Constructive Interaction and Collaborative Work:

Introducing a Method for Testing Collaborative Systems

Mick Tarel © 1997 Artville, LLC

In trying to determine how people use interactive computer systems, many of us invest a considerable amount of time and energy to find out how users work and how they interact both with each other and with the computer. We want to know what they think about the interactive systems we provide for them (and sometimes in cooperation with them) and what we can do to make these systems fit their needs.

This article focuses on how the constructive interaction method helps system designers determine whether the basic concepts underlying a system are well understood by users and whether its implementation, usability, and utility are satisfactory. We describe our experiences in using a form of constructive interaction to test a software package that supports a particular collaborative activity. The difference between our use of the constructive interaction method versus other models is that our test subjects use separate workstations in the same room to discuss their common tasks. Having thus adjusted the setting to the specific characteristics of computer-supported collaborative work, we asked each person to carry out separate sets of predefined tasks that were linked.

Simply asking people whether they are satisfied with a newly introduced system does not suffice, because the reasons they give may not reflect their actual views or behavior [1]. To avoid the shortcomings of such a straightforward approach, the thinking aloud method has been used in order to both gain a more adequate understanding of how a person views a system and test the system's usability [7]. In a standard thinking aloud test, a person has to work on a predefined task while continuously verbalizing his or her thoughts. This method yields a set of verbal utterances combined with actions about the task. The behavior of the person tested can be audio- or videotaped, and analyzing several of those tests may reveal how people understand or misunderstand the computer system and how to reduce misunderstanding. However, this method clearly has drawbacks. First, interaction is limited because the user

Helge Kahler
Institute for Computer Science III, Research Group HCI & CSCW (ProSEC)
University of Bonn,
Roemerstrasse 164, D-53117 Bonn, Germany
Tel.: 49 228 73 4299
Fax: 49 228 73 4382
kahler@cs.uni-bonn.de
http://www.cs.uni-bonn.de/~kahler/index_gb.html

mainly reports his or her experiences to the researcher. Second, the setting may seem unnatural to many people and make them feel they are being observed. And third, the researcher might interact too much with the person tested and bias the result.

To circumvent the difficulties of usability testing involved in the thinking aloud test, the constructive interaction method is carried out using two subjects. The two subjects are asked to perform a task together, which usually leads to arguments about what to do next and how to do it and explanations to each other of why they did what they did. Since this type of interaction is more natural than that in the thinking aloud test, and since interaction between researcher and tested individuals is minimal, the results can be considered to be of a relatively high ecological validity.

This method of observing two people in the solution of a common task in order to better understand learning processes, mental models, and aspects of a system's usability has been labeled with different names, depending on the focus of the researchers involved. Miyake [6], for instance, calls the method "constructive interaction," whereas Kennedy [3], who applied the method in the context of usability testing, refers to it as "co-discovery learning." When applied to usability testing, other names have been introduced, such as "paired-user testing" [12] or "co-participation" [13]. It should be noted, however, that the last two terms usually imply an environment in which two people work together at the *same* workstation, whereas the form of constructive interaction presented in this article requires two people to perform a collaborative task on two separate workstations. To differentiate between these two approaches, I will use the term "constructive interaction for testing collaborative systems" (CITeCS) for the setting with *two separate* workstations (see Table 1 for an overview of the different forms of constructive interaction).

