>by Application Name</strong></td>
<td>Decreases with a reduction of inf. object usage through applications</td>
<td>Decreases with a shift of data from peripheral to core applications</td>
</tr>
<tr>
<td><strong>(Type 2)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>7 Interface Usage</strong></td>
<td>Number of interface usages by applications</td>
<td>Concentration of interface usage by interfaces</td>
</tr>
<tr>
<td><strong>by Interface Name</strong></td>
<td>Decreases with a reduction of the interface usage by applications</td>
<td>Decreases with a shift from special to (reusable) standard interfaces</td>
</tr>
<tr>
<td><strong>(Type 2)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>8 Platform Usage</strong></td>
<td>Number of platform usages by applications</td>
<td>Concentration of platform usage by platforms</td>
</tr>
<tr>
<td><strong>by Platform Name</strong></td>
<td>Decreases with a reduction of the platform usage by applications</td>
<td>Decreases with a shift from special to standard platforms</td>
</tr>
<tr>
<td><strong>(Type 3)</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Measurement results
Development of complexity measures (from as-is to target)

Legend
- N (delta in %)
- H (delta in %)
## Results

### Interpretation of measure variation (1)

<table>
<thead>
<tr>
<th>Complexity Aspect</th>
<th>As-is / to-be</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 Application System by Application Vendor</strong> (Type 1)</td>
<td>(51; 3.90) → (23; 3.08)</td>
<td>Elimination of EUC components and special solutions results in a substantial reduction of N. H remains high as the target architecture is still mainly based on individual applications.</td>
</tr>
<tr>
<td><strong>2 Application Interface by Interface Type</strong> (Type 1)</td>
<td>(40; 3.36) → (39; 0.66)</td>
<td>Interface consolidation and introduction of new interfaces keep the balance. Introduction of centralized integration platforms (EAI / ETL) results in a significant reduction of H.</td>
</tr>
<tr>
<td><strong>3 Function Implementation by Application Name</strong> (Type 2)</td>
<td>(148; 3.09) → (108; 2.50)</td>
<td>Functional disentanglement (mainly in Technical Accounting) results in considerable reduction of N. Centralization of functionality in few core applications results in a reduction of H.</td>
</tr>
<tr>
<td><strong>4 Application Usage by Application Name</strong> (Type 2)</td>
<td>(65; 3.88) → (44; 3.12)</td>
<td>Application consolidation (mainly EUC) results in a substantial reduction of N. Elimination of local applications (e.g., GL) results in minor reduction of H (regional consolidation).</td>
</tr>
</tbody>
</table>
## Results

### Interpretation of measure variation (2)

<table>
<thead>
<tr>
<th>Complexity Aspect</th>
<th>As-is / to-be</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5 LoB Support</strong> by Application Name (Type 2)</td>
<td>(91; 3.90) → (42; 3.11)</td>
<td>Application consolidation (mainly EUC) results in a substantial reduction of N. Elimination of LoB specific applications (e.g., in Exposure Management) results in reduction of H.</td>
</tr>
<tr>
<td><strong>6 Information Usage</strong> by Application Name (Type 2)</td>
<td>(182; 3.33) → (120; 2.43)</td>
<td>Application consolidation and disentanglement of data usage results in considerable reduction of N. Concentration of data usage in core applications incl. reporting warehouse results in reduction of H.</td>
</tr>
<tr>
<td><strong>7 Interface Usage</strong> by Interface Name (Type 2)</td>
<td>(40; 3.69) → (67; 3.45)</td>
<td>Introduction of new applications and interfaces (automation) results in an increase of N. Increased reuse of application interfaces results in a minor reduction of H.</td>
</tr>
<tr>
<td><strong>8 Platform Usage</strong> * by Platform Name (Type 3)</td>
<td>(29; 2.18) → (31; 1.78)</td>
<td>Introduction of new applications (e.g., CRM, Solvency II) results in minor increase of N. Technical standardization of applications results in significant reduction of H.</td>
</tr>
</tbody>
</table>

*) excluding client platforms
Agenda

1. The Role of Complexity
2. Measuring EA Complexity
3. Case Example: Application Landscape Planning
4. Lessons Learned and Recommendations
The calculated measures **matched very well** with the qualitative judgments of the architects involved. In summary, the approach was rated as follows:

<table>
<thead>
<tr>
<th>Benefits:</th>
<th>Drawbacks / to be noted:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>holistic</strong> / multidimensional approach</td>
<td>focus on <strong>structural complexity</strong> only</td>
</tr>
<tr>
<td><strong>complete coverage</strong> of all metamodel aspects</td>
<td><strong>modularity</strong> (one of the major concepts to master complexity) is difficult to capture</td>
</tr>
<tr>
<td><strong>flexible application</strong> at all architecture layers / levels of detail incl. possibility to <strong>drill-down</strong></td>
<td><strong>complexity aspects</strong> need to be defined and interpreted carefully</td>
</tr>
<tr>
<td><strong>unified interpretation</strong> and (potentially) simplified aggregation of measures</td>
<td></td>
</tr>
<tr>
<td><strong>complete metamodel- und tool-neutrality</strong></td>
<td></td>
</tr>
<tr>
<td><strong>simple application</strong> based on existing architecture data</td>
<td>there is a certain risk that complexity is <strong>optimized ‘out of sight’</strong> (e. g., by moving it to the subsystem level; this can be handled by extending the model view appropriately)</td>
</tr>
<tr>
<td>higher <strong>precision</strong> through the use of concentration measures</td>
<td><strong>global minimization of complexity may lead to a local increase</strong> (which can be detected by means of a drill-down and must be handled accordingly)</td>
</tr>
</tbody>
</table>
Possible extensions
Adding measures from network analysis

Measures from the domain of **Network Analysis** may be a good complement to the CEAR approach [SF12]. They are well suited to analyze the **communication** between active system elements (e.g., applications and interfaces).

