

# Practical Rationale for Describing Software Architecture

## *Beyond Programming-in-The-Large*

Kari Smolander<sup>1</sup>, Tero Päivärinta<sup>2</sup>

<sup>1</sup> *Lappeenranta University of Technology, Telecom Business Research Center (TBRC), Department of Information Technology, Laboratory of Information Processing, P.O. Box 20, FIN-53851 Lappeenranta, Finland, kari.smolander@lut.fi*

<sup>2</sup> *Agder University College, Dept. of Information Systems, Serviceboks 422, 4604 Kristiansand, Norway, tero.paivarinta@hia.no*

**Abstract:** The practical rationale for architecture design and description of software-based systems were inquired in three software-producing organizations. In addition to the traditional programming-in-the-large view that dominates the mainstream research on software architecture, this grounded-theory based study suggests that other important purposes for software architecture, especially such related to decision-making and mutual understanding among the stakeholders of software development, exist as well in practice. Their emergence depends on several factors associated with the software development organization in question: such as the variety of customers and other stakeholders, the characteristics of software products, and the nature of the business.

**Key words:** software architecture, qualitative research, architecture description, organizational aspects of software architecture

## 1. INTRODUCTION

Despite of the fact that architecture has become a well-recognized issue of the software engineering research, a dearth of understanding of how and why architectures are designed and described in actual software-producing organizations continues. The mainstream literature on software architecture has largely concentrated on formal models and related constructive works (e.g. Kazman et al. 2001), especially manifested in the large body of archi-

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35607-5\\_15](https://doi.org/10.1007/978-0-387-35607-5_15)

ecture description languages (see e.g. Medvidovic and Taylor 2000). Accordingly, the literature seems to make an assumption that architecture serves mainly as a means for further design and implementation, “programming-in-the-large” (DeRemer and Kron 1976) from the perspective of the “system designer” (cf. Hofmeister et al. 1999; Kruchten 1995).

Other possible rationales for architecture, such as organizational knowledge creation or achievement of mutual understanding among diverse stakeholders of software development, are rarely discussed in the research. The state-of-the-art reports discuss little about these questions (cf. Shaw 2001), and the other than the technical purposes of architecture are often referred to with ad-hoc lists (IEEE 2000; Garlan 2000; Bass et al. 1998) without explicit empirical grounding or verification of their relative importance. Bosch (2000) presents a case study on the possibilities for organizing software product lines, but discusses only little about the varying rationales in architecture design and description. Robbins and Redmiles (1998) consider the idea of diverse knowledge and the theory of reflection-in-action (Schön 1983), but focus on the design work of a single architect leaving other interests untouched. Kruchten (1999) draws on consulting experience, illustrating the diversity of architects’ role, and recommends how architecture teams should be built and operated. The rare empirical studies with the organizational connotations of software architecture include Grinter’s (1999) study on architects in one company. It indicates the communicational and political skills required from architects, as they must communicate with (and influence to) other stakeholders in numerous situations.

Our goal was to examine the rationale for architecture design and description in practice, seeking also for explanations for the possible variation between software-producing organizations. By the (description) rationale, we mean an underlying reason for making and using architecture descriptions. For instance, a designer may emphasize implementation support as the rationale, whereas an executive may need architecture descriptions for strategic decision-making and a project manager for project planning. Our use of the concept of description rationale should not be mixed with the concept of design rationale (Potts and Bruns 1988; Conklin 1989), which is associated with recording the reasons for design decisions of particular systems.

We also wished to associate architecture descriptions with the related stakeholders and the architectural viewpoints (cf. IEEE 2000) they need. In this report, we shall concentrate on resolving the prevalent and emphasized rationales in particular organizations and finding out the causes for the variation observed in the emphases. Another report (Smolander and Päivärinta 2002) explains the role of various stakeholders in detail.

An architecture description can be a document, presentation, plan, sketch, or any related genre of communication used in the architecture design process. This represents an idea somewhat broader from that of the IEEE 1471 recommended practice, which emphasizes the documentation of architecture (IEEE 2000).

A stakeholder is a person or a group of people involved in creating or using architecture descriptions in a way or another: e.g. designer, project manager, architect, general management, customer. The repertoire of stakeholders varies between individual organizations.

