Pers Ubiquit Comput (2008) 12:255–267
DOI 10.1007/s00779-007-0154-3

ORIGINAL ARTICLE

Using job-shop scheduling tasks for evaluating collocated collaboration

Desney S. Tan · Darren Gergle · Regan Mandryk · Kori Inkpen · Melanie Kellar · Kirstie Hawkey · Mary Czerwinski

Received: 10 February 2006 / Accepted: 14 October 2006 / Published online: 24 February 2007

Abstract Researchers have begun to explore tools that allow multiple users to collaborate across multiple devices in collocated environments. These tools often allow users to simultaneously place and interact with information on shared displays. Unfortunately, there is a lack of experimental tasks to evaluate the effectiveness of these tools for information coordination in such scenarios. In this article, we introduce job-shop scheduling as a task that could be used to evaluate systems and interactions within computer-supported collaboration environments. We describe properties that make the task useful, as well as evaluation measures that may be used. We also present two experiments as case studies to illustrate the breadth of scenarios in which this task may be applied. The first experiment shows the differences when users interact with different communicative gesturing schemes, while the second demonstrates the benefits of shared visual information on large displays. We close by discussing the general applicability of the tasks.

Keywords Job-shop scheduling task · Evaluation · Collocated environments · Computer-supported collaborative work · User study

1 Introduction

As computing technologies move off the desktop and into the everyday world, methods for examining group interactions in computer-supported collaboration environments are becoming ever more essential [1]. Existing evaluation methods for examining group interaction consist of methodologies that range from ethnographic methods for observing users in real-world environments to laboratory experiments that examine particular attributes of group interaction in a controlled setting.

Many computer-supported collaborative work researchers have chosen the ethnographic approach. Ethnographic methodologies typically emphasize users' points of view and experiences rather than aggregate measures of performance. Through subjective interpretation of observational data, researchers obtain a rich picture of how technologies are adopted in the real world. For example, Heath et al. [2] describe sociological workplace studies concerned with work, technology, and interaction in organizational environments. They discuss how these studies can provide deeper understanding of technology within fields such as cognitive science, computer-supported collaborative work, and human-computer interaction. While the ethnographic

D. S. Tan (✉) · M. Czerwinski
Microsoft Research, One Microsoft Way,
Redmond, WA 98033, USA
e-mail: desney@microsoft.com

M. Czerwinski
e-mail: marycz@microsoft.com

D. Gergle
Northwestern University, Evanston, IL 60208, USA
e-mail: dgergle@northwestern.edu

R. Mandryk · K. Inkpen · M. Kellar · K. Hawkey
Dalhousie University, Halifax, NS B3H 1W5, Canada

R. Mandryk
e-mail: regan@cs.dal.ca

K. Inkpen
e-mail: inkpen@cs.dal.ca

M. Kellar
e-mail: melanie@cs.dal.ca

K. Hawkey
e-mail: hawkey@cs.dal.ca

approach is essential to gleaning a real world understanding of collaborative work, it is often difficult to include the tight control of variables that laboratory studies can provide.

Hence, other researchers have chosen to employ laboratory experiments in which tasks and variables are carefully crafted and controlled. This allows researchers to study specific phenomena of interest. Designed properly, laboratory experiments can eliminate alternate explanations to observed phenomena and allow for much stronger causal interpretability of the results. They are also generally more replicable than in-situ observations. However, results from poorly designed experiments run the risk of missing important contextual variables and drawing incorrect conclusions.

We share Dourish's [3] belief that the two paradigms provide researchers with complementary information and when wielded accordingly, can be used to optimally understand the role of technology in collaboration settings. Unfortunately, while ethnographic methods are well understood, there is currently a shortage of useful experimental paradigms for evaluating collaborative technologies. In a recent flurry of workshops and papers, researchers have begun work to address this shortage (e.g. [4, 5]). In particular, it would be useful if common tasks could be established to evaluate computer-supported collaborative environments, thereby allowing the community to iterate upon and refine appropriate tasks and metrics. Utilizing a consistent task across studies will help provide validity to the task, and will allow researchers to better assess the significance of their results in comparison to other work.

While we focus our current work on filling this gap in experimental tasks for laboratory experiments, we utilize certain methods that traditionally fall closer to the qualitative tradition, such as inspecting conversation transcripts and interpreting the information that lies therein. In designing our tasks, we have also tried to maintain sensitivity to external validity, ensuring that the tasks are somewhat representative of real-world tasks, so that generalizing these tasks does not drastically alter any experimental laboratory findings.

In this paper, we propose job-shop scheduling as a general task that can be used to evaluate computer-supported collaborative environments. In addition, we present metrics that can be utilized, as well as show how the task and methodology can be adapted to specific research objectives.

1.1 Task framework

In our work, we use the eight class types described in McGrath's [6] task taxonomy to provide a conceptual framework that facilitates discussion of existing experimental paradigms. Working within this framework, we uncover a design space that indicates the need for developing classes of tasks to evaluate collaborative technologies.

In our literature review, we have found that generative tasks aimed at examining group *planning* and *creativity* are fairly well represented with existing experimental paradigms. These tasks assess mediated group performance by focusing on the generation of ideas and plans, as well as the subsequent selection and execution of chosen alternatives. Examples include the automated post office design task [7] and furniture layout tasks [8].

Similarly, the number of executable tasks such as *contests/battles* or *physical performance* tasks has seen recent growth. These task areas are traditionally viewed as those that involve physical behavior, as opposed to symbolic, mental, or verbal manipulations. In a computational world, the pipe construction task [9], the collaborative Lego construction task [10], and collaborative puzzle construction task [11] are all representative of this category.

Many negotiation tasks have also been explored. Researchers have examined *mixed-motive* tasks, which generate tension between individual and collective rationality, across a range of technologically mediated environments [12]. These tasks include bargaining, coalition formation, or game theoretic social dilemmas. *Cognitive conflict* tasks are another type of negotiation task, except that the conflict resides in participant viewpoint rather than participant interest. An example is the desert survival task, recently used by Setlock et al. [13].

The final group of tasks are *intellective* and *decision-making* tasks. These tasks involve problem solving with demonstrably correct answers or consensually agreed upon solutions. Most of these tasks, such as logic problems, are useful for eliciting group discussion and negotiation, but are typically designed such that information resides solely in an individual's head and only becomes group knowledge through discourse. Other tasks focus solely on shared physical objects and their manipulation without the group negotiation and decision making component. While many collaborative systems assume that supporting interaction and communication with shared physical or visual objects is useful, there are surprisingly few tasks that can be directly applied to investigate these tools. In our review of the literature, we found that there exists a major gap in this task space, which we attempt to fill with our work.

