A Study of Collaboration in Software Design

James Wu T.C.N. Graham
School of Computing
Queen's University
Kingston, Ontario, Canada, K7L 3N6
{wuj,graham}@cs.queensu.ca

Paul W. Smith
IBM Canada
pwsmith@ca.ibm.com

Abstract

This paper presents a study of collaboration in software design at a large software company. Ethnographic studies of development teams in the field are relatively rare, so this paper contributes to a small, but growing, body of knowledge about the collaborative activities involved in such design work. Five separate development groups were studied over a six-week period. The methodology included shadowing, interviews and communication event logging. A novel PDA-based application was used for real-time data collection. The results of the study indicate that designers communicate frequently, using a wide variety of communication and collaboration modalities. Designers prefer general-purpose tools to domain specific applications. In support of communication, designers frequently change their physical location throughout the day. Finally, designers frequently change the ways in which they communicate, changing their communication modalities and styles.

1. Introduction

Numerous studies point to the importance of communication in the design and development of software [1,3,5,7,14]. However, relatively little literature is available to show how software designers actually collaborate in performing their tasks, and what tools they choose to aid in this collaboration.

The object of this study was to investigate collaboration and tool use in software design. This goal was motivated by a perception that many existing tools intended to support software design address only a limited subset of the tasks and activities involved, and do so in a manner that is often inappropriate to the design task as actually carried out. For example, the Rational Rose design tool [12] provides excellent support for graphical modeling, code generation, and testing, but limits support for collaboration to coarse-grained, asynchronous sharing of design models. It has been suggested [5,4] that this leads to a mismatch between design activities and the functional support provided by such tools, and that this mismatch has resulted in low tool adoption rates and a general aversion to the use of domain-specific software design tools.

Little empirical data, however, exists to support the assertion that such a mismatch indeed exists. We therefore designed this study to examine the degree of collaboration in software design, the form that this collaboration takes, and the tools that software designers use to communicate. The study was intended to identify any common activities that are not well supported in existing tools, and so guide new research directions in tool design for software engineering. Towards this goal, the study investigates four hypotheses:

1. Software design is a highly collaborative activity in which team members frequently communicate.
2. Software designers prefer general purpose and informal tools over domain specific tools for both design and communication.
3. Team members frequently change their physical location throughout the day.
4. Team members frequently change the ways in which they communicate.

These hypotheses are motivated by existing literature, which is surveyed in the next section. We then introduce our methodology for this study, and present our results. Finally, we discuss the implications of our findings and how they might be applied to future design of tools supporting collaboration in software development.

1.1. Existing research

Curtis, Krasner and Iscoe [2] highlighted the need to examine software development at a team level, rather than focusing on the work practices of the individual. They studied problems in designing large software systems through interviewing personnel from 17 large projects, employing a behavioral model to facilitate analysis. The focus of this analysis was on three different problems - the thin spread of application domain knowledge, fluctuating and conflicting requirements, and communication bottlenecks and breakdowns.

Kraut and Streeter [7] surveyed inter-group coordination practices in 65 different projects at a single large software company. The study employed

questionnaires and interviews and focused on coordination practices, structural characteristics of projects that affect coordination techniques and project success as it related to coordination practices. An interesting result of their study was that developers cite discussion with peers as the most important of 18 ways in which teams coordinate activities.

Norcio and Chmura [1] present the results of an investigation into design activity of software engineers working on the Software Cost Reduction project with the Naval Weapons Center. This study intended to investigate the design process in general, and to identify characteristics of the process that can predict progress. Their method involved collection and analysis of weekly activity reports, as well as activity logs. In this study, they find that discussion among software engineers is correlated with progress in design

Seaman and Basili [13] addressed productivity aspects of communication by empirically studying the organizational and process characteristics that influence the amount of effort software developers at a large software company spend in communication activities. The study used both quantitative and qualitative methods for data collection and analysis, including real-time observation and structured interviews. The results of this study indicate that several organizational factors do affect communication effort, including physical proximity; i.e. communication becomes more difficult at a distance.

In a study by Tang [16], the work activity of small groups of people was videotaped and analyzed in order to understand collaborative work and to guide the development of tools that can support such work. The investigation focused on shared drawing activities, specifically listing, drawing, gesturing and talking around a shared drawing surface. This study differs from others mentioned in this section because it employs observational techniques in a controlled laboratory setting rather than in the field. Of importance among the results was the finding that the process of creating and using drawings conveys much information that is not captured within the drawing itself.

Singer and Lethbridge [15] reported on their experiences studying the work practices of professional software engineers at a telecommunications company. They utilize real-time observation of individuals and groups, as well as questionnaires. The study focuses on work practices in order to determine existing work patterns. The paper also introduces the technique known as *synchronized shadowing* for recording real-time observational data. The results of the study were used as design input for a software maintenance tool.