The IEEE recommended practice (IEEE 2000) regards a viewpoint as a specification from which individual views are developed by establishing the purposes and audience for the views and the techniques for their creation and analysis. Instead of necessitating any existing specification to identify a viewpoint in our analysis as such, we did not require a view and its specifying viewpoint to be explicitly defined beforehand, as we hypothesized that the organizations likely had shortcomings in such definitions, and our prejudice on the viewpoint definition might have constrained rich data collection and analysis.

This paper is organized as follows: the research process, in which we interviewed architects, designers, and managers in three software-producing organizations, is described in Chapter 2. Chapter 3 contains the descriptions of emphasized rationales and the causal relationships found in particular organizations. The results, suggesting that architecture, in practice, represents wider organizational connotations than the traditional “programming-in-the-large” metaphor would include, are discussed in Chapter 4. After that, we end the paper with conclusions.

## **2. RESEARCH PROCESS**

The research followed the grounded theory method, a qualitative approach to creating new theory from data collected around the research theme in question (Strauss and Corbin 1990). Preliminary discussions with representatives of the target organizations had already implied that the concept of architecture would carry ambiguous connotations (Smolander et al. 2002). Hence, we analysed the connotations related to the concept of architecture by the stakeholders from the transcripts, which were produced from 19 theme-based and tape-recorded interviews in the target organizations, hereinafter referred to as Alfa, Beta, and Gamma (Table 1).

The basic idea of the grounded-theory-based data analysis resides in finding conceptual categories and abstractions related to the research goal from a rich set of interviewees’ mentions and other data, and in combining these

categories meaningfully to provide theoretical insight into the phenomenon in question. For the analysis, the interviews were tape-recorded and fully transcribed. A software tool designed for grounded-theory-based data analysis (ATLAS.ti™) was used for managing and analyzing the data (all in the digital form), including also the documents about software processes and the examples of architecture descriptions. We started the analysis simply by searching for mentions about rationales and different stakeholders related to the architecture description process – we also had decided beforehand to focus on the mentions about the rationale of those stakeholders for producing or using architecture descriptions. The conceptual categories of ‘rationale’ and ‘stakeholder’ can thus be regarded as the high-level “seed categories” (Miles and Huberman 1984) of the data analysis. This “open coding” phase (Strauss and Corbin 1990) proceeded iteratively in parallel with the “axial coding” phase, in which relationships between the identified categories were built. The analysis progressed (in parallel with previous phases) with “selective coding” in which the core of the research results was scrutinized to form the report here.

*Table 1.* Characteristics of the target organizations

	Alfa	Beta	Gamma
Personnel in sw development	200	200	400 + 600 (in two divisions)
Typical products	Integrated software for mobile terminals	Software-based telecom services	Tailored software, also for web and mobile users
Typical customers	Manufacturers of mobile hardware	In-house customers	Telecom operators, public administration
Other characteristics	Strategic commitment to specific technology	Extensive and anonymous user base	Role of integrator in large projects
Interviews conducted	1 architect, 1 designer, 4 managers	2 architects, 2 designers, 2 managers, 1 manager of an in-house customer	3 architects, 1 designer, 2 managers

### 3. ARCHITECTURE DESIGN AND DESCRIPTION IN THREE ORGANIZATIONS

Table 1 declares some basic characteristics of the target organizations. During the selective coding phase, we regarded the following conceptual categories and the varying causal relationships observed between them in the target organizations as particularly relevant in relation to the research goal:

- Product – the characteristics of software developed by the organization in question. The characteristics ranged from product-based to tailored deliveries and from hardware-integrated mobile applications to enterprise-wide information systems.
- Customer orientation – the degree of interaction between a development organization and its customers. The degree varied from a customer relationship based on formal requirement documents with low level of interaction, to a consulting-oriented attitude and intense interaction with several groups inside customer organization, including users, business responsables, and data administration organizations.
- Business orientation – the amount of reasoning conducted within a development group about the business area of the software system in question. This may be further divided into the orientation towards understanding customer's business and the orientation towards software organization's own business. We interpreted that some development groups with a strong engineering culture had no interests on reflecting business problems, whereas for some groups the consideration of business problems was considered a responsibility.
- Other business-related and organizational factors characteristic for each organization. These included for instance strategic choices of technologies and partners, and the position of the organization from the perspective of the whole enterprise.
- Diversity of stakeholders – the number of stakeholder roles with different backgrounds, organizations, and professions involved in the software development process and projects. The number of roles varied and especially the external roles (like those related to a customership) contained much variance.
- Rationale for architecture design and description (as declared above)

In the following, we shall describe how these categories appeared in the architecture design and description practices at the target organizations.

