Systems Architecture: Product Designing and Social Engineering

Rebecca E. Grinter
Bell Labs, Lucent Technologies
263 Shuman Blvd
Naperville, IL 60566 USA

beki@research.bell-labs.com
http://www.bell-labs.com/~beki

ABSTRACT
The production of large and complex systems usually requires the coordination and collaboration of many individuals spread among numerous divisions of a corporation. However, much research examining coordination has focused on the subtleties of interactions between individuals who may work together in the same department. In this paper, I present a study of systems architects and the work that they do to coordinate design across organizational and institutional boundaries. I also describe the processes and tools that the architects use to support their work. The implications of the social processes involved in coordinating the design of large complex systems on the product and those involved in its production are discussed.

Keywords
Systems architecture, empirical studies of design and development, grounded theory, coordination and collaboration..

1 INTRODUCTION

The production of large and complex systems usually requires the coordination and collaboration of many individuals including developers, marketing staff, standards experts, and customer representatives. Often, these individuals are often spread among numerous divisions of a corporation. This paper focuses on one aspect of that production: the early stages of design known as architecture. Specifically, I examine the how architects coordinate the design of systems across organizational boundaries.

The role of architect is relatively new especially when it comes to software design. However, the idea of an architect has evolved from an older profession: systems engineering. In the late 1980s it was perceived that something was missing from systems engineering, an attention on the up front part of the process [22]. Specifically, it was perceived that the source of failure in many systems was that no one was explicitly focusing on the overall architecture — structure — of what was being designed. The role of architect evolved to fill that gap.

At the same time as the role of architect was being established within some corporations, software architecture emerged as a research area. Software architecture researchers have focused on a number of topics including architectural description languages for expressing the design, codifying architectural experience into principles, and domain-specific frameworks with formal methods for reasoning about architectures [13, 25].

While much of the research in software architecture has focused on the outcome of the process — the architecture itself — there is an increasing concern being shown for understanding how architecture work happens in practice [1, 13]. One reason why there has been a turn to understanding practice is it is often the relationship between the design and the organization that presents challenges. As one architect in a computer company explained:

> The job {of architect} isn't so much thinking up new architectures but getting them accomplished. That's the larger part of the effort. Lots of people have intellectual architectures and not too many people can translate those info actions and agreement and creation, that's really where the rubber meets the road.

Bass, Clements and Kazman [1] highlight the relationship between architecture and the organization that produces it in something they call the architecture business cycle. They identify a number of influences on the process including the development organization, the customer organization, any maintenance organizations, the technical experience of the architects, and the technical environment itself.

The process of building architecture also reveals these kinds of influences on design (for example [3, 11]). For example, there are clients who drive the design process through their financial support, governmental organizations that regulate the shape, height, and placing of buildings through laws, permits, zones, and so forth, and the architecture profession itself that sets standards for design, and encourages innovations. This design context spans organizations and even countries through the context of financial, legal, and professional attachment.

Systems architects face similar challenges, but little is known about how they manage them, and how they

produce technically working and organizationally workable designs. Empirical studies of engineering design work have often taken a different focus looking what happens during meetings [7, 17, 20] and individuals' design work [24]. Other studies have highlighted the challenges of the overall development process of which design forms one component [2, 9, 12], and how tools support their work [15]. More recently, studies of design have begun to explore the challenges in coordinating work across organizational and geographical boundaries [10, 19].

This paper builds on these studies by drawing on their insights about how design happens. However, it also follows the example set by studies of building architects to expand the analysis away from the design work done by a team of individuals sharing common goals, to examine the heterogeneous network of actors who shape the design of the architectures. In the next section, I introduce the methods used to collect and analyze the data and site of study. The following sections discuss how and why the architects studied bring a wide variety of participants into the design process and describe the technologies they use to support their work. I conclude by arguing that these architects coordinate the work of multiple groups and that they do this by negotiating in an ever-changing environment.