Aspects of Constructive Interaction

The concept of constructive interaction was introduced by Naomi Miyake [5, 6], who asked test subjects to discuss and solve a

Table 1. CONSTRUCTIVE INTERACTION AND COLLABORATIVE WORK

Researchers	Motivation	Number of Workstations to Perform Task	Recording Techniques	Data Analysis
Miyake	Interest in the iterative process of understanding that takes place when people discuss a problem and pass through several levels of understanding	0	Videotape, audiotape	Transcription, double protocol coding, categorization of actions, marking breaks in utterances
O'Malley et al.	Explore the potential of constructive interaction for human–computer interaction	1	Not described	Transcription, diagram of session showing main topics
Wildman	Usability testing	1	Videotape, logging of activities	Analysis facilitated by extra workstation for test subjects to take notes
Kahler – CITeCS	Test system for collaborative work	2	Taking notes, audiotape	Qualitative analysis

problem—in her case, how a sewing machine works. Miyake was interested in the iterative process of understanding that takes place when people discuss a problem and pass through several levels of understanding. Her study proved the existence of consecutive levels of understanding, the sequence of which followed a certain pattern. Miyake also showed that having a pair of individuals discuss a topic and work collaboratively on a solution revealed much about their underlying assumptions, mental models, and understanding of the topic.

O'Malley et al. [8] were the first to explore the potential of constructive interaction for human–computer interaction and the conditions under which it might be effective. They conducted two studies, each of which involved two participants. The first study was a tutorial session in which an experienced user introduced a novice to a system. The session revealed several sources of confusion for the novice. In the second study, two people were asked to find out how a particular command interpreter worked. They discussed possible strategies and tried out various aspects of the system's functions to support their points of view.

Mayes et al. [4] used constructive interaction by asking pairs of subjects to make collaborative decisions about how to proceed through a hypertext. In their study, the authors drew conclusions about the lack of benefits of hypertext learning systems relative to human–computer interaction and reported evidence that constructive interaction itself can promote learning.

Kennedy [3] was the first to describe constructive interaction as a usability testing method in a commercial setting at Bell-Northern Research. Since then, constructive interaction, improved and modified in various ways, has been explicitly and widely used in usability testing.

The main advantage of constructive interaction is that it yields a rich set of qualitative data that provide valuable insight into how people perceive situations, how they go about solving problems, and,

CONSTRUCTIVE INTERACTION AND COLLABORATIVE WORK

Embedment	Results Concerning Method	Variations
Subjects watched tape with experimenter to help clarify issues	CONSTRUCTIVE Interaction suitable to reveal iterative process of understanding.	None
Not described	SUBJECTS show different points of view about common problem. EMPHASIS should be on understanding or developing concepts, as opposed to learning procedures.	Mayes et al.: using constructive interaction to explore learning from hypertext.
Not described	PAIRED-USER testing low-cost collection of data, users' reasoning processes become visible, friendly ambience of users solving problems together, increased testing time for increased data volume and reduced post-test transcription.	KENNEDY: usability testing of telephones, laborious quantitative analysis, post-test interviews. WILSON: collection of expert statements about co-participation. WESTERINK: after the test the pairs are asked to describe their experience to a third person.
Pre: field study with interviews Post: questionnaire after test	TWO workstations in one room well suited for testing collaborative systems and distributed tasks. PREVIOUS results about method confirmed.	None

in particular, how they perceive the conceptual framework and usability of a given system. Sasse [10] suggests that constructive interaction is particularly well suited for exploratory studies. A study by O'Malley et al. [8] revealed that constructive interaction can be quite useful for exploring users' understanding of system concepts. The reason is that differences of opinion lead test subjects to articulate the rationale behind their hypotheses, thereby enabling the observer to understand how the subjects perceive the system. Mayes et al. [4] argue that constructive interaction differs from many other methods because it does not aim at reducing data but rather at exposing as much of the underlying cognition as possible. According to Wildman [12], constructive interaction is a good method for early usability testing when the design process focuses on general issues of navigation, representation, organization, and functionality. Kennedy [3] reports that video recordings of experiments involving constructive interaction provide more interesting, informative, and convincing data than video material from thinking aloud sessions. Kennedy also used the method in her interaction with developers. Seeing users interact in a video about their trouble with using the product was much more convincing than detailed descriptions of usability test results and statistics.

