An Engineering Approach to Enterprise Architecture Design and its Application at a Financial Service Provider

Stephan Aier*, Stephan Kurpjuweit*,
Otto Schmitz**, Jörg Schulz**, André Thomas**, Robert Winter*

* Institute of Information Management
University of St. Gallen
Müller-Friedberg-Straße 8
CH-9000 St. Gallen
{stephan.aier, stephan.kurpjuweit, robert.winter}@unisg.ch

** Deutsche Leasing AG
Fröhlingstraße 15-31
DE-61352 Bad Homburg v.d. Höhe
{otto.schmitz, joerg.schulz, andre.thomas}@deutsche-leasing.com

Abstract:
In analogy to classical engineering disciplines, this contribution discusses characteristics and requirements of an engineering approach to enterprise architecture design and proposes components and a top-level structure of an approach to address these requirements. The proposed components can partially be realized by existing work; partially they lay out the research path towards a mature engineering discipline of EA design. Core components of the proposed approach have been applied and evaluated at Deutsche Leasing AG, a globally operating financial service provider based in Germany.

1 Introduction

Organizations[1] are subject to constant evolution: changing business models, technological innovations, increasingly individualized products and services, the globalization of sourcing, sales, and operations as well as deregulation are only a few drivers of transformation [RWR06]. Due to the differences in impact, organizational change can be distinguished into incremental change (optimization) and fundamental change (transformation). While most functional business administration methods such as finance or human resources provide support for optimization, innovative and fundamental change requires systematic approaches to design, plan and implement the transformation [Wi08]: It is essential for an organization to systematically analyze the impact of upcoming changes. A prerequisite to achieve this is a thorough understanding and therefore an explicit documentation of all structures of interest as well as their relationships. Struc-

[1] In the following *organization* refers to companies, public administration, etc. and comprises the entirety of business related and IT related components of an organization. Therefore non-IT related components are referred to as business components.

tures of interest typically include product structures, business processes and structures, the relationship between business objects and data structures, application landscapes and software architectures as well as the supporting IT infrastructure systems and technologies.

For many organizations enterprise architecture management (EAM) is an important means to ensure the correct and up-to-date documentation of and the alignment between the various structures [AW09]. Enterprise architecture is defined as the fundamental structure of an organization from a holistic perspective as an aggregate model [WF07]. While there is a broad variety of EA literature focusing on evaluation [Sc04] and generalization [If99] of EA frameworks or discussing EA modeling [ABB07], only few publications address how EA should be managed, designed and analyzed systematically to facilitate innovative and fundamental change.

In this contribution we analyze mature engineering disciplines to characterize the role that EA should play in the systematic transformation of organizations. Based on this positioning, we derive requirements for an engineering approach to EA design (section 2). The scope of EA documentation is discussed which is necessary to fulfill such requirements (section 3), and components as well as a top-level structure of an appropriate EA design approach are proposed (section 4). Such components can partially be realized by existing work; partially they lay out the research path towards a mature engineering discipline of EA design. Core components of the approach have been applied and evaluated according to a design science approach at Deutsche Leasing AG, a globally operating financial service provider based in Germany (section 5).

2 Characteristics of Classical Engineering Disciplines and Requirements for an Engineering Approach to EA Design

Shaw analyzed the development of classical engineering disciplines [Sh90]. She found that engineering disciplines produce cost efficient solutions for relevant problems by using scientific knowledge in the artifact design process in service to society. These aspects are now further characterized:

1) "Cost efficient solutions": Engineering does not only imply the construction of suitable solutions, but emphasizes reasonable handling of given resources and conditions.

2) "For relevant problems": The constructed solution addresses problems relevant in practice.

3) "By using scientific knowledge": The construction process is comprehensible and traceable based on scientific construction languages, methods, and frameworks so that the solutions will most likely fit the requirements.

4) "In service to society": The engineer acts in a responsible way by providing useful innovations to society and environment.

The following subsections outline how classical engineering disciplines address these characteristics and which requirements for an engineering approach to EA design can be derived.

2.1 Standardized Construction Plans and Construction Languages

Mature engineering disciplines produce a high level construction plan (architecture) of the artifact under construction.[2] This plan depicts the main components of the artifact and the relationships between these components. The architecture explains how the artifact achieves the desired behavior. All mature engineering disciplines have developed standardized construction languages for architectural descriptions. For example, in mechanical engineering detailed standards exist on how to document construction plans [GMS08].