2 THE STUDY

2.1 Methods

The focus of this study was understanding how one corporation's architects go about their design work, which led to a choice of qualitative research methods [18]. The study itself consisted of two data gathering phases. The first phase involved interviewing the architects. Initially I conducted a group interview with three of the architects. The purpose of the interview was to elicit information about the range of their activities, the people that they interacted with, the history of their department, their place in the bigger organization, and other information of interest.

The information gathered from the group-interview was used to devise a guide for the semi-structured interviews. The guide developed focused on three important components of their work:

- questions about what architecture is;
- descriptions of the work that the architect does in the department including a typical project life cycle; and
- details about the resources (materials, people, technologies) that the architects use in their work.

The questions were designed to encourage the participants to talk at some length about their own work, what they produced, and the resources they used. Enough time was allowed for the interview to follow specific lines of questioning pertinent to an individual architect's work. For example, one architect interviewed reviewed proposals for a standards agency. This was a component of the job that some other architects interviewed did not have. By asking the architect a number of specific questions about this aspect of the work, I was able to learn about it.

I interviewed seventeen architects in the two departments. Typical interviews lasted around 45 minutes with several going over an hour. Each interview was subsequently transcribed. I used grounded theory — a method for developing theories from qualitative data — to analyze the data [14]. The grounded theory approach uses cycles of data gathering and analysis to develop an understanding of what is taking place and why. These cycles involve collecting data, breaking it down into conceptual units, reassembling the parts into a substantive explanation of why events occur, and then using the current understanding to drive the next phases of data gathering [27].

The second phase of the study consisted of following three architecture projects. The projects selected were designed to be a cross-section of the work going on, but were all small projects so that it would be possible to observe a substantial amount of the process unfolding. After following each of the projects to conclusion, I wrote up a report describing architecture work and sent it to subset of the architects I interviewed for their comments and feedback as a way of verifying whether I captured their work practices.

2.2 Site

The data in this study are drawn from two departments of architects that work for a telecommunications equipment vendor. The telecommunications domain has some unusual characteristics that influence the character of the products designed. First, reliability is critical in voice networks, and so designs have to provide fail-safe procedures, minimize downtime, and so forth. Second, most telecommunications systems have real-time performance requirements. Both reliability and real-time requirements can often stand in conflict with other principles of good design, and part of architecture work involves resolving those issues as favorably as possible.

Other characteristics of the telecommunications domain that architects work with come from customers and regulators. First, many of the products are configurations of smaller hardware and software components that customers can purchase collectively or selectively. Further, customers expect to be able add more pieces to their products over time. Second, telecommunications products also need to be designed to work with the phone systems of many countries that have different standards. Third, the telecommunications market has been changing rapidly with the growth of the Internet, deregulation of monopoly markets, and growth of wireless technologies. Not only is there increased demand for products generally, but features that used to be conceptually different are becoming increasingly blurred as definitions of data, voice, networks, broadcast, and routing change. Finally, both governmental and international standards organizations regulate parts of the telephony domain. Furthermore, the standards change over time, forcing changes to be made to the products affected to accommodate the revisions.

These characteristics of the telephony domain along with the size of the development efforts — often taking hundreds of people to build — are the reasons why the company employs architects: to maintain and extend the high-level design of the product. Specifically, their architects design new features and enhance their existing range of products as well as design new ones for emerging markets. Some of

the architects work on very high-level designs of networks of products, while others specialize in designing components and additions to existing product lines. Some architects work on ideas that may be realized within the next decade, while others focus on products that need to be built and released as soon as possible. The common ground these architects' share is that they all work on providing technological solutions to the problems that they are presented with.

In addition to providing architectural solutions to problems the architects studied also shared some other unique corporate demographics. All of them have worked in the company for a long time, with most of them serving ten years or more. They are expected to work on problems without close supervision and they are trusted to produce appropriate solutions that development groups can implement. In this sense they operate much like researchers: fleshing out the details and complexities of a problem, gathering data and resources to inform the choice of solution, weighing alternatives, and then writing up solutions.

