Design as a Minority Discipline
in a Software Company:
Toward Requirements for a Community of Practice*

Michael J. Muller and Kenneth Carey
IBM Research / Collaborative User Experience
1 Rogers Street
Cambridge MA 02142 USA
+1 617 693 4235 and +1 617 693 5187
mullerm@acm.org and Kenneth_Carey@us.ibm.com

ABSTRACT
This paper provides a description of designers’ work practices in a software company. We describe a participatory analysis of the diversity of working relations and roles of designers of IBM’s Lotus software products. Designers are an example of a minority discipline – that is, a discipline whose members are often isolated in their work teams among coworkers with different training, backgrounds, and career paths. We explore differences between the practices of designers of Lotus software products and the published reports of design practices in group settings.

Keywords
Work analysis, design techniques, user studies, design, community of practice, knowledge management

INTRODUCTION
For a number of years, the IBM Research / Collaborative User Experience group (previously Lotus Research) has contributed design practices and design concepts to IBM Software Group’s Lotus software organization and its products, as well as a host of technology concepts and prototypes. Recently, there has been corporate and industry interest in communities of practice as key forms of intra- and inter-enterprise knowledge management (KM). We have combined this timely commercial focus with our history of design support, and we have begun to support a community of practice for designers working on Lotus software products [16]. In this report, we summarize a series of ethnographic explorations of the work and roles of designers in Lotus software organizations, and we contrast our findings with published accounts based on studies of groups of designers, as well as published advice regarding knowledge management.

DESIGN AS A MINORITY DISCIPLINE
The “Double-Knit” Organization
We have used McDermott’s knowledge management work as part of the justification and positioning of our project. McDermott makes a persuasive case for supporting communities of practice in large organizations [11]. In his analysis of the “double-knit” organization (Figure 1), McDermott first describes the formal structure of the organization, which often puts people into multifunctional product teams. McDermott notes that these teams often work well for the first or second version of a product. Ultimately, however, they begin to share attributes and problems with the more traditional structures of functional organizations, which the product teams replaced. Both structures tend to work well for short-term, focused activities. Both structures tend to lose coordination and commonality across organizational boundaries. Often, the first people to experience the problems are what we have termed the minority disciplines [16] – i.e., people who are the sole practitioners of their discipline within a multifunctional product team. In the software field, examples of minority disciplines are product team members who work in disciplines such as human factors, user assistance, and business planning.

Designers provide an example of a minority discipline in many product organizations. Designers are typically con-

Figure 1. The “double-knit” organization. The first direction of knitting is in multifunctional product teams. The second direction of knitting is in networks of practitioners – i.e., communities of practice.
cerned with the visual and dynamic design of products—and sometimes, as well, with the usability of products. Designers typically work within teams composed of software professionals. They are managed by software professionals, they are evaluated with the criteria used to evaluate software professionals, and they are often expected to follow career paths similar to those of software professionals.

Like other minority disciplines, designers often lack the support of other designers. They cannot easily ask for a review of a design by other designers. They cannot easily get help on a new graphical design software tool. They cannot easily get advice about how to make a convincing case to people from engineering or business disciplines—people, that is, who do not share a designer’s vocabulary or perspective or criteria for what constitutes a “good” design.

McDermott claims that people in the situation of designers can benefit by being members of loose networks of practitioners with similar backgrounds. This approach initially looks like a matrix management plan: Designers could be members of a product team and of a design team.

McDermott is at pains to show otherwise. He observes that communities of practice provide different kinds of benefits from those provided by product teams (see also [9]). Typically, the benefits of communities of practice are invisible to formal organizations (see [20]). Communities of practice do not produce products, and may not even produce reports or other documents. What they do help to bring about is continuing professional development for their members, as well as coordination and collaboration for people who lack such opportunities within their teams. McDermott makes a particularly strong contrast as regards career paths: He suggests that members of majority disciplines (e.g., software professionals in software product teams) are likely to pursue their career growth through their teams, whereas members of minority disciplines may more effectively pursue their career growth through their communities of practice.