EA can be regarded as the central construction plan for transformation in a "business-to-IT" approach. The EA describes the main business and IT components as well as their relationships and explains how these components interact. Despite some standardization and unification endeavors like TOGAF [Op07] and GERAM [If99], no comparably accepted and powerful standard language to design, communicate and teach EA designs exists. While TOGAF describes, how to develop IT related aspects of an EA, it does neither comparably comment on business related structures, nor does it specify a standardized construction language. GERAM defines a meta-framework to relate EA frameworks like TOGAF to each other. However, GERAM remains abstract to a large extend and does not provide implementable guidance to EA description and development.

2.2 Reuse of Engineering Knowledge

Classical engineering disciplines distinguish innovative construction from routine construction.[3] Innovative constructions address new solutions while routine construction involves reusing existing solution patterns for known problems [Zw48].

Routine construction is the typical design task in classical engineering disciplines, while innovation is rather rare. To make the construction process as efficient as possible, the collection, organization, and packaging of knowledge is necessary to make it available to less experienced engineers. All disciplines found appropriate media for this knowledge transfer, e.g. engineering handbooks [ABS07; DKB94] and tool support for collaborative engineering [MKW93].

---

[2] Some engineering disciplines including civil engineering and software engineering use the "architectural blueprint" or "architectural design" (short "architecture") as central construction plan. In the following the term "architecture" is used synonymously for the central construction plan of all engineering disciplines.
[3] Please note that the distinction between innovative construction and routine construction is orthogonal to the distinction between optimization and transformation introduced in section 1. For example, organizational transformation can be achieved by means of both routine construction and innovative construction.

Documentation standards which foster the reuse of existing architectural solutions (e.g., architectural patterns or styles) to known problem classes must be found also for EA. Additionally, like in other engineering disciplines there will be no one-size-fits-all solutions. Hence it is important to specify the context factors for which solutions are applicable. It is necessary to adapt the generic architectural solutions to specific situations (e.g., company-specific requirements) and to integrate partial solutions into a complete architectural design.

2.3 Division of Labor

Besides structuring the system to be designed, the construction plan is also used to structure the development process: the components of a system are typically constructed in separate teams and then assembled in order to become a whole according to the architecture. The division of labor during the construction process is a core feature of classical engineering disciplines, since it is the only way to construct complex systems in large teams.

In the context of organizational transformation, division of labor takes place in individual transformation projects which are typically carried out independently in disjoint teams. The role of EA is to ensure overall consistency of project results.

2.4 Systematic Design and Analysis

Designing the architecture is a critical task in engineering, because it involves the transformation of requirements (problem space) into a high level blueprint of the system to be designed (solution space). Architecture design thus involves fundamental design decisions which impact the quality attributes of the system under construction (e.g. Which changes to the system can be made easily, which not? What is the system's performance?). Typically the requirements of different groups of stakeholders must be observed in a system's design. These requirements often contradict each other, so that tradeoff decisions must be made [KKC00].

Great attention must be paid to architecture and typically the most experienced engineers are involved in the architecture design process. By involving experts as well as complex analysis frameworks, engineers seek to ensure the quality of the architectural blueprint so that the architecture satisfies all relevant requirements.

EA is also the result of design decisions which determine fundamental characteristics of the organization such as strategic positioning, business process efficiency and effectiveness, business/IT alignment, and information systems capabilities. Indirectly, EA therefore implies an organization's capability to rapidly launch new products, to adapt to changed regulations, or to exploit business potentials of IT innovations.

Concrete requirements of internal and external stakeholders must be the starting point for EA design. There are different types of requirements. One type focuses on the functional development of the organization. Examples are the development of new markets and sales channels or business process outsourcing. Another type focuses on the optimization of current structures, e.g. by consolidating redundant structures or reusing existing resources to improve flexibility and to prepare the organization for future changes.

The requirements tend to involve tradeoffs which must be incorporated in an overall architectural design. Architecture analysis models should be available that determine the prospective quality attributes of the organization. Only such analysis models allow for the transformation of organization with predictable properties and are thus a prerequisite for an engineering discipline of EA design.

3 Width and Depth of Enterprise Architecture

Enterprise architects often have difficulties answering the following questions: Which objects and which relationships between these objects should be documented? How detailed should the documentation be? From the engineering perspective discussed above and from our experience gained in various EA projects, the following heuristics can be stated.

3.1 Criterion of Width

Concerns of a large and diverse group of stakeholders must be addressed in organizational transformation projects. These include systems architects, project managers, sponsors, implementers, and change agents who are participants in the transformation project, as well as customers, employees, managers, system operators, outsourcing partners, or the workers' council which are stakeholders concerned with the properties of the organization. For software and information systems engineering, catalogs of – mostly technical – concerns have been published [Al00; CE00]. These include quality concerns like system performance [Al00] as well as design related concerns like the structure and representation of data [CE00]. In the context of organizational transformation, also business concerns like business service implementation and business process efficiency should be considered [DIL04]. Based on the definition suggested by [SR01], we define a concern as a matter of interest in an organization. Accordingly, a stakeholder is defined as a person who has a certain concern [KW07].

