

## Essential Layers, Artifacts, and Dependencies of Enterprise Architecture

By Robert Winter and Ronny Fischer

### Abstract

*After a period where implementation speed was more important than integration, consistency and reduction of complexity, architectural considerations have become a key issue of information management in recent years again. Enterprise architecture is widely accepted as an essential mechanism for ensuring agility and consistency, compliance and efficiency. Although standards like TOGAF and FEAF have developed, however, there is no common agreement on which architecture layers, which artifact types and which dependencies constitute the essence of enterprise architecture. This paper contributes to the identification of essential elements of enterprise architecture by (1) specifying enterprise architecture as a hierarchical, multilevel system comprising aggregation hierarchies, architecture layers and views, (2) discussing enterprise architecture frameworks with regard to essential elements, (3) proposing interfacing requirements of enterprise architecture with other architecture models and (4) matching these findings with current enterprise architecture practice in several large companies.*

### Keywords

enterprise architecture, architectural components, architectural layers, architectural views, interfaces

---

### ENTERPRISE ARCHITECTURE: DEFINITION

According to ANSI/IEEE Std 1471-2000, architecture is defined as the “fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution” (IEEE 2000). Enterprise architecture (EA) therefore is understood as (1) the fundamental organization of a government agency or a corporation, either as a whole, or together with partners, suppliers and / or customers (“extended enterprise”), or in part (e.g. a division, a department, etc.) as well as (2) the principles governing its design and evolution (Opengroup 2003).

While an EA model is a representation of as-is or to-be architecture of an actual corporation or government agency, an EA framework provides (Opengroup 2003)

- One or more meta-model(s) for EA description,

- One or more method(s) for EA design and evolution,
- A common vocabulary for EA, and maybe even
- Reference models that can be used as templates or blueprints for EA design and evolution.

The components of an EA framework should be applicable for a broad range of corporations and government agencies.

Traditionally, architecture in the information systems context is focusing on IT related artifacts like IT platforms, software components and services, applications, IT processes, and maybe IT strategy in order to support more efficient IT operations, better return on IT investment, and faster, simpler and cheaper IT procurement (Opengroup 2003). In contrast to this approach which better should be designated information systems architecture (ISA), EA should also include business related artifacts like organizational goals, products and services, markets,

business processes, performance indicators, etc. (Braun/Winter 2005). Only when 'purely' business related artifacts are covered by EA, important management activities like business continuity planning, change impact analysis, risk analysis and compliance can be supported.

## ENTERPRISE ARCHITECTURE: REPRESENTATION

The aforementioned definition of enterprise architecture restricts included components to be 'fundamental'. Due to the broad range of relevant component types, EA may nevertheless comprise a huge number of such artifacts. As a consequence, most EA frameworks distinguish several architecture layers and architecture views in order to reduce the number of artifacts per model (Scheckerman 2004, Tang et al. 2004). When several architecture layers and architecture views are differentiated, design and evolution principles have to address consistency and integration issues. The theory of hierarchical, multilevel systems (Mesarovic et al. 1970) provides a conceptual foundation for such methods.

For EA, a hierarchical approach usually applies the 'IT follows business' principle, starting with strategic positioning from the business management point of view, then deriving appropriate organizational processes and structures on this basis, and finally specifying the information system, i.e. the interaction between human and technical information system components that appropriately support business requirements (Braun/Winter 2005).

Most frameworks differentiate the following EA layers:

- **Business architecture:** The business architecture represents the fundamental organization of the corporation (or government agency) from a business strategy viewpoint. Typical artifacts represented on this layer are value networks, relationships to customer and supplier processes, targeted market segments, offered services, organizational goals, and strategic projects (Weill/Vitale 2001, Hedman/Kalling 2003). Design and evolution principles for business architecture can be derived e.g. according to the market based approach (Porter 1998) or the resource based approach

(Hamel/Prahalad 1990) to strategic management.

