Using Orientor Theory for Coherent Decision Making for Application Landscape Design

Alexander W. Schneider and Florian Matthes

Technische Universitt München
Munich, Germany
Email: {schneial|matthes}@in.tum.de

Abstract. More than 30 years have past since the first enterprise architecture (EA) management framework has been published. While the field has reached a certain maturity in science and practice, a paradigm shift is just beginning. The emerging trend to regard an enterprise as a complex adaptive system might allow to redefine and achieve the holistic nature of EA management approaches. Thereby, the focus on static aspects of the architecture might be extended by taking behavioral aspects into account. In this paper, we argue (1) that currently available EA management approaches fall short in achieving the desired holism, i.e. viewing the system (application landscape) as a whole, (2) apply orientor theory to support decision making by deriving coordinated goals and principles for application landscape design and (3) show how a system-of-systems approach of applied orientor theory could look like. We conclude the paper by sketching out an appropriate research agenda.

Keywords: enterprise architecture, complexity, environment, orientor theory, decision support

1 Introduction

For quite some time, researchers regard enterprises as open dynamic systems [34]. Thereby, they overcome the self-centering view of traditional economics and organizational theory as well as the reductionism prevalent in recent enterprise architecture research. Thereby, they also extend their view to take the environment and respective interactions into account [18]. It seems that this extended viewpoint allows for a deeper understanding and development of new methods in the context of accelerating changes requiring steady adaption. Disruptive technologies, increasing regulation and changing customer demands are just a few general examples. It is obvious that enterprise architects have to design the enterprise and especially the application landscape to optimize fitness with respect to these challenges. Therefore, science has to provide methods to support this process. As a first step into this direction, we propose to search for other disciplines which already established similar thinking to develop solutions for similar problems. In this paper we describe one of those modeling approaches which has been successfully used in ecology and economics, namely orientor theory. We show how orientors might be used to create a formal and deeper understanding of the behavior of application landscapes as a whole taking also their environment into account. Thereby, we not only provide a conceptual integration of different aspects of relevant behavior but also show their dichotomies. Grounded on the assumption that agents with deeper understanding of the system behavior make better decisions the application of orientor theory is a promising tools to model the behavior of enterprise architectures. Finally, we conclude the paper by describing a conceivable road-map for enterprise architecture research that accounts also for dynamics of the enterprise.

2 Problem Statement

2.1 General Goals of EA Management and its Claim for Holism

Even a short literature review reveals that holism is frequently attributed to EA management approaches. A search for the terms "enterprise architecture" AND "holistic" in Google Scholar performed in April 2014 resulted in more than 4.670 articles and book chapters. It is more often than not the case that holism is considered to be a mandatory attribute of EA approaches [25]. Therefore, there is general agreement on the scope of EA initiatives and models. Nevertheless, this is not the case for concrete goals EA management initiatives should pursue. In literature, EA benefits such as improved change management and improved risk management are mentioned frequently [25], some authors also promise an increase of market value [32], better customer orientation and improved alignment with business partners [16]. It becomes obvious that EA initiatives have a wide-ranging scope covering the whole enterprise as well as the enterprise as a whole in its environment.

2.2 Reductionism and Complex Systems

The fundamental idea behind modeling static non-living aspects of an organization with entities and attributes, i.e. EA documentation, to understand or design the system is called reductionism. This philosophical position holds that a system can be completely explained and understood by looking at its constitutive elements and their relationships. Thereby, each phenomenon can be explained in terms of relations between other more fundamental phenomena. In contrast, the science of complex systems teaches us that one inherent characteristic of complex systems is their emergent behavior [14] which is also applicable to organizations [23]. Nowadays, emergent phenomena are considered to be the exact opposite of reductionism, they can hardly be traced back to the interaction of phenomena of single elements. Colloquially, this notion is formulated as "the whole is more than the sum of its parts" which dates back to Aristotle. The problem that arises from the prevalent reductionistic approach of EA management is that the focus on the micro-level is not appropriate to completely explain or even design the behavior on the macro-level. This is especially surprising since, as shown before, the macro-level is the area of interest of holistic