More generally, organizations may find communities of practice useful to keep members of minority disciplines aligned organizationally with product teams, while maintaining strong and vibrant professional communities and practices across teams and divisions.

**Studying Designers as a Minority Discipline**

We have begun a participatory analysis of the work of designers on Lotus software products [16]. Our activities included the following:

- **Informal ethnographic interviews** at 20 designers’ sites
- **CARD sessions** for participatory analysis of typical activity flows in nine designers’ work.

**Ethnographic Interviews**

The informal ethnographic interviews were conducted in designers’ offices (n=18) or studio spaces (n=2). We asked a series of questions about designers’ backgrounds, career paths into design, working relations with their product teams, and working relations with other designers. We asked designers to describe in detail how they worked with team members and organizations, and how (and if) they made use of artifacts (e.g., sketches, databases of specifications, semi-public displays) that are sometimes used by designers in most Lotus software organizations. Whenever possible, we asked designers to show us examples of their work or to give us access to on-line repositories of design materials, and we often continued our discussion around the materials or around a screen image (on-screen or printed...
to paper for annotation). While we had our own “agenda” about information that we were expected to collect, we maintained an open interview style that could and did follow the designers’ senses of what was appropriate, important, or worthy of discussion.

**CARD**

The CARD sessions were conducted in a modification of the established CARD practice [8] – for details of the innovations, see [17]. CARD is a participatory work practice that can be used to analyze human work, to design a high-level or “task-level” system for supporting that work, or to critique an existing work method or software system. Participants (including end-users) use a deck of cards, each of which represents an aspect of work.

As described in [8] and earlier publications, CARD has made use of two types of work representations (layers 1 and 2 of Figure 2):

- Most of the cards showed work objects or work activities (layer 1) – the kinds of observable or formal activities that might be part of a training manual, behavioral assessment, or workflow. In the vocabulary of knowledge management, these objects and activities (and the knowledges associated with them) would be called “explicit.”

- A subset of the cards showed mental, cognitive, or emotional events (layer 2) – the kinds of skills and craft that distinguish an experienced practitioner. These events would be invisible to a conventional, behaviorally-oriented task analysis, including most cognitive task analyses. In the language of knowledge management, these activities (and the knowledges associated with them) would be called “tacit.”

An example of one of the many CARD-based analyses is shown in Figure 3.

One of the strengths of CARD has been that it supports discussions of goals, plans, strategies, motivations, knowledges, skills, and so on – crucial determinants of the success of many human undertakings -- that are difficult or impossible to observe directly, and that have been difficult to represent in conventional analytic approaches (e.g., [15]).

The new approach that we used in this study included a third type of representation (layer 3 in Figure 2). The first two layers represent activities and events that take place during the work that is being analyzed or designed. For the work with designers, we added a third set of cards to support descriptive and critical discussions that take place during the CARD session – i.e., as a set of afterthoughts or reflections upon the work. These reflective or analytic representations addressed issues such as contextualizing the designers’ work within a product lifecycle, and noting circumstances of work that were particularly advantageous or particularly troubling.

Using this third level of analysis, designers served not only as reporters of their own work (level 1 of Figure 2), and not only as co-interpreters of their work (level 2), but also as co-analysts of their work (level 3). In this way, we moved closer to the goals of participatory analysis [2, 14]. When designers contextualized their activities within product life-cycles, they helped us to see commonalities and contrasts in practices across product groups and across corporate divisions. When designers noted strengths or weaknesses of their practices (or of the product stories that they narrated), they helped us to see problems that we could help them solve through the community software system that we are planning to provide.