- **Process architecture:** The process architecture represents the fundamental organization of service development, service creation, and service distribution in the relevant enterprise context. Typical artifacts represented on this layer are business processes, organizational units, responsibilities, performance indicators, and informational flows (Davenport 1993). Design and evolution principles for this layer focus on effectiveness (creating specified outputs) and efficiency (meeting specified performance goals) (Österle 1995).
- **Integration architecture:** The integration architecture represents the fundamental organization of information system components in the relevant enterprise context. Typical artifacts represented on this layer are enterprise services, application clusters, integration systems and data flows. Design and evolution principles for this layer focus on agility (e.g. by service orientation (Douglas 2003)), cost efficiency (e.g. by reduction of interfaces (Linthicum 2000)), integration (e.g. by analysis of data coupling (IBM 1984)), and / or speed (e.g. by straight-through processing (Kuhlin/Thielmann 2005)).
- **Software architecture:** The software architecture represents the fundamental organization of software artifacts, e.g. software services and data structures. A broad range of design and evolution principles from computer science is available for this layer.
- **Technology (or infrastructure) architecture:** The technology architecture represents the fundamental organization of computing / telecommunications hardware and networks. A broad range of design and evolution principles from computer science is available for this layer too.

According to the hierarchical, multi-level systems theory approach, results on each architecture layer reduce the degrees of freedom of the subsequent layers (Mesarovic et al. 1970).

Most of the artifacts classes represented in EA can be represented as aggregation hierarchies, i.e. can be modeled on various level of aggrega-

tion. For example, the organizational goals of a corporation (or government agency) that are defined on a very aggregate level in a balanced scorecard, are subsequently decomposed into more and more specific performance indicators, resulting in a multi-layer goal/indicator aggregation hierarchy. Such aggregation hierarchies are commonly used not only for goals/indicators, but also for products/services, business processes, organizational units, information objects, or software artifacts.

Figure 1 provides a schematic illustration of an EA comprising the above mentioned five hierarchical layers. On each layer, aggregation hierarchies are used to represent artifacts on different levels of aggregation.

Alongside the formation of architecture layers and aggregation hierarchies, views are often used in order to master complexity (Sowa/Zachman 1992). In a multi-layer architecture, views can be layer-specific or cross-layer. Examples for layer-specific views in EA are the structural view (organizational units, responsibilities) and the process view (business

processes, performance indicators) on the organization/process layer. Examples for cross-layer views are security architecture and information architecture.

Based on the concepts of multi-layer representation, aggregation hierarchy and cross-layer view, EA can be defined as the view that represents all aggregate artifacts and their relationships across all layers (Fig. 1). This is due to the fact that only the most aggregate artifact representations can be 'fundamental', and that all more decomposed artifact representations have to be covered by specialized architectures.

The remainder of this paper is organized as follows: In Section 2 we analyze several EA approaches with regard to the core artifacts they propose. Interfaces to other corporate architectures and models are discussed in Section 3. In Section 4, we compare our recommendations against several EA case studies in large companies from different countries and industry sectors. In Section 5, conclusions regarding the level of detail of EA core artifacts and their interfaces to other architectures are drawn, and an outlook to future research in this area is given.

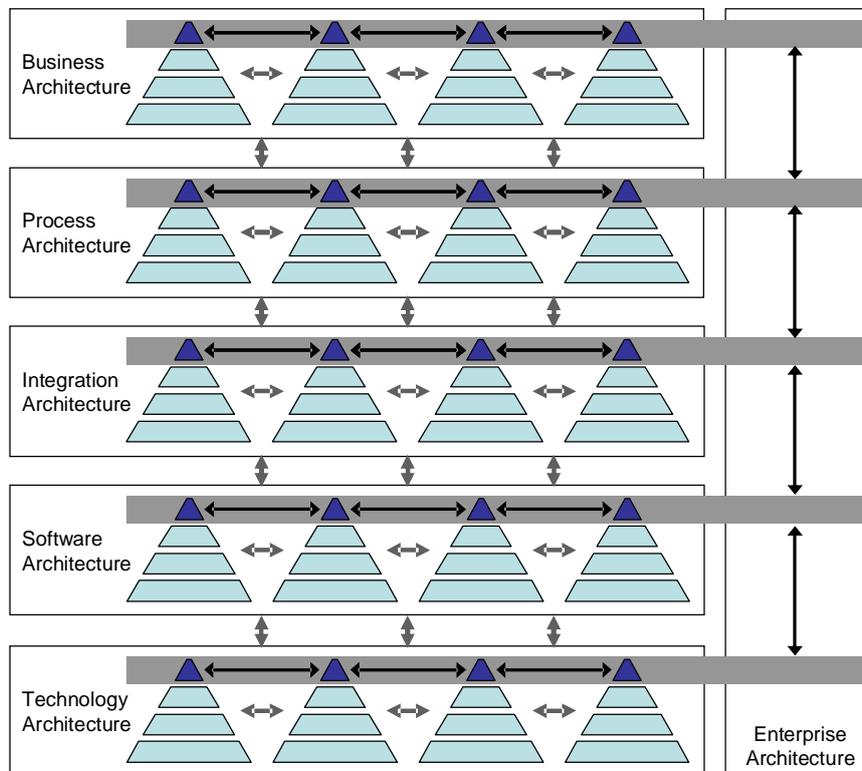


Figure. 1. Enterprise Architecture as a Cross-layer View of Aggregate Artifacts

## CORE ARTIFACTS OF EA

In this section, we discuss which core artifacts should be covered on the five regarded EA layers. As a basis for consolidating artifact types that are considered as being important for EA, we analyze widely used EA frameworks such as

