2008

IT Architecturing: Reconceptualizing Current Notions of Architecture in IS Research

Martin Scheil
IT University of Copenhagen, mscheil@itu.dk

IT ARCHITECTURING – RECONCEPTUALIZING CURRENT NOTIONS OF ARCHITECTURE IN IS RESERACH

Martin Scheil Corneliussen, IT University of Copenhagen, Rued Langgaards vej 7, DK-2300 Copenhagen S, Denmark, mscheil@itu.dk

Abstract

The concept of architecture in relation to information systems research and practice is ambiguous and lacks a common agreed upon definition that combines the different areas of use. This paper investigates the origins of the concept and some of its common uses and conceptions. It is argued that the different conceptualizations of architecture are treated in an unreflectived manner and the effect hereof is a lacking understanding of IT. As an alternative this paper introduces the notion of sensemaking as a theoretical lens to understand the commonality of various conceptualizations of architecture and shift the focus from the architecture to the architect.

Keywords: concepts of architecture, IT-architecture, Information system design, Sensemaking

1 INTRODUCTION

This paper explores some issues concerning the conceptualization of architecture in relation to IS research and practice. The emphasis of the paper is put on an investigation of the origins of architecture and a consideration of how metaphors have shaped our understanding(s) of IT artefacts. From this, a discussion of how a greater awareness of these metaphors can promote a reflective orientation towards the design and use of IT (Dahlbom and Mathiassen 1995) in complex organizational settings is presented. The paper analyses different notions of architecture within the IS research field in order to identify two streams of IT conceptualization. It is argued that architecture can be considered a boundary object (Star and Griesemer 1989; Bowker and Star 2000) and that the negotiation of this object is not adequately understood. In order for such an understanding to surface an alternative notion of architecture – IT architecturing - is presented. The emphasis here is put on the architect and sees the act of architecturing as a sensemaking process (Weick 1995).

The notion of architecture in relation to IS research and practice is intriguing. Its many different uses within this area appear at first glance to be disparate. However, similarities do exist: Whether we think of architecture as a tool for decision making (Weill and Ross 2004; Hjort-Madsen and Burkard 2006), communication (Abowd, Allen et al. 1995), documentation (Zachmann 1987) or software modelling (Garlan 2000), the notion of architecture provides a certain perspective on information systems. Even though this notion, in relation to IS, has existed for roughly half a century, it seems that a deeper understanding of how the concept influents our way of thinking about various aspects of information system development still needs to be addressed. We need to reflect upon the notion of architecture, why we use it and what consequences this have for our understanding of information systems. As noted in (Bernard 2005), there are many areas concerning architectural work, that academia can and should address such as “determining the true qualitative and quantitative value of [Enterprise Architecture]” . Bernard (2005) does not elaborate on this or other subjects, but merely states that the areas of social sciences, management sciences and natural sciences all have something to offer. Though Bernard (2005) focuses on enterprise architecture, he has a point, when he states that different areas of science still have important contributions to make. However, this applies for all conceptions of architecture related to IT.

The paper is structured as follows. The next section presents a discussion of how metaphors (partly) shape our understanding, language and actions in everyday life (Lakoff and Johnson 1980; Morgan 1998), which also applies to IS as a research and practice field. That section concludes that the architectural metaphor shapes the ontological and epistemological assumptions of the field and that an awareness of these metaphors is at best vague (Smolander 2002). Hereafter, the paper analyses how the notion of architecture is conceptualized in IS research and two streams are then identified and compared. Finally, the paper discusses how architectural work can be seen as a sensemaking process (Weick 1995) shifting focus from viewing architecture as a product to a focus on the role of the architect as an agent that tries to understand the socio-technical reality of organizations in order to adopt and align new IT systems to this reality.

2 BACKGROUND OF THE ARCHITECTURAL METAPHOR

Since architecture by no means is a common agreed upon concept neither in academia nor practice an investigation of the history and origins of the metaphor and some contemporary conceptualizations of architecture is introduced in this section.

