A Visual Calendar for Scheduling Small Group Meetings

TR88-020

April 1988

David Beard and Alan Humm
with
David Banks
Anil Nair
Muru Palaniappan

The University of North Carolina at Chapel Hill
Department of Computer Science
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175

A VISUAL CALENDAR FOR SCHEDULING SMALL GROUP MEETINGS

David Beard and Alan Humm with David Banks,
Anil Nair, and Muru Palaniappan

ABSTRACT

Scheduling small group meetings is difficult. Scheduling requires access to everyone's calendar, typically located in scattered pockets or purses. Often, even when schedules are finally collected, no mutually open time slots are available. One could place everyone's calendar on-line and use an automatic scheduler to find a time slot. But it often is difficult to trust the results, because correct scheduling rules are elusive, varying with the people and agenda of a particular meeting. What's needed is an automatic scheduling system that presents and summarizes the available information to the user, points out time-slots of varying mutual acceptability, and still allows the user to look "behind" the results to see who is blocking a time-slot with which event.

We have developed a prototype visual scheduler (VS). Imagine an individual's weekly calendar as a grid on an overhead-projector transparency. Events are darkened blocks with the event's priority denoted by the block's translucency. A user can quickly locate mutually acceptable time slots by stacking together the transparencies of the participants, holding the stack to a light, and looking for openings. Even if no open slot exists, the user can still find the least disruptive time slot by locating those slots admitting the most light and examining which meeting participants are blocking the slot with what events. Users can directly manipulate events and modify them through functions in pop-up menus.

We evaluated VS with both a within-subject controlled experiment and a field study. The controlled experiment compared automatic to manual scheduling using both visual and text schedule representations. These evaluations demonstrated that visual automatic scheduling is beneficial when scheduling small group meetings. Users thought both priority-rated events and access to scheduling decision reasoning was advantageous. VS is still used regularly to schedule meetings by several research projects at UNC-CH, and over 75 "standing" schedules are stored online.

1. INTRODUCTION

Scheduling small group meetings is difficult. Scheduling requires access to everyone's calendar, typically located in scattered pockets or purses. Often, when schedules are finally collected, no mutually open time slot is available. One could place everyone's calendar on-line and use an automatic scheduler to find a time slot. But "black-box", automatic tools hide the scheduling process, so it is difficult to trust the results. Further, correct scheduling rules are elusive, varying with the people and agenda of a meeting. What is needed is a scheduling system that presents and summarizes the available information to the user, points out time-slots of varying mutual acceptability, and still allows the user to look "behind" the results to see who is blocking a time-slot with what event.

We have developed a prototype visual scheduler (VS) for rapidly scheduling meetings. With it, an individual's weekly calendar is imagined as a grid drawn on an overhead-projector transparency. Events are darkened blocks, with the event's priority denoted by the block's opacity. Users quickly locate time slots by stacking transparencies together, holding the stack to the light, and looking for openings. If there is no open slot, the least disruptive slot corresponds to the one admitting the most light. The user can also point to a time slot and see a list of the users with blocking events.

We evaluated VS with both a within-subject controlled experiment and a field study. The controlled experiment examined the relative merits of automatic visual scheduling and manual scheduling. The field study considered the merits of visual display of composite schedules, priority rating of events, and most importantly, examined whether VS was actually useful for performing real scheduling work. From the field study we also developed a list of problems with the current implementation that should be corrected in future versions.

These evaluations demonstrated that visual scheduling is beneficial when scheduling small group meetings. Both assigning priorities to events and access to scheduling decision reasoning were considered advantageous by users. At this time, VS is regularly used to schedule meetings and over 75 "standing" schedules are stored online.

We first describe how people use calendars and schedule meetings, and briefly review on-line automatic schedulers. Then we present the visual scheduling system VS along with a discussion of why we think it is effective. Next we detail the controlled experiment and the field study. Finally we describe our conclusions and plans for future work.

2. BACKGROUND

2.1 The Scheduling Problem

How office workers keep – or "should" keep – their personal calendars is the subject of a great deal of study ([MacKenzie76], [Kelly82], [Kincade85], [Malone83]). While there are many individual differences, certain patterns are evident. Most office workers keep a calendar on their person. Quite often this calendar is very small, as its portability, and therefore its accessibility, is critical. A few individuals keep a desk calendar as well as a pocket version. They find a larger calendar's additional space advantageous, despite the maintenance problems two calendars present.

While portable pocket calendars are advantageous for individuals, they create problems for those trying to schedule group meetings. First, when scheduling a meeting, user need to view simultaneously each participant's schedule. Accessing individual pocket calendars may require many minutes if not hours. Second, accurate calendars are needed; often only inaccurate and outdated copies of individual calendars are available to others. Finally people have various priorities for the events on their schedules and it is difficult to evaluate the priority of another's event.

