

An Integrated Enterprise Architecture Framework for Business-IT Alignment

Novica Zarvić* and Roel Wieringa

University of Twente, Department of Computer Science, Information Systems Group
P.O. Box 217, 7500 AE Enschede, The Netherlands
`n.zarvic,roelw@cs.utwente.nl`

Abstract. When different businesses want to integrate part of their processes and IT, they need to relate their enterprise architecture frameworks. An enterprise architecture framework (EAF) is a conceptual framework for describing the architecture of a business and its information technology (IT), and their alignment. In this paper we provide an integration among some well-known EAFs (Zachman, Four-domain, TOGAF and RM-ODP) and produce an integrated EAF (IEAF) that can be used as common framework to communicate about EAFs of different businesses and relate them to each other.

1 Introduction

Businesses describe their enterprise architecture using an Enterprise Architecture Framework (EAF), which is a structure for documenting the architecture of their IT systems. Usually, each business uses its own EAF, which may or may not be documented. If undocumented, the EAF is a kind of implicit conceptual metamodel of the architecture of their IT systems. However, when different businesses want to cooperate, they have to relate their EAFs to each other, and this means they should document their EAFs. While doing so a common understanding of each others EAFs is needed. We do not claim that businesses should replace the EAF they work with and that all change to the same EAF, but as far as existing EAFs are built on different abstraction mechanisms it is necessary to understand each others frameworks in order to be able to communicate in a reasonable way. An EAF, capable to integrate existing frameworks, is useful for this task, because it can show how the integrated EAFs relate to each other.

Very little has been written to date about EAFs. Langenberg and Wegmann [1] map the research field but make no attempt at comparing EAFs. Schekkerman [2] lists a lot of EAFs without comparing them. Tang et al. [3], finally, compare EAFs but do not attempt to integrate them. In this paper we compare and integrate some of the well-known EAFs. Section 2 discusses and defines the terms Enterprise Architecture and Enterprise Architecture Framework. Section 3 presents a particular framework, called GRAAL framework, and

* supported by the Netherlands Organisation for Scientific Research (NWO), project 638.003.407 (Value-based Business-IT Alignment)

shows how the other frameworks relate to it. Section 4 then presents our integrated framework and discusses the implications for business-IT alignment in value constellations.

2 Defining Enterprise Architecture and Frameworks

We start from the concept of a system as any coherent collection of elements [4]. Software systems are systems, the set of all applications of an organisation is a system, and organisations are systems too. We define *architecture of a system* as “the structure of a system, consisting of the relationships among its components, the external properties of those components, and the way these create emergent properties with added value for the environment”. Like the IEEE architecture definition [5], we consider there to be a single architecture of a system. There can be many different views of an architecture [6], each of which documents a different aspect of the architecture, but we think that there is a single structure which is the combination of all views. Note also that the architecture of a system is not just the structure of the system, but it includes the way in which this structure creates an added value for the environment of the system [7].

The concept of Enterprise Architecture (EA) is defined by various sources as the structure of the IT systems of an enterprise, or even of the entire enterprise, or sometimes as an analysis and documentation of this structure rather than the structure itself [8,9,10,11]. We define an EA simply by restricting ourselves to IT systems in an enterprise context: “An *enterprise architecture* is the structure of an enterprise, consisting of the relationships among its ICT systems, the external properties of those ICT systems, and the way these create emergent properties with added value for the enterprise”. The term EAF, finally, is used mostly to indicate a list of important abstraction mechanisms, such as perspectives, viewpoints, architectures, dimensions, etc. To be neutral with respect to the abstraction mechanisms used, we define an *enterprise Architecture Framework* as “a documentation structure for Enterprise Architectures”. A company can use an EAF to structure descriptions of architectures in such a way that these descriptions can be compared in a meaningful way, to control the design of interfaces among IT systems, to create a repository for storage and retrieval of EA documentation, or as a set of guidelines that assists the development of an EA [12,13,11,14]. In this paper, we look at its role in connecting IT systems of different enterprises.