In [14], Sim presents the results of long-term structured interviews involving four "software immigrants", i.e. newcomers to a software team maintaining a legacy system at a large software company, in order to characterize their "naturalization" process. Sim investigated the difficulties "immigrants" encountered while acclimatizing themselves to the new project and processes. An important result was the finding that a lack of appropriate documentation forced immigrants to rely on communication with mentors within the team in order to gain an understanding of the system.

2. Methodology

Our study was undertaken at a large software development company over a six-week period. The subject pool included designers from five separate development groups within four different departments. Departments ranged in size from 14 to 31 members, and each department was responsible for a different product or aspect of a product. The development groups were at different stages in their respective design processes, from early design to code development, as well as pre- and post version release. Each development group was responsible for mandating its own process. Some groups were developing mature products, and followed defined processes, while other groups were developing experimental products and therefore used more flexible processes. Therefore, despite being in one company, it was possible to observe a wide variety of styles of work.

We employed three primary methods of data collection: interviews, shadowing and communication event logging. Seventeen interviews were conducted within the various groups to probe individual impressions regarding collaborative and design tool use. Members of each group were shadowed, i.e. silently observed in a manner similar to that described by Singer and Lethbridge [9]. A total of twenty-five hours of observation were performed providing subjective and contextual information about collaborative team activities, communication patterns and tool use. Finally, eighteen people recorded their communication patterns for a period of one workday in order to provide quantitative data about the interaction between team members, as well as insight into the typical communication patterns of these developers.

Due to the nature of the activities under observation, as well as the inherent nature of such observations, we knew that conclusive results about these activities were unlikely to be captured. The use of three different methods of data collection helped to provide corroborating evidence. Furthermore, interview data provides insight into the motivations and reasoning behind different aspects of

behavior that cannot be accurately deduced from observation or event logging. As will be seen in section 3, each method provided similar data, increasing our confidence in the results. Nevertheless, it needs to be emphasized that this study included designers from only a single company, over a relatively short period of time. Their behavior during this time is not necessarily reflective of that of the entire company, or of the software industry in general.

There are challenges associated with data collection in such observational studies [9]. The ability to record large amounts of mixed data in real-time is essential to successfully capturing sufficient information about activities, tool use and context. The need to be fast, flexible and mobile is a challenge that eliminates many sophisticated, PC-based methods. Furthermore, doing so in a manner that supports subsequent analysis in a sufficient and convenient way eliminates many alternative methods such as video or audio recordings.

In order to meet these challenges, we developed and used a PDA-based database application, facilitating stylus-based recording of predetermined details about the use of design tools, collaborative activities and Workstyle [18]. Additionally, these forms supported various categories of textual input to more flexibly record general and contextual information about these points of interest, or to capture details regarding unforeseen points of interest. A portable PDA keyboard was used to support rapid textual input. Figures 1 and 2 depict examples of the database forms used for shadowing and interviews. Hypothetical information is included in the forms as an example of their use. Figure 3 depicts an example of the paper form provided to subjects in order to track their communication throughout a workday.

In order to process and analyze the extensive amount of data captured during the study, data from all three information sources was transferred to PC based tools. PDA databases were automatically imported into Microsoft Access [10], and details of the communication logs were manually transcribed into Microsoft Excel [11]. These tools permitted automated analysis of quantitative aspects of the data, as well as providing a better interface to the data than was possible with the PDA. However, much of the data collected required significant human interpretation in order to be as thorough and representative as possible in the analysis presented in this paper.

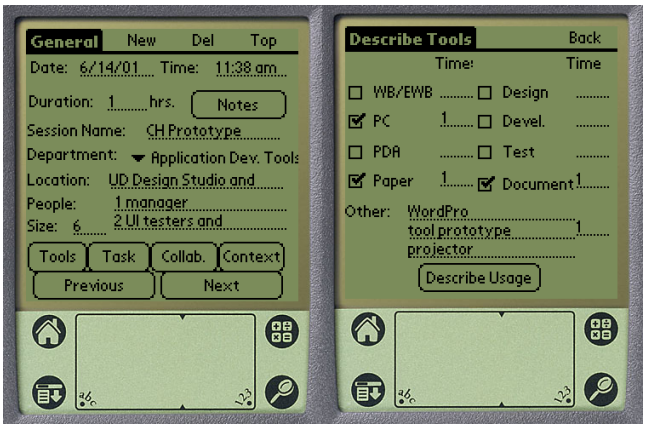

Figure 1: Shadowing data input form examples. On the left is the top-level form. Data about tool use can be recorded in the form on the right. Analogous forms can be used to record data about Task, Collaboration and Context.