Smolander (2003) argues that a plausible reason for why it is difficult to define architecture as a concept is that the source domain i.e. building architecture is equally ill defined. Instead of continuously trying to redefine the concept of architecture we could instead put emphasis to the metaphorical nature of the concept and trace back to the novel meaning of the concept in order to improve our understanding of the mindset that we use to grasp IT with. Onians (1992) presents an overview of the history of architecture in which he argues that "[…] we use metaphors from architecture to articulate our thoughts because the processes of design and construction and the experience of using building[s] relate to basic mental operations. […] [T]here is a uniquely close relationship between building and thinking" . Onians (1992) conclude that contemporary usages of the architectural metaphor is characterized by its "richness and authority" . This suggests that the metaphor we use has agency toward the object of which it is applied. Thus, in IS research and practice the usage of the architectural metaphor does not only provide us with a figurative way of speech, but also shapes our language, mind and discourse (Lakoff and Johnson 1980). Though the history of architecture dates back to ancient Egypt (Onians 1992) the oldest preserved text on architecture is Vitruvius' *De Architectura* (see (Morgan 1914) for an English translation) dating back to 27 BC. An interesting thing about this text is the communality with present-day conceptualization of the architectural metaphor in IS research and practice. Vitruvius (1914) states that "The architect should be equipped with knowledge of many branches of study and varied kinds of learning, for it is by his judgment that all work done by the other arts is put to test" . This suggests that architecture can be regarded as an interdisciplinary field that brings together work done in many different areas which also seems to apply to the conceptualization of architecture within the IS field. To put it another way, Vitruvius' thoughts on architecture pose a truth that can be recognized within the realm of information systems research and practice. This is not accidental. In fact, a plausible reason for why this is the case is that (one of) the underlying metaphor(s) used to grasp how we understand information technology, is indeed the one of architecture. As argued, the metaphor can be seen through the language that we use, when we talk about information systems, e.g. when we *build* a system, talk about the *structure* of a system or when we are making a *plan* for the design. However, the awareness of the underlying metaphor, and its consequences, is at best vague (Smolander 2002). Danesi (2001) argues that concepts, e.g. "architecture", are mental forms. He distinguishes between two types of concepts; concrete and abstract, where the difference between the two is whether or not the concept holds an external reference. That is if the external referent is sensible. Based on Danesi's use of layering theory, it could be argued that the concept of architecture in relation to IS research and practice is meta-symbolic, that is: A "complex metaphorical idea" , that shapes the categories, discourse and language used. What is important to recognize about the use of metaphors, is, that they are created with a purpose in mind and that they "… exerts a formative influence on language, on science, on how we think, how we see, and how we express ourselves on a day-to-day basis" (Morgan 1998).The architectural metaphor therefore can be seen as a temporal displacement in interpretations and practice, which has resulted in a multifaceted understanding of the concept of architecture.

Though the architectural metaphor provides a certain perspective towards the phenomenon of interest, the metaphor itself has also been subject to further deconstruction (Smolander 2002). The architectural metaphor is understood through various views provided by another metaphorical layer (Danesi 2001). These metaphors provide insights to some aspects of the architectural metaphor, but also distort the image created (Morgan 1998). Smolander (2002) analyzes which connotations certain stakeholders - designers, architects and managers - have toward the architectural concept. From the analysis four different metaphors for architecture emerges: "Architecture as literature", "architecture as language", "architecture as decision" and "architecture as blueprint". In the "architecture as blueprint" metaphor architecture is seen as "the structure of the system" . That is, architecture is seen a plan of some future IT artefact. In the "architecture as literature" metaphor architecture is seen as the documentation of an existing IT artefact. It is a way to explicate the knowledge of the implementation of an IT artefact. The "architecture as language" sees architecture as a vehicle for communication. Architecture here is a way for different stakeholders to create a common understanding of some IT artefact. Finally, in the "architecture as decision" metaphor architecture is seen as the basis of a rational decision making process. The blueprint and literature metaphor can be seen as supporting an engineering approach, and were widely recognized by designers and architects whereas the language and decision metaphors were acknowledged by managers. The language and decision metaphors primarily adopt a business oriented paradigm. Smolander (2002) concludes that "Architecture is a plastic concept whose meaning changes according to the stakeholder, situation, and the phase of the project. Even though architecture is plastic, it is robust enough to maintain its identity among stakeholders" .

