Architecture-driven Information System Development – Toward a framework for understanding

H.A. (Erik) PROPER
IRIS, University of Nijmegen
Nijmegen, The Netherlands, EU
E.Proper@acm.org

PUBLISHED AS:

H.A. Proper. Architecture-driven Information System Development – Toward a framework for understanding. In N. Callaos, D. Farsi, M. Eshagian-Wilner, T. Hanratty, and N. Rish, editors, *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics*, volume XII, pages 144–149, Orlando, Florida, USA, July 2003. ISBN 9806560019

ABSTRACT

This article discusses a conceptual framework for architecture-driven information system development. Rather than defining a completely new framework, the conceptual framework is synthesized out of relevant pre-existing frameworks for system development and architecture.

Before discussing the actual framework, we briefly discuss the necessity for an architecture-driven approach to system development.

Keywords: Information System Development, Architecture, Information Architecture, Stakeholders, Requirements

1 INTRODUCTION

In recent years, several approaches to the development of (large-scale) information systems have emerged that depend highly on the use of "*architectures*" [Kee91, TC93, Boa99b, Boa99a, Zac87, SZ92]. The rationale behind the use of architecture is that it provides a number of important benefits [BCK98], such as:

- It is a vehicle for communication among stakeholders.
- It captures early design decisions, both functional aspects as well as quality aspects.
  These early design decisions are important since their ramifications are felt in all subsequent phases.

Some of the mentioned publications use the term "information architecture" while others refer to the same concept as "enterprise (IT) architecture". The use of architecture in the development of information systems, is what we refer to as architecture-driven information systems development.

For information systems, their development, as well as the concept of architecture itself, several frameworks which define the underlying fundemental concepts are already in existance:

- The *IEEE recommended practice for software intensive systems* [IEE00] for architecture.
- The *framework of information system concepts* for systems and information systems [ISO87, FHL+98].
- The *information services procurement library* [FV99, Pro01] for development processes.

In our view, however, none of these frameworks covers the field of architecture-driven information systems development in its entirity. Each of these frameworks focusses on a specific area of architecture-driven information systems development. Nevertheless, since partial conceptual frameworks do indeed exist, the aim of this article is not to develop "yet another" conceptual framework, but rather to synthesize a more complete integrated framework for architecture-driven information system development out of these pre-existing frameworks. A detailed version of this framework is discussed in [Pro04].

We start out by briefly touching upon our view on information systems (section 2) and the motivations behind architecture-driven information system development (section 3). We then continue with the discussion of a conceptual framework for information system development in general (section 4) and the use of viewtypes/viewpoints to relinquish the information needs of stakeholders (section 5). This framework is then extended further with the notion of architecture and its role in system development, leading to a framework for architecture-driven information system development (section 6). The concept of architecture is not discussed until section 6, since we first need to develop a conceptual framework covering (information) system development in general.

Note that for some of the concepts we have borrowed from the *framework of information system concepts* in [ISO87, FHL+98] we have provided a less precise definition in this article. More specifically, the distinction between a system domain and the conception of this domain in terms of a model (the concieved system) has been omitted. We have done so for reasons of compactness. In [Pro04], the full definitions have been used.

2 INFORMATION SYSTEMS

In this article we take the viewpoint that, in line with [FHL+98], "*information systems*" concerns the use of "*information*" by individuals or groups of people in organisations, in particular by means of computer-based systems. We use the term "*organisation*" in the most general sense. In other words, it does not only refer to, for example, large companies. One-man companies, profit- and non-profit-oriented organisations, clusters of companies interacting with each other, even the community of all Internet users and similar communities, may all be considered organisations.

The concept of information system can roughly be defined as that aspect of an organisation that provides, uses and distributes information. An information system *may* contain computerised sub-systems to automate certain activities. Some information systems may not even be computerised at all. A filing cabinet used to store and retrieve several dossiers is, in essence, a manual information system. What we may perceive to be information systems, may vary highly in terms of their scope. Some examples would be:

- Personal information appliances, such as electronic agenda's, telephone registries in mobile phones, etc.
- Specific information processing applications.
- Enterprise wide information processing.
- Value-chain wide information processing.