1.2 Our contribution

The primary contribution of this article is the introduction and validation of the job-shop scheduling paradigm as a useful task for evaluating coordination tools in collocated collaborative environments. Additionally, we present a variation of the traditional job-shop scheduling task, which we call distributed job-shop scheduling, in order to broaden its applicability. This task helps fill the void found in current tasks used for exploring group problem solving with visual information. We describe properties that make the tasks useful, as well as evaluation metrics that can be used.

We also describe two experiments that illustrate analysis methods and demonstrate feasibility of this task for evaluating collaborative tools. We picked experiments in which the outcomes were somewhat intuitive so that we would not be distracted by surprising findings while trying to validate the usefulness and sensitivity of the task. However, the results are interesting and form a significant secondary contribution. These results highlight the benefits of software-based gesture support as well as concurrent viewing of shared visual information in coordination tasks.

2 Job-shop scheduling

2.1 Traditional and distributed tasks

The traditional job-shop scheduling task consists of a finite set of jobs, each composed of a chain of ordered operations. For example, in Fig. 1, the jobs are uniquely indicated by color, and the ordered operations are shown as blocks containing sequential letters. Each operation must be performed in strict sequence and cannot begin until the previous one has been completed. Furthermore, operations must be performed in a single uninterrupted period of time using a specific resource (e.g., a shop tool such as a 'hammer'). There exists a finite set of these resources, and each resource is capable of performing no more than one operation at a time. In other words, operations cannot temporally overlap on a given resource.

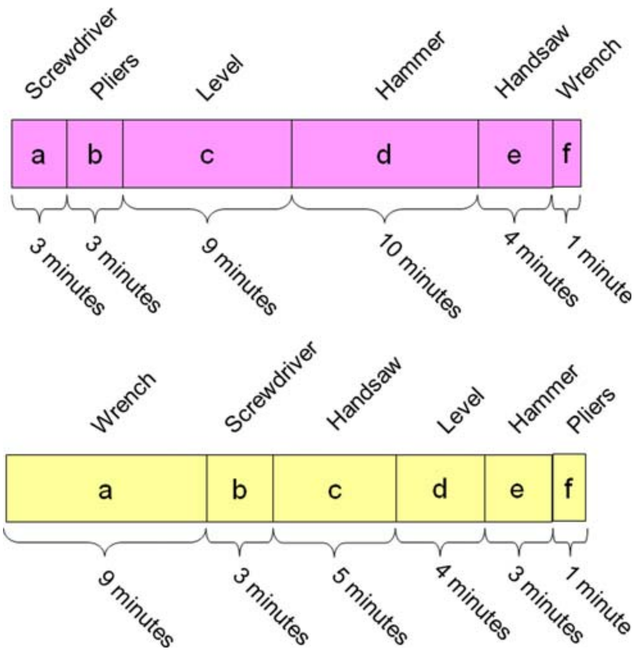

Fig. 1 Two example jobs. Each job (*color*) comprises strictly ordered operations (*blocks*) requiring specific resources (*shop tool*) for some time

To solve the task, the user must schedule all operations while observing the operation ordering and overlapping constraints. An optimal solution is one in which the last operation is completed at the earliest possible time (for an example of an optimal solution to a problem set, see Fig. 2). For a more detailed review of scheduling problems as well as computer-based solution techniques, see [14].

The traditional job-shop scheduling task can be used as a shared group activity that allows researchers to observe collocated collaboration. However, designers may wish to further test collaboration in scenarios where knowledge or access to the task is divided between users, and where information sharing is an intrinsic part of collaboration. To simulate these scenarios, we extend the current task into a distributed job-shop scheduling task by assigning each user in the group explicit responsibility and control over a subset of the jobs. This puts the users in a situation where they need to collaborate in order to integrate their information for the final joint solution. In this task, users are forced to coordinate scheduling operations using whatever shared resources are available in order to get all their jobs completed in the shortest amount of time for the group as a whole.

2.2 Useful properties

The job-shop scheduling task has several nice properties. First, it is simple to explain, easy to learn, and compelling

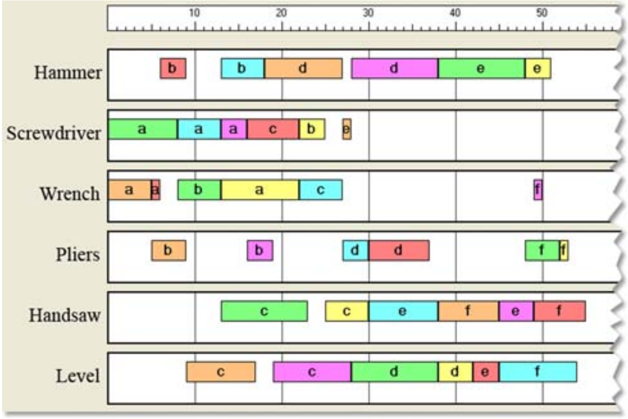

Fig. 2 Complete view of software used and optimal solution to the problem

to perform. Second, interaction with content contains many co-dependencies. In fact, rescheduling one operation typically requires having to move many others around it. This is important because it requires tightly integrated coordination even if users have access to all information. Third, since the task cannot be solved by a simple algorithmic strategy, iterative improvement and coordination are required. Finally, the task has an optimal solution, as well as other metrics that may be useful while evaluating group performance with novel collaboration systems.

The distributed job-shop scheduling task has an additional advantage. Its formulation is representative of many real world collaborative tasks in which information is distributed among group members and knowledge must be integrated in order to successfully formulate a joint solution. For example, collaborative resource planning and calendar scheduling have many of the same properties of this task.

2.3 Evaluation metrics

In order to get as complete a description of performance and process as possible, we devised four classes of evaluation metrics that can be used with the job-shop scheduling tasks. We describe these metrics in this section and, within our two case studies, demonstrate how they may be used.

Outcome measures of performance include the number of times blocks are moved (efficiency), the number of sequenced letters placed out of order (ordering error), the number of times any resource is scheduled to simultaneously perform more than one task (overlap error), and the degree to which the groups optimally schedule the group of jobs (measured by solution length). These metrics can be observed at various points while users perform the tasks in order to examine process and strategy, but should be measured when a solution is reached in order to derive useful performance measures.

Outcome measures of communication efficiency represent low-level communication mechanics that might affect task performance. One such group of measures is a relatively simple count of linguistic components. More complex measures such as discourse analysis using video and audio transcripts as well as log data representing patterns of tool usage can also be undertaken.

Process measures of communication, on the other hand, look at higher-level strategies used to solve the task. For example, we count communicative gestures, as well as look for verbal error correction strategies in our two studies. These measures require some amount of semantic interpretation. Furthermore, analysis can be done to explore social effects such as dominance and leadership, patterns of scheduling and submitting solutions, as well as general use of software tools and interfaces.