3 TWO STREAMS OF ARCHITECTURAL CONCEPTS IN INFORMATION SYSTEMS RESEARCH

In the two next subsections a description of two streams of architecture in IS research is given; the software architecture stream and the business architecture stream. The idea is not to give an exact account of the characteristics of these two, but instead to give a basic impression of the two streams. This is followed by a comparison of these two different ways of conceptualizing architecture.

3.1 The software architecture stream

Software architecture surfaced from the software crisis in the 60's, where it became clear that the increasing complexity of information system design had to be taken into consideration, when designing new applications. As a close companion to - at the time - rising field of software engineering, the concept of architecture proposed a new way of perceiving information system development. A move towards an abstract and idealized practice has become synonymous with the concept of architecture (Coplien 1999), where focus is on the interrelationship of different subcomponents of the application. Garlan (2000) notes that architecture "… exposes certain properties, while hiding others." Though not addressing the underlying reason for why this is the case, it seems that the concept of software architecture can be regarded as a mindset for interpreting structural properties of software. This line of thought is also apparent in (Shaw and Garlan 1995; Mathiassen, Munk-Madsen et al. 2000; Plfeeger 2001), where emphasis is put on the overall structure of the software design. In (Mathiassen, Munk-Madsen et al. 2000) it also becomes apparent that the concept of software architecture is a tool for "[…] understanding the [software] system in itself" . The concept of software architecture can be seen as a 'reverse black box'; it does not address how or why requirements for the application are captured, but is purely seen as an instrument for understanding the software independent of the context it is designed for and situated in. From the four metaphors presented by Smolander (2002), it could be argued that the concept of software architecture primarily supports "architecture as literature" and "architecture as blueprint". Thus software architecture is largely seen as a tool for describing and documenting the high level implementation of the application.

3.2 The business architecture stream

In recent decades the concept of business architecture, in particular denoted as enterprise architecture (EA) (Spewak 1992; Bernard 2005; Hjort-Madsen and Burkard 2006; Ross, Weill et al. 2006) and to some extent as information systems architecture (Zachmann 1987; Sowa and Zachman 1992), has transpired. In contrast to the software architecture stream, business architecture proposes an approach that takes the context of the application into consideration, and is less occupied with the structuration of the application itself. Business architecture could be described as an external view and software architecture as internal view seen from an application point of view, where focus is on how the application generates value for the business and not as much how the internal workings of an application is functioning.

Zachmann (1987) proposes a framework (ISA) (see Figure 1) for creating multiple and simultaneous perspectives and understandings of information systems architecture from a set of descriptions, instead of just a singular one. The framework draws heavily on the original metaphor of architecture, and in (Sowa and Zachman 1992), the framework is extended further with more perspectives and also given a higher degree of formalization. In contrast to the concept of software architecture Zachmann's ISA framework, acknowledges that the software itself is part of a larger context that needs to be addressed when designing it. Though Zachmann describes the relationships between the various perspectives, he does not describe precisely how the framework is meant to be constructed, or by whom. If one were to use the ISA Framework proposed by Zachmann, it would require a variety of skills concerning business, organization, technical and human aspects. This is the core of architectural work as proposed by Vitruvius. The emphasis is put on interdisciplinary work in such a way, that architecture emerges from the interplay between these different perspectives. Though Zachmann addresses many of the points of this paper, he leaves out a discussion of the role of the architect and furthermore a description of how the architectural work is carried out. Spewak (1992) addresses these issues by introducing the concept of enterprise architecture planning (EAP) which – according to Spewak – "[…] results in a high-level blueprint of data, applications, and technology that is cost-effective, long-term solution; not a quick fix" (Spewak 1992). EAP can be seen as a process for defining the top level layers ("Scope" and "Enterprise Model") of the Zachmann ISA framework, but is less occupied with the lower levels e.g. "System Model" or "Technology Model". This introduces a gap in the architectural work since the lower levels seems of less importance. This line of thought is also apparent in (Bernard 2005). Bernard tries to integrate Spewak and Zachmann's work into a method containing both a framework and processes supporting it. However Bernard is – as Spewak – mainly focused on the top level issues e.g. business goals, strategy, governance etc. Compared to the four metaphors presented by Smolander (2002) the concept of business architecture seems to support mainly "architecture as language" and "architecture as decision". Especially "architecture as decision" seems to be of significance (Weill and Ross 2004; Ross, Weill et al. 2006). This contrasts software architecture, where the main purpose can be seen as a form of documentation, or description, of the main characteristics of a particular IT artefact.