The EA documentation must address the information needs of its stakeholders. These information needs can be derived from the stakeholders' concerns. Following the criterion of width, all objects and relationships required to address these information needs must be part of the EA documentation; the sum of all information needs therefore determines the maximum scope of EA documentation. Consequently, the scope of EA must thus be broader than solely the IT architecture of an enterprise. For instance, [WF07] identify five enterprise architectural domains which are also the basis of the work presented in this contribution: business architecture, process architecture, integration architecture, software architecture, and technology architecture.

3.2 Criterion of Depth

When the EA documentation is solely based on the criterion of width, chances are high that a large number of detailed design and implementation structures or detailed inventories of single artifact types are included. Therefore another criterion must be applied to sort out detail structures which do not have architecture impact.

EA documentation must depict how architectural strategies have been applied to address the requirements. Architectural strategies affect the design of the overall system or of a group of congenerous objects such as all core business processes, all domain-spawning data flows, or all products which are distributed over a certain channel. Structures which only focus on implementation details of one object and which are only relevant for this object, should not be considered part of the EA. Exceptions might however be acceptable in certain situations, e.g. in order to support specific concerns of a key stakeholder.

Whether an object should be considered part of the EA or not is indicated by the impact that a change to an object of that type has on other EA objects. If a change to an object does not influence other EA objects, it should most likely not be considered part of EA. Following the idea that EA is the blueprint for transformation projects, problems can arise from over-specifying design decisions which should better be made in the context of an individual project. Therefore, details such as class structures, detailed data structures, mapping information of network adaptors to servers, workflow specifications, or product variant configuration should usually not be considered part of the EA. Figure 1 illustrates this "broad and aggregate" understanding of EA.

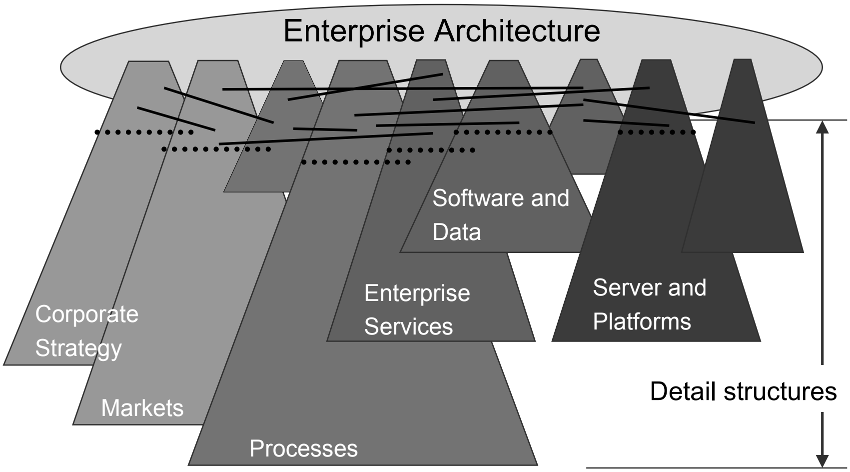

Figure 1: Enterprise Architecture is Broad and Aggregate

In two cases it can nevertheless be useful to include detail structures in the EA documentation. In both cases, changes to the detail structure may cause changes to other EA objects – which means that the heuristic introduced above remains valid:

1) If relationships to other design objects occur at a detailed level: Examples can be found when deploying single software components on servers or assigning sub-goals of a balanced scorecard to the responsible business units. A detailed relationship (e.g. between software components and servers) can always be considered at an aggregate level (e.g. between applications and server platforms). Detail structures should only be considered part of the EA when they codify design decisions that impact properties of the overall system. This is for instance true for the deployment of software components on servers, since the design of these relationships might have considerable impact on the ability of the organization to keep its business running in case of EDPC failures. An example for a relationship on detailed level without significant impact on the properties of the overall system is the assignment of application functions to activities of business processes. In this case, the aggregate relationship between applications and business processes delivers sufficient information to answer all architectural questions, while the detailed documentation is usually misleading.

2) If objects on a detailed level can be reused in multiple objects: The detail level should only be taken into account, if reuse has significant impact on the properties of the overall system. This is for instance the case when reusing product components as part of a platform strategy. It is generally not the case when reusing software libraries in multiple software components.

Moreover, it cannot be recommended to include many objects of a detail structure that all have similar topological relationships within the architecture. This is for example the case when considering all client computers of an enterprise (as an inventory).