<table>
<thead>
<tr>
<th>Designer’s Role</th>
<th>Span of work</th>
<th>Working materials</th>
<th>Work site(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual design</td>
<td>Typically limited to graphical expression of ideas or product identity</td>
<td>Software tools for graphical design (e.g., commercial products supporting image creation and modification)</td>
<td>Own office, offices of team members</td>
</tr>
<tr>
<td>Usability engineer</td>
<td>Typically limited to formative evaluation of iterative prototypes and products</td>
<td>Usability methods (inspections and informal small-sample usability problem analyses)</td>
<td>Usability lab, offices of team members</td>
</tr>
<tr>
<td>Interaction design</td>
<td>Often the entire product, including graphical design, “look and feel,” dynamics, user experience, and various aspects of product design, project/program management, client contact, and marketing.</td>
<td>Eclectic use of diverse materials and practices, including visual tools, usability techniques, and databases (specifications, bug reports) to organize and coordinate the team’s work</td>
<td>Own office, team members’ offices, product managers’ offices, clients’ sites</td>
</tr>
</tbody>
</table>
RESULTS

General Orientation
Designers of Lotus software products work in a diverse settings and places. The largest numbers are in Cambridge MA USA and Westford MA USA. However, there are also designers in Atlanta Georgia US, Lexington Kentucky US, and in Rehovot Israel, and IBM employees from other organizations in Toronto Ontario Canada – in fact, one design team has members distributed in Westford, Lexington, and Rehovot. Designers in Lotus Professional Services (a consulting arm of IBM) may work from customer sites anywhere in the world. The community of designers spans multiple organizations, locations, time zones, countries, and continents.

Designers’ Isolation
Individual designers of Lotus software products are often the only members of their teams who have formal responsibility for design issues and/or who have design expertise. They are thus the sole holders of design knowledge. Some designers may go weeks without speaking with another designer. Many designers said that they wished they could ask other designers for advice – even as simple as “am I doing this right?”

The few designers who worked in pairs clearly valued their opportunities to collaborate. In one case, two designers had given up separate cubicles and crowded themselves into a single-person office, which became a kind of informal design studio. In this cramped space, they had arranged their desks so that their screens were at right angles. Their chairs bumped together at the back corners, and they shared one telephone extension. When we spoke sympathetically about the discomforts of their situation, they told us that they had deliberately arranged their computers and work areas in this way, so that they could easily glance at one another’s screens.

Three other designers, when they had the chance, made sure that design assistants were located directly across the hall from them – again, the theme appeared to be a clear line-of-sight view of one another’s work. This theme is also very clear in accounts of design studios and design education, where visual access to work is a carefully considered and negotiated attribute of work areas and work practices [1, 6, 7, 12, 13, 23, 25].

Designers’ Roles
Designers and their work fall into three categories, as summarized in Table 2. In general, graphical designers and usability engineers tend to stay within their own domains, using their own specialized tools or methods. Their assignments are often of relatively short durations (one to three weeks). Their role in product teams is often relatively reactive.

By contrast, interaction designers often work out of role, using whatever tools were required for collaboration, coordination, or legitimation with other team members, managers, or clients. They were often the “project glue” that maintained a set of documents that were used as points of articulation by diverse teams. Their role was more often one of leadership and initiative.

Designers’ Artifacts
Specifications Documents
Designers make use of certain characteristic artifacts in their work with their product teams. The principal artifact is the specification or “spec.” Designers’ self-positioning with regard to the database of specs was often deeply thought-out and intensively discussed – and quite variable from designer to designer, or perhaps from product to product, or perhaps from team to team.

Specs are stored in databases – typically Notes databases, which can be replicated from central servers to individuals’ laptops upon need, with relatively low-effort synchronization of any changes to the various replicas of the databases. A database may contain hundreds of spec documents, as well as discussions and other annotations on those documents. Keeping informed of relevant changes in a spec database is a complex but necessary task for product team members. For the organizations that we studied, specs were crucial points of definition (what will be developed?), coordination (who will do what?), and collaboration (what are the options and alternatives in doing this?)

In some cases, designers become the owners of all of the product specs, serving as the recorders and coordinators for their teams. This appears to be a position of power, responsibility, and respect. Over several years, some designers have worked with their teams to develop notifications or subscriptions to specs, by which team members can register as being interested in changes to particular specs, and designers can automatically send out change notices to these subscribers whenever they make a major update.

In other cases, designers perform a kind of conceptual “triage” on the spec database, choosing which documents to own, which documents to check periodically for changes, and which documents to ignore. Based on the limited number of cases in our study, designers appear to adopt the “triage” approach when they join a project later than the developers, or when a large product has been composed from already-formed components that were derived from previous products.