Finally, *self-report measures* using questionnaires, surveys, or interviews are useful for exploring such factors as the level of satisfaction with tools, the perceived distribution of contribution from various users, as well as overall interest in the task.

3 Case study 1: supporting communicative gestures

In this case study, we present an experiment aimed at illustrating the feasibility of the traditional job-shop scheduling task for examining how communicative gesture support affects collocated group interactions in scenarios where participants share complete information.

Software support for on-screen gesturing is not a new concept. In fact, researchers have explored a myriad of techniques (e.g., mouse trails [1], gesture traces [15], and spotlights [16]). However, many of these techniques have been designed for distributed, sometimes asynchronous collaborative environments and evaluations have largely aimed at measuring levels of activity awareness. In systems designed for collocated settings, gesturing techniques have been designed to allow a single presenter to direct audience focus more effectively.

There has been much less attention on multiple people gesturing in a shared visual space while engaged in collocated collaboration. This is a much richer environment in which users can make use of a wider range of cues such as verbal exchanges, physical gestures, and a variety of nonverbal coordination mechanisms. In this experiment, we examine the effects of simple software support for gesturing on a shared display while coordinating actions within the job-shop task.

3.1 Hypotheses

In this study, we chose conditions to see how participants would collaborate and communicate without any gesture support, and with the bare minimum gesture support. Since the gestures should help to reduce the ambiguity involved in referring to particular shared objects in the workspace, we expected that the gesture conditions would increase overall performance. Specifically,

> Hypothesis 1A: Groups will produce more optimal solutions (fewer errors and shorter solution length) on the scheduling task when gesture support is provided.

In addition to the basic outcome measures, we hypothesized that groups would adjust their communication processes to take advantage of the ability to remotely gesture in the space. Clark and Wilkes-Gibbs' [17] principle of least collaborative effort states that both speakers and listeners will attempt to minimize the effort they exert during a conversation as they establish a shared understanding. Thus, we expected,

> Hypothesis 1B: Groups will increase their use of efficient referring expressions as demonstrated through their increased use of deictic pronouns when gesture support is provided.

Finally, since virtual gestures are provided in the gesture support condition, we expected to see different levels of gesture use. Specifically,

> Hypothesis 1C: Groups will increase their use of gestures when provided with virtual gesture support.

3.2 Participants and setup

Thirty-six (12 females) university students or staff volunteered for the study in groups of three. The groups consisted of two all-female groups, four all-male groups, and six mixed gender groups. All participants used a computer daily and none had prior experience with the experimental software. Users were screened for color-blindness. The study took about 1 h and users were given a small gratuity for participating.

Users from each group sat at a table facing a large screen plasma display, which measured 54 by 30 inches and ran at a resolution of 1,024 × 768. Each user had a mouse, which they could use in one of the conditions to gesture at various visual components on the shared display. In both conditions, users could not directly manipulate objects on the screen. Instead, they verbally instructed a confederate driver, who sat off to the side and controlled the input mouse, when they wanted to reposition an object on the screen. The driver dragged bars representing each operation along a time line (interface was similar to that seen in Fig. 2). We used the same confederate throughout the experiment to ensure consistency in the interaction. This setup encouraged communication between group members and mimicked a common workplace scenario of a single user application being used in a collaborative setting.

3.3 Task

We created several job-shop scheduling tests consisting of six resources and six jobs, each with six operations, similar to the Fisher and Thompson [18] benchmark tests commonly used in validating online scheduling algorithms. For similar benchmark tests, see [19]. Two tests were used for the actual testing, and simplified tests (two jobs with six resources and three operations) were used in training exercises. Users had equal access and control of all operations and jobs and had to coordinate in order to find an agreeable best solution.

3.4 Manipulation and procedure

We examined collaboration under two gesture support conditions: No Support versus Mouse Cursors. In the No Support condition, users could only utilize physical gestures and verbalizations to communicate their ideas. In the Mouse Cursors condition, users could each gesture on the shared display using uniquely colored mouse cursors, in addition to the physical gestures and verbalizations. We chose this minimal augmentation rather than a more drastic manipulation to see if the collaborative task and metrics would be sensitive enough to distinguish between the two. The experiment was a within-groups design, with each group performing the tasks in both conditions. We counterbalanced the order in which groups saw the conditions, but kept the order of tasks constant so that they would be balanced across gesture support conditions.

In real world coordination scenarios, participants typically have a finite amount of time to negotiate the best possible solution. This solution may not be optimal. In fact, it may not even meet all constraints. To mimic this, we had users work for a fixed amount of time and measured the quality of solutions attained rather than trying to measure the time it took groups to obtain optimal solutions.

Prior to beginning the test, we explained the experimental task and had users independently practice a sample on desktop computers. Once they were comfortable with the task, we explained the gesture support for the first condition and had them perform a small group training task to familiarize them with the gestures and group dynamic, as well as the process of communicating with the driver. They then performed the test task in the first condition. Users had 15 min to find the best solution they could. They were verbally provided with 8-, 5-, and 1-min warnings so that they could keep track of remaining time. This procedure was repeated for the second condition. Users filled out a preference questionnaire after each condition, as well as a final questionnaire soliciting comments at the end of the experiment.

3.5 Results

3.5.1 Outcome measures: task performance

We analyzed the performance data using a 2 × 2 multivariate analysis of variance (MANOVA). Gesture Support (No Support vs. Mouse Cursor) was a within-subjects factor and order of presentation of condition was between-subjects. Our performance metrics included the number of overlap errors, the number of ordering errors, the total number of blocks moved, and solution length.

The MANOVA revealed no significant differences in task performance. Although average solution length tended to be shorter in the Mouse Cursor condition, and there tended to be fewer overlap errors in the Mouse Cursor condition than the No Support condition, these differences failed to reach significance. When we examined the data according to the order in which groups performed the conditions, we found ordering effects for the number of blocks moved ($F_{1,10} = 15.2$, $p \leq 0.003$, $\eta^2 = 0.61$). This result is summarized in Fig. 3. Post-hoc pairwise comparisons revealed that the groups who performed the Mouse Cursor condition first used significantly more block moves in the Mouse Cursor condition than in the No Support condition ($p \leq 0.002$). There were no differences in the number of blocks moved ($p \leq 0.203$) in each condition for those who started with no gesture support. This difference in collaboration efficiency is not reflected in the success of the solution. Making more moves may be reflective of a richer collaboration rather than a more efficient one. We explore this in more detail using the communication efficiency and process measures.