One drawback of such an approach is that the abundance of data cannot be easily evaluated quantitatively. If you want to go beyond purely qualitative statements and perform detailed error analyses or compare different pairings, the data must be carefully transcribed and analyzed. Given the richness of the information, this is likely to be time consuming.

An important issue in constructive interaction is the relationship between the individuals paired. Often it is reasonable to have two individuals who have the same level of knowledge or expertise and whose communication will therefore be marked by an exchange of opinions on how to work on a task. Sometimes, however, it might be helpful to choose individuals with different levels of knowledge in order to create a situation in which the interaction is guided by one person. However, differences in expertise or verbal style (e.g. outspoken or talkative versus shy or restrained) or a hierarchical relationship between the individuals may hamper feedback. Wilson [13] cites positive experiences with recruiting two individuals as a pair, for example, by asking a willing participant if she or he would like to bring along someone to do the test with.

Several suggestions have been made to increase the number of individuals involved in constructive interaction. Westerink et al. [11] have proposed settings in which three people have to interact with each other. In such a situation, the two people taking the usability test were asked afterwards to describe their experience to a *listener,* whose task was to elicit a summary of their impressions. Wilson [13] reports an interesting case in which two system administrators and two users took part in a session during which the administrators explained the product to the users.

Constructive Interaction for Testing Collaborative Systems

I began to explore the topic of constructive interaction when my colleagues and I researched tailorable systems and collaborative work. In our research, we investigated how collaborative tailoring of off-the-shelf applications can be supported by technical mechanisms.

Setting

System

In the study we conducted, we decided to work with a word processor, because it is a good example of a widely and extensively used software. In order to learn more about how groups of users tailor their tasks collaboratively, we carried out a field study at four different organizations. The study yielded a number of different collaborative tailoring scenarios, all of which focused on the exchange of document templates and toolbars, that is, graphical representations of functionality. By analyzing these scenarios, we developed requirements for the design of a tool to be used as an add-in to the word processor, or rather, an extension of its functionality using the programming interface. This add-in, henceforth called *the tool*, provided some functionality for collaborative tailoring, that is, for sharing and distributing changes to the functionality or appearance of the word processor or document templates that could then be used or modified by other people (see [2] for information on tailoring and the tool).

The basic functions included loading and saving document templates and toolbars. It was also possible to combine a document template and several toolbars in a package in order to support specific word processing tasks, such as the design of a Web page or the writing of a mathematical paper. The collaborative aspect was added by the functionality we provided for sharing document templates and toolbars between the creator and other persons by both a sending and an access mode. In order to support centrally administrated environments, adaptations could be sent to groups of end users. The access mode allowed users to simply store the tailored artifacts in a shared workspace. If another user were searching for a certain adaptation she could access the required templates or toolbars in this shared workspace, the *public folder*.

Method

To test the tool, constructive interaction was an obvious choice because we wanted to have pairs of users perform tasks collaboratively. The test had two goals. On the one hand, we wanted to find out if and to what extent the users taking part in the test understood the concept of sharing tailored artifacts and how it was implemented in the tool. On the other hand, we expected the experiment to yield clues for improving the usability and utility of the tool.

Enhancing the constructive interaction setting as outlined previously, we set up two workstations, which the two test subjects were to use to perform their common task. This setting reflects the nature of asynchronous distributed work, whereby two individuals, A and B, take turns in performing a sequence of actions. Unlike most computer-supported collaborative work situations, however, the two individuals were located side by side in the same room so that they could talk to each other face-to-face and so that each was able to see what was happening on the other's monitor. All test subjects had participated in the field study. The subjects teamed up in each of the two pairs knew each other but had not worked closely together before. The tests took place in one of our offices.