3 DESIGN NEGOTIATION: ALLIANCE AND INFORMATION GATHERING

The design process begins with an architect being assigned a problem to work on. "Problems" — features or enhancements to the existing product lines, or entirely new products — are generated by the architects, their management, people in the development organizations, and field representatives on behalf of their clients. These problems eventually arrive to the architect who has the most expertise in the area.

Assuming that the problem gains support in the form of resources and staffing — which does not always happen — the lead architect assembles a team of colleagues to work on the problem. Although individuals are assigned problems, the architects almost always work in groups. The architects distinguish between two kinds of team participants: core members and consultants.

The core team members form the heart of the group. Core group members are expected to attend all the team meetings and contribute to the design of the solution for either as long as the project lasts or until they leave their current job. The lead architect aims to have all key aspects of the problem covered by members of the core group. The architects explained to me that they try to draw as many people as they can from their department because it is easier to negotiate for core members' time from people in their management chain. The lead architect turns to their architecture division and then the rest of the corporation — in that order — if there are areas of expertise missing from the department.

Consultants contribute to the project by attending specific meetings and providing feedback to the team about the current design. The architects draw the consultants from anywhere in the corporation. Finding and requesting time from consultants requires the architects to maintain and use extensive social networks that span the entire corporation. These networks have to be especially large to find consultants who may be geographically and functionally removed from the architects. All the architects I interviewed had worked in development before joining their departments. Several of them were well known throughout the corporation for having designed certain features of products. During their time with the company the architects had met many people, kept in contact with them, and now as architects were turning to them for advice.

Architects try to bring experts into the design process for more than just their technical knowledge. Architects have responsibility for the design of the solutions that they produce, but they do not build their designs. The implementation of the designs is given to development organizations, usually those that are working on similar products or have had responsibility for the product line to date.

Critically, these development organizations have some autonomy in deciding whether to implement the solution. This puts the architects in the position of needing to convince developers that these solutions are worth building. This process begins in the team formation stage when architects try to ensure that they have developers on the team in some capacity who will recommend the solution to their respective groups. Furthermore, the people that they bring into the architecture process will get to know the architects. This means that during development if problems arise with the architecture the developers are more likely to approach the architects and ask for help. This is important because it lets the architects know how their designs are working out in development, and makes sure that their next design fits the current product reality.

Architects also work with the development organization to get support and start the process of aligning the design to the implementation schedule. Development groups have their own schedules for building products. Architects need to find opportunities in those schedules where the developers can start working on their products, something that has to be planned months and even years in advance of the time when development starts. So, in addition to bringing developers into the team, the architects give presentations to development divisions as a mechanism for socializing the design and gathering information about whether and when the organization could build the design.

The development organizations are only one group among several that the architects need to socialize their solutions with. Much of this group socialization of architectures is conducted through presentations of the design. I found that by attending presentations given by architects I met the groups that need to "buy-in" to the final design.

Most projects have problem owners — people who generated and provided resources for the design work — and the architects present their solutions to those owners. Some architects like to have routine presentations scheduled in the early stages of the project. Others wait until they have reached a point where they believe that they should present their current working solution to the problem owners. The architects use these presentations to get feedback and support for further work. These presentations to problem owners serve as an acknowledgment of continued commitment to the architecture work by the problem owners.

Architects also give presentations to sales teams and customers. The purpose of presenting to the sales team is to get support to sell the products that result from their architectures. Architects are often bought in to present products to customers as the technical expert. The presentations, question and answer sessions, and time with the customers exposes the architects to the clients' concerns.

Customers are only one external source of information that architects need exposure too in order to build successful products. Governmental and international agencies directly influence the equipment vendors by generating standards for communications, and the media that handle that traffic. Some of the architects work with the government agencies, reviewing their standards and following their latest discussions about standards and regulations. This provides the architects with current information that they then use in their architectures to ensure that when the standard or regulation is implemented their design will be compliant. In some cases, it also provides the architects with an opportunity to influence the direction of future standards in ways that support their design work.