3.5.2 Outcome measures: communication efficiency

In order to examine how participants communicated verbally, we transcribed conversations from both conditions. Because responses from users in each group were likely to be correlated with one another, we analyzed the conversation at the group level. From the transcripts of the participants (excluding the confederate), we classified references to blocks and locations as absolute (e.g. ‘‘red A’’ or ‘‘time 20 on the hammer row’’) or deictic (e.g., ‘‘that block’’ or ‘‘put it here’’). In addition, we classified references to blocks as either initial (first reference) or follow-up references. We expected that access to a gesturing mouse would facilitate communication by allowing more use of deictic terms and references. Since there were no differences in the total number of words spoken between the two conditions (No Support: 1,586 words, Mouse Cursor: 1,540 words; $t = 0.47$, $p = 0.645$), we did not normalize the counts.

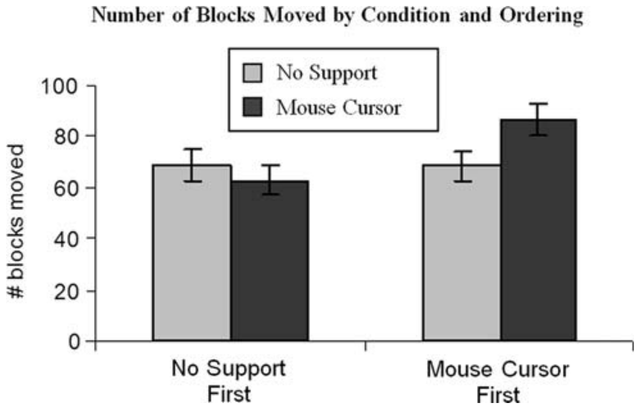

Fig. 3 Groups starting with the Mouse Cursor condition had fewer block moves when they switched to the No Support condition

We analyzed this data using a 2 × 2 MANOVA. The level of Gesture Support (No Support vs. Mouse Cursor) was a within-subjects factor, while order of presentation of condition was a between-subjects factor. Dependent measures included the number of initial block references, follow-up block references, total block references, location references, and the percentage of references for each of these categories that used deixis. Table 1 contains a summary of these results. We found a greater percentage of deictic references used to initially reference a block when users had access to a mouse. For follow-up references, the use of deictic references increased in both conditions to encompass almost half of the references, but did not significantly differ between conditions. None of the other measures differed between conditions.

Although there was no difference in the total number of location references, the percentage of location references that used deictic references was significantly greater in the Mouse Cursor condition than in the No Support condition. As expected, participants were able to utilize the mouse to facilitate their communication, especially when establishing the point of discussion, yielding a more efficient collaboration. We found no order effects on any of the dependent measures.

3.5.3 Process measures

We also examined how users utilized non-verbal communication channels in the two conditions. We expected that access to a simple gesture tool in the Mouse Cursor condition would enhance the use of non-verbal gestures to facilitate communication. From video recordings of the experimental sessions, we counted the number of times that each user made physical gestures as well as mouse cursor gestures.

In the No Support condition, there were a total of 615 physical gestures made, 577 (93.8%) for communicative purposes. Non-communicative gestures included affect displays, or body movements conveying emotional state, and beat gestures, or small movements in space that did not explicitly convey information about the task. Also, 78.9% of the physical gestures included the dominant hand.

In the Mouse Cursor condition, there were a total of 278 physical gestures, 258 (92.8%) for communicative purposes. Again, 78.0% of the physical gestures made included the dominant hand, meaning that participants removed their hand from the mouse to gesture. In addition to physical gestures, there were also 685 virtual gestures

Table 1 Mean block and location counts for both conditions

	No support mean (SE)	Mouse cursor mean (SE)	F	Significance	η^2
Initial block	175.4 (13.7)	191.1 (13.4)	1.908	0.197	0.160
Follow-up block	51.3 (6.3)	46.2 (5.8)	0.578	0.465	0.055
Total block	226.7 (16.2)	237.3 (16.8)	0.515	0.489	0.049
Initial using deixis (%)	2.2 (0.4)	10.2 (2.1)	15.237	0.003	0.604
Follow-up using deixis (%)	45.6 (6.4)	43.6 (6.1)	.118	0.738	0.012
Total location	73.2 (9.0)	83.3 (7.2)	1.751	0.215	0.149
Location using deixis (%)	33.7 (4.1)	49.0 (4.3)	22.089	0.001	0.688

Percentage of references using deixis was significantly greater in the mouse condition for initial block and location references

made with the mouse cursors, 18% of which were made concurrently with each other.

We analyzed the sum of the communicative physical and virtual gestures in a repeated-measures 2 × 2 MANOVA, with Gesture Support as a within-subjects factor and order of presentation of condition as a between-subjects factor. The MANOVA revealed that while there were significantly more communicative physical gestures made when the participants did not have a mouse cursor with which to communicate ($\text{mean}_{\text{mouse}} = 21.5$, SE = 6.9; $\text{mean}_{\text{control}} = 48.1$, SE = 8.4; $F_{1,11} = 14.8$, $p \leq .003$, $\eta^2 = 0.57$), there were significantly more total communicative gestures made when participants had access to a mouse cursor ($\text{mean}_{\text{mouse}} = 78.6$, SE = 9.1; $\text{mean}_{\text{control}} = 48.1$, SE = 8.4; $F_{1,11} = 23.6$, $p \leq 0.001$, $\eta^2 = 0.68$, see Fig. 4). There were no order effects on the amount of physical or virtual gesturing.

3.5.4 Self report measures

After each condition, users rated the gesture support in terms of how easy it was to accurately gesture, to physically gesture, to interpret their partners' gestures, and to communicate their ideas, each on a five-point scale (strongly disagree = 1, strongly agree = 5). We analyzed these ratings using a Wilcoxon signed ranks test across Gesture support conditions. We found that although the Mouse Support condition was rated higher on average for all four factors, the differences were not significant. See Table 2 for mean results and statistical analyses.

After the experimental session was complete, users ranked the conditions using the same four statements. We used a chi-squared test to determine whether there were differences in which condition was ranked first (see Fig. 5 for results). Although more participants preferred the Mouse Cursor for each of the factors, only the difference in ratings of accuracy reached statistical significance (Accuracy: $\chi^2(36) = 5.44$, $p \leq 0.020$; Physical: $\chi^2(36) = 0.44$, $p \leq 0.505$; Interpret: $\chi^2(36) = 1.00$, $p \leq 0.317$; Communicate: $\chi^2(36) = 0.11$, $p \leq 0.739$), and this difference is only marginally significant if correction for multiple tests is applied.