The test subjects had to work collaboratively on two tasks, each of which involved several subtasks. The tasks consisted of jointly creating and refining a word processor's document template, including a toolbar. Before testing the individuals, we explained to them the basic functions of the tool and the aim of the experiment, which was to test the tool's usability and utility. In a first step, the task of Person A was to create a document template, modify a toolbar, and incorporate another toolbar that she received from Person B. Afterwards, she had to save all of these elements in a document template connected with a toolbar in her private folder and send it to Person B. Person B, in turn, had to create a toolbar with specific icons and send it to Person A for further usage. The second task required Person A to define a group and send a document template to the group. She then had to change a toolbar, save it in the private folder, and make the toolbar available in the public folder. In this phase of the test, Person B had to copy the toolbar from the public folder to his private folder and then load it using the preview mode. Both participants had the same written task description, which was divided into two sections, "Tasks for Ms. A" and "Tasks for Mr. B." From the task description they could see when it was their turn to interact with their workstation. We encouraged the test subjects to read the task description and to check with each other whether they knew what to do; they were also encouraged to discuss the next step that each had to take in the course of the task. Work on these tasks lasted about 30 minutes.

METHODS & TOOLS
COLUMN EDITORS
Michael Muller
Lotus Development Corp.
55 Cambridge Parkway
Cambridge, MA 02142 USA
+1-617-693-4235
fax: +1-617-693-1407
mullerm@acm.org

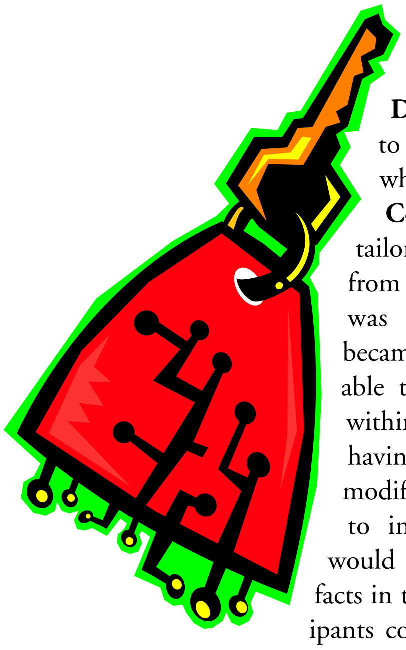
Finn Kensing
IT University of Copenhagen
Glentevej 67
2400 Copenhagen NV
Denmark
+ 45 3816 8888
fax: + 45 3816 8899
kensing@itu.dk

A researcher and the developer of the tool were present to observe each of the test pairs. Both took notes during the tests. Moreover, the test was audiotaped to support those notes in cases of doubt and to be able to extract quotes. After the test we reviewed a brief questionnaire concerning aspects of working with the tool that could not be dealt with in the tasks.

Results

System

The results of our constructive interaction sessions concern different levels. First of all, the test showed clearly that the interface of the tool needed to be improved. Some buttons caused misunderstandings and had to be renamed. The name of one button, for instance, had to be changed from **Delete** to **Deactivate** because its function was to hide a toolbar. Another button, which had originally been labeled **Copy** and allowed users to move a tailored artifact that was sent to them from the inbox to their private folder, was renamed **Adopt**. Moreover, it became clear that users should also be able to delete a tailored artifact from within the word processor rather than having to use the file manager. This modification also resulted in a proposal to introduce an administrator, who would be allowed to delete tailored artifacts in the public folder. All of the participants considered it an advantage to have the possibility to save, combine, and distribute tailored artifacts. Although not all test subjects were expert users they were all able to use the tailoring function and the sharing function. The users perceived the overall usability of the tool to be good.



Two participants, one a network administrator and the other an experienced user of the particular word processor being used, said that such a distribution of tailored artifacts would be quite helpful for their organizations. The constructive interaction sessions revealed that the participants' conceptual model of how the distribution of files worked was close to how we, the designers, had intended and carried out the distribution. This is an important result insofar as a misperception of the underlying model (for example, about how links work or who can see and change which elements) often leads to a user's inefficient use or lack of acceptance of the system. This holds in particular for more complex work group settings.