In yet other cases, designers are assigned a subset of the specs as their own responsibilities. Based again on the limited number of cases in our study, designers tend not to remain in this disempowered position. Most designers who receive such a partial assignment develop their reputations within their product teams, and come to have greater and greater responsibilities for the specs.
Prototypes and Graphic Images

Designers communicate with product teams in a variety of settings and through a variety of media. One of the strongest media is prototypes or (in the case of visual designers) graphics. Prototypes may take a number of forms, from deliberately informal, hand-drawn, comic-like stories, to scenario-like presentations (showing overall flow, more as a narrative than as a demonstration per se), to elaborately hyperlinked sets of images in a presentation file (simulating end-user actions with moderate fidelity) to high-fidelity, labor-intensive click-by-click simulations using advanced graphical prototyping products. Designers choose among these different methods based on a complex analysis that includes factors such as

- The stage of the product development process
- The current problem to be solved
- What needs to be expressed
- (For designers in the consulting organization) the client’s specific requests, if any

Some highly influential designers use very low-technology prototyping tools, such as basic painting programs. Others dig deeply into what appears to be simple business software, reinventing presentation systems into elaborate prototyping environments. Still others develop competences that are very similar to those of programmers, albeit in environments that support rapid idea expression rather than production software performance.

Evolutionary Prototypes

Designers in the consulting organization used prototypes at an earlier stage of their work than most other designers. In some of their engagements, they introduced a prototype as a means of both testing their understanding of client requirements, and of “selling” the client on their team’s understanding, creativity, and client-focus. When appropriate, they moved through a succession of prototypes, from “hands-off” (consultant-operated) to “hands-on” (customer-operated) to production code. Unlike the prototypes made by designers in product teams, the consultants’ prototypes often involved high production values and considerable attention to marketing and related issues.

Designers’ Practices

As noted above, most designers work in isolation from other designers. Their everyday working practices focus on effective communication with their product teams. Interaction designers were particularly concerned with team communication. However, their choices among practices vary considerably, based in part on their roles and in part on their team structures.

Visual designers appear to listen more than speak, in keeping with the reactive nature of much of their work (as noted above). Visual designers tend to communicate largely through providing graphical designs upon request, and often through an interaction designer. Also as noted above, some visual designers have relatively short-term relationships with product teams, moving from one product to another in as brief a period as a month. Many visual designers work on more than one problem, project, or product during the same period, juggling demands from multiple teams.

Usability professionals also tend to have a reactive and short-term relationship to product teams, but their work often involves more contact with product teams, and more advocacy for their work and their findings. (These issues have been discussed many times in the literature – see, e.g., [22]. Because of the extensive CHI and UPA literature on usability practices, we will not describe them in detail here.)

For some product teams, usability professionals are the only design staff. In these cases, their involvement tends to be longer and more expansive. For other product teams, the role of the usability professional is sometimes buffered or brokered to the team through the interaction designer. We heard some interaction designers talk about the work of usability professionals as a kind of partnership; We heard other interaction designers talk about the work of usability professionals as if they were contractors or consultants, external to the team, and sometimes a source of uncertainties, misunderstandings, and difficulties.

The heart of design practice occurs with interaction designers. As we stated in the introduction to this section of the paper, interaction designers often become selective generalists, taking on major roles in communication and coordination for the entire product team. In this way, they serve a role similar to that of a software architect – another role in Lotus software teams that involves deep technical commitment (albeit in a different technical field) and intense communication and negotiation skills in one-on-one and group settings (see also [10, 18] for analyses of similar roles). The following examples give some idea of the variety of practices that interaction designers have adopted – sometimes strategically, sometimes opportunistically, and sometimes because there appeared to be no other choice:

- Several interaction designers became de facto or declared product designers, moving fluidly among perspectives of interaction, features, application programmer interfaces, back-end processes, product plans, and market analyses.