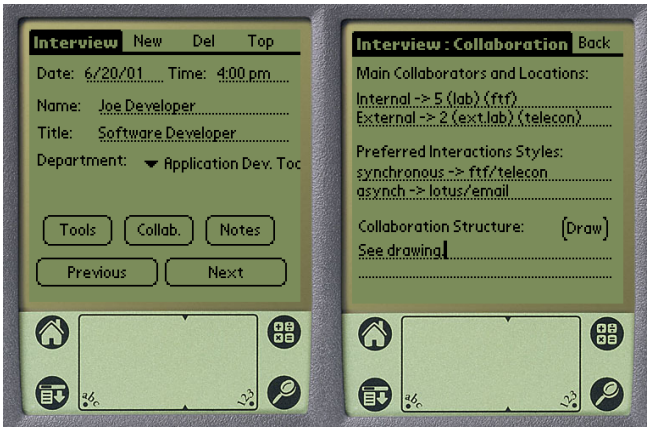

Figure 2: Interview data input form examples. On the left is the top-level form. Data about collaboration styles is recorded in the form on the right. Analogous forms can be used to record data about Tools, as well as general interview notes.

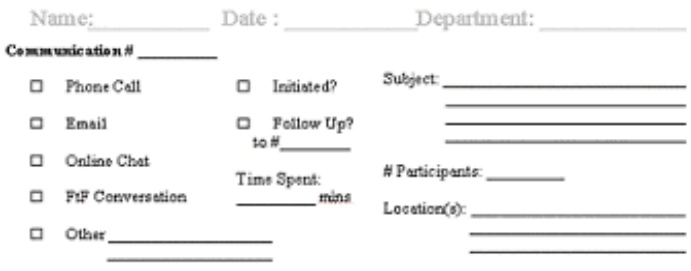

Figure 3: Example communications log entry form.

3. Results

In this section, we present our results from the observational study. It is important to reiterate that these results represent a single data point – i.e. a set of groups observed within a single company over a single six-week period. The results should be considered not as proof, but

Metric	Sample Size	Mean w/ 95% Confidence	Standard Deviation
Communication Events per Day	274 Events	15.2 ± 3.4	7.4
Communication Minutes per Day	2241 Minutes	124.5 ± 46.8	101.3
Group Size per Communication Event	274 Events	3.11 ± 0.40	3.43

Table 1: Interactivity between team members based on communication logs from 18 developers

as contributors to a body of evidence about the nature of collaborative design. It should also be noted that the statistics presented here should be considered as conservative, as the self-reporting used to collect much of the statistical data may result in under-reporting of collaborative and communication events.

In the following sections, we address each of the four hypotheses presented in the introduction.

3.1. Interactivity between team members

The study revealed that team members maintain a high degree of interaction in order to support their work. Team members communicated frequently and extensively, and often switched between communication modalities. For example, we observed that, designers averaged 15 different *communication events* a day. A *communication event* is considered to be a single, continuous interaction such as a single email, telephone call or meeting. These events ranged over sending email, telephone conversations, impromptu hallway interactions, and scheduled group meetings. (This number does not include emails received from others.) Furthermore, on average these events consumed 124 minutes, or over 2 hours of the workday, and involved an average of 3 different people. Findings are summarized in Table 1. Another point of interest is the high variance found in the results. Not only is interaction extensive and frequent, but also patterns of communication vary widely between individuals

Regardless of the variance, it is clear that considerable time is devoted not to production of design related artifacts, but to simple communication tasks. These results are consistent with earlier studies by DeMarco [3] and Jones [6].

Shadowing observations provide contextual information about these interactions. Consider the following two excerpts from the general commentary recorded while shadowing two designers over a period of approximately one hour, each working independently in their offices. Details in brackets indicate paraphrasing for privacy, clarity or brevity.

1. *[Jack] receives phone call to confirm meeting. Uses Lotus email to discuss design and implementation details with [colleagues]. Uses browser to get information from [a database] and responds with an email. Checks for more email. Receives telephone call, checks email while on phone. Returns to [working]. Writes multiple emails to ask questions of different colleagues, and checks for new email frequently (20mins). [Jill] drops by to ask a question regarding low-level dependencies. [Mike] drops by, but leaves because [Jill] is tying up time. Says he will come back later. [Jack] checks email when [Jill] leaves, Within seconds [Mike] returns to ask question.*

2. *[Sue] dropped by [Fred's office] for less than 5 seconds, asked a [question], received an [acceptable] answer and left. Subsequently, [Fred] paid a return visit, but [Sue] was then busy and promised a return visit. Extended periods of silence occur [between bouts of discussion] when both [Fred and office mate Sam] [go about what] they are doing (may be 3-5 minutes at a time). [Sue] returns, bringing a paper document (application output log file) to discuss – [Fred] diverts his attention to the new discussion. [Fred] leaves office with [Sue] to go to her office and review her work at her PC. [Mary] drops by and interrupts [Sue] with unrelated question. It is quickly answered and [Mary] leaves.*

These anecdotal descriptions are indicative of the degree of interactivity between team members. They demonstrate the amount and frequency of communication and synchronous interaction involved in ostensibly asynchronous or independent activities. Collaboration does not only happen in long, planned meetings, but rapidly and frequently in unplanned or impromptu interactions. These anecdotes also indicate the frequency with which designers move between different communication mechanisms (in this case, primarily email and face-to-face discussion.)