As well as conforming to standards, and offering features that their customers want, the architects must also follow what other telecommunications equipment vendors are developing. The architects I interviewed all subscribed to trade journals such as *Computer Telephony*. They kept journals, magazines, and books around their offices. It is hard to describe the volume of these subscriptions that most architects seemed to have. One architect referred to his extensive collection as a fire hazard.

This continual attention to broader market trends permeates the entire architecture organization. The architects' bosses, also meet routinely and share information about work going on inside the company, and innovations in the marketplace. Sharing news about where the telecommunications business is headed is a topic of conversation for everyone in the architecture organization whether it's over lunch, in a meeting, or at a conference or trade show.

The architects work in ways that allow them to accomplish their dual mission of designing technically possible and organizationally feasible products. They bring people from all over the corporation in to consult on their technical knowledge, and at the same time learn about other groups' priorities and schedules. The architects present their work to different groups to ensure that the solutions are attractive to build, buy, and sell. Finally, they continually look outside the company to align their work with standards agencies and competitors. All of this is necessary design work, without it, solutions might work, but could not be built or sold.

4 DESIGN SUPPORT: TOOLS AND PROCESSES FOR ARCHITECTURE

The architects use a variety of technologies in their work. Clearly, the telephone and electronic mail are vital for maintaining their networks of contacts. In this section, I will describe how two other technologies support their work: the world-wide web (WWW) and a viewgraph package.

The World-Wide Web (WWW) is clearly a source for gathering information. All the architects use the WWW, and many of them use it to gather information about products that might be elements their design must accommodate. However, for one of the two departments the web is much more than a resource as it is also used as a tool for sharing resources and an organizational memory.

The adoption of the WWW was not planned. Instead, the architects discovered the WWW as part of their role to investigate and experiment with new solutions, and bring the results of that into their work. Initially the WWW server was on one of the architects' machines. While he maintained the server, another architect taught herself cgi scripting. Slowly these bottom-up efforts spread out until a number of other architects were using the technology. This was made possible by an early commitment from the head of the department to have a common hardware and software platform in use throughout the department. Eventually, enough people found the WWW useful that the department invested in another machine to house the server. This was mainly because the architect whose machine had held the server was experimenting with other new software packages and in the process kept crashing his machine and consequently taking down the server. The shift to a stand-alone and stable server marked the beginning of the use of the WWW as a departmental tool.

The architects use the WWW to share project materials with others. A number of architects pointed out the advantages of using the WWW instead of other traditional mediums. Sometimes they have meetings with team members who are located in another state or even in another country. In the absence of the WWW, the team members would have to fax the slides or notes to the remote team members. Sometimes the faxes did not arrive, or were simply not arriving fast enough. The architects now put their slides onto their server and the remote members can easily get them. They also put meeting notes and other summary materials on to the WWW rather than photocopying and mailing them out to people. Now, each project has a small space on the server where materials relating to the on-going design can be found. These practices have been further reinforced through the emergence of a new departmental standard for documenting architecture work on the WWW.

The WWW may be a new technology, but the practices that it is supporting inside the architecture department represent a common pattern of adoption. In the beginning, the technology is adopted to support existing processes, and it succeeds when it supports these practices well enough that people continue to use it. However, the technology also provides new possibilities and slowly people develop new practices, and thus the technology begins to influence the character of the processes that it supports.

One of these new practices is something the architects refer to as borrowing. If an architect knows that the architecture she's working on is similar to someone else's then she may use the WWW server to copy the relevant materials. Rather than redoing an architecture by hand, the WWW provides a way for architects to cut-and-paste from others'

work into their own and then make modifications as necessary. They have also found it useful for bringing new people up-to-speed with the current state of design. Rather than spending time in meetings explaining the current design, the architects send people to the project web site. In that sense, the WWW becomes a repository of design rationale for facilitating new design work and bringing others up to date.