One pair of interaction designers on the same product deliberately imported specifications database structures, agent technologies, and working practices from previously successful products in other parts of the company (see “Designers’ Artifacts,” above).

- One interaction designer, newly assigned to a product team, discovered that the product had been assembled from the pieceparts of several other products, and had significant problems in consistency and in inter-module

---

data-sharing. She initiated a change in tactical and medium-range team activities in order to meet these challenges, with forward-looking warnings of renegotiated feature lists and deadlines. This designer became, in effect, the program manager for her product, while maintaining her commitment to design.

• One designer encountered a similar – but better understood – set of product pieceparts that had to be integrated into a single offering. She responded with a social initiative anchored around Cooper’s notion of personae [4]. She used the personae to create common understandings among designers and developers in three locations, spanning two continents, arguably three cultures, and seven hours of time-zone difference. Her team recognized that her efforts had entered the domain of product definition, and asked her to take on the part-time role of product manager, meeting with customers and preparing product briefings and future product vision presentations – while continuing to fulfill her day-to-day design leadership responsibilities.

• Several interaction designers told similar stories of being asked to help a product group design a single user interface feature, and then seeing their roles expand to multiple features, and then the entire product. One of these designers maintained a UI-focused practice, with powerful implications for other components of the product. Another of these designers was asked to do “a little” work in showing how his design could be implemented in HTML, and then was asked to become the de facto UI developer for all of the HTML-based features. In order to make his work more effective, he agreed to this role, and became designer-developer for the course of that project.

The Exceptions: Designers in Design Groups
The preceding descriptions apply to designers working as the sole design-oriented member of a product team. We discovered other work and activity patterns among designers who were members of two small design groups.

Within design groups, there was a more open sharing of draft design ideas. One group maintained the design studio practice of relatively formal critiques of one designer’s work by another designer. A second group held weekly meetings which were often devoted to a review of current draft design work on a product. Our impression was that these shared practices were made possible not by the physical proximity of the designers, nor by the fact of their shared organization, but rather by the work culture that was actively promoted and protected by leaders of those groups. People in these groups spoke of skills in listening and in critiquing the work of others. We do not yet understand the specific skills that designers valued in these areas, but we hope to learn more.

CONTRASTS AND CONUNDRUMS
When we compared our findings to conventional design research, we encountered surprises. We discuss one surprise in each of three categories of contrasts and contradictions:

Artifacts
According to the design research literature, design studios and other groups of designers usually share some form of semi-public displays of informal works in progress [1, 12, 13, 25, 26]. These “pin-up boards” appear to provide awareness of one another’s work, and opportunities for coordination and critique [12, 13]. Similar practices have been applied to areas of software engineering [7] and mathematics education [23].

While designers of Lotus software products often make displays of designs for intra-team work, they generally do not use of such displays on a public basis for sharing design ideas from one team to another. Indeed, when asked about collaborative activities with other designers, roughly half of the informants immediately talked instead about collaborative activities within their software teams. On an anecdotal basis, one of us has observed a similar team/product focus in design at other companies. We now wonder: Is the low emphasis on visual sharing that we observed a result of software culture? of the minority status of design work within a more technology-focused team? of the text-oriented form of most collaborative software? It will be interesting to see if we can apply the lessons from virtual architectural design studios [12, 26] to UI design problems (e.g., in a related domain, [7, 25]).

Practices
According to design research literature, design studios and other groups of designers make extensive use of a work practice called a critique or “Crit” [1, 7, 12, 13]. A Crit is a designer-to-designer review that may be given by a master designer to an apprentice, or may be exchanged among peer designers. Among the organizations that use them, Crits are well-understood practices with defined roles. With the exception of two working groups, designers of Lotus software products are generally reluctant to engage in Crits or similar peer reviews with designers from other teams. Again, on an anecdotal basis, one of us has observed similar reluctance to engage in inter-team Crits at other companies. As noted by Kvan [6], the development of trust and dialogue across product groups may be the most challenging part of bringing in these practices. If Crits are a valuable aspect of design practice, then we are challenged to bring about working tools and working relationships that will foster their use in our organizations.