EA management. Furthermore, many researches have shown that organizations qualify to be regarded as complex adaptive systems [34, 12] since they exhibit complex, adaptive and emergent behaviors due to multiple interacting agents [8]. To qualify for a reductionistic approach we need to be aware of the laws declaring how theories on the micro-level, e.g. the resistance to change of a single application, relates to theories on the macro-level, e.g. the resistance to change of an application landscape. As long as science has not accomplished to reveal such laws, a pure reductionistic approach, e.g. by collecting a huge amount of data on the micro-level, is doomed to fail when trying to support design decisions on the macro-level. Although reductionism in general faces criticism in scientific literature (e.g. [15, 20]), we only want to point out here that if reductionism is applied it has to be done on a strong theoretical basis.

2.3 Reductionism is Prevalent in EA Management

A well-known problem in EA modeling is to define the breadth and depth of EA documentations, i.e, relevant elements and their level of detail. If not done correctly issues related to overmodeling and overuse of detail might occur [1]. Typical issues include analysis paralysis, delayed delivery of results and high costs for data collection. The observation that many companies in practice faced such problems clearly indicates that the holistic idea of EA management has often been interpreted by reductionists. Hereby, the assumption seems to be that the more elements and the more details are modeled or documented the closer one gets to a 'holistic' approach. Another way of interpreting holistic in this context would be to look at the whole system, e.g. an application landscape, instead of focusing on its constituting elements.

As mentioned before, (IT) cost reduction is one of the major claims of EA management. Existing literature suggests that EA facilitates building a more standardized IT platform with fewer technologies, leading in turn to simplified interfaces, higher reliability through reduced operating platform complexity, and lower maintenance and support costs [37]. Thereby, the assumed law seems to be that an application landscape's cost is just the sum of the costs of its elements. While this holds true in a narrow context, it might not be true from a holistic point of view. On the one hand, while infrastructure operating costs might decrease, application development costs can increase due to necessary workarounds required in a fixed technology context. On the other hand, cost reductions based on technology standardization and structural complexity reduction are only short term cost optimizations. Risks associated to extensive standardization might cause expenses in the future and have therefore be regarded in respective calculations. To our knowledge, most cost cutting approaches based on EA management neglect the system's of path-dependence. The statically viewed EA has a history which has to be taken into account. For example, investments have been made for some elements in the past while their expected benefits will materialize in the future. Consequently, they cannot be changed without loosing (parts) of the promised benefits. Furthermore, IT systems are not autonomous elements within the application landscape which can be changed easily. For example, an organization employs IT staff with specific knowledge which could get lost in case of standardization. Additionally, standardization activities on the application landscape might require standardization activities within the business layer.

2.4 A Paradigm Shift in Enterprise Architecting

Despite the fact that many existing EA approaches applied a naive reductionistic thinking even in the absence of concrete laws to build reductions, we see evidence for a paradigm shift in enterprise architecting from reductionism to holistic thinking. For example, the co-evolution path model describes how an enterprise as a whole behaves in the context of its environment [18]. While the complexity of the environment increases, the enterprise has to increase its complexity as well. But an overshot could be very dangerous and therefore each enterprise has to maintain an adequate level of complexity over time. An overview of complexity work in EA management can be found in [33]. Additionally to hard facts as provided by existing EA approaches, we also see an opportunity in supporting EA decisions indirectly via creating a better understanding of the system as a whole among EA decision makers. The benefits of shared mental models, conventions and language have already been recognized in the context of virtual teams [22], customer relationship management [27] and lab experiments in general [35]. Therefore, we argue that models facilitating such shared mental models or language should also be established for the domain of enterprise architecting because a shared IT-business understanding allows companies to conceive, implement and use innovative IT applications to improve process performance [28].