Although the focus of this article will mainly be on the development of large-scale (e.g. enterprise-wide) information systems, the resuls are applicable to the development of all sizes of information systems.

3 THE NEED FOR ARCHITECTURE

The prevailing conditions under which most organisations currently operate have a tendency to evolve constantly. Reduced protectionism, the introduction of common currencies, deregulation of international trade, privatisation of state owned companies, increased global competition, cross-border merges, the emergence of new trade blocks, all contribute toward an increasingly dynamic business environment. Developments that are fuelled even more by the advances of eCommerce, Networked Business, Virtual Enterprises, etc. It is suggested [Kee91, TC93] that to improve their chances for survival, organisations need the ability to quickly adapt themselves to such socio-economic developments.

Organisations make use of (largely computerised) information systems to fulfill in their information processing needs. When an organisation evolves, these information systems should be able to co-evolve in a natural way. Ideally, information technology should enable an organisation to go out and seek new challenges. However, one of the current dilemmas of information technology seems to be that in most cases it smothers an organisation's ability to change rather than supporting it. While it is quite reasonable to state that advanced computerised information systems should lead to revolutionary improvements in the flexibility and effectiveness of organisations, organisations still find themselves anchored to their pre-existing information systems. Quite often, these systems are the embodiment of the prevailing cultures and structures of the organisation's past. These systems tend to have an almost tangible monolithic nature that would be a feast to software archologists.

Organisations can deal with changes in their environment in a variety of ways. While some may try and continue their business *as usual*, others may choose to embrace the new developments and try to exploit their potential to their fullest. Neither approach is a guaranteed way to success or failure. Embracing new developments too early may lead to organisational chaos and decline, while waiting too long may result in missed business opportunities.

Already in [Kee91] and [TC93], an elaborate discussion can be found on the changes in context and culture that occur inside organisations as well as in their environments as a result of different socio-economic changes in combination with technological developments in information technology. Tapscott [TC93] was one of the first to propose an architecture-driven approach as a way to deal with the needed (continuous) changes to the organisational structure, the enterprise wide information systems and the underlying information technology.

The use of the concept of architecture in the field of information systems is not new. It can be traced back to the use of the concepts "*computer architecture*" and "*software architecture*", of which the last is most related to the field of information systems as most information systems are to a large extend computerised. During the last decade, the notion of "*software architecture*" has received an increasing amount of attention in the software engineering community; both from research and from industry (see for example [BCK98, SG96]).

Architectures are usually expressed in terms of architectural descriptions, essentially design descriptions pertaining to a (information) systems architecture. In [IEE00] the following potential uses of architectural descriptions, in the context of software engineering, are identified as well:

- Expression of the system and its (potential) evolution.
- Analysis of alternative architectures.
- Business planning for transition from a legacy architecture to a new architecture.
- Communications among organizations involved in the development, production, fielding, operation, and maintenance of a system.
- Communications between acquirers and developers as a part of contract negotiations
- Criteria for certifying conformance of implementations to the architecture.
- Development and maintenance documentation, including material for reuse repositories and training material.
- Input to subsequent system design and development activities.
- Input to system generation and analysis tools.
- Operational and infrastructure support; configuration management and repair; redesign and maintenance of systems, sub-systems, and components.
- Planning and budget support.
- Preparation of acquisition documents (e.g., requests for proposal and statements of work).
- Review, analysis, and evaluation of the system across the life cycle.
- Specification for a group of systems sharing a common set of features, (e.g., product lines).

These advantages are not only limited to software (as it may be found in computerised information systems), but equally well relate to most types of systems, in particular to information systems. In [Rec91, MR02] this relationship is emphasised as well.

4 INFORMATION SYSTEM DEVELOPMENT

The aim of this section is to arrive at a better understanding of concept of "information system development" itself. In this process, we first need to introduce some more basic terminology.