Knowledge Organization
Conventional advice in establishing communities of practice is to develop a single, hierarchical taxonomy of knowledge resources [3, 21, 24]. However, what we have learned from designers argues against such a monolithic way of organiz-
ing and accessing shared resources. Members of the design community adopt the three perspectives summarized in Table 1, plus additional product-oriented views and design-research views. The differences in these perspectives have important implications for the design of shared resources:

- For example, the units in which designers think and produce are different for the different design perspectives. Visual designers tend to focus on icons and graphical displays. Interaction designers tend to focus on documents and/or prototypes. Usability specialists tend to focus on summaries of usability issues in formats that can easily be understood by developers. Most designers can fluidly move from one perspective to another upon need. Nonetheless, when working intensively on a task – usually against a deadline – designers are most comfortable working within a particular perspective. Software support for designers in these different roles should reflect the perspectives and needs of each role.

- As regards practices and supports for practices, visual designers often think in terms of tools (primarily software tools). By contrast, usability specialists often think in terms of methods (practices that may or may not involve specific pieces of software). The difference initially appears to be subtle, but the consequences are profound: Visual designers often take an anti-process stance, whereas usability specialists often find their strength within well-defined and well-structured usability procedures. Usability professionals often adopt the strategy of making minor adaptations to a foundational evaluation process (for example, while accommodating to a new software tool). By contrast, visual designers are likely to define their (less formal) work practices in the terms required by each software tool. Broadly speaking, the difference can be summarized as follows:
  - Usability specialists tend to use the method to give structure to the usage of the tool.
  - Visual designers tend to use the tool to give structure to the usage of the method.

This generalization should be understood as speaking of tendencies, rather than of extremes of practice, as illustrated diagrammatically in Figure 4.

- Finally, the organizational contacts and persuasive rhetorics of the three categories of designers vary widely. Visual designers tend to use graphical displays as communication aids when working with interaction designers and developers and, occasionally, marketers. Interaction designers tend to use story, narrative, and analysis (e.g., of product exposures and needs) as arguments to other members of their teams and/or external clients and customers; graphical displays may fit within narratives and stories, but take a secondary role that contrasts strongly with the primary role of graphical displays for visual designers. Usability specialists tend to use usage evidence (usually in the form of UI problems and issues, not in the form of statistical analysis) as arguments with developers and, from time to time, with interaction designers and visual designers as well.

Our observations are convergent with those of Davis et al. in the LIRÉ project [5]. Subgroups among their designers exhibited differences in perspectives and granularity of actions on documents, leading to the maintenance of an evolving, user-controlled set of multiple organization schemes for their repository.

The dominant KM strategy of relying on a single monolithic knowledge-framework [3, 21, 24] thus appears unlikely to meet the diverse needs of the three major perspectives of designers in the group that we studied (visual designers, interaction designers, and usability specialists). Instead, we appear to need a way to maintain multiple, simultaneous views (ontologies) that can self-reconfigure when new knowledge is added to the community resources [19], perhaps similar to those of the LIRÉ project [5]. In broader terms, we see a need to support multiple ways of organizing distinct forms of interrelated knowledges for multiple constituencies – a step toward more democratic ways of identifying, formalizing, and sharing knowledge in a KM context (e.g., [19]).

We are intrigued by these apparent contradictions between our findings and the published KM research and advice. We look forward to further investigations and technology explorations, to help us and the designers understand whether technologies and practices from the research literature can help refine our design practice – or whether that literature is simply not representative of work, organizations, and undertakings like ours.

CONCLUSION

The work analysis described in this paper is now complete. Our next steps will be a series of interventions intended to support and strengthen cross-product collaborations among designers. Some of these steps will be technological, such as providing a set of on-line resources for sharing among designers. Some of these steps will be social, such as finding opportunities for dialogue and collaboration among designers, and (if appropriate) designing new technologies to support new forms of interaction. We hope to report these interventions at future conferences.
ACKNOWLEDGMENTS
We thank Merry Morse, Rob Corell, David Millen, John Patterson, and Irene Grief for helpful discussions and advice.

REFERENCES