3.2. Use of general purpose and informal tools

Designers observed during this study predominantly used general-purpose tools over domain specific tools, and showed a marked preference for informal interaction

with these tools. The results presented in this section are from both shadowing and interview data. As both methods have limitations, results from both are presented to allow correlation between them. While the amount of use of each tool varies in the shadowing vs. interview data, both methods identified the same set of tools as being important. In some cases, labels on the pie graphs represent sets of similar tools that have been reduced to a single label for clarity. For example, the label "Text Editors" represents a set of text entry tools, including word processors.

Figure 4 shows results on the use of design tools. We distinguish between "creative design tools", which are used in the informal or brainstorming phases of design, versus "formal design tools", which are used to create archival design artifacts.

Not surprisingly, both interviews and observations show that creative design is largely supported by informal media (such as paper and whiteboard). This preference is consistent with observations of other researchers [8, 17]. For formal design tasks, Lotus Notes and text editors figure prominently in both observed and expressed tool use. This is partially due to the internal culture of the company under study: though designers were free to choose tools that suited their preferences, the company had standardized on Notes for document sharing.

These results demonstrate a preference for lightweight tools that place few restrictions on designers, even in later, more formal stages of design. As opposed to a domain specific design or process management tool that embodies assumptions about the artifacts produced or the processes used to produce them, Notes provides increased flexibility as the document database accommodates all manner of design artifacts. This is reflected in the significant use of various text editors for formal design, all of which are compatible with a Notes document repository. Domain specific design tools are entirely absent in both expressed preferences and observed tool use. Some development tools were listed as preferred for low-level design, although such use was not observed in practice. This is a clear indication that flexibility in collaboration is more important to designers than the advantages of domain-specific design tools, such as syntax checking of designs or code generation.

Figure 5 shows tools used by software developers to support collaboration in both distributed and co-located settings. Not surprisingly, for distributed interactions, standard communication tools such as telephone and email are well represented. Furthermore, Lotus Notes figures prominently as it is management-mandated and serves for many as their sole email tool, while

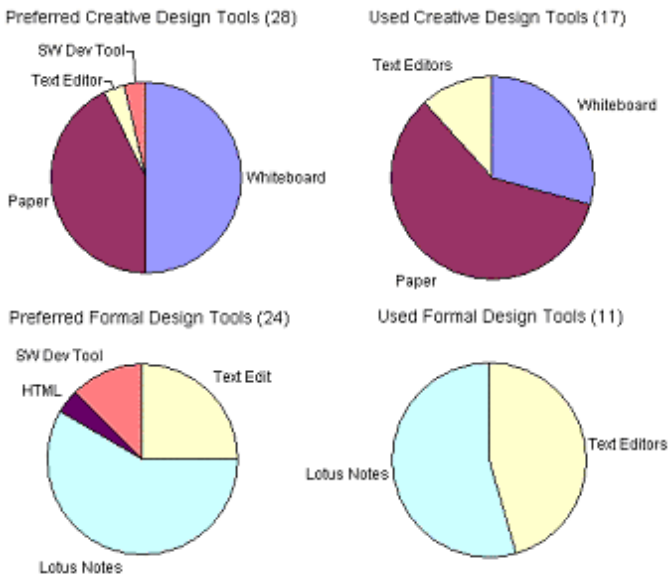

Figure 4: Design tools - Observed use (shadowing data) and expressed preferences (interview data)

others use different email applications that, according to interviews, better suit their preferences. Of interest is the difference between the perceptions and reality regarding use of Notes as a tool that supports collaboration. It may be that many see it simply as a document repository and do not associate it with supporting distributed, asynchronous collaboration. Where co-located interaction is possible, there is an overwhelming preference for and use of face-to-face communication, often in conjunction with whiteboards. Though some designers expressed a preference for the telephone even in situations where face-to-face interaction was possible, such behaviour was not frequently observed. Finally, there was a surprising use of text editors and Notes as co-located collaboration tools. This was typically observed as multiple designers huddled around a PC running Notes (or a text editor) and working collaboratively on the document.

Figure 6 presents results on tools supporting collaboration in both synchronous and asynchronous settings. Again, interviews reveal a distinct preference for face-to-face communication for synchronous interaction. Telephone and whiteboard are also well represented as preferred and observed tools for synchronous interaction. This is further evidence of the preference for lightweight and informal interaction between designers. Electronic tools that support synchronous interaction through application sharing, or shared place implementations, are not represented at all.