4.1 BASIC TERMINOLOGY

We presume information system development to originate from some need to make changes to a pre-existing situation. These needs for change are assumed to originate from people, organisations, etc., that have some stake with regards to the system under consideration. We therefore start the discussion of our framework with the introduction of the concept of *stakeholder*.

To any information system, a set of stakeholders may be associated, if alone those parties that use it to exchange information. A stakeholder is, in line with [IEE00], defined as:

Stakeholder – a party with a specific concern pertaining to a system, its development, its operation, or any other aspects that are critical or otherwise important.

Examples include: Users, operators, owners, architects, engineers, testers, project managers, business management, ...

The concerns of stakeholders are defined as:

Concern – those interests which pertain to the system's development, its operation or any other aspects that are critical or otherwise important to one or more stakeholders.

Note that at the start of a system development project, not all concerns of relevant stakeholders may be explicitly known. Some concerns may lie dormant until they are woken up as a result of some design decission.

The concern of a stakeholder, with respect to some system to be developed, originates from some deeper motivation; the stakeholder's own goals:

Goal – the end toward which effort is directed by a stakeholder, in which the information system (of which the stakeholder is indeed a stakeholder) plays a role.

This may pertain to strategic, tactical or operational end. The role of the system may range from passive to active. For example, a financial controller's goal with regards to a future/changed system may be to control development costs, while the goal of users of the system may be to get their job done more efficiently.

Note that having a clear understanding of the stakeholder goals, which underly the requirements, allows for better prioritisation of requirements relative to a stakeholder's true needs.

The concerns of a stakeholder correspond to a set of *requirements* with regards to the information system. The notion of requirements is, in line with [FV99], defined as:

Requirement – an essential quality property which a system, its description, or its development, has to satisfy.

Traditionally, requirements focussed on functional properties of the desired system ("functional requirements"). However, many, many, more types of ("non-functional requirements") exist. Requirements will usually be expressed in terms quality properties such as: functionality, flexibility, portability, security, maintainability, etc. See for instance: [ISO01].

4.2 DEVELOPMENT

The way we view information system development, is that it originates from a mismatch between the requirements of a system's stakeholders and the actual systemic properties the system exhibits (at that same point of time). This mismatch will prompt a desire to change the system:

System change – a change of the systemic properties of a system. Such changes may require structural changes of the system's internals.

This allows us to define information system development as follows:

Information system development – making *deliberate* system changes to a pre-existing information system with the aim of arriving at an information system which better meets the actual stakeholder's requirements than the original system did.

Note that this change of requirements may not be explicitly known to the stakeholders yet! The true requirements of the stakeholders may not reveal themselves until a proper 'problem analysis' has been performed.

It should also be noted that any exchange of information between actors can already be regarded as a manifestation of an information system. The introduction of such an information system can occur spontaneous if there are some stakeholders that are in need of the exchange of information. Any further changes to such an information system may be done deliberate. Consider, for instance, a situation in which two people working in some organisation have the need to exchange information. They may do so by exchanging this information on a face-to-face basis. However, as time progresses, the need may arrise (a change in requirements), to maintain a history of the information exchange. In this case, it may be decided to further develop the pre-existing information system into an information system involving a proper database system as a sub-system.

Information systems are likely to have multiple stakeholders. Theoretically, one could indeed have an information system with only one stakeholder. For example, the information system consisting of you, a pencil and a piece of paper (allowing you to make notes) only has one stakeholder: *you*. However, in practice information systems have many stakeholders. If we claim that information system development involves the process of "making deliberate system changes to a pre-existing information system with the aim of arriving at an information system which better meets the stakeholder's requirements than the original system did", then we should also realise that the set of "stakeholder requirements" is not that clear cut. Different stakeholders may have conflicting stakes, which are bound to translate to conflicting set of requirements. Consider, for instance, the requirements that may be put forward by a financial controller versus the ones of future users of a system. These requirements are likely to be conflicting. A crucial part of system development is therefore the negotiation of a balanced set of requirements that reflect the requirements of the different stakeholders and the relative weight (importance, political power, etc.) that should be associated to the stakeholder specific requirements (for instance, based on the stakeholder's goals).