### 3.1 Emphasized rationales

#### 3.1.1 Alfa

*"[The designer uses architecture descriptions] for finding out what his own module would contain, how it would communicate [with other modules] ... or what other modules it would possibly use and what other modules would possibly use it ... in fact, that's it."*<sup>1</sup>

*John, software architect*

Figure 1 depicts the causal relationships (the arrows in the picture) between the core categories resulting in the rationale for architecture design and descriptions that were observed in Alfa. As a software producer for mobile terminals, Alfa has made a very strong and binding commitment to a certain technology. Therefore, the development group does not generally need to reflect on the technological risks and the selection of technology actively because the risks and selections are readily embedded in the business strategy. In addition, Alfa as a formal subcontractor and a tool developer typically has low customer orientation since the customers are seldom integrated to the software production process. In addition, the users are mostly anonymous to the development group. This results to low business orientation among the developers, and low diversity of stakeholders with similar backgrounds and education. The emerging rationale for architecture design and description becomes then programming-in-the-large, as little reflection upon business or with other stakeholders is conducted. Architecture description is tied directly to implementation and to the transfer of explicit knowledge from architects to designers (and to some extent to the other stakeholders).

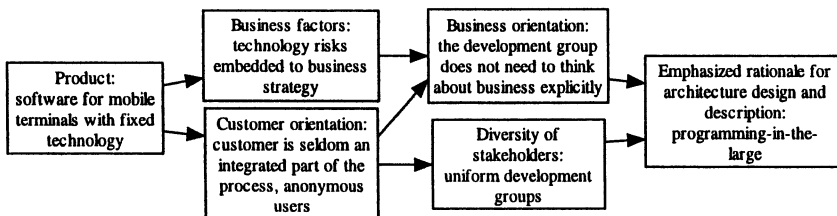


Figure 1. Emphasized rationales at Alfa

<sup>1</sup> All the quotations are translated from Finnish

**3.1.2 Beta**

*“I draw and make the descriptions and documents for the purpose that we could ... with our directory system [...] that we could outline it from the description, what is installed in what parts of the world and what kind of connections we have built from what parts of the world.”*

*Thomas, product manager*

*“Naturally, the descriptions are needed by the production organization that is responsible for coordinating the operation services. They define from whom we buy the operation services, from whom we buy the daily care and what other services we buy for it ...”*

*Timothy, architect*

Beta’s situation (Figure 2) diverges from that of Alfa’s. The developers are business-oriented as the use of the software (for telecom services) produces revenue to the corporation *per se*. To make right decisions, the software engineers must understand also the revenue logic behind the business processes in which the software solutions are being used. The customer orientation and diversity of stakeholders emphasize also slightly different role of architecture descriptions than in Alfa. Instead of further design and implementation as the main function, architecture emerges also as a tool for documenting decisions and making solutions and technologies explicit to the various stakeholders in different business units and technology centers inside the corporation. As the in-house software developer of the company, Beta must reflect on the corporate-wide strategies to software production. Therefore, the reuse of reference architectures and product lines arises as a topic of interest as well.

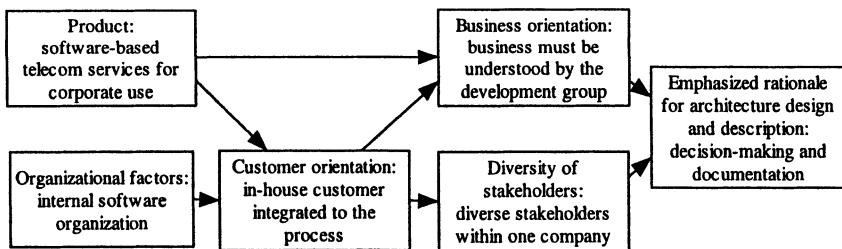


Figure 2. Emphasized rationales at Beta

### 3.1.3 Gamma

*“With customers, I have used architecture descriptions as the basis for telling about the possibilities we have if we do things this way. It is little like innovating or, let’s say, selling. Our aim is to show to the customer that we understand the problem domain [...] If there are technical people present, a logical architecture description is a good way to show our thoughts to the customer.”*