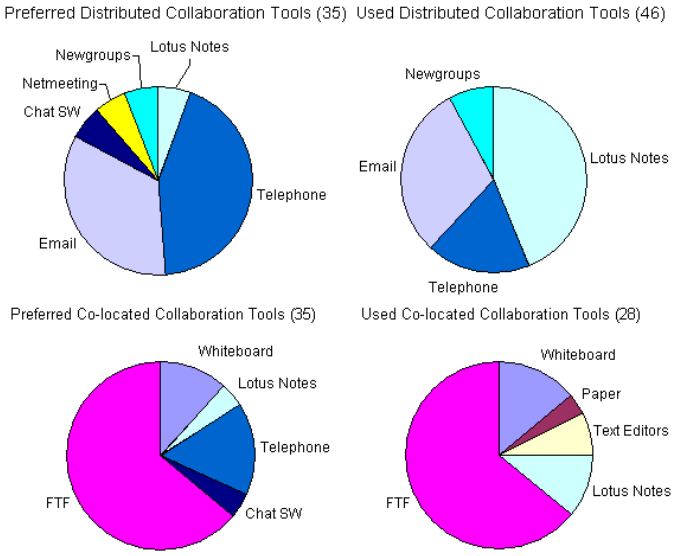

Figure 5: Collaboration Tools – Observed use (shadowing data) and expressed preferences (interview data) based on physical distribution of collaborators

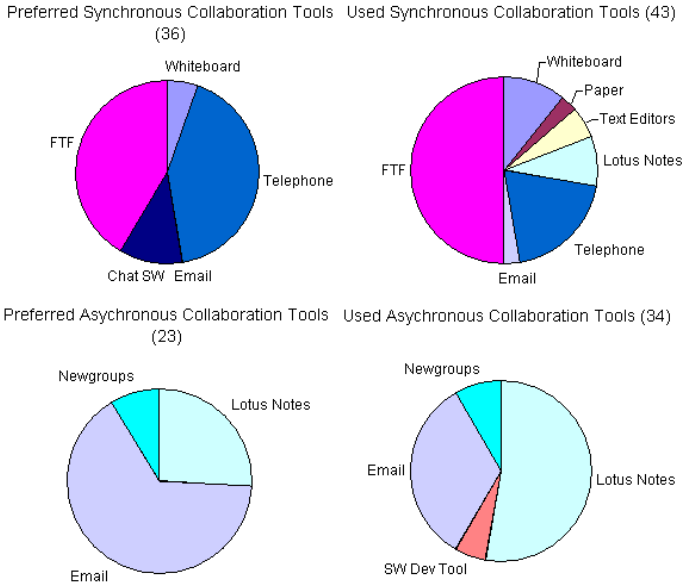

Figure 6: Collaboration Tools – Observed use (shadowing data) and expressed preferences (interview data) based on interaction synchronicity

In one designer's view this is because

"[Face-to-face, telephone and whiteboard] are most efficient. Other [mechanisms] require more time, and can leave confusion and other issues. [For example, electronic] chat is not as rich, and can take much more time to achieve the same results…"

Others described tools for synchronous, distributed collaboration as "too fussy" to set up, requiring too much time without sufficient gains versus traditional means.

The more complex the interaction with the tool, the more attention the tool draws away from the collaboration at hand. Perhaps this is why we see represented in these graphs only the basic tools for accomplishing the design task within given constraints.

3.3. Collaboration involves frequent changes in location

This study also reveals that the designers change location frequently. This involved moving between offices for face-to-face interactions with individual colleagues, or to common meeting areas for pre-planned group meetings. This also involved general movement around the building to access various facilities (printers, copiers, kitchen, cafeteria, etc), which triggered impromptu or unplanned interactions. Our results are based on communication log entries regarding the various locations in which different communication events occurred. We found that designers collaborated, on average, in between six and seven different locations every day. We also looked at movement within *threaded communication topics,* which are communications regarding the same topic that involve multiple events over the day. On average, we found that people move at least once in 85% of communications requiring multiple interaction events, and an average of more than 3 such threads occur each day. These values can be found in Table 2. For threaded communications involving more than 2 events, this rate increases, and people move at least once (1.44±0.86@95% confidence) for every such thread.

Again, the variance is considerable, indicating that there is a wide range of behaviors in this regard. Some people move significantly more or less than others, though most moved at least once per threaded communication topic.

Metric	Total Observed	Mean w/ 95% Confidence	Standard Deviation
Physical Movements per Day	79 Movements	6.6 ± 2.3	4.1
Movements in Threaded Communication Topics	33 Movements	.85 ± .33	1.3
Threaded Communication Topics per Day	56 Threads	3.11 ± 0.89	1.94

Table 2: Location changes in collaboration, based on communication logs from 18 developers

Modality	Total	Mean w/ 95% Confidence	Standard Deviation
Phone	314 Minutes	17.4 ± 14.0	30.4
Email	371 Minutes	20.6 ± 7.5	16.3
E-Chat	68 Minutes	3.8 ± 5.0	10.8
Face-to-Face	1488 Minutes	82.7 ± 38.0	82.3

Table 3: Minutes per Communication Modality per Day based on communication logs from 18 developers

Modality	Total	Mean w/ 95% Confidence	Standard Deviation
Phone	32 Events	1.8 ± 1.2	2.6
Email	130 Events	7.2 ± 2.0	4.3
E-Chat	11 Events	0.6 ± 0.7	1.4
Face-to-Face	101 Events	5.6 ± 2.0	4.4

Table 4: Events per Communication Modality per Day based on communication logs from 18 developers

Another means we have to assess frequency of location change for collaboration is to examine the frequency of face-to-face interactions. These types of communication events inherently imply increased movement, as a location change is often required for one or more of the participants. An exception is when people are already co-located at the time of communication. Figures 5, 6 and 7 all show the importance of face-to-face communication as an interaction modality. Tables 3 and 4 show the frequency and extent of face-to-face communication relative to other modalities. Far more time is spent in face-to-face communication than any other modality. This is particularly interesting when compared to telephone use: in general, people prefer to talk in person rather than on the phone, even when this implies a change in location. Similarly, designers were observed to engage in more face-to-face communications than telephone conversations (table 4.) These comparisons are statistically significant at the 95% confidence level.