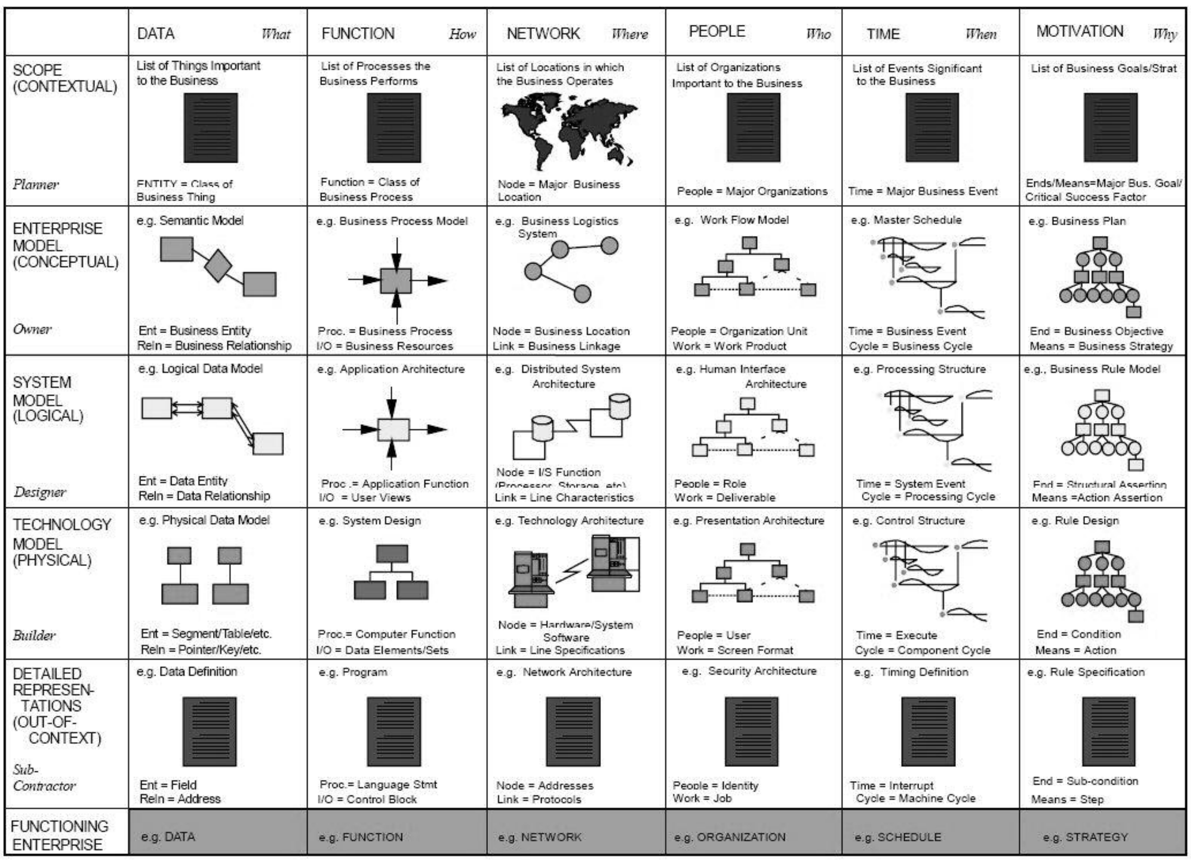

Figure 1: The Zachman ISA Framework from (Sowa and Zachman 1992)

3.3 Comparison of architectural concepts

The two concepts of architecture – software architecture and business architecture – are used to describe very different aspects of an IT artefact. It seems that though both streams do not consider each other, they are not mutually exclusive; their approach towards information system design is in both cases a high level one, where certain details of the design are abstracted away.