Both pocket and desk calendars are used for a variety of purposes. The calendar may be used as a schedule, that is, a reminder of future events and tasks. It may also be used as a diary recording past activities and expenses. Finally, both pocket and desk calendars serve as structured notebooks recording phone numbers, addresses, birthdays and other important dates, etc. While all these uses are important, this paper is only concerned with scheduling and maintenance of schedules.

2.2 On-Line Automatic Scheduling

One can locate a copy of everyone's calendar on-line so that the information can be accessed by everyone ([Greif87]). This approach does aid the scheduler in collecting individuals' schedules. But on-line schedules are stored in computers, not pockets, and many users are reluctant to maintain accurately their on-line calendars. Even with accurate on-line schedules, quickly locating acceptable meeting times is still a major problem for two reasons: rarely are there any mutually available time-slots, even for a small group meeting, and locating them manually is difficult and error prone.

Assuming correct on-line calendars are available for each individual, what sort of tool will help the scheduling-user find optimal meeting times? One could use a "black box" automatic-scheduler – either a rule-based "expert-system" ([Lehner87]) or a programmed equivalent – to locate a meeting time. With such a system, heuristic "scheduling rules" are developed in an attempt to analytically describe how the typical user schedules the typical meeting. By acting on these rules, such scheduling systems quickly produce scheduling solutions that often match the expectations and needs of the scheduling-user.

However, we argue that such "scheduling-rules" cannot really reflect the normal processes individuals use to schedule group meetings. Those processes are either too varying, or too sensitive to be encoded. Different people use different rules and situations vary widely, so it is difficult to develop a generally useful fixed set. For example many experts suggest scheduling certain critical meetings just before lunch or quitting time to insure a decision is quickly made ([MacKenzie76]), while other meetings require sufficient time for reflection. Or depending on the agenda, certain people may not be critical to a discussion and could be left out, simplifying the location of an acceptable meeting time.

Further, schedulers often may use "hidden rules" that they will not allow into common knowledge – which would likely happen if the rules were incorporated in an automatic-scheduler. For example, a manager might intentionally schedule a conflicting meeting in order to insure a disruptive person *can not* attend. A rule for this situation might read, "Choice between two otherwise equal time slots, by selecting the one that Jones cannot attend." Such a rule could not, realistically, be used with an automatic-scheduler.

In short, in the real world, schedulers schedule meetings using a wide variety of complex, constantly changing rules, many of which can not allow into common knowledge, and therefore, cannot place into a rule-based "black box", automatic-scheduler. What is needed is "grey box" (vs black box) solution, in which an on-line automatic-scheduler first calculates a set of possible solutions, including both any mutually available times slots and those with minimal conflict, then displays the results in a quickly assimilated manner, and finally, allows the scheduler to look behind the results to see how they were obtained.

3. THE VISUAL SCHEDULER

3.1 Visual Scheduling Metaphor

Consider the following metaphor: an individual's weekly calendar is a grid on an overhead transparency (Figure 1). Columns indicate days of the week, and rows represent hours. Events are darkened blocks with the event's priority denoted by the block's opacity.

One locates a mutually open time slot by stacking all the transparencies together, holding the stack up to a light, and looking for openings. However, as is often the case with even a small group of busy people, there may be no completely open slot. In this case, the least disruptive time slot should correspond, roughly, to the one admitting the most light.

The above scheduling approach not only locates and presents the optimal meeting time, but also, in the same quick-to-grasp presentation, both shows the relative priority of each composite time slot and allows the user to point to a time slot and acquire additional information about what users and what events are blocking it. Therefore, the user still is able to apply a variety of vague scheduling "rules" without being overwhelmed by detail.

With this metaphor, each potential meeting time can be thought of as a local minimum in a space of points representing time-slots and events. In this space, a point's value is the conjunction of a set of fuzzy-logic or weighted variables determined by the priorities of events at that point ([Zadeh65], [Zadeh73]). While using vision to search a large mathematical space is not common, it is not unknown. Since developing VS, we have learned about an information-retrieval system from the 1960's which used a somewhat similar approach ([Jonker67]). The Jonker system uses 8 1/2" X 11" cards, each representing a keyword or concept, to locate thousands of papers. Each card has positions for up to 10,000 papers. If a card contains a hole at a paper's position, then the card's corresponding keyword applies to that paper. One can quickly locate all the papers for which a group of keywords apply by superimposing the cards corresponding to the keywords, holding the stack to a light, looking for common holes, and finally locating the corresponding papers. Obviously, since a hole is either present or absent, this system, unlike VS, can only calculate the conjunction of Boolean, not fuzzy logic variables. *

* An interesting on-line graphic information retrieval system could be developed by com-