The anecdotes described in section 3.1 (regarding interactivity) also illustrate the nature of the majority of location changes. As the designers showed a marked preference for face-to-face interaction, there was much more movement between offices as communication was required.

3.4. Frequent changes in communication modalities

In addition to frequent changes in location, and in correspondence with the frequency of interaction, team members often change the mechanisms they use to interact. This means that they move frequently and fluidly

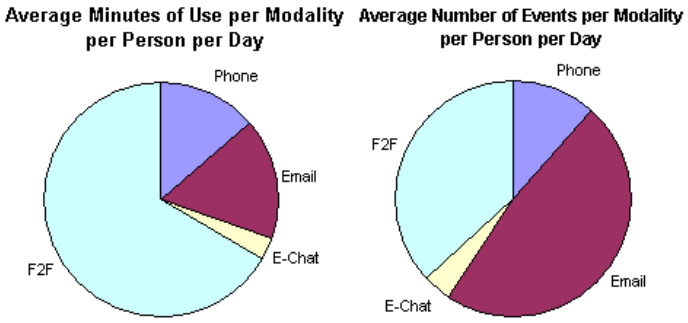

Figure 7: Average daily use of different communication modalities

between different communication modalities, for example between telephone and email. This implies not only a change in mechanism, but often a fundamental change in the way in which the team members interact. Specifically, some changes in modality imply a change to the synchronicity of interaction. For example, following up a face-to-face communication with an email implies a change from synchronous to asynchronous interaction.

Tables 3 and 4 provide some insight into the variety of modalities used to support communication in software design. Each modality is used both frequently and extensively enough to imply a significant amount migration between them. This is demonstrated more clearly in Figure 7, which shows average use of each modality during a day. Although face-to-face interaction is predominant in these graphs, each modality is represented, indicating that movement occurs between them each day. Similarly, Table 5 presents results regarding the extent and frequency of use of synchronous vs. asynchronous interaction. More time is spent in synchronous interaction (approximately 1.5 hours/day), but considerable time is also spent in asynchronous communication (35 minutes). More importantly, the number of events of both kinds of interactions is indicative that movement between these modes of communication occurs.

Table 5 presents more specific results regarding shifts in communication modality that support these claims. However, movement between communication modalities and changes in interaction synchronicity within a single thread of collaboration is more interesting than simply the frequency and extent of use of on a daily basis. To examine this, we again looked at *threaded communication events,* multiple communication events regarding a common topic. As shown in Table 2, we identified 56 such threads, an average of more than three per designer per day, each averaging more than three events. This means with regard to the three separate subjects, there were on average three separate communication events per day. Within these 56 identified

Metric	Total Observed	Mean w/ 95% Confidence	Standard Deviation
Minutes of Synchronous Communication per Day	1870 Minutes	103.8 ± 43.9	95.0
Minutes of Asynchronous Communication per Day	640 Minutes	35.5 ± 24.6	53.3
Synchronous Events per Day	144 Events	7.94 ± 2.78	6.01
Asynchronous Events per Day	137 Events	7.61 ± 2.04	4.42
Synchronicity Shifts per Threaded Communication Topic	44 Shifts	0.77 ± 0.27	1.04
Modality Shifts per Threaded Communication Topic	59 Shifts	1.01 ± 0.35	.77

Table 5: Changes in modality during collaboration based on communication logs from 18 developers

threads, we found 59 shifts in communication modality, an average of once per threaded communication topic. Of these, 44 involved changes in the synchronicity of the interaction, an average of three out of every four threaded communications. Of course, while our observations were limited to a single day, some threaded communications might have longer duration.

Once again, the variance in these results is considerable, indicating a wide range of behaviors in this area. People change modalities with different frequencies, likely due to personal work style or characteristics of their tasks. Regardless, the frequency of these changes is evidence of the need to address such activities in tools supporting collaboration in software design.

4. Analysis

Before discussing these results, it is important to reiterate that they are representative of observations of only five groups at a single software company and may only be truly applicable to other contexts if the situations are sufficiently similar. While the results may not be considered as conclusive about the nature of collaboration in software design, they do contribute to a body of evidence supporting our hypotheses.