	Software architecture	**Business architecture**
Focus	(Inter)relations among software components	(Inter)relations among business components
Main Purpose	Documentation	Strategic planning
Scope	Software products	Enterprise
Approach	Top-Down	Top-Down
Used by	Engineers, Software developers, Software architects	Managers, Enterprise architects
Leading metaphors	Architecture as literature, Architecture as blueprint	Architecture as decision, Architecture as language

Table 1: Two architectural streams: Software architecture and Business architecture

So what do the two streams have to offer each other? As information technology is becoming an increasingly important parameter for enhancing business competitiveness (Weill and Ross 2004), a deeper understanding of how software is constructed could lead to new opportunities seen from a business perspective. At the same time, a thorough understanding of business goals, needs and organization could potentially lead to better design solutions, where the value of a particular piece of software would become more obvious. The challenge is to align these different usages of the concept of architecture in order to take advantage of both concepts strengths and to reduce their individual weaknesses.

It is important to emphasize that these different conceptions are in fact different semantics. They cannot be merged into each other, since the meaning of the particular concept is context depended (Smolander 2002). Instead it could be regarded as a boundary object with numerous simultaneous meanings and interpretations.

4 ARCHITECTURING AS SENSEMAKING

Though the two streams of architecture are supported by different metaphors, they are similar in many ways: Each stream proposes a top down approach to design and is focused on the (inter)relationships between various ‘parts’. Whether we think of an IT as a specific application or think of it in larger terms as the sum of the interaction between various actors, what is at stake is the relations between its parts. What we are trying to grasp by thinking in terms of any system’s architecture is the structure of the system. We want to be able to answer the question: How well do the pieces fit? (Coplien 1999) The answer to this question lies in the eye of the beholder. By focusing on how we interpret these relationships and evaluate these, we emphasize the act of architecturing, that is; trying to make sense of some system’s organization (Weick 1995). Because the architectural metaphor is used in a number of ways within many different areas (Onians 1992) a clarification of the domain of which it is utilized is needed. Therefore I will denote it as IT architecturing in order to clarify that the interest at stake is information technology. IT architecturing seen as sensemaking can be understood as one or more meaningful interpretations of the continuous organizing, emerging from the interrelationships between a socio-technical system’s parts.

This way of understanding architecture offers an interesting perspective. Where traditional views on architecture provide us with a focus on the relationship between various parts of a larger system, the concept of IT architecturing emphasize the human interpretation and understanding of these relationships. Thus architecturing can be seen as a sense making process (Weick 1995; Vaccari 1998; Danesi 2001; Toit 2003), where the phenomenon of interest is the emergent properties of some system (Heylighen 1989). The concept of IT architecturing acknowledges that architectural work holds various intentions, and that these can coexists. Where earlier notions discussed in this paper do not take other conceptions of architecture into consideration, the concept of IT-architecture provides a frame where different semantics can be supported simultaneously. At the same time the concept of IT architecturing emphasize that technology is intentional and subjective in its nature, and not an objective instrument, where the tangible use determines its position (Bowker and Star 2000; Kallinikos 2005).

The three concepts: Business architecture, software architecture and IT-architecturing, provides us with different semantics used to support various aspects of IT and the context it is implemented in. However, where business and software architecture first and foremost is addressing some external phenomenon, IT-architecture is used to create an understanding of how these different uses can be aligned in such a way, that they can offer a better support to each other, driven both by business needs and technological opportunities and restraints. The role of the IT-architect is therefore to try to make sense of the various concepts and unite them into a coherent understanding of the interplay between business and technology. By returning to Vitruvius and his thoughts on the role the architect, it could be argued that what is necessary in order to align the two streams of architecture, is a new role that transcends software architecture and business architecture, and thereby returning to the original intentions of the role of the architect.