3 A Comparison of EA Frameworks

The GRAAL Framework. To compare EAFs, we use one framework as reference, namely the GRAAL framework used in our architecture research [15,12,16,17]. The GRAAL (Guidelines Regarding Architecture ALignment) framework derives from a framework for information systems development methods [18,19] and has

been used in the GRAAL project¹ to compare architecture alignment in different organisations in the Netherlands.

The GRAAL framework divides an organisation and its IT into a number of layers, where each layer contains entities (systems in the general sense) providing services to entities in the layers above it. From the bottom up we distinguish the physical infrastructure layer, the software infrastructure layer, the enterprise system layer (i.e. applications and information systems), the enterprise, and its environment (See the examples in Fig. 1). Each company may add more detail to a particular layer, such as for example distinguish different infrastructure domains, or distinguish enterprise systems into information systems and applications. These are refinements, and the GRAAL framework contains only the greatest common divisor of all these company-specific frameworks. The essential characteristic of the GRAAL layers is that entities at one layer provide services to entities at higher layers.

The second GRAAL dimension is that systems at each level have certain aspects. Foremost among these is that each system provides services (to systems in higher layers); each service should provide some added value (utility) and does this by engaging in behaviour over time, during which data is exchanged with other systems over communication channels. (We restrict our attention to systems that exchange data.) And these services are delivered at a certain level of quality.

The GRAAL framework contains three other dimensions. The *decomposition* dimension says that each system is decomposed into subsystems, that are encapsulated in it. The *system life* dimension says that each system goes through stages in its life, from conception to disposal. The *refinement* dimension says that each system can be described at different levels of abstraction, from very abstract (few details) to very detailed.

The Zachman Framework [13,20]. The rows in the Zachman framework represent the perspectives of different roles on the system (Fig. 1(a)). The planner considers the scope of the system in relation to the environment, the owner considers the role of the system in the enterprise, the designer considers the software needed to achieve the business goals, and the builder considers the infrastructure needed to build the system. So far, this corresponds to layers in the GRAAL framework. The subcontractor role moves however to subsystems of a system, and this is moving along another GRAAL dimension, namely the decomposition dimension. Because decomposition can be done at any level in the GRAAL framework, it should not be placed at the lowest level only, as Zachman does.

The data, function and network aspects of Zachman correspond to the GRAAL aspects of data, service and communication. Zachman's time aspect corresponds roughly to the behaviour aspect in GRAAL, because the behaviour as a functional property of a system is the ordering of product interactions in time [21, p. 40]. The people aspect is represented in GRAAL's enterprise layer, because people are part of the enterprise and therefore of the business aggregation hier-

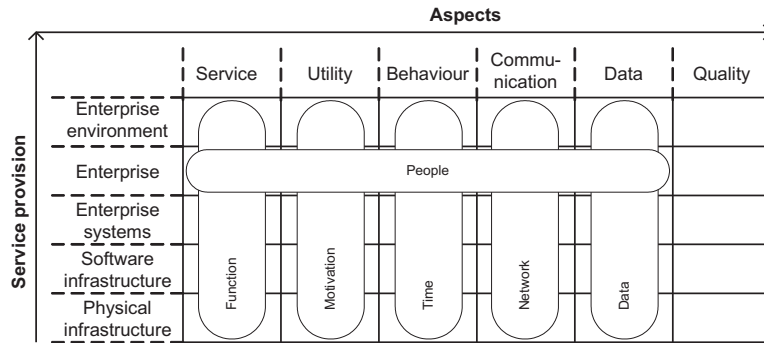
¹ See <http://is.cs.utwente.nl/GRAAL>.

archy. Finally, the motivation aspect from Zachman corresponds to the utility aspect of GRAAL, because the utility of an entity at any layer for entities at higher layers is the motivation why this lower-level entity exists. We conclude that the Zachman framework can be mapped into the GRAAL framework.