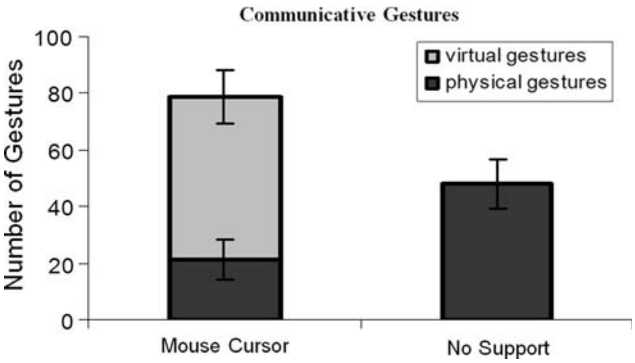

Fig. 4 Mean (±SE) number of gestures. Users used more physical gestures in the No Support condition, but overall gestures were higher in the Mouse Cursor condition

Table 2 Mean preference for each condition (including standard deviations)

	No support mean (SD)	Mouse cursor mean (SD)	Z	Significance
Accurate	3.9 (0.9)	4.2 (1.0)	1.64	0.102
Physical	3.7 (1.0)	3.8 (1.0)	0.86	0.392
Interpret	3.8 (0.9)	4.1 (0.9)	1.43	0.154
Communicate	4.1 (0.8)	4.3 (0.8)	1.15	0.251

Higher preferences are better

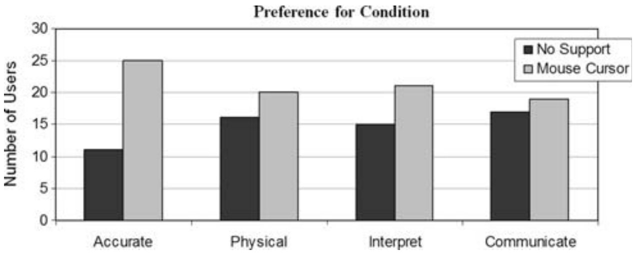

Fig. 5 Preference for the Mouse Cursor condition shown in the number of users who ranked either condition first for each of the four factors

3.6 Summary of case study 1

In this experiment, we used a traditional job-shop task as a shared activity in order to examine user behavior with and without virtual gesture support. Contrary to what we had expected, we found no statistical differences in our task performance measures between the two conditions. However, we did find that users took longer and moved more blocks in the Mouse Cursor condition, which could be interpreted as stemming from richer more engaged interaction. Additionally, users in the Mouse Cursor condition replaced many of their physical gestures with a more efficient coupling of virtual gestures and deictic references to communicate with each other. The participants used more efficient conversational expressions when virtual gestures were available, especially when referring to objects for the first time. This suggests that the gesture tools support initial reference and conversational grounding on the terms used to describe the shared objects.

4 Case study 2: shared visual information

In the second experiment, we focused on evaluating the effects of shared visual information and divided control in collocated collaboration. In this experiment we adapted the traditional job-shop scheduling task so that each user had access to different pieces of information and controlled a subset of the jobs.

We used the distributed job-shop scheduling task to examine differences in performance and communication across two methods of sharing display content in a collaborative scenario. In one method, commonly used in conference rooms and meetings today, users take turns projecting information on a large shared display. In the other, a method becoming increasingly available with new software tools, users simultaneously share visual information from multiple sources on the large display.

4.1 Hypotheses

Research has shown that people utilize shared visual information to support conversational grounding and task awareness [20, 21]. Hence, we expected that providing a method for groups to share their information in a centralized fashion would facilitate group performance. Specifically,

> Hypothesis 2A: Groups will produce more optimal solutions (fewer errors and shorter solution length) on the distributed job-shop scheduling task when multiple group members display information simultaneously.

Prior work has demonstrated that conversational efficiencies typically accompany the availability of shared visual information [10, 11, 22]. However, since shared visual information is available in both conditions in this experiment, we expected to see greater communicative efficiency primarily when the shared visual information was more salient as a conversational resource (i.e., when users were able to simultaneously share visual information). Hence,

> Hypothesis 2B: Groups will use more efficient communication techniques when they can simultaneously display shared visual information from all members of the group.

Finally, we hypothesized that the shared experience would cause users to rate this method more favorably. In fact,

> Hypothesis 2C: Members of the groups will find that simultaneously sharing information is more satisfying and more effective for coordinating information while performing the distributed job-shop scheduling task.

4.2 Participants and setup

Twenty-four (12 females) university students, aged 19–31 years old, volunteered for the study. All users spent more than 30 h a week using a computer, and none had prior experience with the experimental software. Participants were screened for color-blindness and were then divided into eight groups of three people each, with each group consisting either of all male or all female users. Users within each group did not know each other prior to the study. The study took about 1 h and users were paid a small gratuity for participating.

Users from each group sat at three desks each facing a large 95 inches wall-projected display. Each table was 11 inches away from the large display and the two side tables were about 25° off center on either side. See Fig. 6 for an illustration of the setup. Each user interacted through an IBM Thinkpad laptop and Microsoft Intellimouse placed on each of their tables. All users had an unobstructed view of other users and of the large display, but could not see each other's laptop displays. The laptops were connected over a local network to the desktop computer driving the large display.

4.3 Task

In this experiment, we used a job-shop task consisting of six resources and six jobs, each with six operations, as seen in Fig. 2. This test was taken directly from Fisher and Thompson's benchmark tests [18].

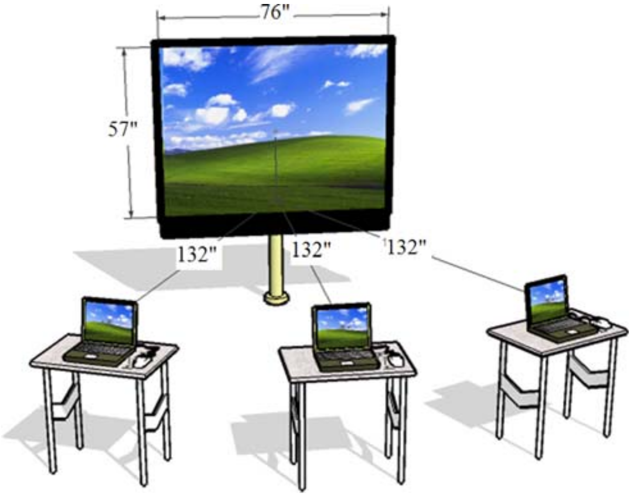

Fig. 6 Representative diagram of the experimental setup used in the two experiments. Measurements included are for study 2 only

Each of the three users was responsible for scheduling two of these jobs. Users had to coordinate schedules because they had to share the six resources available to perform the operations. We built a scheduling program that allowed users to adjust their schedules simply by dragging bars representing each operation along a time line (see Fig. 7 for an example of what each user saw on their personal laptop display).

4.4 Manipulation and procedure

After balancing for gender, we randomly assigned each group to one of two between-group conditions: Serial Presentation or Parallel Presentation. In both conditions, we used the WinCuts and Visitor systems [23] to replicate content that existed on the local laptop displays onto the large projected display, and to allow each user control over the two sets of displays.