4.3 DEVELOPMENT-PROCESS ASPECTS

Information system development processes involve several subprocesses. Some of these sub-processes can be seen to be common to all information system development processes. Some typical processes may involve:

1. Determine what requirements (*what*) and stakeholder goals (*why*) should be met by the finished system.
2. Design the desired system (from the highest levels of abstraction to the fullest details required).
3. Construct the system, i.e. assemble its components.
4. Install the system in its intended operational environment

By numbering these processes, we may have given the reader the feeling that these processes should be executed in a specific order. In practice, however, this is *not* the case. Different strategies do exist with regards to the order in which these processes may be executed [FV99, Som89], such as:

- Linear, i.e. step by step, finishing one step before continueing with the next step.
- Incrementally, i.e. based on some subdivision of the system into sub-systems developing sub-system by sub-system.
- Iterative, i.e. frequently iterating between the four subprocesses during system development.

To stress the fact that the above sub-processes may be executed in any order, we actually prefer to use the term "development-process aspect" rather than sub-process.

In our framework we consider the development of information systems as to involve four key aspects:

Definition aspect – those aspects of system development, which aim to identify all requirements (and goals) that should be met by the system (and its description).

In literature this process may also be referred to as *requirements engineering* [JBR$^+$93].

Design aspect – those aspects of system development, which aim to produce the design of a system which conforms to the stated requirements.

The resulting system design may range from high-level designs, such as a strategy or an architecture, to the detailed level of programming statements or specific worker tasks.

Construction aspect – those aspects of system development, which aim to realise and test a system that is regarded as a (possibly artificial) artifact that is not yet in operation.

Installation aspect – those aspects of system development, which aim to make a constructed system operational, i.e. to implement the use of the system by its prospective users.

The definitions of the construction and installation aspects are conform those used in [FV99]. In [FV99], the definition and design aspects of system development are collectively referred to as the description aspects. We feel, however, a clear distinction should be made between the definition of the future system in terms of its requirements and the actual system design. Nevertheless, it is important to stress the fact that the definition and design aspects aim to provide different *descriptions* of the future system:

Definition description – a description of the definition of the future system. In other words, a description of the requirements, and their motivation in terms of stakeholder goals, that should be met by the future system.

Design description – A description of the design of the future system, including a motivation of design decissions (partly) in terms of the requirements and goals put foward in the definition of the system.

Collectively we will refer to design descriptions and definition descriptions as **system descriptions**.

5 INFORMATION NEEDS OF STAKEHOLDERS

The focus of this article is on architecture-driven system development. As mentioned in the introduction, an important role of architecture is that of a "means of communication and negotiation among the different stakeholders". This specifically comes to the fore during the definition and design of an information system. The descriptions of the definition and design should therefore effectively convey information to the different stakeholders, in other words, they should meet the information needs of the different stakeholders. Note that "information need" should not be interpreted in a uni-directional way, focussing purely on communicating information *to* the stakeholders. Stakeholders do not just have a need to be informed, they also have a need to (or are required to) contribute information to the development process; if only to steer the process in a desirable direction.

While it is desired to develop an information system based on a unified, complete and consistent definition and design, different stakeholders have different concerns, and as a result have different information needs with regards to the definition and design of the system. This requires the introduction of different (stakeholder/concern specific) views on a system's definition and design. A view on a system description is, in line with [FHL$^+$98] and [IEE00], defined as:

View – a description of a system from the perspective of a single concern, or a related set of concerns, of a stakeholder.

Some special views are:

Unified view – the system view which covers all possible concerns with regards to the system.

Definition view – a sub-view of the unified view, which focuses on the system requirements.

Design view – a sub-view of the unified view, which focuses on the system design.

Views may be classified into types of views, where a viewtype may (based on [CBB$^+$02]) be defined as:

Viewtype – a classification of the information provided by views conforming to this viewtype in conjunction with a well articulated way of thinking. The classification will be stated in terms of the (system) concepts used to express the views. A way of thinking may also be referred to as *die Weltanschauung* [Sol83, WAA85], *underlying perspective* or *philosophy* [Avi95].