3 Applying Orientor Theory to Model Application Landscapes in their Environment

As outlined before, we want to extend the scope of EA models towards achieving a holistic view. A literature review conducted in 2014 revealed, that currently the dynamic aspect as well as the subjective aspect of an enterprise's complexity is underrepresented in EA literature [33]. Therefore, we employ orientor theory as a method which might be able to create such shared mental models in the context of application landscape design. Bossel [5] defines orientors as the "set of criteria that are relevant for the evaluation of system development [...] that systems (or their managers) use to orient their decisions and actions regarding the system". Although orientor theory originates from ecology, it has been used to describe complex systems in arbitrary contexts [6].

3.1 Benefits of Orientor Theory for Application Landscape Decisions

The application of orientors is a means to cope with situations in which the desired state of a system is not agreed upon the designers. Due to the large number of different interest groups directly involved in planning and decision-making processes influencing an application landscape, several points of view have to be taken into account. Especially in ecosystems, where orientor theory has been developed, in such situations a single objectively derived best solution does not generally exist [13]. Therefore, orientors form conditions that we can apply to systems in order to judge their sustainability [36]. Because decisions about extensive application landscape (AL) transformations are made in boards where people with different views and goals converge, the application of orientors and especially their dichotomy can help to establish shared understanding of the problem among the decision makers. That shared problem understanding can help establish development priorities and keep senior management focused on generating benefits from new IT capabilities has already be shown in practice [31]. In addition to that, orientors also offer strategic guidance for decision making processes on different levels of governance. A framework based on orientors would allow to balance different interest and therefore ensure the whole system's viability.

Furthermore, the application and modeling of orientors not only allows to assess the current state of a system and better understand opposing goals but also allows to agree upon desired orientors the system should follow. Based on such framework and concrete strategies goal derivation can be facilitated. Therefore, orientor theory should be of interest especially for EA approaches following the Enterprise Ecological Adaptation school of thought [19], whereby sustainability and organizational coherence are major goals.

3.2 Orientor Theory and its Implications for EA Management

According to orientor theory each system orients towards six environmental aspects namely normal state, scarce resources, variety, evolution, variance and other systems. Here, we focus on the AL as the system under investigation including applications, infrastructure as well as the people interacting with them. Figure 1 depicts the six pairwise contradictory orientors and their respective environmental influences.

Existence The existence orientor refers to the normal state of the environment which means that the system has to maintain its state variables constant to enable functioning under given circumstances. This orientor requires:

- A protective shell preserving the systems from threats able to push the system state out of the acceptable range
- No failure of system structure
- No self-destructive behavior

Following the existence orientor an AL would, e.g., install firewalls to protect applications from hackers. To prevent failures of the system structure all relevant stakeholders have to be involved in AL design decisions which in practice is still an issue [21]. IT capabilities for this orientor include IT service management as well as a monitoring capability.

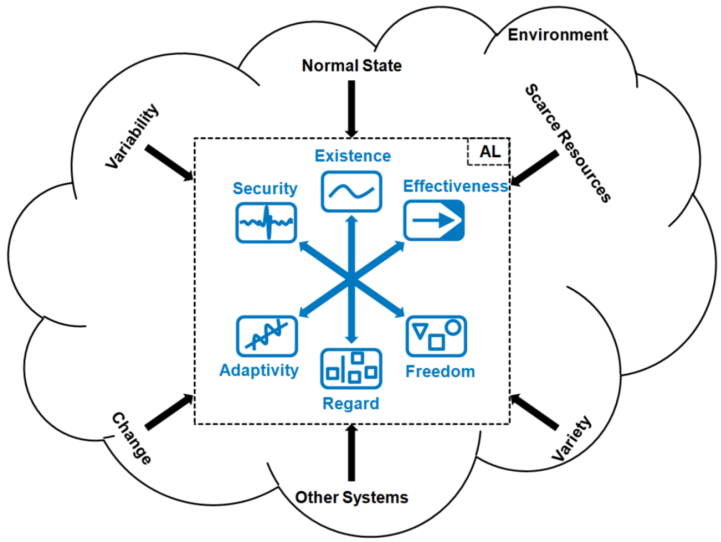

Fig. 1. Application landscapes modeled according to orientor theory [5, 10]

Effectiveness The effectiveness orientor refers to scarce resources in the environment and how they can be secured. Within the system, resources have to be distributed wisely and used in an efficient way, e.g. budget, time and knowledge. Externally, the system has to ensure efficient acquisition of scarce resources by connections to the environment and other subsystems. Knowledge acquisition through hiring new employees can be difficult, for example in the domain of EA management [21]. In order to acquire budget, executive support or management commitment is needed. But this is often rather low regarding EA management which is often seen as an operational initiative rather than a strategic concept with long-term redemption [17].