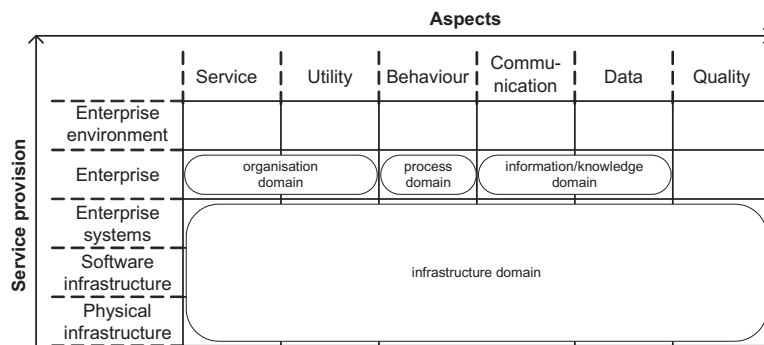
The Four-Domain Architecture [22]. This framework distinguishes four domains (Fig. 1(b)). The process domain includes processes, procedures, business tools and dependencies required to support business functions. This corresponds to the behaviour aspect of entities at the enterprise level. The information/knowledge domain includes business rules, data, all types of information, definitions, interrelationships, etc. and corresponds to the aspects communication and data, in the GRAAL framework, also at the enterprise level. The infrastructure domain includes facilities, hardware, system software, networks, etc. This corresponds to the software infrastructure and physical infrastructure layers from GRAAL. It spans even the quality aspect of GRAAL, because reliability and availability are elements expected to be documented within the infrastructure domain. The organisation domain includes business people and their roles and responsibilities, organisational structure as well as interrelationships to all kind of stakeholders. This corresponds partly to the utility and service aspect of enterprise-layer entities in the GRAAL framework (who does what for whom?). The organisational structure and relationships are part of the system decomposition dimension of GRAAL, which is not shown in our figures. Note that the Four-Domain Architecture was developed for managing EAs and to support frameworks such as the Zachman's. Iyer and Gottlieb also distinguish architecture design from architecture use and split each of the cells of Zachman's framework into two subcells, corresponding to these two stages in the life of an architecture. This corresponds to the system life dimension of GRAAL.

TOGAF [14]. TOGAF is, compared to all other frameworks presented in this paper, the most comprehensive one, because TOGAF offers a complete guide for the development of an EA and comes up with an architectural development method. Such a step-by-step guide is absent at Zachman and GRAAL. TOGAF distinguishes four kinds of architectures, namely business architecture, data architecture, application architecture and technology architecture, where data architecture and application architecture is sometimes referred to as information systems architecture. We consider these architectures as the highest-level building blocks to be documented. The mapping to GRAAL is straightforward (Fig. 1(c)).

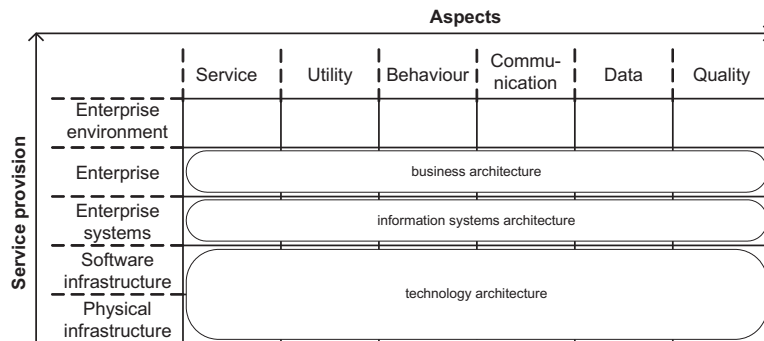
RM-ODP [23]. RM-ODP (Reference model for open distributed processing) provides five viewpoints (Fig. 1(d)), where a viewpoint is "a subdivision of the specification of a complete system, established to bring together those particular pieces of information relevant to some particular area of concern during the design of the system" [24]. A viewpoint in RM-ODP can span aspects of one or more viewpoints in GRAAL and vice versa. The RM-ODP enterprise viewpoint focuses on the purpose, scope and policies of the system and provides the overall