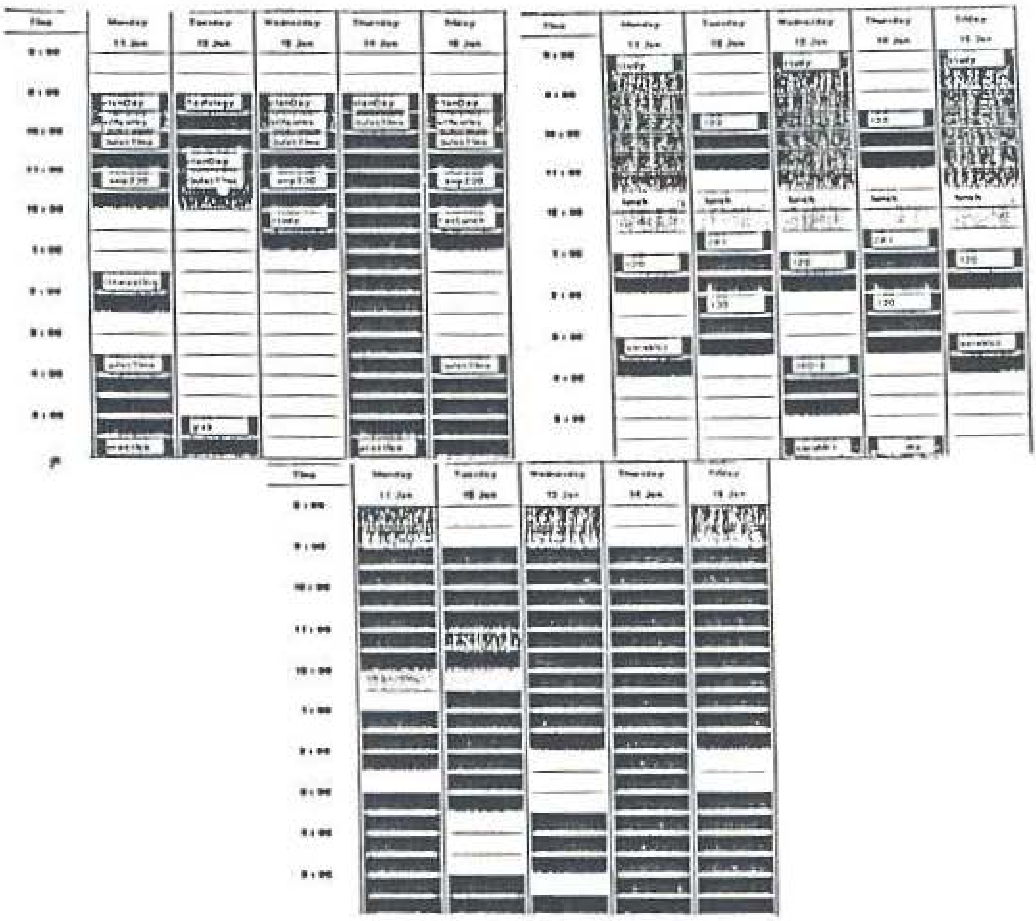

FIGURE ONE: Clockwise from upper right: two schedules and their combination

Metaphors or "mental models" are often used to help users quickly understand and accept a new computer system. Understanding any complex system is a mental process which involves developing expectations of what result will occur when some action is performed. By tailoring the actions and the "visual look" of a system to some existing system already understood by users, the system builder increases the likelihood that users will "understand" how the system works, and therefore will feel "comfortable" and "in charge" ([Olson87]).

3.2 System Description

The visual scheduler prototype VS implements the above visual scheduling analogy. It is written in Xerox SmallTalk and currently operates on networked Sun2 and Sun3 workstations running 4.3 Unix. The system requires either a networked or central file system connecting all users; current implementation uses a ".schedule" file, located in each user's home directory, to store that individual's schedule.

Figure two details the VS workstation screen. The window in the center of the screen

bining the on-line fuzzy-logic search of VS with the Jonker information retrieval method.

displays a user's weekly calendar showing the days of the week, and the time of day from 8:00 am until 5:00 pm. Calendar events are the darkened rectangles on the calendar. The event's physical location denotes its time and duration, while its "greyness" denotes priority. A label is provided to describe the event. The two fields below the schedule allow the user to change the week or person involved in a meeting or to view a particular person's schedule. A user may look at but not modify other users' calendars.