3.3 Pragmatic Criterion

Organizations are subject to constant change. Therefore EA models need to be updated regularly. Many projects show that continuous maintenance efforts lead to high costs. For that reason it must be considered if the benefits from covering a stakeholder's concern exceed the costs of gathering the information continuously. Quantifying the costs and benefits of information needs is not trivial [Sc05]: Benefit analyses often result in "reverse considerations" (What if we did not have this information?). Cost drivers are the type of information, its origin, necessary extraction efforts and frequency.

EA serves as a high level blueprint for the transformation of an organization. High change frequencies typically indicate that the level of aggregation is too low. From our experience, in most of these cases it is sufficient to use more aggregate structures (as proposed in the criterion of depth).

4 Components and Top-Level Structure of an Engineering Approach to EA Design

The proposed approach to EA design contains various components which collectively address requirements of EA design as an engineering discipline. The components can partially be realized by existing work; partially they lead to research needs.

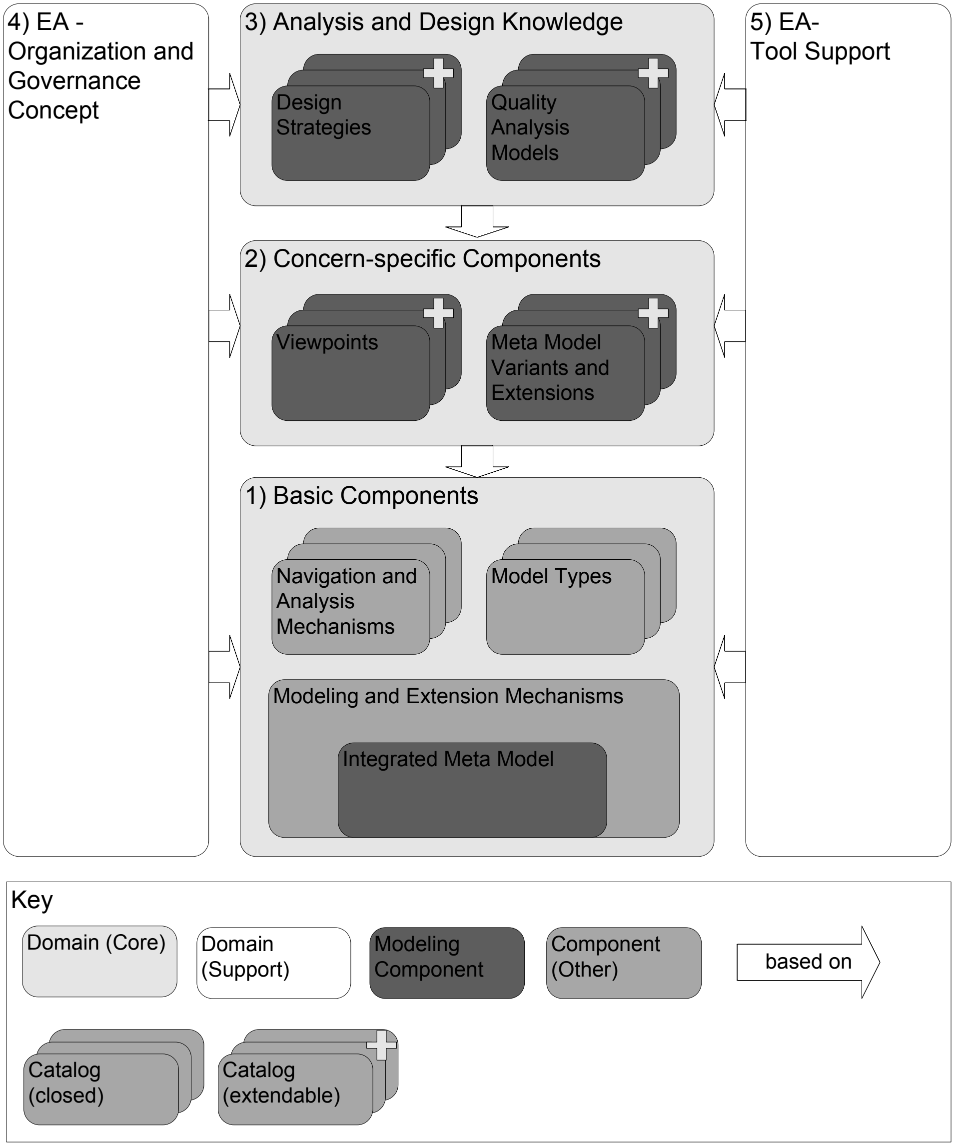

Figure 2: Top-level structure of the proposed engineering approach to EA design

Figure 2 illustrates the top-level structure the proposed approach which is further described in the following subsections.