*Jack, software architect*

In its tailored software business, Gamma interacts intensively with a great variety of customers, including customer’s business responsables, data administration, technical personnel, and users in various positions. In addition, Gamma as an integrator must co-operate intensively with other partners, such as vendors of interfacing systems, and hardware and software vendors. Hence, architecture descriptions are most essentially used as a means for communication and negotiation about the contracts concerning software projects (Figure 3). The understanding of the customer’s business is essential for the developers. Architecture thus becomes a rich description of the reality that must be comprehended and shared among the heterogeneous collection of stakeholders, including the varying assemblies of customers and suppliers. The rationale and role of architecture description resides in communicating meanings and creating common objectives and mutual understanding among the stakeholders.

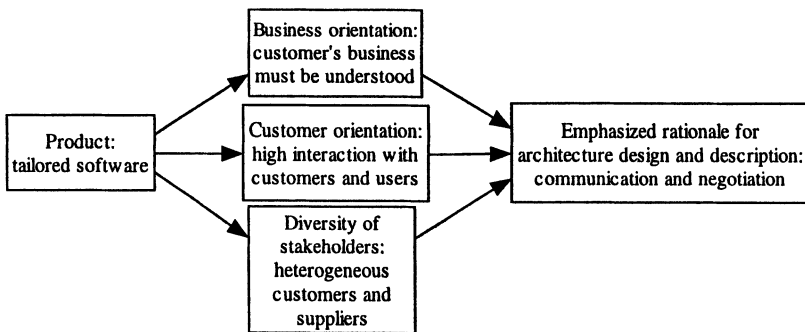


Figure 3. Emphasized rationales at Gamma



### **3.2 Used viewpoints in architecture description**

In an earlier study conducted in the target organizations (Smolander et al. 2002), architects' conceptions about the usage of architectural viewpoint were scrutinized. We found out that especially the business environment and business practices had effects on the selection of architectural viewpoints by the architects. Our original objective here was to extend that earlier study and to find out the general combinations and relationships between rationale, stakeholders, and used viewpoints in the architecture design and description process. As mentioned above, we let the interviewees to define their own points-of-view to architecture and analyzed the meanings given for the concept from the transcripts. The analysis revealed that the concept of architecture carried varying connotations. To some, it included only the structure of software components, whereas to others, it included system's stakeholders and their functionality, packages of business operations, network topologies, among many other features of the system.

With regard to the issue of viewpoints, the interviewees in the target organizations were generally rather unaware of such specific viewpoints recommended or defined in the literature (e.g. Kruchten 1995; Hofmeister et al. 1999; Zachman 1987). Some interviewees mentioned, for example, the 4+1 model (Kruchten 1995), but the concept of a viewpoint was not explicitly present in the target organizations' process documents or architecture descriptions, nor recalled in the interviews. Still, we were able to identify more than 20 diverging viewpoints to architectural level descriptions of software mentioned in the data, ranging, for instance, from viewpoints describing networks, hardware, functional scenarios, used technology, cost-benefit mappings, failure plans, memory allocation, security, laws and regulations involved, stakeholders, systems integration and legacy interfaces, among many other features of the systems requiring separate attention at the design phase. These viewpoints were used in no explicitly pre-planned way; they were selected as the situation at hand required. On the other hand, so-called "rich descriptions", informally constructed and communicated among the stakeholders, were used in all kinds of situations; obeying no explicit viewpoint definition (as defined in IEEE 2000). These "rich descriptions" were used in negotiations with customers as well as in internal meetings and communication among the stakeholders. Typically, they were materialized as PowerPoint™ presentations and extensive textual documents comprising often tens of pages of text and pictures; e.g., imported UML-based models created with design tools such as Rational Rose™, which was used as well for architecture descriptions in all target organizations. The comprehensive textual documents were often regarded as "too specific" for other stakeholders than designers, however.

Furthermore, several interviewees were also rather unaware of all the rationales behind the descriptions. Some created architecture descriptions simply because it was defined obligatory in the process model and some were aware only of a proportion of the uses of their descriptions. These reasons led us to abandon the original idea of trying to relate viewpoints explicitly to particular rationale and stakeholders (at least based on our contemporary amount of data).