While most of the architects in this department are eager and regular WWW users, they have taken on an additional source of negotiation. The architects now have to persuade the people outside the department to adopt their WWW practices. Otherwise, the architects have to send out photocopies of meeting notes, viewgraphs, and documentation. However, over time as a department they have managed to persuade other architects — including those I interviewed in the other department — and others to adopt the WWW.

Another tool that the architects use a lot is a viewgraph package. The need for a viewgraph package reflects their need to have a medium for developing architectures that fits their working style, a style that involves talking with numerous groups about their designs. These viewgraphs were usually produced using a tool like Microsoft PowerPoint™.

Although a viewgraph tool does not carry the prestige of a CAD tool for producing architecture documentation, it offers the architects at least four advantages. First, it is easy to draw with the tool. Compare the simple drawing functions in the viewgraph package with other systems: a complex CAD package or text processing software. Second, the viewgraph package allows the architects to draw their own architectures. Another alternative would be to hand draw the architecture and give it to the art department to draw up professionally. Not only would they require considerable planning, but also if anything needed to be changed afterwards the architects would not have the ability to change the drawing. Third, as everyone in the department uses a standard viewgraph package the architects can share and borrow others' architectures and customize them for their purposes. Finally, slides are highly portable within the corporation. Other individuals work with the package too. The architects can send their viewgraphs to anyone who needs to see them, and know that they'll be able to read them. Given the diverse range of groups the architects work with this portability is critical.

5 DISCUSSION

The last two sections described the work that these architects do to design enhancements or new features and examined the role that technologies play in supporting them. This section discusses the role that the architects play in coordinating work from two perspectives. First, I describe how these architects go about coordinating design across organizational boundaries. Second, I discuss how the architects negotiate in a context that changes and where their ideas are subject to resistance.

5.1 Facilitating Design Across Boundaries

The architects work with people from their own department like other designers. However, these architects must also work with other sections of the organization. This work involves crossing organizational boundaries, both inside and outside the company. A considerable amount of their work involves garnering support and commitment from distant departments, maintaining current information about development schedules, and staying current with new standards, technologies and legislation.

The challenge of coordinating heterogeneous groups has recently become a focus for research [6]. Two mechanisms for coordinating this kind of work have been previously described: boundary spanners and boundary objects. In a study of software development projects, Curtis, Krasner, and Iscoe [12] identify boundary spanners as people who moved among different groups transferring information about the state of the project. These boundary spanners are characterized as informal roles, adopted by individuals with good speaking and listening skills who have contacts that span multiple groups.

Clearly, part of the work of these architects involves having good networks of contacts across the corporation. Unlike boundary spanners, the architects are expected to use these informal networks. It may not be a written requirement of their job, but everyone knows that knowing people is an essential part of being a competent architect in the company. In this sense, the corporation expects architects to work across organizational boundaries. In other words, they are sanctioned by the corporation to facilitate coordination across heterogeneous groups with different priorities and agendas. This makes them different from the boundary spanners described by Curtis who used their networks without any expectations on the part of the corporations they worked for.

Boundary objects are another approach to facilitating cross-group and inter-organizational coordination [26]. Boundary objects are:

> objects that are both plastic enough to adapt to the local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites. ([26] p 103.)

One example of a boundary object is a library index, which allows multiple groups — different groups of researchers with different objectives — to use the same materials, by presenting them in a standard way. A design process that involves multiple groups also generates boundary objects [8].

These architects certainly produce boundary objects. The viewgraphs and web-based documentation describing the designs act as guides for development, let managers see what features are being adding to products, help funders to decide whether to support the project among other uses. Here the portability across platforms and usability of the technology is critical in supporting those boundary objects, because it lets those multiple groups access the design in ways that are resonant with their own environments and skills in using these technologies.