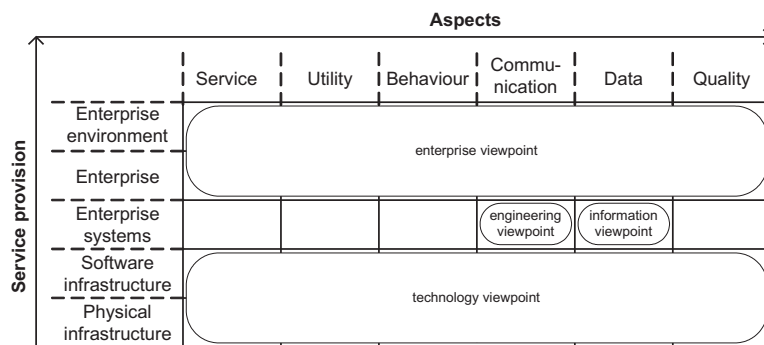
(a) Comparison between GRAAL and the Zachman framework



(b) Comparison between GRAAL and the Four-Domain Architecture



(c) Comparison between GRAAL and TOGAF



(d) Comparison between GRAAL and RM-ODP

Fig. 1. Comparison between GRAAL and other well know frameworks

environment in which the system will be built. This corresponds roughly to the enterprise and enterprise environment layers in the GRAAL framework. However, the RM-ODP enterprise viewpoint can also specify more technical entities like operating systems or database systems [25], which lie in GRAAL on the software infrastructure layer. This is not represented in Fig. 1(d).

The information viewpoint is a viewpoint on the system and its environment that focuses on the semantics of the information and information processing performed. This corresponds roughly to the data aspect on the enterprise system layer in GRAAL. The computational viewpoint enables distribution through functional decomposition of the system into objects which interact at interfaces. In GRAAL this viewpoint corresponds to the decomposition dimension. It is not shown in Fig. 1(d). The engineering viewpoint focuses on the mechanisms and functions required to support distributed interaction between objects in the system, which corresponds roughly to the communication aspect on the same layer. The technology viewpoint focuses on the choice of technology in the system. It is used to build technology specifications of particular configurations of hardware elements, software elements and networks. The technology viewpoint corresponds to the physical infrastructure and software infrastructure layers in GRAAL.

4 Discussion and Further Work

The paper described briefly five frameworks and compared them. The basis of the comparison was the GRAAL framework. This section summarizes our results.

Abstraction mechanisms. The frameworks use different abstraction mechanisms. GRAAL and Zachman use dimensions, but the other frameworks *populate* some of these dimensions. The GRAAL dimensions Aspects and Service layers correspond roughly to the Zachman dimensions Aspects and Perspectives. The GRAAL dimensions Decomposition, System life, and Refinement are not mentioned as such by Zachman, although he does include a decomposition perspective (subcontractor). As we have seen, the abstraction mechanisms of the other frameworks, namely the domains, architectures and viewpoints are mostly a combination of the system aspects dimension and the service layers in GRAAL. The abstraction mechanisms used by GRAAL and Zachman are at a metalevel with respect to the others, which explains the relationship between the different frameworks. EAFs at a higher level of abstraction can integrate frameworks that use lower level abstraction mechanisms.

The Integrated Enterprise Architecture Framework. Our analysis makes clear that to find an integrated framework, we can build on GRAAL. We just extend GRAAL with a number of domains, which we place on the appropriate layers. Figure 2 shows the resulting integrated EAF (IEAF). In the IEAF we merged the two infrastructure layers for simplicity. Many EAFs, like RM ODP or the Four-Domain architecture also do not have a distinction between the two on their

highest level of abstraction. This serves to integrate such frameworks more easily. The *enterprise network* domain contains sets of interacting business actors that are profit-and-loss responsible, such as independent businesses, or business units within a large corporation. Therefore it is especially interesting for networked business constellations. The *organisation structure* domain of the IEAF contains the decomposition of an organisation into whatever elements are recognised, such as units, departments, employee roles, etc. The *business process* domain consists of business processes and the *communication and information* domains consists of human or automated communication channels and the information passed through them. The *services* domain consists of IT services, and the *behaviour* domain consists of software behaviour. The *infrastructure* domain consists of all software and hardware needed to facilitate the higher layers. Note that the IEAF has the advantage that each layer is not fix but flexible. This means that, if necessary, additional domains can be added to an IEAF layer, which goes so far that domains can even span several layers as shown in Fig. 2. Because the layers are not fix they can be viewed as dimensions.