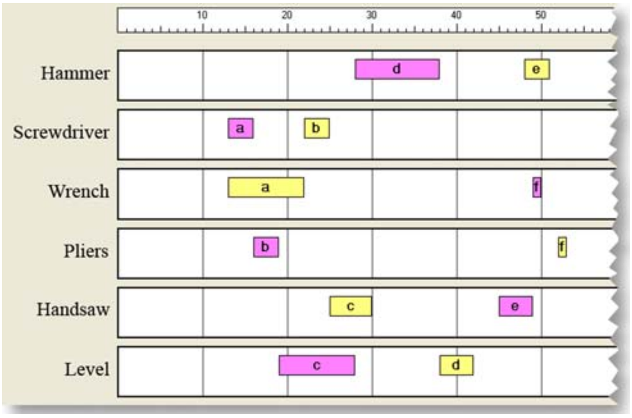

Fig. 7 View of one user's schedule. Each user was responsible for two different jobs

WinCuts is a system that allows users to replicate arbitrary regions of existing windows into independent windows. Each of these new windows is a live view of a region of the source window with which users can interact. Each window can be shared across multiple devices and hence used to send information to the shared display. Visitor is a system that redirects the input stream over the network so that a user can use a mouse and keyboard connected to one computer to control the input on another computer. We used Visitor to allow users to move their cursor off the top edge of their local laptop screen to control the cursor on the shared display. If multiple users did this simultaneously, they would 'fight' for control. Hence they had to socially mediate the use of the cursor on the large display. We saw no instances in which control collisions were not quickly resolved.

In the Serial Presentation condition, groups could see information from only one of the users on the large shared display at any given time. However, any user could move their cursor to the shared display, click on the taskbar there, and switch to a view of another user's solution space. In the Parallel Presentation condition, groups could simultaneously see information from all three users on the shared display. Using WinCuts, all three users could select relevant regions of their local content to send to the large display for simultaneous viewing. Furthermore, users could rescale and lay out content placed on the shared display. Although updates were seen on the large display in real time, users could only control their own content. It is interesting to note that all four groups in this condition decided to divide the display into thirds and to scale and vertically stack their information. This makes sense, since it allows simultaneous-viewing of the information in a way that best aided the task.

Before the test, we gave users written instructions and had them practice on a representative task for 5 min. Once they were comfortable, they had 20 min to work on the actual test. They were warned when they had 10, 5, and 1 min remaining. Following the test, users filled out a satisfaction questionnaire.

4.5 Results

4.5.1 Outcome measures: task performance

We analyzed the performance data using a 2 × 2 analysis of variance (ANOVA). The factors were Presentation Style (Serial vs. Parallel) and Group Gender (male vs. female). Overall we found no impact of group gender and therefore focus our reporting to the impact of presentation style in the following paragraphs. We examined two different performance metrics from the distributed job-shop scheduling task: number of overlap errors, and overall solution length.

We observed significantly fewer overlap errors with Parallel Presentation than with Serial Presentation ($F_{1,6} = 15.47$, $p = 0.007$, see Fig. 8). In other words, groups were more likely to complete the task with fewer errors when they had shared visual information made accessible using WinCuts, rather than having to keep some of this information in their memory or to continually communicate it verbally.

The second dependent variable we examined was possible solution length. While we found that groups using Parallel Presentation had solutions that were shorter (better) on average (73.5 vs. 78.5 units), the difference was not statistically significant ($F_{1,6} = 1.65$, $p = 0.25$).

4.5.2 Outcome measures: communication efficiency

While we found a statistical difference in the quality of the solution (i.e., the groups had less overlapping pieces in the parallel condition), this result tells us little about the communication and strategy used to solve the task. Since we hypothesized that performance improvements with Parallel Presentation would be partially due to increased communication efficiency, we expected lower word counts, lower utterance counts, and increased use of conversationally efficient linguistic references such as deictic pronouns [21].

In order to treat data as independent (belonging to the individual) even though they were correlated with actions within the group, we analyzed the data using the mixed model analysis technique described in Kenny et al. [24] to examine word and utterance counts. In this model, Presentation Style (Serial vs. Parallel) was a between group factor. However, because each individual score was not independent, each group (triads) was nested within Presentation Style and modeled as a random effect.

While the means tended to favor shared visual information across our communication efficiency measures (see Table 3), none of the models reached statistical significance. We found no evidence for a difference between the Parallel Presentation and Serial Presentation conditions ($F_{1,22} = 0.152$, $p = 0.71$). We believe this could be due to the noise inherent in such measures and to the small number of groups we observed.

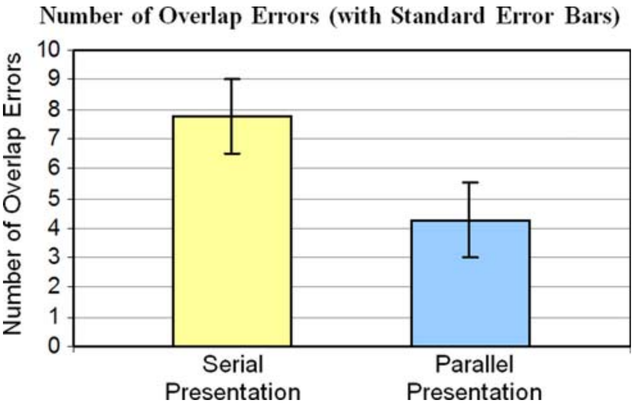

Fig. 8 Users made significantly fewer overlap errors using Parallel Presentation

4.5.3 Process measures

The following transcripts present examples of detailed assessments that can be made using this task in order to establish a deeper understanding of the ways in which the technologies affect performance. We expected groups to be more efficient and less error-prone with Parallel Presentation in which shared visual information was simultaneously available. A detailed exploration of the transcripts and logs seems to confirm this. For example:

Serial (Group 4S—querying)

3: Ok, um, I guess is anyone’s A longer than this?
1: Yea, I have one that’s for 8 min.
3: Ok uh.
2: I guess I’ll be able to move to A.
3: When does your end? When does your A end?
1: Oh mine? I’m sorry, 8.
3: Well my A for red is really small, I’ll show you guys. Do you have a bigger A?

Parallel (Group 0P—demonstrating)

2: Everyone put their A’s down. Move everything else away for now.
1: I’ll have to start this later than I would like to, but that is ok.

In the Serial excerpt, it is evident that when the groups are attempting to identify options for their ‘a’ operation, they use a rather inefficient method of querying one another and then waiting for a verbal response. However, in the Parallel excerpt, one of the users suggests pulling out all ‘a’ operations for everyone to see. This provides a shared visual resource that can be used for grounding subsequent conversation.