As argued, we can consider the two streams of architecture as different semantics which is another way of saying that they address different social realities. Luhmann (1996) describes how different social systems are supported by different semantics in their communication. Briefly described, a semantic is a theme stock reserved for communication i.e. it is the constraints that are necessary in order to determine suitable and unsuitable contributions in theme related communication (Luhmann 1996). Or put more directly, a semantic is a contract of how to understand some communication in relation to the context in which it is situated. Each semantic has a medium, code, program, output and - possibly - an institutionalization. By medium Luhmann (1996) refers to the function that enables communication in the first place. Considering the two streams of architecture, it could be argued that the medium for software architecture is the software itself and for business architecture it is the corporate strategy that acts as a medium. Code refers to a perspective on the semantic at hand. By applying a specific code certain things become visible while others remain hidden. In business architecture the code is competitiveness/uncompetitiveness because architecture in this stream is viewed as a vehicle that can enhance the chances of survival of the business. Program informs *what* constitutes e.g. Competitiveness or through which means it can be obtained. The program associates the semantic with the code in order to give substance to both. The program for the business architecture stream consists of frameworks and governance principles for architecture, whereas the software architecture stream's program consists of applications, models and construction principles. The function can be described as the relationship between the semantic and its environment. That is, the contribution that the semantic produces for the rest of 'society'. Business architecture functions as a mean to increase or decrease the complexity of an organization. This can be seen in the way that e.g. Zachmann's framework (1992) is utilized: The framework is used to describe to states of an organization; one that describes the current state (as-is) and one that describes the desirable future state of an organization (to-be). The as-is is understood as a mechanism that reduces the complexity of an organization through categorization. The same is also the case in the to-be mode; however, the to-be can also be understood as mean to increase the complexity of an organization.

Architectural semantics	Medium	Code	Program	Function	Output
Business Architecture	Corporate Strategy	Competit iveness +/-	EA Framework, Principles	Categorization	Decision support
Software Architecture	Software	Construc tion +/-	Applications, Models, Principles	High level representation	Structurati on
IT Architecturing	Human	Meaning +/-	Understanding	Optimization	Alignment

Table 2: Architectural semantics

What is argued in this paper is that it requires human understanding to interpret and mediate between various architectural semantics in order to align these in a meaningful way. As depicted in Figure 2: Architectural semantics, the code supporting IT-architecture is focused on the creation of meaning of the interplay between business architecture, software architecture and possibly other concepts of architecture as well.

5 IMPLICATIONS FOR STUDYING ARCHITECTURE

As this paper is to be considered as research in progress some thoughts on how to approach the field will be presented in this section. One point that needs to be stressed out is that though the concept of IT-architecture is presented as a novel notion it has very different properties than the two above mentioned streams. First of all, IT-architecturing emphasizes the human factor of architectural work which suggests a praxiographic investigation of the work practices of the architect. In this way IT-architecturing differs from the two other identified streams of architecture since it has a more descriptive character whereas the two others tend to be quite normative in the way they define architecture (see e.g. Kazman and Eden (2003) or Smolander (2003) for a discussion of various definitions). So, by introducing the concept of IT-architecturing is also to leave the battlefield of defining architecture and instead look at *how* architects actually carry out their business in an everyday manner; how they make sense and utilize the dots, lines, boxes and other (symbolic) tools that inhabit their social world in their everyday activities.

Mol (2002) argues that doing a praxiographic study is to "[…] investigate knowledge incorporated in daily events and activities rather than knowledge articulated in words and images and printed on paper." (p. 32). So, in order to study architecture - as a praxiographic endeavour – a qualitative case study is proposed as a research design. Participant observation will be used as the primary technique for constructing data, however, interviews along with document analysis will also be deployed in order to achieve a rich understanding of the context, life world and everyday doings of people doing architecture.

6 CONCLUSION

In this paper two different concepts of architecture have been explored. It has been argued that though they are meant for different purposes, supported in different ways by various stakeholders, the concepts of software architecture and business architecture has something to offer each other in order to drive business competitiveness further and thereby create a sustainable base for both business and technology. In order to encompass the two concepts, the concept of IT-architecture has been proposed and it is argued that it can serve as a mean for mediating between the two other concepts, thus shifting focus from the object at hand and into the human interpretation of the object seen from multiple perspectives. It is emphasized that IT-architecture is a product derived from a sense making process, where the purpose is to align the various uses of the concept of architecture. This requires an interdisciplinary approach where organizational, technical and human considerations are regarded in such a way that they can bring forward a symbiotic understating of how the interrelationship between these considerations can bring forth a sustainable business that can enhance its competitiveness in piecemeal steps.