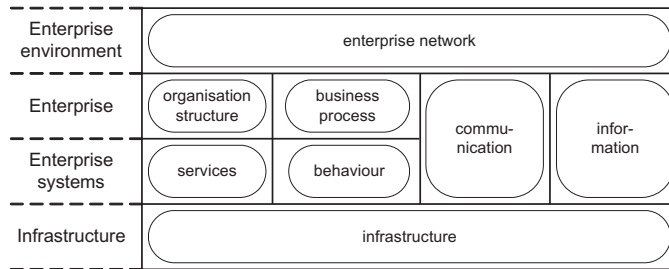


Fig. 2. An Integrated Enterprise Architecture Framework (IEAF) for networked business constellations

The implication of the IEAF for business-IT alignment is that each domain is an area of design knowledge, and that the alignment problem must be decomposed into these domains. In the case of interorganisational business-IT alignment, the additional implication is that the EAFs of the partner companies can be mapped to each other using the IEAF as common reference. Our research implies that each of these EAFs can be mapped to the IEAF and that this is the way the EAFs should be mapped to each other. Further, the IEAF forms a basis for communication between different businesses cooperating in a networked value constellation.

Evaluation. In design science the evaluation phase is very important. “Design science addresses research through the building and evaluation of artifacts” [26]. The GRAAL framework, as our built artifact, was used to analyse the EAFs of five different organisations in the Netherlands, each having a different EAF. All the information from those EAFs could be incorporated without information

loss into the GRAAL framework. We consider this as a first validation of the GRAAL framework. The IEAF, as our simplified artifact, inherited most of its documentation structure from the GRAAL framework. For the evaluation of our artifacts case studies were used, like described by Hevner et al. [26, page 86].

As we have seen during the comparisons in Sec. 3, the abstraction mechanisms, namely domains, architectures and viewpoints of the other frameworks are mostly a combination of the systems aspect dimension and the service layers in GRAAL. Therefore dimensions, like the layers in the IEAF, are at a higher level of abstraction than the abstraction mechanisms used by most of the other EAFs. We also identified that the frameworks use a fixed number of abstraction mechanisms. The number of domains in the IEAF (which is not fixed) was identified and situated on the appropriate layers after having examined the frameworks used by our industrial partners for documenting their enterprise architectures. They covered all the information documented. Therefore it is true that it is an integrated EAF.

Corroboration of the IEAF is also provided by an independent investigation of IT architect roles [27]. The architect roles distinguished by most companies coincide with the domains identified in the IEAF. Further validation is provided by a preliminary investigation of the EAFs of several companies that were compared with the IEAF and could all be mapped to it without loss of information.

Summary and Future Research. The frameworks described in this paper use different abstraction mechanisms, but can nevertheless be mapped into the GRAAL framework with some degree of approximation. The result is an integrated EAF (IEAF). The IEAF serves as a reference point between different organisations, and enables them to understand each others frameworks. The presence of the domains provide an additional useful utility. In the future we will investigate cross-organisational integration problems using the IEAF as our conceptual framework.