4.1 Basic Components

Basic components contain the mechanisms for EA modeling, analysis and design.

- *Integrated Meta Model:* The integrated meta model specifies the vocabulary to consistently describe all domains of EA. In the context of the proposed approach a meta model is a model of a modeling language [Bé05; Kü06]. It specifies the available object types, relationship types as well as consistency constraints on their usage [Si99]. The meta model of the approach is based on the architectural domains introduced in section 3.1. Other EA frameworks and meta models may be applied to realize this component, as long as they satisfy the criteria stated in section 3.

- *Modeling and Extension Mechanisms:* A meta model independent description language encapsulates basic structuring and modeling mechanisms that have turned out to be useful to EA modeling. These include hierarchical refinement of objects using "part-of" and "is-a" relationships. Necessary extensions or adaptations of the integrated meta model (e.g. to company-specific requirements) can be done systematically by means of extension mechanisms [KHO07].

- *Navigation and Analysis Mechanisms:* Generic mechanisms support the analysis of the structural model information using predefined viewpoints and ad-hoc queries.

- *Model Types:* Model types represent the model information appropriately to be useful for the stakeholders. Examples for generic model types include matrix diagram, dependency diagrams, list reports, or spider web diagrams [BEL08; WBF07]. Viewpoints are stakeholder-oriented instances of model types (e.g., a matrix diagram showing the relationship between applications and business processes).

4.2 Concern-specific Components

Concern-specific components apply the generic mechanisms defined in the basic components to address specific stakeholder concerns.

- *Meta Model Variants and Extensions:* Extensions of the integrated meta model support the application of the engineering approach in specific contexts (e.g. industry, company size) and in specific projects (e.g. business driven changes, IT driven changes). The meta model variants and extensions are created by means of extension mechanisms.

- *Viewpoints:* A viewpoint captures an information need of one or more stakeholders (derived from their concerns) and defines an EA view which addresses this information need. The EA view is determined by its model type, the underlying meta model, and an analysis mechanism (specifying how the view is created on basis of a model that corresponds to the meta model) [Ba04]. The viewpoints are organized as an extendable catalog from which appropriate viewpoints can be selected and assembled to a company-specific or project type-specific solution.

4.3 Components of Analysis and Design Knowledge

Components of analysis and design knowledge help to keep record of the engineers' knowledge.

- *Design Strategies:* Proven design solutions (architectural strategies and principles) for known problems are organized as a knowledge repository [BEL08].

- *Quality Analysis Models:* Quality attribute analysis models relate architectural design decisions to quality attribute metrics.

The proposed approach can be understood as interface between organizational concepts for EA design and underlying EA tools: On one hand, the approach defines requirements for software tools and gives guidance how to use them. On the other hand, the approach serves as a "service layer" for EA organization and governance concepts, which anchor EA design within the company by defining processes, organizational roles and interfaces to other activities like strategic planning.

5 Applying the Engineering Approach to EA Design at a Financial Service Provider

The engineering approach to EA design introduced before has been applied and evaluated at Deutsche Leasing AG, a globally operating financial service provider based in Germany. The following sections describe this case study and the lessons learned. The description puts a special focus on how the components introduced in the previous section have been instantiated.

Goal of the project was the initiation of EAM to ensure transparency of all structures in context of a comprehensive outsourcing strategy and to gain indications for consolidation potentials. Therefore a company-specific meta model had to be defined and supported through customization of an EA tool.

5.1 Enterprise Architecture Modeling

As discussed before, there is no standard EA modeling language. Thus, Deutsche Leasing AG had to select an existing modeling approach. Though other approaches could have been applied as well, the St. Gallen Core Business Meta Model (CBMM) [ÖWH07] has been selected as *Integrated Meta Model* for the following reasons: (1) It covers all EA domains from business architecture to technology architecture [WF07]. (2) It conveys the documentation of EA at a high level of abstraction. (3) It can be adapted to the specific requirements. (4) EA tool support for the model is available.

In a series of workshops, the CBMM was validated against the concerns and information needs of the stakeholders within Deutsche Leasing AG. Additional object types (e.g. application environments, virtual servers) and relationships were identified to address the information needs of the stakeholders. Other design objects were eliminated. These activities led to a CBMM *Meta Model Variant/Extension*. This variant can be seen as an adaptation of the CBMM for the project type "IT master plan" with special focus on the IT-related domains of EA. The width and depth of the EA model was a major issue in these discussions. As pragmatic answer to these questions the three criteria presented in section 3 were applied.