TOGAF 8.1 (Opengroup 2003), FEAF version 1.1 (CIO-Council 1999, CIO-Council 2001) and ARIS (Scheer 1999) with extensions from (IDS Scheer 2005). The results are exhibited in Table 1 below.

	TOGAF 8.1	FEAF 1.1	ARIS
organizational units	Business Architecture		Organizational Architecture
business locations	Business Architecture	Technology Architecture*	
business goals and objectives (for each organizational unit)	Business Architecture		Business Performance Architecture
performance metrics			
business functions (as an aggregation hierarchy)	Business Architecture		
internal and external business services	Business Architecture		
business processes (including measures and deliverables)	Business Architecture	Application Architecture*	Business Process Architecture
business roles (including skills requirements)	Business Architecture		Organizational Architecture
programs		Application Architecture	
data (various views)	Information Systems Architecture		
applications (various views)	Information Systems Architecture		
conceptual / semantic data model	Data Architecture	Data Architecture*	Data View
logical data model	Data Architecture	Data Architecture	Data View
data management process models	Data Architecture		
data entity/business function matrix	Data Architecture		
various data related views	Data Architecture	Data Architecture	
'process' systems models	Applications Architecture	Application Architecture*	Function View
'place' systems models (location, org unit)	Applications Architecture	Technology Architecture	Organization View
'time' systems models (events, messages)	Applications Architecture		Control View
'people' systems models	Applications Architecture		Organization View
outputs, products (business logistics model)		Technology Architecture*	Output View
various application related views	Applications Architecture	Application Architecture	
software components		Application Architecture	Function View
hardware models	Technology Architecture	Technology Architecture	Organization View
communications models	Technology Architecture	Technology Architecture	Control View
processing models	Technology Architecture		
other technology models	Technology Architecture	Technology Architecture	

\* also components of Business Architecture

Table 1. Enterprise Architecture as a Cross-layer View of Aggregate Artifacts

TOGAF's business architecture is populated with many artifact types. The five-layer approach presented in the preceding section therefore differentiates between strategy related artifacts (specifying the "what") and organization/process related artifacts (specifying the "how"). TOGAF's IS architecture layer can be compared to the proposed integration/application layer. Data and applications architecture are assigned to different layers in TOGAF, whereas they are treated as views on the software layer in the proposed framework.

FEAF's data architecture, application architecture and technology architecture can be interpreted as cross-layer views, while business architecture comes close to a simplified business & process architecture.

In general, FEAF comprises not many artifact types, and has – like the Zachman framework from which it was adapted – not put much effort into specifying EA-relevant artifact relationships.

Goal/performance related artifacts and product specifications are not covered by FEAF at all.

The five views of the original ARIS framework (Scheer 1999) comprise artifacts that are located mainly on the software component layer of the proposed EA framework. This mainly 'technical' subset of artifacts has recently been extended (IDS Scheer 2005). But even with 'business architecture' extensions, strategy related artifacts are not covered in depth. There is also a lack of artifacts that represent high-level constructs on the integration/application layer (e.g. application clusters, enterprise services). On the other hand, many artifacts are covered that are considered not to constitute an important component of EA (e.g. computer hardware details, machine resources).

When comparing these framework approaches to EA, the following set of artifact types results as a hypothesis for EA core artifacts:

- Strategy specification ("what" questions): hierarchy of organizational goals and suc-

cess factors, product/service model (including partners in value networks), targeted market segments, core competencies, strategic projects, maybe business principles, dependencies between these artifacts.