Following the effectiveness orientor an AL has to continuously balance efforts to make processes and applications more efficient with efforts creating assets which are non-efficient but effective in the long run. The better the ratio of efforts and outcomes the more orientation towards effectiveness. But that does not imply that the ratio has to be good at any point in time. To stay viable the system has to maintain a positive ratio on average over time. Therefore, an integration of EA management and project portfolio management is inevitable [11]. However, the system needs to ensure access to required resources from the environment. For application landscapes this implies, e.g., access to people skilled in programming languages in use or skilled enterprise architects which are still issues in practice [26, 21]. Another means to accomplish efficiency is to increase standardization within the AL or especially within the infrastructure layer.

Freedom The freedom orientor refers to the variety of the environment and describes the system's freedom of action. Thereby, a system needs as much freedom as its environment offers variety [2]. In general, a system has to secure itself from overextension by using one of the following strategies:

- Reacting by using the systems repertoire
- Reacting by influencing the system's environment
- Reacting by searching for a new environment

Following the freedom orientor an AL would try to achieve a maximum of action alternatives and limit the amount of fixed structures. These include but are not limited to long-term contracts with infrastructure or application providers, a high penetration of a single vendor within the AL as well as non-redundant processes and applications. Another ability relevant for ALs is to cope with technological progress, especially with disruptive technologies as they might limit the whole system's viability if the system is constrained in its action alternatives. Some companies even changed their environment in presence of a significant environmental, i.e. external, change by moving their headquarter overseas [3]. Employing diverse workforce, having a modular AL and decentralized governance structures also supports the freedom orientor.

Adaptability The adaptability orientor regards the evolution of the environment. If a system is not able to elude from threatening influences it has to adjust its parameters or even its structure. In general, systems can either adapt their structure or their behavior. Changing the system's structure can result in a new system differing explicitly from the old one. In contrast, changing only the behavior is also considered as co-evolution and which is mostly suitable if small environmental changes occur. Such adaptability requires a certain degree of self-organization. In particular, the following conditions facilitate adaptivity:

- versatile system components
- variety within the system structure
- redundant but physically different processes
- decentrality and partial autonomy
- memory as information storage to enable learning

Following the adaptivity orientor would require, e.g., a modular architecture with loosely coupled applications, data and technology components which allows to set global standards while also allowing regional differences [30]. In order to apply versatile components and variety within the AL applications can be build on different technologies and programming languages and conscious acceptance of functional redundancy. Since adaptivity requires the ability to learn knowledge management becomes vital for adaptive ALs, cf. [9]. An example for a structural change of the AL is a transition to a service oriented architecture. Fostering an open organizational culture and having flexible structures directly supports the adaptivity orientor.

Security The security orientor regards temporal variances of influences and ensures that the system is safe from unforeseen harmful influences. Therefore, the system needs to be mostly independent from unstable environmental factors and dependent only on stable environmental factors. In general, this can be accomplished, e.g. by

- setting up buffers to contain overloads and bypass supply gaps
- establishing self-regulating structures
- defusing potentially harmful threats

For application landscapes following the security orientor would imply, e.g., to keep enough knowledge within the company to overcome potential supply gaps on the market. Furthermore, to establish self-regulating structures an comprehensive monitoring and governance capability is needed. Defusing harmful threats in this context could be achieved, e.g., by setting up uninterrupted power supply units or different connections to the Internet. Setting up resource buffers, e.g. via server virtualization, secures individual applications from request overloads. Especially the implementation of appropriate risk management and continuity management support the security orientor.