References

Abowd, G. D., R. Allen, et al. (1995). "Formalizing Style to Understand Descriptions of Software Architecture." *ACM Transactions on Software Engineering and Methodology* **4**(4): 319-364.

Bernard, S. A. (2005). *An Introduction to Enterprise Architecture*, AuthorHouse.

Bowker, G. C. and S. L. Star (2000). *Sorting things out: classification and its consequences*. Cambridge, MIT Press.

Coplien, J. O. (1999). Reevaluating the architectural metaphor: Toward piecemeal growth. *IEEE Software*. **16:** 40-44.

Dahlbom, B. and L. Mathiassen (1995). *Computers in Context*, Blackwell Publishing.

Danesi, M. (2001). "Layering Theory and Human Abstract Thinking." *Cybernetics & Human Knowing* **8**(3): 5-24.

Garlan, D. (2000). *Software Architecture: a Roadmap*. Conference on The Future of Software Engineering Limerick, Ireland, ACM Press.

Heylighen, F. (1989). *Self-organization, Emergence and the Architecture of Complexity*. 1st European Conference on System Science, Paris.

Hjort-Madsen, K. and J. Burkard (2006). "When Enterprise Architecture Meets Government: An Institutional Case Study Analysis." *The Journal of Enterprise Architecture*(February): 11-25.

Kallinikos, J. (2005). "The Order of Technology: Complexity and control in a connected world." *Information and Organization* **15**(3): 185-202.

Kazman, R. and A. Eden (2003). "Defining the Terms Architecture, Design, and Implementation." *SEI Interactive* **1**(1).

Lakoff, G. and M. Johnson (1980). *Metaphors we live by*. Chicago, Illinois, University of Chocago Press.

Luhmann, N. (1996). *Social Systems*, Stanford University Press.

Mathiassen, L., A. Munk-Madsen, et al. (2000). *Obejct-Oriented Analysis & Design*. Aalborg, Marko Publishing ApS.

Mol, A. (2002). *The Body Multiple: Ontology in Medical practice*. Durham and London, Duke University Press.

Morgan, G. (1998). *Images of Organization*. Toronto, Berrett-Koehler Publishers

SAGE Publications.

Morgan, M. H. (1914). *Ten Books of Architecture*, Harvard University Press.

Onians, J. (1992). "Architecture, Metaphor and the Mind." *Architectural History* **35**: 192-207.

Plfeeger, S. L. (2001). *Software Engineering: Theory and Practice*. Upper Saddle River, New Jersey, Prentice-Hall.

Ross, J. W., P. Weill, et al. (2006). *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution*. Boston, Massachusetts, Harvard Business School Press.

Shaw, M. and D. Garlan (1995). "Formulations and Formalisms in Software Architecture." *Computer Science Today: Recent Trends and Developments* **1000**: 307-323.

Smolander, K. (2002). *Four Metaphors of Architecture in Software Organizations: Finding out The Meaning of Architecture in Practice*. 2002 International Symposium on Empirical Software Engineering.

Smolander, K. (2003). On the Role of Architecture in Systems Development. *Department of Information Technology*. Lappeenranta, Lappeenranta University of Technology. **PhD**.

Sowa, J. and J. Zachman (1992). "Extending and formalizing the framework for information systems architecture." *IBM Systems Journal* **31**(3): 590-616.

Spewak, S. H. (1992). *Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology*, John Wiley & Sons Inc.

Star, S. L. and J. R. Griesemer (1989). "Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology." *Social Studies of Science* **19**(3): 387-420.

Toit, A. d. (2003). "Knowledge: a sense making process shared through narrative." *Journal of Knowledge Management* **7**(3): 27-37.

Vaccari, E. (1998). "Knowledge as Modelling." *Cybernetics & Human Knowing* **5**(2): 59-72.

Weick, K. E. (1995). *Sensemaking in Organizations*, SAGE Publications.

Weill, P. and J. W. Ross (2004). *IT Governance: How Top Performers Manage IT Decision for Superior Results*. Boston, Massachusetts, Harvard Business School Press.

Zachmann, J. (1987). "A framework for information systems architecture." *IBM Systems Journal* **26**(3): 276-292.