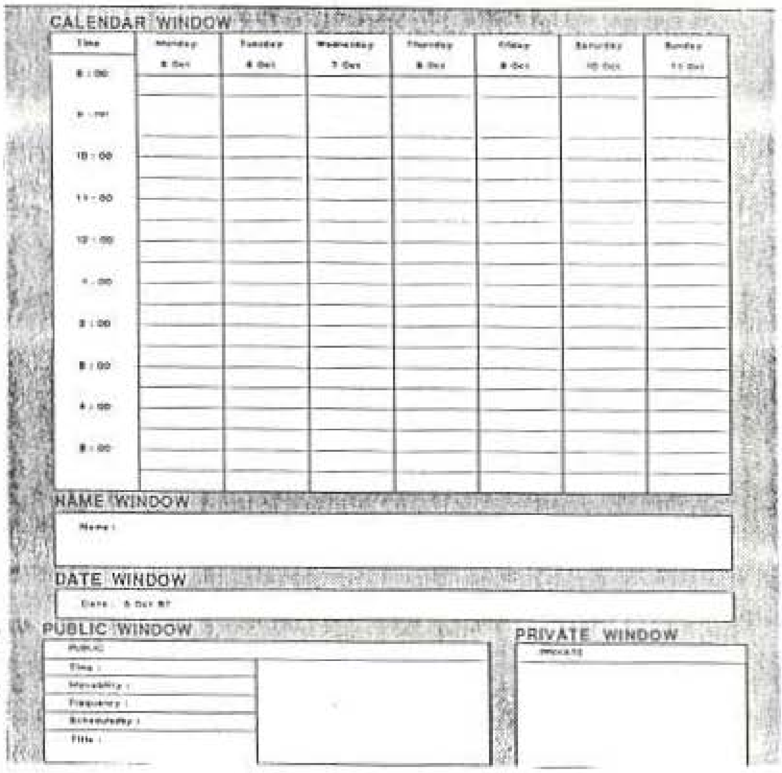

FIGURE TWO: Visual Scheduler display

The event description window, at the bottom of the screen, is used to describe a particular event selected by the user. The window's left hand column includes fields for the event's time and date, whether it repeats, who scheduled it, and its title or label. The center and right hand windows are for public and private general purpose text strings attached to the event. All fields except the private text window have general read permission.

Users can create not only one-time events, but also repeating or "standing" events. Examples of standing events include class-periods, weekly faculty lunch, project team meetings, office-hours, RA/faculty meetings, etc. Direct manipulation (see [Schneiderman83], [Brooks77]) is used to add, delete, move, resize, or copy events; functions available through pop-up menus. The system also provides context-dependent help.

If a user wants to view an individual's schedule, either another's or his own, he enters the individual's name into the name field and the system displays the corresponding calendar showing shaded events with their titles. To see the details of an event, the user "Clicks" on it with the mouse button.

To schedule a meeting, a user enters the participants' names into the name field and the system generates and displays the composite schedule. With a composite schedule, no event labels are displayed. However, "Clicking" on a time-slot results in the display of a list of those participants whos' schedulers have events during that time.

How VS combines event opacities to form the composite schedule – its *opacity algorithm* – is critical to its effectiveness. VS currently treats each event's priority as a percentage of transmitted light: A low priority event transmits 80%, a light grey; a high priority event transmits only 20%; and a maximum priority or black event transmits 0%. Events are superimposed by multiplying the transmitted-light percentages.

4. EVALUATION

We argue that VS is noteworthy for several reasons: it is superior to manual methods; a visual representation is useful for displaying a composite schedule; visually represented priority helps the user quickly locate meeting times; and finally, visual scheduling is useful, that is, actual users choose to use VS to schedule meetings.

In this section, we evaluate VS to determine whether the above assertions are correct. First, a controlled experiment examines whether automatic scheduling of VS is superior to manual scheduling. Then a field study considers, in particular, whether visually represented priority is helpful, and in general, whether the scheduler is a useful tool.

4.1 Experiment

We wished to determine whether automatic visual scheduling is superior to manual scheduling in terms of *total response time, error-free response time*, and *percent error*. For this experiment, all scheduling events were considered to have the same priority. Preliminary pilot tests indicated that considering priority when manually scheduling is too difficult to produce usable results.

People typically use text schedule descriptions, such as those found in a pocket calendar, rather than visual descriptions, when manually scheduling a meeting. Therefore, visual scheduling (VS) was compared to both manually scheduling with visual representations of each participant's schedule (visual manual (VM)), and manually scheduling with text representations (Text-manual (TM)). Example visual and text schedule representations are given in figures one and three.

An automatic visual-scheduling (VS) trial would start with subjects being given a list from four to seven meeting participants. They would manipulate VS to produce and display a composite schedule on the workstation and then indicate, on a blank-schedule answer-sheet, the mutually available time slots.