- Organization/process specification (“how” questions):
  - Specification of structure: organizational unit hierarchy, business location hierarchy, business role hierarchy (including skills requirements), dependencies between these artifacts
  - Specification of behavior: business function hierarchy, business process hierarchy including inputs/outputs (internal and external business services including service levels), metrics (performance indicators), service flows
  - Specification of information logistics: business information objects, aggregate information flows
  - Dependencies between these artifacts, e.g. responsibilities, information requirements
- Application specification (business IT alignment questions):
  - Specification of applications and application components
  - Specification of enterprise services and service components
- Software specification:
  - Specification of software components: functionality hierarchy, event/message hierarchy
  - Specification of data resources: conceptual, logical and physical data models
  - Dependencies between these artifacts, e.g. data usage by software components (CRUD)
- Technical infrastructure specification:
  - Specification of IT components: hardware units, network nodes, etc.
  - Dependencies between these artifacts
- Specification of dependencies between layers, e.g.:
  - Organizational goals/success factors vs. process metrics
  - Products/services vs. process deliverables

- Organizational units vs. applications (ownership)
- Activities vs. applications
- Activities/business processes/information requirements vs. enterprise services (orchestration)
- Applications/enterprise services vs. conceptual data entity types
- Applications/enterprise services vs. software components (composition)

In the following Section, we will first discuss which artifact types on which aggregation levels should be part of EA, and which should be part of other, specialized architectures and models. In Section 4, we will compare our recommendation with several EA case studies that exhibit current EA practice in large companies.

## INTERFACING EA WITH OTHER CORPORATE ARCHITECTURES & MODELS

It is obvious that the complexity of a medium or large corporation (or government agency) cannot be covered by one single EA. In real life, several EAs for different parts of the enterprise might be maintained, and/or EA will co-exist with other, more specialized architectures that cover a subset of artifacts. Hence a useful interfacing between EA and specialized architectures must be specified.

An analysis of the goals of EA seems useful in order to specify this interface. EA should

- Support IT business alignment by providing support for consistent design and evolution of artifacts on different layers and/or in different views,
- Support transformation (business development, process reengineering, IS reengineering) by providing impact analyses, and
- Support maintenance, compliance, risk management etc. by documenting not only structures and direct dependencies, but also allowing the analysis of multi-step dependencies (e.g. server – software service – enterprise service – process deliverable – product – revenue).

As a consequence, EA should be ‘broad’ rather than ‘deep’: For multi-step dependency analyses and holistic coverage of IT business alignment, it is much more useful to cover a large number of artifact types and their dependencies on an ag-

gregate level, than to cover a small number of artifact types and their dependencies in more detail. This understanding of EA is illustrated in Figure 1. We therefore need to identify appropriate interfaces to partial, specialized architectures like:

- Product/service architecture (managed e.g. using a product management tool),
- Metrics architecture (managed e.g. using a performance management tool),
- Process architecture (managed e.g. using a process modeling tool and workflow management systems),
- Information/data architecture (managed e.g. using a data modeling tool and database management systems),
- Software architecture (managed e.g. using a software design tool and a configuration management tool), and

- Technology architecture (managed e.g. using a computer system management tool).

In order to provide an indication regarding the specification of interfacing between EA and specialized partial architectures, four EA case studies are presented in the next section.

## CASE STUDIES

By presenting four case studies of EA initiatives conducted by large companies from different industries and countries, we want to validate our recommendations for essential EA artifact types in Section 3.

The data was collected by desk research, informal interviews with EA project managers and/or personal involvement of the authors in EA projects of these companies. Table 2 below gives an overview of the four analyzed companies.

	Company A	Company B	Company C	Company D
<b>Country</b>	Switzerland	South Africa	Germany	Switzerland
<b>Industry</b>	Financial services	Mining	Banking	Insurance

Table 2. Overview of Case Studies

The description of the cases is structured as follows. We start with outlining the core business of the company. Then we describe the motivation for adopting an EA program within the respective company. After this, the EA model layers of the respective project are specified and mapped to the architecture layers proposed in Section 1. Finally we identify core artifacts within each layer and important intra-layer as well as cross-layer dependencies between artifacts.

### Company A

Company A is a major Swiss financial service provider. The EA program of company A was started in 2005 and is ongoing. The program

was initiated because an aggregate, enterprise-wide view of important entities and dependencies did not exist.

Unlike many other organizations, IT business alignment is not the major driver for EA efforts in company A. Instead, EA is aimed at supporting strategy implementation, in particular at supporting the project selection/project portfolio planning process. In addition, EA is regarded as foundation of business continuity planning, service management and security management.

The enterprise architecture model of company A comprises four architectural layers as is shown in Figure 2 on the next page.