The IEEE recommended practice practice for architectural description [IEE00], defines the notion of viewpoint, as:

> A form of abstraction achieved using a selected set of (architectural) constructs and structuring rules, in order to focus on particular concerns within a system.

a definition which is in line with the proposed notion of viewtype. It, however, adds the notion of structuring rules. In [IEE00] it is also explicitly noted that rules, heuristics and other guidelines that may assist in the construction of a view may be part of a viewpoint as well. In our framework, we prefer to define a viewpoint as follows:

Viewpoint – a viewtype in combination with a *way of working* defining the way in which views may be constructed. This way of working may be stated in terms of rules, patterns, heuristics, guidelines, etc.

In the development of an information system different viewtypes will be used to provide insight into different aspects of the new system being developed. In literature, several frameworks of integrated viewtypes can be found. Some of these frameworks even claim to provide a complete coverage of all relevant concerns with regards to the system. The level of "completeness" of these frameworks is highly dependend on the philosophical stance with regards to an information system and/or its automated parts, which such a frameworks takes. An integrated framework of related viewtypes is what we would like to refer to as a viewmodel:

Viewmodel – an integrated framework of viewtypes (or more elementary viewmodels), in conjunction with an overall way of thinking.

Note that a viewmodel is essentially regarded as being a composed viewtype.

Some of the more well-known viewmodels are:

The Zachman framework [Zac87, SZ92]: named after its creator, is based on the *what*, *how*, *where*, *when*, *who*, *why* interrogatives, leading in their interpretation to the following key aspects of nformation systems: Data, Function, Network, People, Time and Motivation. The framework combines these aspects with five classes of stakeholders: Planner, Owner, Designer Builder and Aspect-contractor This leads a total of 30 viewtypes.

Tapscott & Caston [TC93]: this framework was proposed as a means for organisations which are in the process of introducing architectures as a means to better allign business and IT. The framework distinguishes five viewtypes: business, work, information, application and technology.

RM-ODP (Reference Model of Open Distributed Processing) [ISO98]: provides a framework which identifies five viewtypes: Enterprise, Information, Computational, Engineering and Technology. The Model-Driven Architecture which is currently being developed by the OMG (Open-Management Group) is inspired by the RM-ODP.

Kruchten's 4+1 framework [Kru95]: this framework originates from the domain of object-oriented modelling, and UML in particular. The framework identifies four perspectives on a system: Logical, Process, Development and Physical. This four perspectives are tied together by means of an integrating perspective (the "+1"): scenario's.

Soni, Nord and Hofmeister [SNH95]: this framework resembles the Kruchten framework, and identifies the following viewtypes: Conceptual, Module, Execution and Code. The is no fifth integrating viewtype.

6 ARCHITECTURE-DRIVEN

In [BCK98], architecture is identified as a means to capture early design decisions touching upon both functional as well as non-functional quality aspects. These early design decisions are important since their ramifications are felt in all subsequent phases. In this sense, architecture forms a bridge between a system's definition and a system's design. In an architecture, the essense of both definition and design will meet. In this section we aim to arrive at a better understanding of architecture-driven information system development, by focussing on the definition of *architecture*.

The IEEE [IEE00] recommended practice for architectural description defines architecture as follows:

Architecture (structure-oriented) – the fundamental organisation of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its evolution and design.

This is, in our opinion, a highly structured-oriented definition, as it defines architecture in terms of the elements it should "structurally" contain.