Due to its complexity, the meta model is not suited for discussions with stakeholders. Therefore it has been divided into manageable groups of semantically related objects that are typically designed together ("meta model fragments"). For each of these meta model fragments, a model type has been defined. Figure 3 illustrates example models of the model types that were applied at Deutsche Leasing AG.[4]

The following *Modeling Mechanisms* have turned out to be useful in the context of Deutsche Leasing AG. First, the refinement of core artifacts (products, business processes, applications and software components, servers, data objects) into two refinement dimensions: The "part of" dimension has been used to model the internal structure of objects down to a refinement level as specified by the criterion of depth (cf. section 3.2). The "is a" dimension has been used to include reference models (especially of business processes) and their instantiation within different business units. Second, domain clustering has been applied to aggregate and align design objects at a high level of abstraction.

[4] Although it neither intended nor possible to read the details of each model, these screenshots should give an idea of the model types employed and the various possibilities of visualization. The same is true for figure 4.

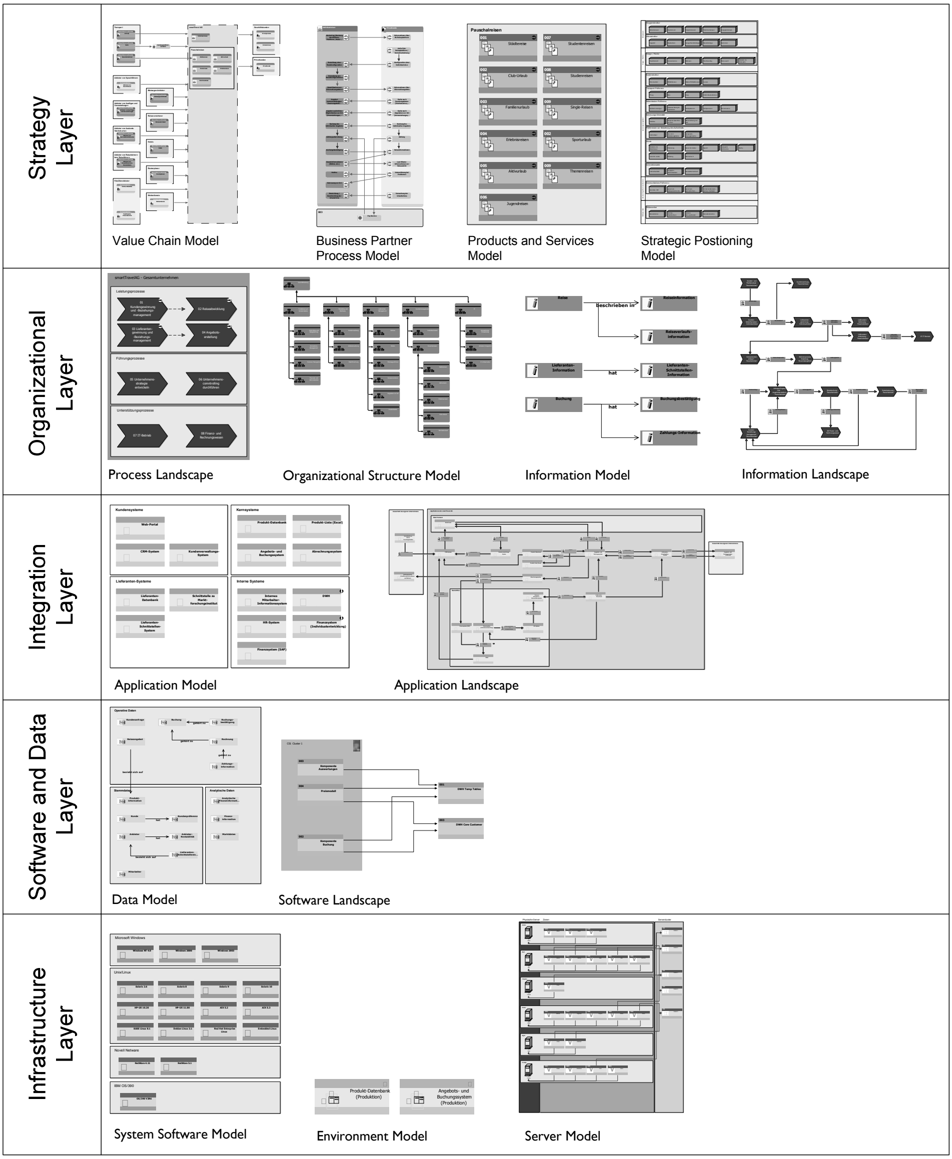

Figure 3: Model types applied at Deutsche Leasing AG

5.2 Enterprise Architecture Analysis

To analyze EA with respect to the concerns of the stakeholders, a set of pre-defined queries (*Viewpoints*) of the structural model information has been defined. Each query answers a question which has been developed in collaboration with a stakeholder. In detail the following questions are answered: How does a business unit interact with its business partners? What is the strategic position of different products? Which applications are used along a process in different organizational units? Which applications are used along a process for different products? Which organizational units participate in which business processes? How are business processes refined hierarchically? How are applications deployed on servers? How are business objects represented as data objects?