#### 4. DISCUSSION: VARYING RATIONALE FOR ARCHITECTURE DESCRIPTIONS

Emerging rationale related to architecture design and description clearly varied among the three target organizations and their stakeholders. In addition to the programming-in-the-large conception that was highlighted in Alfa (but only secondarily present in Beta and Gamma), architecture design comprises decision-making, organizational communication, and documentation for maintenance and reuse. Figure 4 summarizes the findings related to the rationale for architecture design and description in the target organizations. The characteristics of the development organization, including the business and customer orientation and the diversity of stakeholders, evidently affect the rationale for architecture design and description. The rationale then steers the needs for tool and method support in architecture design and description. The development group must have tools and methods for transferring the knowledge about architecture between architects and designers, the descriptions must support and document decisions concerning architecture, the solutions must be made explicit enough to all stakeholders, and some level of common understanding about architecture must be achieved among a number of diverse stakeholders.

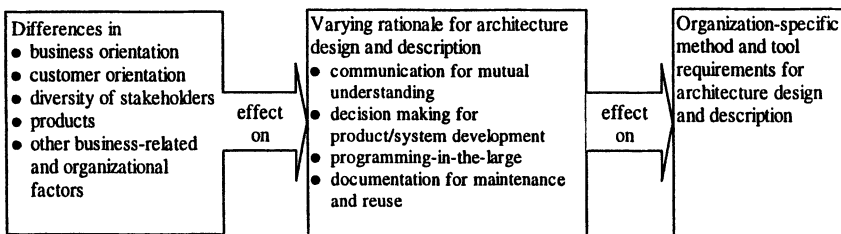


Figure 4. Varying rationale for architecture design and description

We recognized two sets of category combinations that suggested an explanation to the formation of rationale for architecture design and description

in the organizations. The higher business orientation, customer orientation, and the diversity of stakeholders, the more the rationale pointed towards communication and mutual understanding of architecture. Especially the role of a customer seemed to be of high importance emphasizing the communicational aspects of architecture. On the other hand, the lower the business orientation, customer orientation, and the diversity of stakeholders, the more emphasis was put on architecture as programming-in-the-large, i.e. architecture design was seen more solely as the pre-phase for programming and a way to transfer design knowledge from architects to software engineers.

However, most of the current software architecture research recognizes only the second combination. Our study suggests that further research on the software architecture design and description should consider a wider scale of issues than the programming-in-the-large rationale or the plain viewpoint of the software engineer. Current conceptions on architecture design and description, including the proliferation of the formal languages and models that currently dominates the research on software architecture, are somewhat based on such presuppositions about the rationale for architectural design that do not correspond with practice. The research should cover the evaluation of the role of formal vis-à-vis informal models and descriptions in architecture design – to what extent knowledge is created and transferred through formal models and to what extent through informal pictures, textual descriptions, and other social interaction. After that, we can determine effective means and tools to develop the architecture design and description process and methods. These means might slightly exceed the current focus on the formal issues of architecture design.

The simultaneous existence of several rationales for architecture design and description leads to a conclusion that architecture as a concept contains a seed of conflict. Whereas many designers and architects may remain satisfied with formal and detailed architecture descriptions (aimed at supporting implementation), the issues of creating common understanding among diverse stakeholders, enabling decision-making, and documenting the software for re-use, more rich and easy-to-interpret descriptions emerge among the other stakeholders. These diverging descriptions will most probably involve logical conflicts, and the “rich descriptions” may use simplifications and selling glorifications for creation of trust, interest, and understanding among the stakeholders. The consequence of having several rationales should be recognized both in research and practice. Software-producing organizations need different and even partially conflicting descriptions for different purposes and stakeholders. Eventually, the absolute quality of descriptions may appear less important than the quality of interpretations that people make from the descriptions. We thus argue that alongside the formal consistency of architecture descriptions highlighted for supporting

the technical engineering of software as such, the issue of how communicative these descriptions appear to be in the varying contexts of their use would deserve additional research efforts in the future.