Coexistence The coexistence orientor refers to other systems influencing the system and anticipative behavior. Each system has to consider the behavior and interests of other systems for its own interest. Usually, each influence from the environment has an 'unsystemic' component as well as a component consisting of the behavior of other systems. Since sometimes a specific system is of special interest, that system has a special role within the coexistence orientor. It requires:

- the ability to realize that another system is affected by some influence
- the availability of behavioral patterns

While one can imagine a huge amount of relevant systems for an application landscape an important one should be the enterprise or business units. Being informed about current strategies, e.g. to increase the number of customers for a certain product, allows the application landscape to invest in respectively required capabilities such as scalability. Other systems of interest could be providers of infrastructure or applications. Environmental influences like the dissemination of cloud computing might influence the adaptivity of them and therefore indirectly influence the AL. Therefore, sensors as well as analytics capabilities are required for the coexistence orientor.

3.3 Mapping Established Enterprise Architecture Principles to Orientors

In order to steer an enterprise architecture (EA) in general and an application landscape (AL) in particular adopting EA principles is a pervasive means, cf. [29]. Because principles are used to steer an application landscape towards a specific direction, they obviously qualify to be mapped on the six basic orientors. We will do this for three exemplary EA principles formulated in the industry standard TOGAF (The Open Group Architecture Framework) [38].

Common Use Applications "Development of applications used across the enterprise is preferred over the development of similar or duplicate applications which are only provided to a particular organization". This principles directly orients towards effectiveness because the intention is to save costs and time during implementation and operation of different IT systems performing the same tasks. Thereby, it renounces the freedom as well as the adaptivity orientor. If applied, freedom in term of action alternatives will be limited because, e.g., the number of available add-ons is limited if only one solution is used. Setting global standards also prevents or impedes local evolution and therefore limits the adaptivity of the AL.

IT Responsibility "The IT organization is responsible for owning and implementing IT processes and infrastructure that enable solutions to meet user-defined requirements for functionality, service levels, cost, and delivery timing". This principle obviously orients towards the existence of the AL because defined responsibilities ensure that someone really cares about a system or entity. However, the principle reduces the regard to other systems, e.g. the business units, because keeping the AL viable might become more important than keeping the depending business units viable.

Technology Independence "Applications are independent of specific technology choices and therefore can operate on a variety of technology platforms". This principle clearly orients towards adaptivity because if applied it allows for a grater variety of components. On the one hand, e.g., if a new data storage providing better performance is offered by the environment the AL can exploit these performance gains. On the other hand, the orientation towards effectiveness is lowered because defining and managing interfaces and shared protocols which might be subject to evolution increases costs.

3.4 A System of Systems Approach

In the previous section we outlined how orientor theory could be used to model the role of application landscapes (AL) within their environment. But since ALs are designed systems such orientor model could also be used to support design decisions. Therefore, an enterprise architect has to decide which orientor is most important for the AL. But, an EA or AL could also be regarded as a system of systems [24] wherein the behavior of each individual is explained by the structure and arrangement of the lower individuals of which it is composed [7]. Because in such setting each sub-system again can be regarded as a system and modeled with orientors we can use the orientor approach to model ALs, domain landscapes, application clusters as well as single applications. Although such approach has already been proposed in general [4], we want to elaborate the relationship of systems' and sub-systems' sustainability here. Until now, the system-of-systems orientor approach assumed that the system's sustainability depends on the sustainability of each sub-system. The challenge is to identify the different orientors responsible for the system's sustainability, i.e. viability of the company. In the context of ALs, this could mean that the sub-system consisting of all applications supporting production processes have to orient more towards effectiveness and therefore standardize protocols and vendors whereas the sub-system consisting of all customer-facing applications has to orient more towards adaptivity since the environment is rapidly changing, e.g. mobile devices. But we also want to point out, that the system's viability can also be achieved by explicitly leaving sub-systems to die. This includes, e.g., IT carve-outs or discontinuance of a whole line of business.