However, I would also argue that these architects are a boundary object themselves through which other groups can coordinate enough to accomplish design work. Problem owners, funding groups, and developers coordinate

their needs through the architects. The architect becomes a centrally located resource available to these groups. In the corporation these architect are boundary objects known by their title and organized around their expertise with identifiable pieces of products or skills with standards and other technologies. Conversations and meetings serve as boundary "occasions" when the architects work to establish the shared understanding of the design. The architects share these understandings among all the groups involved, through their presentations.

It is the institutionalization of the use of networks of contacts, the title of architect, and the organizational scheme by which others locate appropriate architects that allow heterogeneous groups to design products. All the technologies that architects use in their work support this need to span multiple groups as seamlessly as the architects themselves do. Architects are a kind of organizational diplomat, helping the many groups it takes to build a technically working and organizational workable product come together to agree on a design.

In this corporation, architects are one group of individuals who support intra and inter-organizational coordination. They do this by having work that takes them across large sections of the company. While architecture may not play this role in other corporations where the architects may not work in these ways, studies of other corporations suggest that where there are organizational divisions one mechanism of managing the coordination required is through the institutionalization of a work role like the one these architects have.

For example, people responsible for release management also work with many sections of the organization to find the right pieces and assemble them into the final product [16]. Away from the software development world, Pycock and Bower's study of fashion design [21] also highlighted the interorganizational nature of that work and how groups of individuals have the role and authority to facilitate that coordination.

Studies of these kinds of individuals hold much promise for researchers interested in the challenges of intra and inter-organizational coordination. These individuals can be interviewed and observed using field research techniques. These kinds of people offer those interested in building systems to support coordination that spans divisions of a corporation insight into the kinds of technologies that can work.

5.2 Gradients and Contexts in Design Work

In recent years a number of studies have illustrated how the context in which work takes place influences and shapes the kinds of technologies that can be adopted and used. The design work here presents a particularly interesting case, because these architects find themselves working with two contexts, that of the customers and the development organization. They work to include the context of their customers — their users — while designing within the development context that surrounds them. Each context creates challenges that are worked out through the design itself.

The development context creates a number of challenges Bowers and Pycock [5] have described how the existence of a prototype created a *gradient of resistance*, which made changes in line with the prototype easier to negotiate for than radical alternations. These architects work with that gradient of resistance. Much of their work consists of adding new pieces to existing product lines or changing current technologies. These kinds of projects begin with an existing base of software and hardware. The architects could champion a solution that involves designing from scratch, but it takes more energy to convince the development organizations.

However, this is not just a gradient of technological resistance. The development organizations are structured and scheduled to build certain products. Over time they have built up expertise in the products that the build, and defined their relationships with other groups. If the architects suggest that the development groups build things that demand changes in resources, expertise, scheduling, and bring them into contact with other groups in new ways they often find that the gradient of organizational resistance grows.

In turn, the architects use their own *gradient of influence* given to them by their organizational position. Simply put designs that require organizational change requires more effort. Architects must negotiate every aspect of their design with every group involved. This negotiation lies somewhere between the two extremes of inter-organizational coordination. The print shop workers in Bowers, Button and Sharrock's [4] study had little choice in using a new system in their work, because it was part of their contractual obligation. In other words, the print shop workers were powerless to change their circumstances. While these architects are not powerless, they can not enforce their solutions using managerial authority, because they do not manage the implementation. Instead, these architects must to convince others on the strengths of their merits that designs are worthy of implementation.

A further complexity in managing these dual contexts is that they change each other over time. Changes occur in the market or standards that affect the corporation's direction and their customers' priorities. Changes inside the corporation also influence the products that architects design, and so change the nature of what can be sold to customers. Architects — especially those who work on projects that unfold over years — find themselves making changes to their technical solutions to accommodate these updates, and renegotiating agreements and support from other groups. One critical variable in this is the degree to which the technology in question is in a state of flux. More mature products tend to be stable in the sense that new changes are often alterations to what exists. However, other products may be much more susceptible to radical changes in areas where standards are evolving rapidly, or the underlying technologies are changing dramatically. A final source of change is the corporation itself, and when that happens then even the most stable of project can need modification to accommodate the reorganization.