Participant#1:

Mon.	8-11:30, 1-2, 3-4
Tue.	9:30-11, 11:30-3:30
Wed.	8-12:30, 1-2, 3:30-5, 5:30-6
Thu.	9:30-11, 11:30-3:30, 5:30-6
Fri.	8-12:30, 1-2, 3-4

FIGURE THREE: Sample Schedule in *text manual* form

A manual scheduling trial was started with subjects being given the schedules for a set of meeting participants. With the visual-manual (VM) scheduling task, the 4" X 4" schedules were printed on separate sheets of paper and spread out in front of the users for simultaneous viewing. With text-manual (TM) scheduling, the schedules of all participants were typed on the same sheet of paper(see figure three). The subjects were instructed to indicate on a blank-schedule answer-sheet the mutually available time slots. Subjects were told they could place any marks they wished on the answer-sheet, as long as the mutually-available times slots were clearly indicated. The answer sheet thus was scratch paper that could serve to augment the subject's working memory.

Hypothesis: We predicted that VS would be faster than both TM and VM for total response-time, and error-free response-time. We also predicted that VS will also have a lower percent error than either VM or TM.

4.1.1 Method

The entire experiment was conducted in a room with controlled light and isolated from external noise. A large table placed in front of the subject contained a Sun3/180 workstation (keyboard, mouse, and display screen) used to run VS, and sufficient additional room for the visual manual and text manual tasks. A stop watch was used to time all events. A pilot study was conducted to debug the experimental procedure.

Subjects: Six Computer Science graduate students, four males, and two females, participated in the experiment. Computer science graduate students may seem inappropriate for evaluating a tool to be used by a general office staff. However, pilot evaluation suggested that the task of manually scheduling meetings without errors is very difficult; Computer Science Graduate students were chosen to insure that a sufficient number of trials would be correctly completed. All subjects were familiar with VS.

Procedure: On being admitted to the experiment laboratory, subjects were briefed on the purpose of the experiment and their roles. They were informed that their participation was voluntary and that all results would be kept confidential. Each subject was instructed to work as quickly as possible without making errors. They were told they could write anything they wished on the blank-schedule answer-sheet as long as they clearly indicated the mutually available time slots for the meeting participants. After this introduction, each subject scheduled meetings using sets from four to seven individuals' schedules. The experimenter recorded verbal protocol and observed subjects to learn what manual-scheduling procedure they used.

Design: The experiment was designed as a one-way repeated measures analysis of variants (ANOVAR) with three scheduling methods (text-manual (TM), visual-manual (VM), and visual-scheduler(VS)). Each subject scheduled six meetings for each method. Each subject's score for each method was the average of the six meetings. Order of the methods of scheduling was counterbalanced across subjects. Each of a subject's 18 meeting-scheduling trials used a different set of meeting participants. Each of these sets was scheduled once by each subject, ordered (Latin square) to insure the sets of subjects did not affect the results.

The independent variable was the scheduling method. Dependent variables were total trial time, error-free time, and percent error. A meeting trial answer was graded "incorrect" if it had one or more mistakes. Separate analyses were conducted to determine the effect of scheduling method on total response time, percent correct, and error-free response time.

4.1.2 Results

Total Time: The effect of scheduling method on total time was significant, CF(3,15)= 16.8, $p < .01$. Post-hoc testing using the Newman-Keuls [Winer62] procedure (p=.01), revealed that AS was faster than both VM and TM, which did not significantly differ from each other. *Percent Error:* The effect of scheduling method on percent error was significant, CF(3,15)= 9.1, $p < .01$. Newman-Keul's (p=.01) revealed that AS resulted in fewer incorrectly scheduled meetings than both VM and TM. VM and TM did not significantly differ from each other. *Error Free Response Time:* The effect of scheduling method on error-free response-time was significant, CF(3,15)= 15.5, $p< .01$. Newman-Keul's revealed that AS was faster than both VM and TM (p=.01), and VM was faster than TM (p=.05).

All six subjects successfully completed the experiment. However, the subjects had difficulty determining the exact time for an event on the visual-scheduling forms as well as on VS's display screen. Both protocol and observation indicate that it was easy to misread an event's time on the current visual schedule and misplace an event on the blank schedule.

With the visual and text manual tasks, four of the subjects independently developed the following "complement" algorithm to schedule meetings: first the subjects systematically marked the *unavailable* times for each participant on the blank-schedule answer sheet. Then, when finished, they circled or otherwise indicated the mutually available time slots, that is, those time slots on the blank-schedule answer-sheet that previously had not been marked unavailable. Verbal protocol indicated that every subject realized that manual scheduling was quite difficult and that a systematic method was required.

Subjects five and six developed an interesting method for manual-scheduling with visual schedules, apparently based on their exposure to VS. Both subjects realized that the time-consuming and error-prone aspect of the visual manual scheduling-task was transferring each participant's schedule events to the blank schedule. They both developed the following method for transferring events: For each participant's schedule they first superimposed the schedule on top of the blank answer sheet schedule, then held the combination up to the light, next marked the unavailable time-slots on the answer sheet where corresponding events showed through, and finally circled the available time slots, indicated by the absence of any unavailable markings.