**Centrality**

- **Closeness Centrality**: shortest path of a node to all other nodes
- **Betweenness Centrality**: number of shortest paths crossing a given node
- **Eigenvector Centrality**: extent that a node is connected to other central nodes

**Modularity**

- **Modularity**: number of edges that fall into groups minus the expected number in an equivalent network with random edges
- **Cluster Coefficient**: average share of neighbours of a node, which are neighbours themselves
Application scenarios
Possible applications for complexity measures

1. **Decision support / simulation** for (major) architecture decisions (e.g., target application landscape planning and landscape transformation)

2. **Comparative analyses within an organization** (e.g., to explain differences in costs and flexibility)

3. **Cross-organizational architecture benchmarkings** (complementing existing cost benchmarkings, etc.)

4. **Systematic planning of target values** as part of a continuous architecture management (e.g., differentiated by architecture layers and domains)

5. **Data delivery** to IT risk management and the **management of operational risks** (both, complexity models and measures)

...
1. Stakeholder Orientation

- the stakeholders of complexity analyses and their respective concerns should be identified from the very beginning
- the calculation of measures needs to be aligned with stakeholder requirements
- complexity assessments should be performed on the back of management questions / drivers only
- comparative as-is analyses (e.g., between different business domains) should not be conducted without a clear top management mandate
## General recommendations

### Complexity measurement and management

#### 2. Rigorous Data Management

- The reporting should be based on the **EA repository** only and be fully **automated**
- All stakeholders need to be informed about the **impact** of architecture data on the measures and be in a position to **update** that data in a timely manner
- A method needs to be defined of how to deal with **missing data** in a reliable and uniform way
3. Qualitative Analysis and Interpretation

- Complexity assessments require in-depth analyses, which can only be carried out by skilled and experienced architects.
- The analysis often requires drill-down operations and supplemental research.
- The results of complexity assessments should always be commented and interpreted qualitatively (e.g., in a report).
- Architecture decisions should never be based on complexity measures alone.
4. Compensation for Local Complexity Increase

- the global minimization of complexity may lead to an increase on the local level
- in particular, application consolidation will usually result in increasing dependencies of the target system thereby causing higher coordination and change efforts
- this should be detected in advance using appropriate analysis methods (e.g., drill-down)
- for the implementation to be successful, the resulting changes need to be communicated properly and additional resources be provided
5. From Business to IT Architecture

- to achieve maximum effectiveness, complexity optimizations should **begin with the business**
- this will not be possible without a respective problem **awareness** in the business and the **willingness** to act (esp. in the top management)
- a comprehensive approach to **business architecture management** is the ideal means to achieve this
Outlook
Still a lot of things to figure out

A long way of research ahead…

…but the time seemed right to make a start
References


References


### About us

#### Our business segments

<table>
<thead>
<tr>
<th>1. Development of EA Capabilities</th>
<th>2. Enterprise Architecture Office</th>
<th>3. IT Strategy and IT Governance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developing your core competencies</td>
<td>Expanding your operating range</td>
<td>Aligning your IT with the business</td>
</tr>
<tr>
<td>setup of EA practices</td>
<td>capturing of as-is landscapes</td>
<td>implementation of governance frameworks (esp. COBIT)</td>
</tr>
<tr>
<td>EA health checks / realignment of EA practices</td>
<td>target landscape planning</td>
<td>development of the IT strategy</td>
</tr>
<tr>
<td>development of the business architecture</td>
<td>management of transformation projects / programs</td>
<td>setup of portfolio management practices</td>
</tr>
<tr>
<td>introduction of EA tools</td>
<td>development of technical standards and reference architectures</td>
<td>optimization of the IT organization</td>
</tr>
<tr>
<td>architect trainings</td>
<td>development of solution architectures</td>
<td>implementation of risk management frameworks</td>
</tr>
<tr>
<td>EA team coaching</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
About us
Memberships and research cooperation

Organizations:

Communities:

Research Partners:

Research Projects:

CEAR
Complexity in Enterprise Architectures

CALM³
Complexity of Application Landscapes – Models, Measures, Management
Thank you for your attention

Dr. Christian Schmidt
Managing Partner

Scape Consulting GmbH
Westhafen Tower
Westhafenplatz 1
60327 Frankfurt

P +49. 69. 71 04 56 463
F +49. 69. 71 04 56 58 463

schmidt@scape-consulting.de
www.scape-consulting.de