In addition to the "structure-oriented" definition of architecture, we prefer to also include a "pragmatics-oriented" definition focussing on what the use of architecture [HP02]. At the end of section 3 we have already listed a number of potential uses for architectural descriptions, as identified in the IEEE standard. It is, in our opinion, in these uses where the true definition of architecture should be sought. Architecture in an (information) system context should really be regarded as a means of communication, negotiation and guidance for future system developments, in terms of essential properties of a future system. Properties that may indeed refer to "the fundamental organisation of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its evolution and design". This leads to the following "pragmatic" definition of architecture:

Architecture (pragmatics-oriented) – the "*fundamental organisation of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its evolution and design*", where the descriptions of this fundamental organisation are explicitly used during system development as a means:
- of communication & negotiation among stakeholders,
- to evaluate and compare design alternatives,
- to plan, manage, and execute further development,
- to verify the compliance of a system's implementation.

Having a structure-oriented and a pragmatics-oriented definition of architecture, does raise the question of a semantics-oriented definition. Providing such a definition is part of ongoing research, in which we aim to more precisely define the communication & negotiation processes surrounding architecture and its impacts on ensueing design decissions. It is in this relaying of results from the communication & negotiation processes to the ensueing design decissions where we aim to define the semantics of architecture.

Finally, title of this article refers to the concept of "architecture-driven system development". With the above pragmatics-oriented definition of architecture, we can finish our framework with the following definition of architecture-driven system development:

Architecture-driven information system development – the development of information systems using architecture (and their descriptions) using architecture as a means:
- of communication & negotiation among stakeholders,
- to evaluate and compare design alternatives,
- to plan, manage, and execute further development,
- to verify the compliance of a system's implementation.

7 CONCLUSION

In this article, we have discussed a conceptual framework for architecture-driven information system development. This framework was developed out of a number of pre-existing conceptual frameworks, in a attempt to avoid developing "yet another conceptual framework". We realise that in this short article we could only touch upon some of the key concepts. A more elaborate discussion may be found in [Pro04].

In this article, we did not discuss the concepts which would actually have to be used to discribe the architecture of information systems. We do acknowledge the fact that such a framework is relevant and needed. Such a framework is part of future research within the ArchiMate consortium (see Acknowledgements), where we again, will start out from pre-existing frameworks of concepts in order to synthesize a framework that best fits our needs for information system architectures.

ACKNOWLEDGEMENT

This paper results from the ArchiMate project, a research initiative that provides concepts and techniques to support architects in the visualisation, communication and analysis of integrated architectures. The ArchiMate consortium consists of ABN AMRO, ABP, the Dutch Tax and Customs Administration,

Ordina, Telematics Institute, Centre for Mathematics and Informatics, University of Nijmegen, and the Leiden Institute of Advanced Computer Science.

REFERENCES

[Avi95] D.E Avison. *Information Systems Development: Methodologies, Techniques and Tools*. McGraw-Hill, New York, New York, 2nd edition, 1995. ISBN 0077092333

[BCK98] L. Bass, P. Clements, and R. Kazman. *Software Architecture in Practice*. Addison Wesley, Reading, Massachusetts, USA, 1998. ISBN 0-201-19930-0

[Boa99a] B.H. Boar. *Constructing Blueprints for Enterprise IT architectures*. Wiley, New York, New York, USA, 1999. ISBN 0-471-29620-1

[Boa99b] B.H. Boar. *Practical steps for aligning information technology with business strategies*. Wiley, New York, New York, 1999. ISBN 0471076376

[CBB$^+$02] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley, Reading, Massachusetts, 2002. ISBN 0201703726

[FHL$^+$98] E.D. Falkenberg, W. Hesse, P. Lindgreen, B.E. Nilsson, J.L.H. Oei, C. Rolland, R.K. Stamper, F.J.M. Van Assche, A.A. Verrijn-Stuart, and K. Voss, editors. *A Framework of Information Systems Concepts*. IFIP WG 8.1 Task Group FRISCO, 1998. ISBN 3-901-88201-4

[FV99] M. Franckson and T.F. Verhoef, editors. *Introduction to ISPL*. Information Services Procurement Library. ten Hagen & Stam, Den Haag, The Netherlands, 1999. ISBN 9076304858

[HP02] W.-J. van den Heuvel and H.A. Proper. De pragmatiek van architectuur. *Informatie*, 44(11):12–14, 2002. In Dutch.