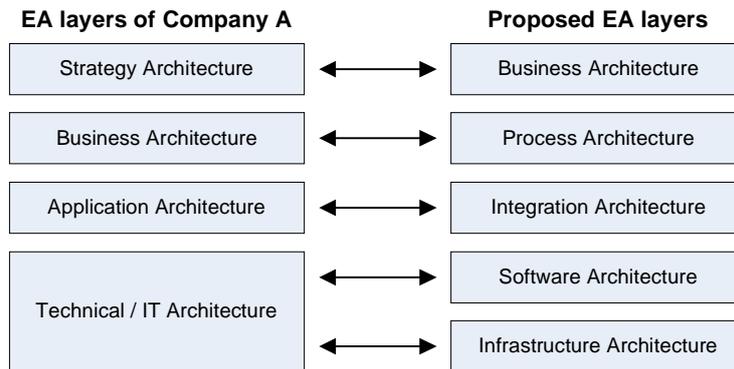


Figure. 2. Mapping between EA layers of Company A and proposed EA layers

The strategy architecture represents organizational goals as well as projects aiming at implementing these goals, and project results linked to these goals. Company A intends to extend the strategy architecture by adding success factors related to organizational goals and performance indicators for these success factors.

The business architecture corresponds to the process architecture mentioned in Section 1 (Figure 2) and represents core business functions, organizational units, business locations and service level agreements which are all linked to high level business processes. The services offered by company A are also represented on this layer. This is a major difference to our proposal in Section 1 where we assigned business services to the strategy layer. Organizational units are depicted by a hierarchical model which defines organizational units broadly at first and then with increasing detail. Furthermore, core business functions are linked to strategy implementation projects (as part of the strategy architecture).

Application services, applications, application clusters and information objects are artifact types assigned to the application architecture. Here the most important dependencies regarding artifacts from the same layer as well as from superordinate layers are:

- Dependencies between application services and business processes as well as between application services and core business functions,
- Dependencies between information objects and applications, and

- Dependencies between information objects and business processes.

The IT architecture comprises deployed applications which are linked to application services on the superordinate layer and IT components called "configuration items" (servers, databases, etc.). IT components are represented by a hierarchical model.

With respect to the development of enterprise architecture content, Company A strongly relies on information from models which already exist within the enterprise. A single EA repository is used to integrate meta data on all EA artifact types. Artifact meta data from various sources is cleaned, and redundancies are eliminated during the repository update process. There is no need for specific EA modeling activities except for representing aspects that are not covered by existing models.

### Company B

Company B is one of the world's leading producers of precious metals with exploration and mining activities in South Africa, Canada, Russia, Brazil, and Zimbabwe.

The adoption of an EA program at company B was motivated by the need for accurate, timely and consistent information regarding the corporate structure. Main reasons for dealing with EA at company B have been poor IT business alignment, absence of an effective IT governance, and unification of modeling methods, tools and standards across the enterprise.

Company B's EA model is subdivided into five layers, as is shown in Figure 3 on the next page.

The business architecture represents strategic objectives, business principles, offered products, business roles, organizational units and business locations as well as risk items. All of these artifact types are linked to high level business processes. The business processes model itself is a hierarchical model consisting of enterprise processes (level 0) which are decomposed into

value chains (level 1). These value chains are then decomposed into processes (level 2). Summarizing these findings, company B's business architecture can be understood as a combination of the proposed business architecture and selected parts of the process architecture from Section 1, as shown in Figure 3 below.

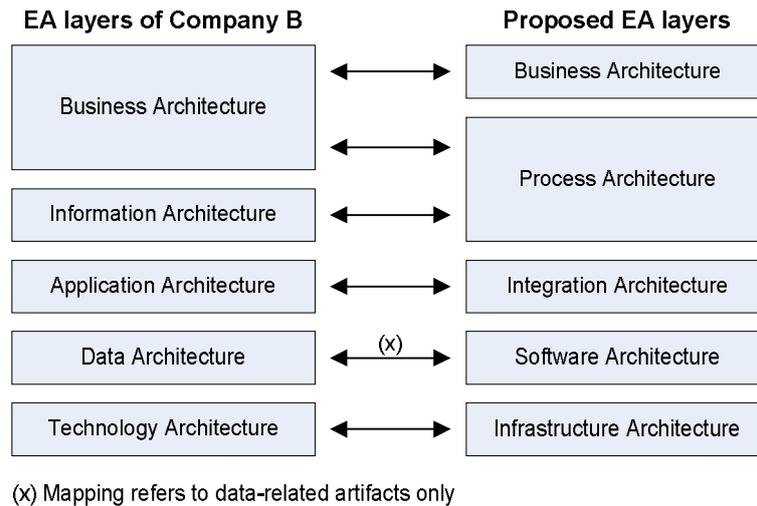


Figure. 3. Mapping between EA layers of Company B and proposed EA layers

The information architecture is comprised of an information object hierarchy and a metrics hierarchy as well as dependencies between information objects and metrics on the one hand and processes, applications and data models on the other hand.