4.1.3 Discussion

The automatic visual scheduling produced fewer errors (all six subjects), and required less time, both total and error-free. Additionally, the visual manual treatment had shorter error-free response time than the text manual method. After examining individual subjects' results, we feel it is due to the unexpected method used by subjects 5 and 6.

We believe these experimental results can be extrapolated to larger meetings and provide the following argument: Analyzing the "complement algorithm" used by most subjects indicates that it is monotonically increasing with the number of meeting participants; so we would expect that more participants would require the subject to take more time. On the other hand, VS should take about the same amount of time regardless of the number of participants, so we would not expect more time to be needed for larger groups. In fact, we might expect that with more participants, subjects would take less time to complete our scheduling task, because more participants would, on average, have fewer mutually available time slots needing to be transfered to the answer sheet.

4.2 Field Evaluation

While controlled experiments can produce clear, detailed answers to narrow, precise questions, field studies can consider the entire system in a real setting with real users doing real work. The purpose of the field study was to determine how VS was used, how well a visual representation of a composite schedule worked, whether visually-represented priority helped locate acceptable meeting times, and finally whether VS actually was useful for real work. We also wanted to locate problems with the current implementation and determine how they could be corrected.

The study was conducted for six weeks during the fall term 1987 at the UNC-CH computer science department. For the study, we focused on scheduling activity by small research teams of graduate students, faculty, and research associates. Interviews, a focus group, and limited user observation were used to gather data.

4.2.1 Method

4.2.1.1 Environment

Several aspects of the UNC-CH Computer Science department make it a powerful environment in which to examine office automation tools. First, our facilities include a state-of-the-art communication plant; offices have access to a variety of communication media including Ethernet and RGBS. Second, faculty, staff, and graduate students have access to more than 80 high performance workstations. Finally, electronic mail (e-mail) and electronic news are the predominant means of communication in the department; homework and class assignments are both given out and turned in with e-mail, which is also used to "task" graduate students, researchers and secretaries; a sizable portion of the department's scheduled-event opportunities arrive through either e-mail or electronic news.

4.2.1.2 Motivation

Too much time is needed to access VS –implemented in the SmallTalk rapid prototyping language – because SmallTalk does not function within the Suntools windowing environment used in the UNC-CH Computer Science department. To use VS, users have to destroy their window work-environment, bring up VS, interact with the displayed calendar, return to the window system, and try to rebuild their original window work-environment, a process that can take several minutes. Pilot analysis indicated that this was too slow for VS to serve as a user's primary calendar. Therefore, we decided to focus on standing schedules.

There are good reasons for thinking standing schedules may be viable in a university department, despite not containing all the scheduled events of meeting participants. Graduate students' standing weekly schedules comprise a large majority of their scheduled Monday-Friday events. Students and faculty have classes, office hours, standing research meetings, etc., which reflect their highly structured working days. One faculty member's weekly calendar is 80% filled with standing events alone (8am - 6pm). Since users' standing schedules vary little during a semester, little work is needed to maintain their accuracy. These observations may not be applicable to all office settings.

While we did not expect busy faculty to participate in the field study, their presence is critical to oral examinations, faculty reviews, and research teams. Therefore the standing schedules for all the faculty members were obtained and entered into the system by the VS system staff.

To motivate student use, we provided bait – a tool which prints a desirable copy of the user's standing schedule on a laser printer, complete with priority-shaded events. We previously observed that many students would laboriously draw a copy of their standing schedule for frequent reference. Many graduate students who had no need to schedule meetings entered their standing schedule solely to obtain a "pretty" hard copy.

Placing prototype systems into field use is risky, because of poor documentation, interface design problems, lack of training, and the potential for many bugs. Therefore, we added an on-line context-dependent help system, an on-line manual page, and a maintenance staff to answer questions and fix critical bugs.

4.2.1.3 Data Collection

The primary method for gathering data was the interview. After VS had been in the field for four weeks, an announcement was made asking users to meet with the authors and discuss their scheduling experiences. The purpose of the interview was to develop answers to several questions: first, what sort of meetings were scheduled? Were they standing or one-time meetings, and how many participants were to attend? Second, what method did schedulers use to locate an acceptable meeting time, and in particular, how often did the users look behind the results of VS to determine which participants were blocking a time slot with what events? This is a critical question, because the need to look at the users and events blocking a time slot indicates, to some extent, the need for more information than is typically provided by an automatic scheduler.

We were also interested in the merit of assigning a priority to an event and the visual display of the composite schedule. Additionally, we asked users to list problems with VS needing correction. Finally, we asked whether VS was useful, that is, did it help the users work, and would they continue to use it? We observed several users to verify the results of the interviews.