Different model types have been applied in the viewpoints: two-dimensional matrixes (relating two objects types by means of a cross in the matrix cells), three-dimensional matrixes (relating instances of two object types by means of a third object type, figure 4), reports (lists of selected objects together with attributes of these objects), dependency diagrams, and cluster maps. *Navigation and Analysis Mechanisms* define how the viewpoints are derived from the instance of the integrated meta model.

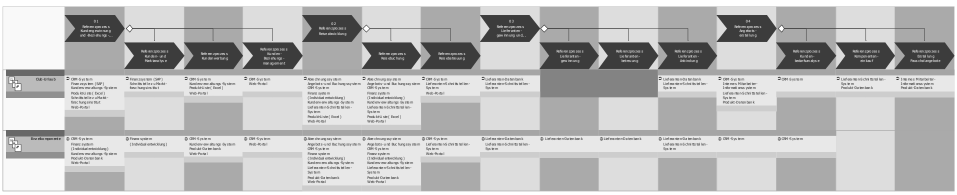

Figure 4: Three-dimensional matrix analysis showing product lines (y-axis), business processes (x-axis) and applications which are used in a process for a product line (matrix cells)

Quality Analysis Models help to interpret the viewpoints in the light of quality attributes. In the project at hand, early experiences with quality models could be made. For example, metrics on media breaks along the process, business continuity considerations in case of server failures, and consolidation potentials of applications could be considered. Especially matrix analyses have turned out to be a valuable tool to foster and rationalize the communication between the IT unit and the business units.

5.3 Analysis of Potential and Mapping with Design Knowledge

The analysis results are a basis to apply architectural *Design Strategies*. In the project at hand, early experiences with simple design strategies were made. For example, the analysis of the application landscape revealed manual data flows between applications for which automatic interfaces would be valuable. By means of a mapping between business processes and the supporting applications, redundancies and gaps were identified.

5.4 Enterprise Architecture Organization and Tool Support

Deutsche Leasing AG has implemented a decentralized approach to EA management. The EA domains are maintained by different stakeholder groups. For example, the business units maintain their business processes and products, while application managers maintain the software architecture of their applications. This decentralized maintenance concept implements the engineering principle of division of labor. EA maintenance processes and roles were defined in an *EA Organization and Governance Concept*.

The *Integrated Meta Model*, *Modeling Mechanisms*, *Model Types*, *Navigation and Analysis Mechanisms*, and *Viewpoints* have been implemented using the *EA Tool* ADO-ben®.[5] The screenshots shown in Figure 3 and 4 are created using this tool.

6 Conclusions and Future Work

The application of the engineering approach to EA design at Deutsche Leasing AG has led to the following conclusions from a practitioners' point of view: The EA should be positioned as a planning tool, not as a tool focused on operative tasks (like for example a CMDB system). To achieve this, the three criteria defining EA scope have proven to be valuable. The criterion of width requires that the EA meta model and the viewpoints are developed in close collaboration with stakeholders of the EA. To get the buy-in of the stakeholders, the introduction of EAM should be taken as a chance to revise the documentation processes within the organization in order to ensure that the EAM organization concept is integrated seamlessly and does not cause additional work load for the stakeholders.

From a research point of view, the top-level structure of the proposed approach leads to the following research opportunities: In particular the components of analysis and design knowledge (figure 2) must be fleshed out. First, EA design strategies must be gathered and packaged to provide hands-on guidance for enterprise architects. Though EA frameworks provide a good starting point on what should be considered part of an EA, they do not provide concrete guidance how to address concrete requirements by making architectural design decisions. Complementary, quality analysis models must be developed that relate the structural EA design decisions to the quality attributes of the organization under construction.

7 References

[ABB07] Arbab, F. et al.: Integrating Architectural Models. Symbolic, Semantic and Subjective Models in Enterprise Architecture. In: Enterprise Modelling And Information System Architectures 2 (2007) 1, pp. 40-56.

[ABS07] Avallone, E. A.; Baumeister, T.; Sadegh, A.: Marks' Standard Handbook For Mechanical Engineers. 11th edition, Mcgraw-Hill Professional 2007.

[Al00] Aldrich, J.: Challenge Problems for Separation of Concerns. In: Proceedings, OOPSLA 2000 Workshop on Advanced Separation of Concerns, Minneapolis, USA 2000.

[AW09] Aier, S.; Winter, R.: Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. In: Business & Information Systems Engineering 51 (2009, forthcoming) 2.