4 Conclusion and Outlook

In this paper, we argued why existing EA management approaches fall short in providing a holistic approach and introduced orientor theory as a means to describe a system, i.e. application landscape, as a whole in the context of its environment. By linking aspects of existing EA management approaches to respective orientors we put them into a more general and coherent framework and thereby identified opposing forces. Furthermore, for each of the six orientors we derived implications for AL management and outlined how a system of systems approach could be used to apply orientor theory for sub-systems like domain landscapes or application clusters as well. In addition, we mapped well-known EA principles to the six basic orientors, identified that their application shifts an AL towards one orientor while dismissing another and thereby demonstrated the benefits of applying orientors for AL management. In order to underpin the applicability of the proposed modeling method researchers have to observe its application in practice. If the framework should be used to assess and steer application landscapes concrete measures have to be defined for each orientor. In case of a sufficient data base we suggest to analyze, for example, if companies within the same industry branch or of equal size orient their application landscape development towards the same orientors. It would also be worthwhile to examine the use of domain-specific orientors empirically. Furthermore, we suggest to analyze other approaches which are able to model system behavior, such as causal loop diagrams, in the context of application landscape management in order to model behavior in more detail.

References

1. Armour, F.J., Kaisler, S.H., Liu, S.Y.: Building an Enterprise Architecture Step by Step. IT professional 1(4), 31–39 (1999)
2. Ashby, W.R.: An introduction to cybernetics. Methuen, London (1976)

3. Birkinshaw, J., Braunerhjelm, P., Holm, U., Terjesen, S.: Why do some multinational corporations relocate their headquarters overseas? Strategic Management Journal 27(7), 681–700 (2006)
4. Bossel, H.: Assessing viability and sustainability: a systems-based approach for deriving comprehensive indicator sets. Integrated Natural Resource Management: Linking Productivity, the Environment and Development pp. 247–266 (2003)
5. Bossel, H.: Systeme, Dynamik, Simulation: Modellbildung, Analyse und Simulation komplexer Systeme. Books on Demand, Norderstedt (2004)
6. Bossel, H.: Systemzoo, Systemzoo, vol. 3. Books on Demand, Norderstedt (2004)
7. Boulding, K.E.: General systems theory–the skeleton of science. Management science 2(3), 197–208 (1956)
8. Brown, S.L., Eisenhardt, K.M.: The Art of Continuous Change: Linking Complexity Theory and Time-Paced Evolution in Relentlessly Shifting Organizations. Administrative science quarterly 42(1), 1–34 (1997)
9. Buckl, S., Matthes, F., Schweda, C.M.: Future research topics in enterprise architecture management–a knowledge management perspective. Journal of Enterprise Architecture 6(3), 16–27 (2010)
10. Burger, M.: Modeling Application Landscapes as Dynamic Systems. Master Thesis, Technische Universität München (2013)
11. Dern, G.: Management von IT-Architekturen: Leitlinien für die Ausrichtung, Planung und Gestaltung von Informationssystemen. Praxis, Vieweg + Teubner, Wiesbaden, 3., durchges. aufl edn. (2009)
12. Fuller, T., Moran, P.: Small Enterprises as Complex Adaptive Systems: a Methodological Question? Entrepreneurship & Regional Development 13(1), 47–63 (2001)
13. Gnauck, A.: Applying Ecological Goal Functions: Tools for Orientor Optimization as a Basis Decision Making Processes. In: Müller, F., Leupelt, M. (eds.) Eco targets, goal functions, and orientors, pp. 511–525. Springer (1998)
14. Goldstein, J.: Emergence as a construct: History and issues. Emergence 1(1), 49–72 (1999)
15. Horst, S.W.: Beyond reduction: Philosophy of mind and post-reductionist philosophy of science. Philosophy of mind, Oxford University Press, Oxford (2007)
16. Jonkers, H., Lankhorst, M.M., ter Doest, Hugo WL, Arbab, F., Bosma, H., Wieringa, R.J.: Enterprise architecture: Management tool and blueprint for the organisation. Information Systems Frontiers 8(2), 63–66 (2006)
17. Kaisler, S.H., Armour, F., Valivullah, M.: Enterprise Architecting: Critical Problems. In: 38th Annual Hawaii International Conference on System Sciences (2005)
18. Kandjani, H., Bernus, P., Nielsen, S.: Enterprise Architecture Cybernetics and the Edge of Chaos: Sustaining Enterprises as Complex Systems in Complex Business Environments. In: 46th Hawaii International Conference on System Sciences (HICSS). pp. 3858–3867 (2013)
19. Lapalme, J.: Three schools of thought on enterprise architecture. IT professional 14(6), 37–43 (2012)
20. Lewontin, R.C.: The triple helix: Gene, organism, and environment. Harvard University Press (2001)
21. Lucke, C., Krell, S., Lechner, U.: Critical Issues in Enterprise Architecting – A Literature Review. In: Proceedings of the 16th Americas Conference on Information Systems (AMCIS) (2010)
22. Maynard, M.T., Gilson, L.L.: The Role of Shared Mental Model Development in Understanding Virtual Team Effectiveness. Group & Organization Management 39(1), 3–32 (2014)