[IEE00] Recommended Practice for Architectural Description of Software Intensive Systems. Technical Report IEEE P1471-2000, IEEE Standards Department, The Architecture Working Group of the Software Engineering Committee, September 2000. ISBN 0-738-12518-0
`http://www.ieee.org`

[ISO87] *Information processing systems – Concepts and Terminology for the Conceptual Schema and the Information Base*, 1987. ISO/TR 9007:1987.
`http://www.iso.org`

[ISO98] *Information technology – Open Distributed Processing – Reference model: Overview*, 1998. ISO/IEC 10746-1:1998(E).
`http://www.iso.org`

[ISO01] *Software engineering – Product quality – Part 1: Quality model*, 2001. ISO/IEC 9126-1:2001.
`http://www.iso.org`

[JBR$^+$93] M. Jarke, J.A. Bubenko, C. Rolland, A. Sutcliffe, and Y. Vassiliou. Theories Underlying Requirements Engineering: An Overview of NATURE at Genesis. In *Proceedings of the IEEE Symposium on Requirements Engineering, RE'93*, San Diego, California, January 1993. IEEE Computer Society Press.

[Kee91] P.W.G. Keen. *Shaping the Future - Business Design Through Information Technology*. Harvard Business School Press, Boston, Massachusetts, USA, 1991. ISBN 0875842372

[Kru95] P. Kruchten. The 4+1 view model of architecture. *IEEE Software*, 12(6):42–50, November 1995.

[MR02] M.W. Maier and R. Rechtin. *The Art of System Architecting*. CRC Press, Boca Raton, Florida, 2nd edition, 2002. ISBN 0849304407

[Pro01] H.A. Proper, editor. *ISP for Large-scale Migrations*. Information Services Procurement Library. ten Hagen & Stam, Den Haag, The Netherlands, EU, 2001. ISBN 9076304882

[Pro04] H.A. Proper. *Da Vinci – Architecture-driven Information Systems Engineering*. Nijmegen Institute for Information and Computing Sciences, University of Nijmegen, Nijmegen, The Netherlands, EU, 2004.

[Rec91] E. Rechtin. *Systems architecting: creating and building complex systems*. Prentice-Hall PTR, Upper Saddle River, New Jersey, 1991. ISBN 0138803455

[SG96] M. Shaw and D. Garlan. *Software Architecture: Perspectives on an Emerging Discipline*. Prentice Hall, Englewood Cliffs, New Jersey, 1996. ISBN 0131829572

[SNH95] D. Soni, R.L. Nord, and C. Hofmeister. Software architecture in industrial applications. In *Proceedings of the 17th International Conference on Software Engineering*, pages 196–207, Seattle, Washington, USA, April 1995. ACM Press. ISBN 0-89791-708-1

[Sol83] H.G. Sol. A Feature Analysis of Information Systems Design Methodologies: Methodological Considerations. In T.W. Olle, H.G. Sol, and C.J. Tully, editors, *Information Systems Design Methodologies: A Feature Analysis*, pages 1–7. North-Holland/IFIP WG8.1, Amsterdam, The Netherlands, EU, 1983. ISBN 0-444-86705-8

[Som89] I. Sommerville. *Software Engineering*. Addison-Wesley, Reading, Massachusetts, USA, 1989.

[SZ92] J.F. Sowa and J.A. Zachman. Extending and formalizing the framework for information systems architecture. *IBM Systems Journal*, 31(3):590–616, 1992.

[TC93] D. Tapscott and A. Caston. *Paradigm Shift – The New Promise of Information Technology*. McGraw-Hill, New York, New York, USA, 1993. ASIN 0-070-62857-2

[WAA85] A.T. Wood-Harper, L. Antill, and D.E. Avison. *Information Systems Definition: The Multiview Approach*. Blackwell Scientific Publications, Oxford, United Kingdom, EU, 1985. ISBN 0-632-01216-8

[Zac87] J.A. Zachman. A framework for information systems architecture. *IBM Systems Journal*, 26(3), 1987.