## 5. CONCLUSION

In addition to the programming-in-the-large conception, a more holistic rationale of architecture description should be recognized in software-producing organizations. The rationale for architecture design and description should be carefully scrutinized considering a number of factors in the organizational context in question; such as the type of produced software, business orientation, customer orientation, and the diversity of stakeholders. Architecture descriptions can be deeply involved in requirements and customer management, organizational communication, and knowledge creation and management. In connection to the development of architectural practices in a particular software-producing organization, this fact should be respected and attention targeted on making the rationale for architecture descriptions explicit. The process of architecture design emphasizes different descriptions and viewpoints if the rationale resides in achieving understanding between diverse stakeholders compared to the aim of technical implementation within a uniform development group. The categories and causal relationships identified here contribute as a potential conceptual basis for improved tool and method support for architecture design.

## REFERENCES

- Bass, L., P. Clements and R. Kazman (1998). *Software Architecture in Practice*, Addison-Wesley.
- Bosch, J. (2000). *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*, Addison-Wesley.
- Conklin, J. (1989). "Design Rationale and Maintainability", *Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences*, 3-6 Jan 1989, Kailua-Kona, HI, USA, 533-539.
- DeRemer, F. and H. H. Kron (1976). "Programming-in-the-Large Versus Programming-in-the-Small". *IEEE Transactions on Software Engineering* SE-2(2): 80-86.
- Garlan, D. (2000). "Software Architecture: a Roadmap". *The Future of Software Engineering*. A. Finkelstein, ACM Press.
- Grinter, R. E. (1999). "Systems Architecture: Product Designing and Social Engineering". *ACM SIGSOFT Software Engineering Notes* 24(2): 11-18.

- Hofmeister, C., R. Nord and D. Soni (1999). *Applied Software Architecture*. Reading, MA, Addison-Wesley.
- IEEE (2000). *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, IEEE, IEEE Std 1471-2000.
- Kazman, R., P. Kruchten, C. Verhoef and H. van Vliet, Eds. (2001). *Proceedings of the Working IEEE/IFIP Conference on Software Architecture, WICSA2001, Amsterdam, The Netherlands, 28-31 August 2001*. Los Alamitos, CA, IEEE Computer Society.
- Kruchten, P. (1999). "The Software Architect - and the Software Architecture Team", *Proceedings of WICSA1: 1st Working IFIP Conference on Software Architecture*, San Antonio, TX, USA, 22-24 Feb 1999, Kluwer Academic Publishers, 565-583.
- Kruchten, P. B. (1995). "The 4+1 View Model of Architecture". *IEEE Software* 12(6): 42-50.
- Medvidovic, N. and R. N. Taylor (2000). "A Classification and Comparison Framework for Software Architecture Description Languages". *IEEE Transactions on Software Engineering* 26(1): 70-93.
- Miles, M. B. and A. M. Huberman (1984). *Qualitative Data Analysis: A Sourcebook of New Methods*. Beverly Hills, Sage.
- Potts, C. and G. Bruns (1988). "Recording The Reasons for Design Decisions", *Proceedings of The 10th International Conference on Software Engineering*, Singapore, 418-427.
- Robbins, J. E. and D. F. Redmiles (1998). "Software Architecture Critics in the Argo Design Environment". *Knowledge-Based Systems* 11(1): 47-60.
- Schön, D. (1983). *The Reflective Practitioner: How Professionals Think in Action*. New York, Basic Books.
- Shaw, M. (2001). "The Coming-of-Age of Software Architecture Research", *Proceedings of the 23rd International Conference on Software Engineering (ICSE 2001)*, 657-664a.
- Smolander, K., K. Hoikka, J. Isokallio, M. Kataikko and T. Mäkelä (2002). "What is Included in Software Architecture? A Case Study in Three Software Organizations", *Proceedings of 9th annual IEEE Conference and Workshop on Engineering of Computer-Based Systems*, Apr 8-11 2002, Lund, Sweden.
- Smolander, K. and T. Päivärinta (2002). "Describing and Communicating Software Architecture in Practice: Observations on Stakeholders and Rationale", *CAiSE'02 - The Fourteenth International Conference on Advanced Information Systems Engineering*, May 27 - 31, 2002, Toronto, Ontario, Canada.
- Strauss, A. L. and J. Corbin (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Applications*. Newbury Park, CA, Sage Publications.
- Zachman, J. A. (1987). "A Framework for Information Systems Architecture". *IBM Systems Journal* 26(3).