Applications, application clusters, application modules and application interfaces are artifacts represented by the application architecture. In order to enable IT business alignment, applications are connected 'upward' to business processes and organizational units (on the business layer).

The data architecture depicts (logical) data models embracing data subject areas, entities, tables and their relationships to business processes and organizational units/business roles.

The technology architecture comprises infrastructure applications and network nodes. Both are linked to applications, databases, organizational units and business locations.

### Company C

Company C is a large German full-service bank with more than four million private customers and almost half a million corporate clients.

Company C developed EA to enhance the transparency of process architecture and integration architecture. By that means, potentials for optimization across business segments should be revealed, and information required for sourcing decisions should be provided.

Company C's EA approach distinguishes between business architecture and technical/IT architecture, with both architectures being subdivided into several layers. The mapping between company C's EA layers and the EA layers proposed in Section 1 is illustrated in Figure 4.

For each business segment, the business model layer represents value networks, targeted market segments and offered services. The service model is a hierarchical model comprising three levels: level 1 (product categories) is decomposed into level 2 (product groups) which then is decomposed into level 3 (products).

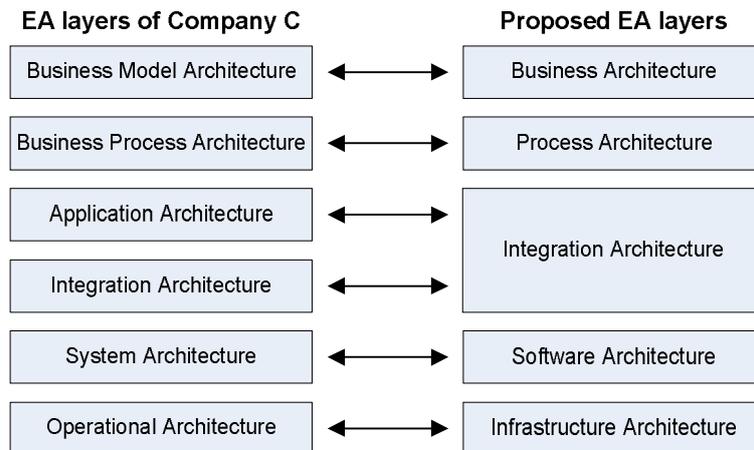


Figure. 4. Mapping between EA Layers of Company C and Proposed EA layers

The breakdown of enterprise processes/value chains into sub-processes and their relationships are represented on the business process layer. By means of a org unit hierarchy, the organizational structure of the company is represented on the business process layer too. In addition, the following dependencies are specified on the business process layer regarding artifacts from the same layer as well as from other layers:

- Dependencies between business processes and organizational units
- Dependencies between business processes and application clusters as well as single applications
- Dependencies between offered services and corresponding business processes
- The application layer is comprised of logical application clusters, while the integration layer describes principles and technologies for application integration. Technical components related to applications are represented on the system layer. The operational

layer represents mandatory principles for application operation.

#### Company D

Being a major player in the Swiss insurance market with more than one million customers and focusing on non-life insurance, Company D has launched its EA initiative more than four years ago.

Investment in EA has been justified by a lack of transparency regarding dependencies between IT and service offerings / business processes. As a consequence, an EA model was created as a means for enterprise planning, to eliminate redundancies, and to standardize business processes as well as information systems.

Company D's EA model is comprised of three architectural layers. Alongside these three layers, two architecture views exist. Fig. 5 depicts the mapping of this approach to the EA layers proposed in Section 1.