After finishing the individual interviews we conducted a focus group. A focus group is a meeting that interviews users as a group. A combined interview, as opposed to separate interviews, allows users to hear and stimulate each other. This is important because it helps generate lists of design problems and possible solutions. While we asked the same questions as in the interviews, we focused on problems with the system and how the users felt they might be corrected.

To develop an indication of a standing schedule's accuracy, we also asked several users to compare, for a given week, their standing schedule to their pocket one. Additionally, data on how many people used VS was gathered automatically.

4.2.2 Results and Discussion

Schedulers reported that the system helped them schedule meetings. They felt that the event priority rating was particularly useful. They particularly appreciated being able to see not only the open time slots, but also other possibly open time slots, represented by various grey levels, and second, being able to look behind the automatically generated results to determine which meeting participants were blocking a time slot with what events at what priority.

More than 55 students input their standing schedules into the system, so that, along with the 20 faculty schedules entered by the VS systems staff, over 75 schedules were available. Many people used VS to schedule meetings during the field evaluation and continuing after the study was completed.

We also gathered data on the accuracy and completeness of the standing schedules stored on-line in VS. Six individual's pocket calendars were compared to the standing schedules in VS to determine the usefulness of maintaining only standing schedules on-line. These pocket calendars contained, on average, 2.2 hours per week of additional events not included in the individual's standing schedules. While this is a small sample, we believe this indicates that, for the environment of a university graduate computer science department, standing schedules are sufficient for scheduling meetings. Comments from users seemed to confirm this. They felt that a "critical mass" of standing schedules was available during and after the field study. While they may not have always been able to locate all the schedules of the participants, locating several possible time slots for part of a group was most often sufficient for the whole. One possible explanation might be that due to shared class periods and the like, there is a great deal of overlap among the standing schedules in an academic department,

Over 15 people used VS to schedule meetings during the field evaluation period. Typical scheduling tasks ranged from scheduling a student project meeting involving three participants, to trying to locate a time when all the graduate students could meet and discuss an important issue, to locating standing-meeting times for a large research group. For many of these scheduling sessions, there were no mutually open time slots available.

In general, users reported scheduling meetings as follows: first, they would generate a composite schedule of all meeting participants who had on-line schedules. Often there would be no mutually-open time-slots. Then users would focus on the light-grey time slots displayed on the composite schedule. They asked questions such as: "Who was blocking a possible time slot?, Did that participant have to be at the meeting?, What was the blocking person doing during the time slot in question? and could it be moved or skipped? Did the blocking person overrate the priority of a blocking event?" When one or two possible meeting times were located, scheduling-users would send electronic mail to all the meeting participants asking whether the time-slot choices were acceptable.

Developing a set of rules or a program to implement the above scheduling method would be quite difficult, as it depends on the particular meeting, the personality and roles of the meeting's individual participants, and the nature of the blocking events. This seems to give weight to our notion that users need to be able to see behind the automatically-generated results and understand the composite schedule.

Despite finding VS generally useful, there were several problems reported by the users. First, "priority" has two possible meanings: How *important* an event is, and hard would it be to *move* it? A meeting between a student and his facility advisor is very important, but can often be moved. On the other hand a lower priority meeting between six faculty members may not be important, but would be very difficult to reschedule. The scheduler wants to know how much trouble it would be to move an event. To this end, we use the label "movability" instead of "priority". Despite this, users often rated a meeting by its importance, rather than its movability. Second, the method used by VS to combine event priorities into a composite priority is also a problem. The current algorithm produces too dark a time slot when combining several low priority, highly movable events.

Third, people often rated the priority of an event too highly. This seemed to increase the number of occasions when schedulers had to look behind a time slot's visual combination to see who had events then. Fourth, while schedulers could point to a time slot on a composite schedule and see a list of the users blocking the time slot, they reported that that they also would like to know the title and priority of each event blocking a time slot. One scheduler went to the trouble of displaying the calendar of each blocking user to help in determining whether that time slot could be made available. Finally, all the users stated that they wanted VS to execute under the workstation windowing software so that their schedule could be instantly accessed without disturbing their other workstation tasks. They also suggested that future versions be integrated with electronic mail, electronic news, and any on-line personal information system in use ([Malone87]).

5. CONCLUSIONS AND FUTURE WORK

Visual scheduling is clearly superior to manual scheduling based on both the results of the controlled experiment and the field evaluation. Further, VS appears to be a strong alternative to typical computer-based schedulers because events have priority, users can quickly "see" and understand the composite result-schedule, and because users feel they can look behind the automatically-generated results, understand how the scheduling decisions are made, and locate possible alternate times. Users found VS helpful in doing their scheduling work, and both individuals and small research groups continue to use the system. Incompatibility with the department's windowing environment precluded VS's use as a personal calendar, but a critical mass of over 75 standing schedules were entered and over 15 people scheduled meetings with the system.