Additionally, the shared visual information provides less ambiguous information than linguistic descriptions. In fact, we observed several instances where errors that would have been caught in the Parallel Presentation condition were missed in the Serial Presentation one. For example:

Serial (Group 4S—undetected mistake)

2: We are always using it until 33
1: Yea ok that’s fine. Can you put yours after mine? (Error: 3 puts the operation at 36 instead of intended 33; the group moves on)

Parallel (Group 0P—detected mistake)

1: Oops, haha, move yours to start at the end of mine, at 35, no 36. (Error: 2 moves the incorrect operation to 36)
1: No no no, not the level, the hammer, yeah that.

4.5.4 Self report measures

Finally, we analyzed users' perceptions of satisfaction and their overall level of confidence with their final solution using five-point scale (strongly disagree = 1, strongly agree = 5). This analysis used the same mixed model analysis described in Sect. 4.5.2 in order to control for correlated responses within groups and replaced the dependent variable with the appropriate self-report metrics. The data showed that the Parallel condition [Mean (SE) = 3.92(.24)] was viewed as significantly more satisfying ($F_{1,22}$ = 10.62, p = 0.004) than the Serial condition [2.83(.24)]. The Parallel condition [3.42(.35)] was also considered marginally easier ($F_{1,22}$ = 2.88, p = 0.10) than the Serial condition [2.58(.35)]. These results are summarized in the lower portion of Table 3.

4.6 Summary of case study 2

In this experiment, we used our distributed job-shop task to examine user behavior when groups have various levels of shared visual information and control. We found, as expected, that the shared visual information led to better performance, as measured by fewer overlap errors. Results also suggest that the shared information may have led to more efficient communication, though these measures did not reach statistical significance. This is further supported by qualitative evidence extracted from process measures, as well as self-report measures suggesting that users were more satisfied and found the task easier to complete when they had shared visual information.

5 Discussion

Overall, the findings from these two studies suggest that our task provides a useful platform for investigating coordination in a wide range of collaboration scenarios and that the task and associated methodology can be adapted to best fit the research question being investigated. In this section we address the areas we believe the tasks worked well and describe some ways in which we believe they may be improved.

In order to evaluate the potential of our tasks, we decided to use rather stringent testing criteria. First, we used a relatively small number of groups (12 and 8 respectively, in the two studies). Since group studies are notoriously difficult to run and require greater resources than studies of individuals, we chose what we felt was a lower bound on an acceptable number of groups.

Using a within-subjects design in the first study and a between-subjects design in the second illustrates differences that exist between the two paradigms. The within-subjects design increases statistical power of comparisons since variance between groups can largely be accounted for in analysis. However, researchers should be careful to examine ordering, learning, and contamination effects when groups perform the task in more than one condition, as seen in the first study. Conversely, the between-subjects design has a strong advantage in that users are not contaminated by exposure to additional levels of the independent variable. However, doing so makes it rather difficult to find statistical differences between the groups unless the effect sizes are large and the individual differences are minimal.

Overall, our results suggest that the tasks are fairly sensitive at detecting differences on various dependent variables. For example, in the second study, we were able to obtain mean values of overlap errors that we could claim to be different with over 99% confidence. While the measure of overall solution length was not as sensitive at detecting differences, we feel that it may still be a practical measure to collect.

Given the small number of groups run in the second study and the fact that we were interested in assessing whether or not this measure might be of value in the future, we performed a power analysis to investigate its sensitivity. While caution must be taken when interpreting the findings

Table 3 General benefits of communication efficiency with Parallel Presentation (top). Users were significantly more satisfied ($F_{1,22}$ = 10.62, p = 0.004) and borderline more confident ($F_{1,22}$ = 2.88, p = 0.10) with Parallel Presentation (bottom)

	Mean utterances per individual (SE)	Mean words per individual (SE)	LS mean usage of diectic pronouns (SE)
Serial	113.82 (18.07)	753.83 (93.17)	104.18 (12.04)
Parallel	97.42 (18.07)	702.33 (93.17)	110.07 (12.04)
	Mean Satisfaction (SE)	**Mean Ease (SE)**	**Mean Confidence (SE)**
Serial	113.82 (18.07)	753.83 (93.17)	104.18 (12.04)
Parallel	97.42 (18.07)	702.33 (93.17)	110.07 (12.04)

of power calculations [25], we use it to guide future studies by generating a least significant number (LSN). The LSN indicates the number of observations expected to be needed in order to achieve significance given the existing (or expected) standard deviation, effect size, and alpha-value. The parameters for our analyses were: $\sigma = 5.51$, $\delta = 2.5$, and $\alpha = 0.05$. This analysis revealed that we would have found a significant difference ($p < 0.05$) with 21 groups. This suggests that while our measure of solution length is clearly not as sensitive as the overlap measure described above, it is not completely infeasible as a performance metric (11 groups in each condition).

We should caution that even though it did not happen in our experiments, there could exist an error-optimality trade off. For example, a group could create a really short solution by overlapping all the operations, thus creating a large overlap error. We believe that the usefulness of either of these metrics must be carefully examined in the context of the specific interface and instructions provided. In our experiment, we explicitly instructed users to aim for the shortest possible valid answer. While we did not see any tradeoff effects in our experiment, we would advise that other researchers using this task be aware of this possibility.

Our raw measures of communication efficiency, as reflected in word counts, utterances, and deictic references, were more successful in the first study than in the second. Measures of communication efficiency are highly variable and group specific. If researchers are particularly interested in using this task to analyze such communication efficiency measures, we would strongly suggest a within-subjects approach in order to control for individual (or in this case group) communication preferences. In addition, given that group dynamics significantly impact communication processes, we also recommend longer practice sessions to help groups establish work strategies.

Finally, in both studies, a review of the communication and action transcripts was useful in providing descriptive events that demonstrate how groups adapt their communication to the available collaboration tools. In the first study, counts of gestures, when correlated with linguistic analysis, revealed interesting patterns of communication. Similarly, in the second experiment, users often used the visual space to ‘demonstrate’ their available task objects [10] in the Parallel Presentation condition. However, while in the Serial Presentation condition, they simply used language to describe the potential objects rather than switching views of the workspace. If researchers plan to use this task to perform dialogue or discourse analysis, we would also suggest that they consider within-subjects manipulations in order to help account for the individual differences inherent in communication patterns.

We believe that evaluation tasks like this one are particularly important with technologies geared towards distributing computing resources in various form factors throughout the environment. For example, there has been a recent interest in tabletop displays that allow multiple people, clustered around the horizontal display surface, to view and coordinate a common set of information. There has also been much work done on sharing and interacting with information on physically large displays such as wall projections. For reviews of work done in both these areas see [26–28]. Finally, this task could also be useful for evaluating infrastructures that integrate these technologies into coherent environments, which typically allow multiple users to interact across multiple devices and hence collaborate with each other (e.g., [29]).