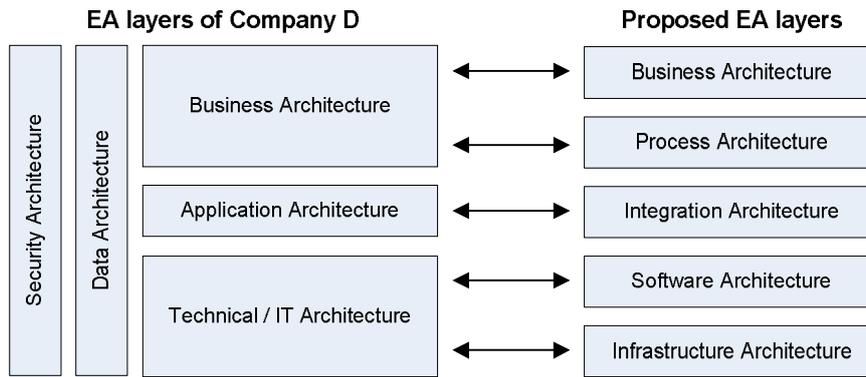


Figure. 5. Mapping Between EA Layers of Company D and Proposed EA Layers

The business architecture is aimed at representing corporate strategy, services, business principles, business scenarios, business processes and corresponding activities as well as business components and business objects. Business processes are depicted by means of an aggregation hierarchy. Elementary processes are decomposed into activities. Business scenarios are mapped to services.

Dependencies between business activities and applications, application components, application services and application interfaces are represented by the application architecture.

The technical/IT architecture is comprised of two sub-layers designated as “implementation technology” and “runtime environment”. Software systems and their decomposition into software components, software modules as well as software interfaces are represented by the “implementation technology” sub-layer. Software interfaces are linked to application interfaces. Platforms, databases, integration systems and network nodes required to run the systems are represented by the “runtime environment” sub-layer. In addition, dependencies between platforms and integration technologies on the one hand, and software components on the other hand are specified on the runtime environment sub-layer.

The data architecture encompasses data objects, entity types and tables. Data objects are linked to business objects represented within the business architecture.

Business risks, non-functional requirements and technical precautions are represented by the security architecture. Dependencies between

business risks and business activities are also represented as part of the security architecture.

## CONCLUSIONS AND OUTLOOK

Based on the discussion of current EA frameworks, we proposed a set of EA layers, artifacts and dependencies in Section 2 that we consider as essential for a business-oriented approach to EA. In Section 3, we presented arguments for differentiating between EA and other, specialized architectures and models in enterprise modeling. Since the basic layer design “from business to IT”, most of the artifacts and many dependencies can be identified in various actual EA cases summarized in Section 4, it is justified to propose our recommendation of EA essentials as a hypothesis. Of course, broad empirical studies need to validate this proposition.

We believe that an important success factor for EA initiatives is to clearly distinguish between the broad, but aggregate EA on the one hand, and a set of specialized, detailed partial architectures on the other. Enterprise modeling can achieve its goals only if interfaces between EA and specialized architectures have been conceptually specified and efficiently implemented.

With reference to the essential EA artifacts proposed in Section 2 and our findings from the case studies in Section 4, we suggest that interfaces between EA and specialized architectures should be specified as follows:

- **Business Architecture:** While product/service categories, product/service groups and products/services should be comprised in EA, more detailed prod-

uct/service representations like variants, versions/releases and components should be represented by a product sub-architecture, and managed by a product management tool. Furthermore, projects aiming at realizing strategic goals should not be decomposed in EA. Project portfolio tools and / or project management tools are more appropriate for this purpose.

- **Process Architecture:** Business processes should not be decomposed further than to the sub-process level. Detailed process descriptions including specifications of activities and work steps are out of EA scope and should be maintained by using specialized business process modeling tools and / or workflow design tools. As a consequence, process performance management tools are much better suited to represent performance indicators related to activities, while the aggregate performance information represented in balanced scorecard tools might be a valuable part of EA.
- **Integration Architecture:** While aggregate dependencies / data flows between applications or application components are belonging to EA and should be managed by appropriate EA tools, detailed interface descriptions for data exchange, remote procedure calls etc. should be maintained by using tools like integration repositories of enterprise application integration (EAI) tools.
- **Software Architecture:** Detailed descriptions of data objects (e.g. attribute lists) are not essential for EA purposes and should be managed e.g. by using a data modeling tool. In addition, structural and behavioral aspects of single software components are not covered by EA and should be managed by using software design tools.
- **Infrastructure Architecture:** Detail specifications of IT components (hardware units etc.) are not essential for EA; Asset management tools should be used for managing such meta data, and appropriate interfaces have to maintain consistency between the different repositories.

In addition to a broader empirical validation of the proposed essential layers, artifacts and dependencies of EA, further research will be needed to identify and validate a reference meta

architecture that specifies architecture components (EA, performance management, product management, project management, workflow design, data management, EAI configuration, software design, IT asset management, etc.) and interfaces between those components.

Another interesting aspect that has not been covered in depth is the differentiation of EA scenarios, e.g. by clustering EA approaches based on EA goals, enterprise size, enterprise structure, etc. Based on an appropriate scenario model, the recommendation of reference structures and reference processes for EA could then be refined by scenario-specific configuration.