Managing change and negotiation are core elements of the architects' work in this corporation. Their practices of

involving people in the design process, giving presentations and soliciting feedback, that manages to lower the gradient of resistance, while simultaneously increasing the gradient of influence. All of this continues throughout the project because at any time changes from inside or outside the corporation can have a major effect on the design.

6 CONCLUSION

The design and development of large and complex products is a coordination intensive activity requiring hundreds if not thousands of individuals to synchronize their efforts. While some of this work takes place among groups with a common goal, a considerable amount of coordination is required that spans multiple groups of an organization that have their own particular concerns. In this corporation architects own part of the responsibility of intra- and inter-organizational design. Thus, they represent an opportunity to study the challenges of coordinating design work across boundaries.

In this study, I have shown that these architects work in a web of social forces like their building architecture counterparts. These forces do not create additional work that needs to be taken care of in order to produce the technical design, they are what product architecture is about. It is the articulation work necessary to bring about the opportunity to design technically a new product or feature enhancement [23].

Their architecture work involves reconciling these social forces in ways that produce products that are both technically possible and organizationally feasible. This means that these architects are not only technical experts. Their practices suggest that they are also good communicators and listeners able to move among multiple groups, extract their concerns, and present a solution that reaches a compromise among the participants.

These skills and needs are reflected in the processes and technologies that the architects studied use in their design work. Specifically, these architects have evolved and standardized a set of practices that ensure that they get information and feedback essential for the design process. They have adopted technologies that allow them to share their work with all the interested parties readily. In addition to this, the organization has supported their collaborative activities by institutionalizing the role of architect. In conclusion, this study contributes to our understanding of what the challenges of coordinating design work in a large development corporation, and the kinds of processes and technologies that make this possible.

ACKNOWLEDGMENTS

I would like to thank all the architects who gave willingly explained their work to me and made sure that I was "in the loop" during projects as well as those who made the study possible. I would like to thank Paul Clements, Jim Herbsleb, Sanjoy Mazumdar, David Weiss, and the reviewers for their comments and suggestions on previous drafts.

REFERENCES

1. Bass, L., P. Clements, and R. Kazman, *Software Architecture in Practice* Addison Wesley Longman, Inc., Reading, MA, 1998.
2. Bellotti, V. and S. Bly. Walking Away from the Desktop Computer: Distributed Collaboration and Mobility in a Product Design Team. In *Proceedings of ACM Conference on Computer Supported Cooperative Work CSCW '96* (Cambridge, MA, November 16-20, 1996), New York, N.Y.: ACM Press, 209-218.
3. Blau, J. R., *Architects and Firms: A Sociological Perspective on Architectural Practices* MIT Press, Cambridge, MA, 1987.
4. Bowers, J., G. Button, and W. Sharrock. Workflow from Within and Without: Technology and Cooperative Work on the Print Industry Shopfloor. In *Proceedings of European Conference on Computer-Supported Cooperative Work* (Stockholm, Sweden, 10-14 September, 1995), Dordrecht, Netherlands: Kluwer Academic Publishers, 51-66.
5. Bowers, J. and J. Pycock. Talking Through Design: Requirements and Resistance in Cooperative Prototyping. In *Proceedings of ACM Conference on Human Factors in Computing Systems CHI'94* (Boston, MA., April 24-28, 1994), 299-305.
6. Bowker, G. C., *et al.*, *Social Science, Technical Systems, and Cooperative Work: Beyond the Great Divide* Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey, 1997.
7. Bucciarelli, L. L. An Ethnographic Perspective on Engineering Design. *Design Studies. 9*, 3 (1988), 159-168.
8. Bucciarelli, L. L., *Designing Engineers* MIT Press, Cambridge, MA, 1994.
9. Button, G. and W. Sharrock. Project Work: The Organisation of Collaborative Design and Development in Software Engineering. *Computer Supported Cooperative Work: The Journal of Collaborative Computing. 5*, 4 (1996), 369-386.
10. Carstensen, P. H. and C. Sørensen. From the Social to the Systematic: Mechanisms Supporting Coordinating in Design. *Computer Supported Cooperative Work: The Journal of Collaborative Computing. 5*, 4 (1996), 387-413.
11. Cuff, D., *Architecture: The Story of Practice* MIT Press, Cambridge, MA, 1991.
12. Curtis, B., H. Krasner, and N. Iscoe. A Field Study of the Software Design Process for Large Systems. *Communications of the ACM. 31*, 11 (1988), 1268-1287.
13. Garlan, D. and D. E. Perry. Introduction to the Special Issue on Software Architecture. *IEEE Transactions on Software Engineering. 21*, 4 (1995), 269-274.
14. Glaser, B. G. and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research* Aldine de Gruyter, Hawthorne, N. Y., 1967.