References

1. Langenberg, K., Wegmann, A.: Enterprise Architecture: What Aspects is Current Research Targeting? Technical report, EPFL (2004) http://icwww.epfl.ch/publications/documents/IC_TECH_REPORT_200477.pdf.
2. Schekkerman, J.: How to survive in the jungle of Enterprise Architecture Frameworks. Trafford (2004)
3. A. Tang and J. Han and P. Chen: A Comparative Analysis of Architecture Frameworks. Technical report, Swinburne University of Technology (2004) <http://www.swin.edu.au/ict/research/technicalreports/2004/SUTIT-TR2004.01.pdf>.
4. Blanchard, B.S., Fabrycky, W.J.: Systems Engineering and Analysis. Prentice-Hall (1990)
5. IEEE Architecture Working Group: Definition of Architecture. http://www.pithecantropus.com/~awg/public_html/ieee-1471-faq.html (2002)
6. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley (1998)

7. Wieringa, R.: Architecture is Structure plus Synergy. <http://graal.ewi.utwente.nl/WhitePapers/Architecture/architecture.htm> (2004)
8. West, D., Bittner, K., Glenn, E.: Ingredients for Building Effective Enterprise Architectures. http://www-128.ibm.com/developerworks/rational/library/content/RationalEdge/nov02/EnterpriseArchitectures_TheRationalEdge_Nov2002.pdf (2002)
9. Schekkerman, J.: The Economic Benefits of Enterprise Architecture. Trafford (2005)
10. Perks, C., Beveridge, T.: Guide to Enterprise IT Architecture. Springer (2003)
11. Bernard, S.A.: An Introduction to Enterprise Architecture. Authorhouse, Bloomington, Indiana (2004)
12. Wieringa, R.: The GRAAL Architecture Framework. <http://graal.ewi.utwente.nl/WhitePapers/Framework/framework.htm> (2004)
13. Zachman, J.: A framework for information systems architecture. IBM Systems Journal **26**(3) (1987)
14. The Open Group: The Open Group Architecture Framework (TOGAF) - Version 8, Enterprise Edition (2002)
15. van Eck, P., Blanken, H., Fokkinga, M., Grefen, P., Wieringa, R.: A Conceptual Framework for Architecture Alignment Guidelines. http://graal.ewi.utwente.nl/GRAAL_whitepaper_20021017.pdf (2002)
16. Wieringa, R., Blanken, H., Fokkinga, M., Grefen, P.: Aligning Application Architecture to the Business Context. In: Conference on Advanced Information System Engineering (CAiSE 03), Springer (2003) 209–225 LNCS 2681.
17. Wieringa, R., van Eck, P., Krukkert, D.: Architecture Alignment. In Lankhorst, M., ed.: Enterprise Architecture at Work. Springer, Berlin (2005)
18. Wieringa, R.: Requirements Engineering. John Wiley, Chichester, England (1996)
19. Wieringa, R.: A Survey of Structured and Object-Oriented Software Specification Methods and Techniques. ACM Computing Surveys **30**(4) (1998) 459–527
20. Sowa, J., Zachman, J.: Extending and formalizing the framework for information systems architecture. IBM Systems Journal **31**(3) (1992) 590–616
21. Wieringa, R.: Reactive Systems. Morgan Kaufmann, San Francisco (2003)
22. Iyer, B., Gottlieb, R.: The Four-Domain Architecture: An approach to support enterprise architecture design. IBM Systems Journal **43**(3) (2004) 587–597
23. International Organization for Standardization: RM ODP - Open Distributed Processing Reference Model - ISO/IEC 10746-1, ISO/IEC 10746-2, ISO/IEC 10746-3 and ISO/IEC 10746-4. (http://isotc.iso.org/livelink/livelink/fetch/2000/2489/Ittf_Home/PubliclyAvailableStandards.htm)
24. Vallecillo, A.: RM-ODP: The ISO Reference Model for Open Distributed Processing. (citeseer.ist.psu.edu/277001.html)
25. Joyner, I.: Open distributed processing: Unplugged! <http://homepages.tig.com.au/~ijoyner/ODPUnplugged.html> (1997)
26. Hevner, A., March, S., Park, J.: Design Science in Information Systems Research. MIS Quarterly **28**(1) (2004) 75–105
27. Wieringa, R.: An inventory of architect roles and competencies. In: IT Practitioners Conference, Barcelona (2006) <http://www.opengroup.org/public/member/proceedings/q106/>.