## AUTHOR BIOGRAPHIES

Dr. Robert Winter is full professor of information systems at the University of St. Gallen (HSG), director of HSG's Institute of Information Management and academic director of HSG's Executive Master of Business Engineering program. He received master's degrees in business administration and business education as well as a doctorate in social sciences from Goethe University, Frankfurt, Germany. After eleven years as a researcher and deputy chair in information systems, he was appointed chair of information management at HSG in 1996. His research interests include business engineering methods and models, information systems architectures / architecture management, and integration technologies / integration management.

Ronny Fischer is a research assistant and doctoral student at the University of St. Gallen (HSG). He holds a master's degree in business administration and information systems (concentration in financial analysis, application systems, and operations research) from the University of Leipzig, Germany. His research interests include enterprise architecture modeling, enterprise architecture management, and IT/business alignment. He has been a project manager of bilateral projects with industry partners conducting applied research in the areas of process modeling, enterprise architecture modeling and enterprise architecture management. Mr. Fischer is currently writing his doctoral thesis on enterprise architecture management.

## REMARK

This paper is an extended version of a paper previously presented at the First Workshop on Trends in Enterprise Architecture Research (TEAR 2006) within the Tenth IEEE International EDOC Enterprise Computing Conference (EDOC 2006), Hong Kong, China, 16 October 2006.

## REFERENCES

- Braun, C. and Winter, R. (2005). A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform, GI-Edition Lecture Notes in Informatics (LNI), Enterprise Modelling and Information Systems Architectures, Proc. of the Workshop in Klagenfurt, Klagenfurt, 24.10.2005, P-75, pp. 64-79.
- CIO-Council (1999). Federal Enterprise Architecture Framework, Version 1.1.
- CIO-Council (2001). A Practical Guide to Federal Enterprise Architecture.
- Davenport, T. H. (1993). Process Innovation - Reengineering Work through Information Technology, Harvard Business School Press, Boston.
- Douglas, K. B. (2003). Web Services and Service-Oriented Architecture: Your Road Map to Emerging IT, Morgan Kaufmann Publishers,
- Hamel, G. and Prahalad, C. K. (1990). The Core Competence of the Corporation, in: Harvard Business Review, 68, No. 3, pp. 79-91.
- Hedman, J. and Kalling, T. (2003). The business model concept: theoretical underpinnings and empirical illustrations, in: European Journal of Information Systems, 12, No. 1, pp. 49-59.
- IBM (1984). Business Systems Planning - Information Systems Planning Guide, 4th ed. Ed., Atlanta.
- IDS Scheer (2005). Enterprise Architectures and ARIS Process Platform, Saarbrücken.
- IEEE (2000). IEEE Recommended Practice for Architectural Description of Software Intensive Systems (IEEE Std 1471-2000), IEEE Computer Society, New York, NY.
- Kuhlin, B. and Thielmann, H., eds. (2005). The Practical Real-Time Enterprise: Facts and Perspectives, Springer Verlag,
- Linthicum, D. S. (2000). Enterprise Application Integration, AWL Direct Sales, Reading, Massachusetts.
- Mesarovic, M. D., Macko, D. and Takahara, Y. (1970). Theory of Hierarchical, Multilevel Systems, Academic Press, New York, London.
- Opengroup (2003). TOGAF "Enterprise Edition" Version 8.1, <http://www.opengroup.org/architecture/togaf8-doc/arch/>, last accessed on 27.11.2004.
- Österle, H. (1995). Business in the Information Age - Heading for New Processes, Springer, New York.
- Porter, M. E. (1998). Competitive Advantage: creating and sustaining superior performance, 2nd ed. Ed., Free Press, New York.
- Scheer, A.-W. (1999). ARIS – Business Process Frameworks, 3rd Ed., Springer, Berlin et al.
- Schekkeman, J. (2004). How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework, 2nd Ed., Trafford Publishing, Victoria, British Columbia.
- Sowa, J. F. and Zachman, J. A. (1992). Extending and Formalizing the Framework for Information Systems Architecture, in: IBM Systems Journal, 31, No. 3, pp. 590-616.
- Tang, A., Han, J. and Chen, P. (2004). A Comparative Analysis of Architecture Frameworks, SUTIT-TR2004.01, Swinbourne University of Technology, Swinbourne.
- Weill, P. and Vitale, M. R. (2001). Place to Space: Migrating to eBusiness Models, Harvard Business School Press, Boston, MA.