15. Grinter, R. E. Doing Software Development: Occasions for Automation and Formalisation. In *Proceedings of Fifth European Conference on Computer-Supported Cooperative Work ECSCW '97* (Lancaster, UK, September 9-12, 1997), Dordrecht, Netherlands: Kluwer Academic Publishers, 173-188.

16. Grinter, R. E. Recomposition: Putting It All Back Together Again. In *Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW '98)* (Seattle, Washington, November 14-18), ACM Press,

17. Herbsleb, J. D., *et al.* Object-Oriented Analysis and Design in Software Project Teams. *Human-Computer Interaction. 10*, 2-3 (1995), 249-292.

18. Lofland, J. and L. H. Lofland, *Analyzing Social Settings: A Guide to Qualitative Observation and Analysis*. 3rd Ed Wadsworth, Belmont, CA, 1995.

19. Olson, J. S. and S. Teasley. Groupware in the Wild: Lessons Learned from a Year of Virtual Collocation. In *Proceedings of ACM Conference on Computer Supported Cooperative Work CSCW '96* (Cambridge, MA, November 16-20, 1996), New York, N.Y.: ACM Press, 419-427.

20. Potts, C. and L. Catledge. Collaborative Conceptual Design: A Large Software Project Case Study. *Computer Supported Cooperative Work: The Journal of Collaborative Computing. 5*, 4 (1996), 415-445.

21. Pycock, J. and J. Bowers. Getting Others to Get it Right: An Ethnography of Design Work in the Fashion Industry. In *Proceedings of ACM Conference on Computer Supported Cooperative Work CSCW '96* (Cambridge, MA, November 16-20, 1996), New York, N.Y.: ACM Press, 219-228.

22. Rechtin, E. and M. W. Maier, *The Art of Systems Engineering* CRC Press, Inc., Boca Raton, FL, 1997.

23. Schmidt, K. and L. Bannon. Taking CSCW Seriously: Supporting Articulation Work. *Computer Supported Cooperative Work (CSCW): An International Journal. 1*, 1-2 (1992), 7-40.

24. Schön, D. A., *The Reflective Practitioner: How Professionals Think in Action* Basic Books Inc., New York, NY, 1983.

25. Shaw, M. and D. Garlan, *Software Architectures: Perspectives on an Emerging Discipline* Prentice Hall, Inc., Upper Saddle River, N.J., 1996.

26. Star, S. L., Cooperation Without Consensus in Scientific Problem Solving: Dynamics of Closure in Open Systems, in *CSCW: Cooperation or Conflict?*, S. Easterbrook, Editor. 1993, Springer-Verlag: Heidelberg, Germany. 93-106.

27. Strauss, A., *Qualitative Analysis for Social Scientists* Cambridge University Press, New York: N.Y., 1987.