4.1. Methodology

The first major obstacle in this study was determining an appropriate method for recording details about the activities of the designers. We learned that it is difficult to study field behavior in such broad activities as software design. The requirement that our observational activities not interfere greatly with developers ruled out controlled experimental studies, so we adopted an ethnographic approach. This approach required us to capture large amounts of heterogeneous data in real time. As discussed in the section on methodology, this led us to develop and use a PDA-based database. This facilitated mobility and rapid data entry, as well as subsequent data analysis.

We believe this approach shows promise for empirical studies in software organizations. A common alternative approach is to use video and audio recordings to capture behavioral data. As is likely typical of production environments, the software company was unwilling to permit such recording. Similarly, we felt that the synchronized shadowing methodology as presented by Singer and Lethbridge [15] was focused at too low a level of activity for our purposes, and may have interfered to an unacceptable degree with the activities of the designers in their daily routine. In order for the database approach to be effective, the input forms must be designed to quickly capture both predetermined aspects of interest, as well as details about unanticipated behavior. Furthermore, multiple streams of data from different sources must be collected such that they may corroborate each other and strengthen any claims that may be made based on them.

Much of the data was automatically imported into standard tools, and reports and statistics generated. This is certainly an improvement over the degree of automation available for the analysis of video and audio recordings. Considerable work was nevertheless required to analyze free-form transcriptions of observations. Further research into this approach would no doubt increase the degree to which the data collected could be automatically analyzed.

4.2. Hypotheses

This study has also provided us with an increased confidence in our original hypotheses about the nature of collaborative software design. These hypotheses were listed in section 1.

(H1) Team members were observed to be highly interactive, spending on average more than two hours per day on communication tasks. Communication was predominantly face to face or via telephone or email. The results corroborate Kraut and Streeter's [7] claims regarding the importance of communication in the coordination of team activity, as well as Norcio and Chmura's [1] findings that discussion is correlated with progress in design. Also, team members often changed

various aspects of their interaction such as location, synchronicity or modality of communication. These findings, in combination with the demonstrated preference for face-to-face interaction, can be viewed as supporting Seaman and Basili's [13] assertion that communication becomes more difficult at a distance; a preferred mechanism (face-to-face) is unavailable, forcing the use of other, less preferred, communication means.

(H2) We saw that designers preferred lightweight, general-purpose tools for design versus domain-specific design tools. In the early stages of design, not surprisingly, paper and whiteboard were frequently used. In later stages of design, general-purpose tools such as text editors and Lotus Notes were still preferred over domain-specific design tools. This lends support to Tang's [16] claim that the process through which an artifact is created is as important to design as the content of the resulting artifact itself. Similarly, many designers were observed to use tools that, while remaining appropriate and sufficient for the task at hand, minimized the complexity of the interaction. An interesting example of this was an observed use of Microsoft Paint in combination with screen-capture functionality for rapid user interface prototyping. Rather than use a more complex, domain specific, prototyping tool, some designers simply cut-and-pasted images of required interface components that already existed in other applications (e.g. tool bars, menu items, etc) and assembled images of the new interface. These images were used in slide presentations to prototype basic behaviors of the new interface, and provided an effective means of evaluating the look-and-feel of the developing application. We hypothesize that the advantages of easy communication (e.g., putting a text or PowerPoint document into Notes) outweighed the benefits of domain-specific tools.

This preference for general-purpose tools held also for tools used to communicate, where the most commonly used tools were telephone, email, whiteboard, and face-to-face discussion. Interviews related this preference to the overhead associated with using such tools versus that associated with the special-purpose collaboration facilities in software design tools.

Despite the fact that the results reveal a general preference for lighter weight, general-purpose tools over domain-specific tools, some people expressed opposing preferences. For example, some designers preferred formal design tools, and desired more formality in available tools. In some cases, atypical preferences were a result of a specific external factor. For example, one designer never used a whiteboard because of the smell of the dry-erase markers used with the board supplied in his office. Another designer never used a whiteboard because he couldn't reach it in his office due to space constraints that required an additional desk be placed directly in front of the board.

(H3) We found that developers change locations frequently in order to collaborate, showing that on average, developers collaborated in more than 6 locations per day. According to interview data, this was due to a strong preference to work face-to-face. Many designers felt it was simpler, quicker and generally more efficient to use standard communication tools, including meeting face-to-face, than to establish remote interaction though tools. This often meant that people would walk up and down multiple flights of stairs numerous times each day to meet in person rather than use a telephone or some other collaboration tool.

However, despite the variability in the nature of the interactions between designers, individuals still expressed different preferences for particular collaboration styles. Some predominantly used face-to-face (when possible), some telephone, and others made use of email even in situations where a face-to-face interaction may have been available and equally appropriate. The high standard deviations in our results reflect variations in individual preferences in various contexts. Although face-to-face is on average strongly preferred, interviews revealed individuals who avoided co-located interactions as much as possible.

(H4) Designers frequently change the way in which they communicate, and carry on multiple, simultaneous threads of collaboration. Shadowing revealed that it is typical for designers to attend a face-to-face meeting on a topic, then follow up with email, ask a supplementary question by telephone, follow up with more email, etc.