Observation of individuals scheduling meetings indicated that it was extremely useful to be able to examine *why* a composite time slot was unavailable, that is, which individuals were unavailable, for what reason, and at what priority. Future versions of VS should have two additional features to aid in this effort: first, the ability to display quickly the names, event labels, and event priorities of individual calendar events for a given time slot on the composite schedule; and second, the ability to quickly point to and display the schedule for a participant. While one can perform all these tasks with the current system, it is too slow really to benefit the scheduling process.

Users told us that event owners often placed too high a priority on an event, as compared to their willingness to later allow its usurpation. We suspect that either, at the time the event was created, the user mistakenly believed it was of higher priority, or that for security or political reasons, the event was assigned too high a priority. For example, students always gave computer science class periods the maximum priority, even if they often skipped those same lectures. One partial solution to this problem would be to label the middle of the priority range "normal", thus encouraging users to pick these lower priorities . We feel however, that the best solution is to allow schedulers to examine easily the labels of events blocking a time slot, and determine for themselves the priority by the event's title. Schedulers can then make a decision and poll all the meeting participants for their agreement or comments. No matter what the scheduling algorithm, users will insist on having sufficient data to make their own decisions.

When scheduling a meeting, the "darkness" of a combined event should indicate how difficult it would be to usurp that time slot. But the current opacity-algorithm reaches "black" too quickly by over-rating the priority of a time slot occupied by many low priority events. If five individuals have overlapping events each with only the lowest priority, the combined schedule would still display that time slot as very dark or "unavailable". We expect that an algorithm which is more a function of the maximum priority and less a function of the total number of participant events would be superior.

We wish to thank Professors Coggins and Brooks for their assistance in developing the prototype, for the support of the faculty, staff, and students of UNC-CH Computer Science Department in its evaluation, and for Dr. Mantei's helpful critique of the manuscript.

REFERENCES

[Brooks77] Brooks, F.P., "The Computer 'Scientist' as Toolsmith - Studies in Interactive Computer Graphics",*Information Processing* 77, B. Gilchrist, ED. IFIP North-Holland Pub. Co. 1977.

[Greif87] Grief, I. and Sarin, S., "Data Sharing in Group Work", *ACM Transactions on Office Information Systems*, Vol. 5, No. 2, Apr. 1987, pg. 187-211.

[Jonker67] *The Jonker Information Systems Manual* , Jonker Corporation, Gaithersburg, Maryland, 1967.

[Kelly82], Kelly,J.F. and Chapanis,A., "How Professional Personnel Keep Their Calendars: Implications for Computerization,", *Journal of Occupational Psychology*, 1982, 55, 241-256

[Kincade85] Kincaid,C.M., Dupont,P.B., Kaye,A.R., "Electronic Calendars in the Office: An Assessment of User Needs and Current Technology,"*ACM Transactions on Office Information Systems*, Vol. 3, no. 1, Jan. 1985, pog. 89-102.

[LehnerR87] Lehner, P.E., and Zirk, D.A. "Cognitive Factors in User/Expert-System Interaction",*Human Factors*, 1987, 29(1), pg. 97-109.

[MacKenzie76] MacKenzie, R.A., *Time Trap: How to get more done in less time*, MacGraw-Hill, 1976

[Malone87] Malone,T.W., Grant,K.R., Lai, K.Y., Rao, R., Rosenblitt,D. "Semi Structured Messages are Surprisingly Useful for Computer-Supported Coordination," *ACM transactions on Office Information Systems,* Vol. 5 No. 2, pp. 115-131 ,April 1987.

[Malone83] Malone, T.W., "How People use Office Systems", *ACM Transactions on Office Systems*, Vol. 1, No. 1, pp. 99-112. January 1983,

[Olson87] Olson, J.R., Carrol, J.M., and Anderson, N., "What does the User of Software Know? Mental Model and Software Human Factors", *Proceedings of the Workshop on Software Human Factors: Users Mental Models*, (Available from: Committee on Human Factors, National Research Console, 2101 Constitution Avenue, NW, Washington, DC 20418)

[Schneiderman83] Schneiderman, B., Direct Manipulation: A Step Beyond Programming, IEEE *Computer*, Vol. 16, pp 57-69, 1983.

[Winer62] Winer, B.J., *Statistical Principles in Experimental Design* McGraw-Hill, NYNY, 2ndEd., 1962

[Zadeh65] Zadeh, L.A., "Fuzzy Sets," *Information Control,* Vol. 8 pp. 338-353, 1965

[Zadeh65] Zadeh, L.A., "Outline of a New approach to the Analysis of Complex Systems and Decision-Making Approach," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SME-3, No. 1, pp. 28-45, January 1973.