[Ba04] Bayer, J.: View-based Software Documentation. Ph.D. Thesis, Universität Kaiserslautern, Kaiserslautern 2004.

[5] ADOben® is a registered trademark of BOC AG, additional information: http://adoben.iwi.unisg.ch/.

[Bé05] Bézivin, J.: On the Unification Power of Models. In: Software and System Modeling 4 (2005) 2, pp. 171-188.
[BEL08] Buckl, S. et al.: Enterprise Architecture Management Pattern Catalog. Munich 2008.
[CE00] Czarnecki, K.; Eisenecker, U.: Generative Programming: Methods, Tools, and Applications. Addison-Wesley 2000.
[DIL04] ter Doest, H. et al.: Viewpoints Functionality and Examples. Technical Report TI/RS/2003/091, Telematica Instituut 2004.
[DKB94] Dubbel, H.; Kuttner, K. H.; Beitz, W.: Dubbel. Handbook of Mechanical Engineering. Springer, Berlin 1994.
[GMS08] Giesecke, F. E. et al.: Technical Drawing. 13th edition, Pearson Education, Denver, CO 2008.
[If99] IFIP–IFAC Task Force: GERAM: Generalised Enterprise Reference Architecture and Methodology, Version 1.6.3. http://www.cit.gu.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/GERAMv1.6.3.pdf, Date of Access: 03.12.2007.
[KHO07] Kurpjuweit, S.; Höning, F.; Osl, P.: Metamodell-Erweiterungsmechanismen. St. Gallen 2007.
[KKC00] Kazman, R.; Klein, M.; Clements, P.: ATAM: Method for Architecture Evaluation. Technical Report, Software Engineering Institute Carnegie Mellon University 2000.
[Kü06] Kühne, T.: Matters of (meta-)modeling. In: Software and Systems Modeling 5 (2006) 4, pp. 369-385.
[KW07] Kurpjuweit, S.; Winter, R.: Viewpoint-based Meta Model Engineering. In: Proceedings, Enterprise Modelling and Information Systems Architectures – Concepts and Applications, Proceedings of the 2nd Int'l Workshop EMISA 2007, Bonn 2007, pp. 143-161.
[MKW93] McGuire, J. G. et al.: SHADE: Technology for Knowledge-based Collaborative Engineering. In: Concurrent Engineering 1 (1993) 3, pp. 137-146.
[Op07] The Open Group: The Open Group Architecture Framework TOGAF - 2007 Edition (Incorporating 8.1.1). Van Haren, Zaltbommel 2007.
[ÖWH07] Österle, H. et al.: Business Engineering: Core-Business-Metamodell. In: Wisu – Das Wirtschaftsstudium 36 (2007) 2, pp. 191-194.
[RWR06] Ross, J. W.; Weill, P.; Robertson, D. C.: Enterprise Architecture as Strategy. Creating a Foundation for Business Execution. Harvard Business School Press, Boston, MA 2006.
[Sc04] Schekkerman, J.: How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework. 2nd edition, Trafford Publishing, Victoria, British Columbia 2004.
[Sc05] Schekkerman, J.: The Economic Benefits of Enterprise Architecture: How to Quantify and Manage the Economic Value of Enterprise Architecture. Trafford Publishing, Victoria, British Columbia 2005.
[Sh90] Shaw, M.: Prospects for an Engineering Discipline of Software. In: IEEE Software 7 (1990) 6, pp. 15-24.
[Si99] Sinz, E. J.: Architektur von Informationssystemen. In: Rechenberg, P.; Pomberger, G. (eds.): Informatik-Handbuch. 2nd edition, Hanser, München, Wien 1999, pp. 1035-1046.
[SR01] Sutton, S. M.; Rouvellou, I.: Issues in the Design and Implementation of a Concern-Space Modeling Schema. In: Proceedings, Advanced Separation of Concerns Workshop, Toronto, Canada 2001, pp. 119-124.
[WBF07] Winter, R. et al.: Analysis and Application Scenarios of Enterprise Architecture – An Exploratory Study (Reprint). In: Journal of Enterprise Architecture 3 (2007) 3, pp. 33-43.
[WF07] Winter, R.; Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. In: Journal of Enterprise Architecture 3 (2007) 2, pp. 7-18.

[Wi08] Winter, R.: Business Engineering – Betriebswirtschaftliche Konstruktionslehre und ihre Anwendung in der Informationslogistik. In: Dinter, B.; Winter, R. (eds.): Integrierte Informationslogistik. Springer, Berlin, Heidelberg 2008, pp. 17-38.

[Zw48] Zwicky, F.: Morphological Astronomy. In: The Observatory 68 (1948), pp. 121-143.