6 Conclusion and future work

We have described both traditional and distributed job-shop scheduling tasks, both of which we assert can be usefully applied to evaluate interfaces that support coordination in computer-supported collaboration environments. We have discussed evaluation measures and have shown examples, grounded in two experiments, of particular analyses that could prove useful. Results from the experiments demonstrate benefits of rich digital communicative gestures as well as shared visual information when performing coordination tasks.

We believe that these tasks can benefit others exploring technology to support collaboration, and further use of it within the community will help establish a common understanding of the tasks, and appropriate metrics to measure impact in collaborative environments. We think it would be interesting to explore further variations of the tasks to test specific properties of group interactions. For example, we could explore versions in which users do not have equal access to all information (i.e., a hidden information task). We could also explore other analyses, such as correlations between communication and tool usage, more detailed strategic analyses such as the amount of time or effort spent on planning versus execution, measures of contribution by individual users, and social effects such as leadership and dominance. Furthermore, we could explore how tasks like this scale to larger numbers of people, as well as whether or not they allow us to adequately measure the trade offs that exist between the overhead of managing information and the benefits of shared gestures and visual spaces.

Acknowledgments We thank Patrick Baudisch, Andrew Faulring, James Fogarty, Jana Foster, Susan Fussell, Teresa Janz, Brian Meyers, Jeff Nichols, Judy Olson, Randy Pausch, Daniel Robbins, George Robertson, Greg Smith, and Ryder Ziola for their discussions of the tasks and assistance with the experiments.

References

1. Mennecke BE, Wheeler BC (1993) An essay and resource guide for dyadic and group task selection and usage. Institute for research on the management of information systems working paper #9309, Indiana University, Bloomington
2. Heath CC, Knoblauch J, Luff P (2000) Technology and social interaction: the emergence of workplace studies. Br J Sociol 51(2):299–320
3. Dourish P (2006) Implications for design. Proceedings of CHI 2006 conference on human factors in computing systems, pp 541–550
4. Inkpen K, Mandryk R, DiMicco JM, Scott S (2004) Methodologies for evaluating collaboration in co-located environments. Workshop presented at the ACM conference on computer-supported cooperative work 2004
5. Pinelle D, Gutwin C, Greenberg S (2003) Task analysis for groupware usability evaluation: modeling shared-workspace tasks with the mechanics of collaboration. ACM Trans Hum Comput Interact 10(4):281–311
6. McGrath JE (1984) Groups: interaction and performance. Prentice Hall, New Jersey
7. Olson JS, Olson GM, Storrøsten M, Carter M (1993) Groupwork close up: a comparison of the group design process with and without a simple group editor. ACM Trans Hum Comput Interact 11:321–348
8. Scott SD, Carpendale MST, Inkpen, KM (2004) Territoriality in collaborative tabletop workspaces. Proceedings of the ACM conference on computer-supported cooperative work 2004, pp 294–303
9. Gutwin C (2002) Traces: visualizing the immediate past to support group interaction. Proceedings of graphics interface 2002, pp 43–50
10. Clark HH, Krych MA (2004) Speaking while monitoring addressees for understanding. J Mem Lang 50:62–81
11. Kraut RE, Gergle D, Fussell SR (2002) The use of visual information in shared visual spaces: informing the development of virtual co-presence. Proceedings of the ACM conference on computer-supported cooperative work 2002, pp 31–40
12. Rocco E (1998) Trust breaks down in electronic contexts but can be repaired by some initial face-to-face contact. Proceedings of CHI 1998 conference on human factors in computing systems, pp 496–502
13. Setlock LD, Fussell SR, Neuwirth C (2004). Taking it out of context: collaborating within and across cultures in face-to-face settings and via instant messaging. Proceedings of the ACM conference on computer-supported cooperative work 2004, pp 604–613
14. Karger D, Stein C, Wein J (1997) Scheduling algorithms. In: Atallah MJ (ed) Handbook of algorithms and theory of computation. CRC Press, Boca Raton
15. Gutwin C, Penner R (2002) Improving interpretation of remote gestures with telepointer traces. Proceedings of the ACM conference on computer-supported cooperative work 2002, pp 49–57
16. Khan A, Matejka J, Fitzmaurice G, Kurtenbach G (2005) Spotlight: directing users' attention on large displays. Proceedings of CHI 2005 conference on human factors in computing systems, pp 791–798
17. Clark HH, Wilkes-Gibbs D (1986) Referring as a collaborative process. Cognition 22:1–39
18. Fisher H, Thompson G (1963) Probabilistic learning combinations of local job-shop scheduling rules. In: Muth J, Thompson G (eds) Industrial scheduling. Prentice Hall, Englewood Cliffs, pp 225–251
19. Beasley JE (1990) OR-Library: distributing test problems by electronic mail. J Oper Res Soc 41(11):1069–1072. http://www.brunel.ac.uk/depts/ma/research/jeb/info.html
20. Endsley MR (1995) Toward a theory of situation awareness in dynamic systems. Human Factors Special Issue Situation Awareness 37:32–64
21. Kraut RE, Fussell SR, Siegel J (2003) Visual information as a conversational resource in collaborative physical tasks. Hum Comput Interact 18:13–49
22. Gergle D, Kraut RE, Fussell, SR (2004) Action as language in a shared visual space. Proceedings of the ACM conference on computer-supported cooperative work 2004, pp 487–496
23. Tan DS, Meyers B, Czerwinski M (2004) Wincuts: manipulating arbitrary window regions for more effective use of screen space. Proceedings of CHI 2004 conference on human factors in computing systems, pp 1525–1528
24. Kenny DA, Mannetti L, Pierro A, Livi S, Kashy DA (2002) The statistical analysis of data from small groups. J Pers Soc Psychol 83:126–137
25. Hoenig JM, Heisey DM (2001) The abuse of power: the pervasive fallacy of power calculations for data analysis. Am Stat 55:19–24
26. Ryall K, Forlines C, Shen C, Ringel-Morris M (2004) Tabletop design: exploring the effects of group size and table size on interactions with tabletops shared-display groupware. Proceedings of the ACM conference on computer-supported cooperative work 2004, pp 284–293
27. Scott SD (2005) Territoriality in collaborative tabletop workspaces. Doctoral Dissertation, Department of Computer Science, University of Calgary
28. Tan DS (2004) Exploiting the cognitive and social benefits of physically large displays. Doctoral Dissertation, School of Computer Science, Carnegie Mellon University
29. Johanson B, Fox A, Winograd T (2002) The interactive workspaces project: Experiences with ubiquitous computing rooms. IEEE Pervasive Comput Mag 1(2):67–74