23. Morel, B., Ramanujam, R.: Through the looking glass of complexity: The dynamics of organizations as adaptive and evolving systems. Organization Science 10(3), 278–293 (1999)
24. Morganwalp, J., Sage, A.P.: A system of systems focused enterprise architecture framework and an associated architecture development process. Information, Knowledge, Systems Management 3(2), 87–105 (2003)
25. Niemi, E.: Enterprise architecture benefits: Perceptions from literature and practice. In: Niemi, E., Ylimäki, T., Hämäläinen, N. (eds.) Evaluation of enterprise and software architectures: critical issues, metrics and practices. University of Jyväskylä (2008)
26. Paulson, L.D.: IT hiring growth modest, but steady. IT Professional 8(1), 6–9 (2006)
27. Plouffe, C.R., Williams, B.C., Leigh, T.W.: Who's on First? Stakeholder Differences in Customer Relationship Management and the Elusive Notion of Shared Understanding. Journal of Personal Selling and Sales Management 24(4), 323–338 (2004)
28. Ray, G., Muhanna, W.A., Barney, J.B.: Competing with IT: The Role of Shared IT-Business Understanding. Communications of the ACM 50(12), 87–91 (2007)
29. Richardson, G.L., Jackson, B.M., Dickson, G.W.: A principles-based enterprise architecture: Lessons from Texaco and Star Enterprise. MIS Quarterly pp. 385–403 (1990)
30. Ross, J.W.: Creating a Strategic IT Architecture Competency: Learning in Stages. MIS Quarterly Executive 2(1), 31–43 (2003)
31. Ross, J.W.: Enterprise architecture: driving business benefits from IT. Massachusetts: Massachusetts Institutue of Technology (2006)
32. Schekkerman, J.: The economic benefits of enterprise architecture: how to quantify and manage the economic value of enterprise architecture. Trafford Publishing (2005)
33. Schneider, A.W., Zec, M., Matthes, F.: Adopting Notions of Complexity for Enterprise Architecture Management. In: Proceedings of the 20th Americas Conference on Information Systems (AMCIS) (2014)
34. Schneider, M., Somers, M.: Organizations as Complex Adaptive Systems: Implications of Complexity Theory for Leadership Research. The Leadership Quarterly 17(4), 351–365 (2006)
35. Selten, R., Warglien, M.: The Emergence of Simple Languages in an Experimental Coordination Game. Proceedings of the National Academy of Sciences 104(18), 7361–7366 (2007)
36. Spangenberg, J.H.: Biodiversity pressure and the driving forces behind. Ecological Economics 61(1), 146–158 (2007)
37. Tamm, T., Seddon, P.B., Shanks, G., Reynolds, P.: How Does Enterprise Architecture Add Value to Organisations? Communications of the Association for Information Systems 28, 141–168 (2011)
38. The Open Group: TOGAF Version 9.1 (2011), `http://pubs.opengroup.org/architecture/togaf9-doc/arch/`