We saw that in the course of a single day, in threaded topics, people change modality more than once on average, and that these changes often involve a change in synchronicity. Moreover, developers on average carried out more than three simultaneous threaded interactions in the course of a single day.

4.3. Impact on tool design

The results of this study have clear implications for the design of tools supporting team-based software design in large companies. Clearly, a tool that does not support communication to some degree ignores a large part of the daily activity of designers. Furthermore, a tool that supports only asynchronous communication, via email or document repositories, does not address the predominantly synchronous interactions in which designers engage.

An implication of these results is the importance of flexibility with respect to how a tool supports collaboration. Changes in physical location, synchronicity and communication modality are frequent, and current tools do not sufficiently support such changes, if at all. In most existing tools, changes in synchronicity and location require a change in modality as well, imposing additional overhead on designers that choose to use them. Furthermore, tools should be tailorable in the way in which they support interaction in order to accommodate variations in individual preferences under the variety of conditions under which teams collaborate. This means providing flexibility not only in the collaboration styles supported, i.e. co-located/distributed, synchronous/ asynchronous, but also in the interaction mechanisms available to support individual collaboration styles.

5. Conclusion

This paper presented results of an observational study of collaboration in software design undertaken at a large software development company. We described a novel approach for performing such studies. Specifically, we used a combination of real-time shadowing, interviews and communications logging to present a multi dimensional view of the behavior witnessed. Additionally, we developed a PDA-based database that facilitated quick, easy and portable data recording, as well as subsequent analysis. The design of this database helped to guide the research while in the field by prompting the recording of details about specific aspects of behavior relevant to our hypotheses.

We also identified a variety of behavioral patterns regarding collaboration and tool use at this software development company. Team members communicated frequently and extensively with each other, and a considerable amount of an individual's time was spent in communication. Furthermore, the nature of these communications changed regularly in terms of synchronicity, location and modality, though face-to-face interactions were strongly preferred. Finally, designers preferred to use general-purpose tools that suited their needs, whether for design or communication, rather than use domain specific tools that imposed significantly more overhead on their task or interaction.

6. Acknowledgements

This work was partially supported by Communications and Information Technology Ontario (CITO), the Natural Science and Engineering Research Council (NSERC) and IBM's Center for Advanced Studies.

We would like to acknowledge the contributions made to this work by Turid Horgen and Amina Razvi of the Massachusetts Institute of Technology, as well as all of the software designers and developers who took part in this study.

7. References

[1] Chmura, L. and A. Norcio (1986). Design Activity in Developing Modules for Complex Software. Empirical Studies of Programmers.

[2] Curtis, B., H. Krasner and N. Iscoe (1988). "A Field Study of the Software Design Process for Large Systems." Communications of the ACM 31(11).

[3] DeMarco, T. and T. Lister (1987). Peopleware. New York, Dorset House.

[4] Iivari, J. (1996). "Why are CASE Tools Not Used?" Communications of the ACM **39**(10): 94-103.

[5] Jarzabek, S. and R. Huang (1998). "The Case for User-Centered CASE Tools." Communications of the ACM 41(8): 93-99.

[6] Jones, T. C. (1986). Programming Productivity. New York, McGraw-Hill.

[7] Kraut, R. E. and L. Streeter (1995). "Coordination in Software Development." Communications of the ACM 38(3): 69-81.

[8] Landay, J. A. and B. A. Myers (1995). "Interactive Sketching for Early Stages of Design". CHI '95, Denver, CO, ACM Press.

[9] Lethbridge, T., Sim, S. and Singer, J. (1998 July), "Studying Software Engineers: Data Collection Methods for Software Field Studies", Submitted May 2000 to Empirical Software Engineering

[10] Microsoft Corp. – Access, http://www.microsoft.com

[11] Microsoft Corp. – Excel, http://www.microsoft.com

[12] Rational Corp. – Rose, http://www.rational.com

[13] Seaman, C. B. and V. R. Basili (1997). "Communication and Organization in Software Development: An Empirical Study." IBM Systems Journal 36(4).

[14] Sim, S. E. and R. C. Holt (1998). The Ramp-up Problem in Software Projects : A Case Study of How Software Immigrants Naturalize. International Conference on Software Engineering, Kyoto, Japan, University of Toronto

[15] Singer, J. and T. Lethbridge Studying Work Practices to Assist Tool Design in Software Engineering. International Workshop on Program Comprehension, Ishia, Italy, NRC.

[16] Tang, J., C. (1991). "Findings from Observational Studies of Collaborative Work." International Journal of Man-Machine Studies.

[17] Wagner, A. (1990) "Prototyping: A day in the life of an interface designer". The Art of Human-Computer Interface Design. Addison-Wesley, Reading, MA, pp. 79-84

[18] Wu, J., Graham, T.C.N, Everitt, K., Blostein, D. and Lank, E. (2002) "Modeling Style of Work as an Aid to the Design and Evaluation of Interactive Systems". Proceedings of CADUI'02. Valenciennes, France.