Method

We found that, for our purposes, constructive interaction for testing collaborative systems proved to be effective. The topic of collaborative work that this version of constructive interaction focused on was, for the first time, connected to the method. The tests showed that our enhancement of constructive interaction methodologically suited the questions raised by computer-supported collaboration. The collaborative nature of the task and the fact that the system was a medium for the test subjects' collaboration made CITeCS the method of choice.

Constructive interaction as it was employed in earlier studies was a useful framework to start developing ideas about testing a collaborative task, because it already involved communication between two people working on a common task. The new aspect that we added with our enhanced model is the distribution of parts of the task among the two people. Introducing a second workstation, while still allowing face-to-face communication, combines the advantages and the natural communicative setting of constructive interaction with the main features of collaborative work. In our tests, this approach resulted in lively discussions among the participants, which provided valuable insights into the problem-solving process as well as into the interests of the partners and the different roles they assumed in their collaboration. Our setting proved to be well chosen because either user in a test pair could ask the other person what impact their actions would have on the other person's work. This approach made it possible for each user to understand both sides of the collaborative process. Its benefit was thus twofold: (1) it helped test users to understand the system and (2) it allowed us to gain a number of interesting insights into how the users' perspectives on their particular part of the common task was influenced by our tool and how our design influenced their collaboration.

Aside from the quite awkward option of employing one tester as a dummy user, an alternative setting for this kind of task would be to let the two users work in separate rooms, each of them observed by a tester. However, such an approach not only would involve more resources but, in order to create a more realistic collaborative environment, would unnecessarily hamper the communication between the test subjects. Using two workstations with distinct but strongly connected tasks for the two subjects, as we did, instead of the alternative of one workstation with two virtual screens, prevented one of the paired individuals from assuming a dominant role.

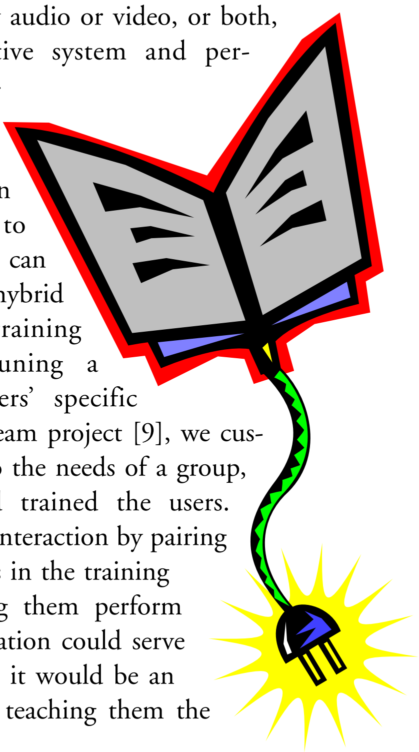
Although the setting with two workstations in one room proved particularly useful for examining collaborative work, several aspects of our test can be related to findings of other researchers who employed constructive interaction or paired-user testing. Like others before us, we found that having two users discuss and perform a common task was a useful means for understanding the users' perception of the system concepts and for uncovering usability flaws. Compared with thinking aloud sessions, which we had used previously to explore other issues, the discussion between the two test subjects seemed much more "real" than the utterances from thinking aloud test takers.

For both the implementation of the tests and the evaluation of the data, we chose a simple setting that required neither laboratory nor sophisticated video equipment. Furthermore, we did not transcribe the tapes in detail or perform a quantitative evaluation because we felt that the extra work involved would have been disproportionate to the potential benefits for our research goal. The richness of the data shows that such a technologically modest approach can be useful in academic or other settings where resources are limited. This modest approach also has the advantage of providing a more natural setting for testing collaborative activities because it can be carried out at people's work places where individuals feel more comfortable than in the artificial "workplace" constructed for them at a research institute.

Our CITeCS activities were connected to other forms of learning about the usability of our tool in two ways: before the test, we conducted individual interviews about tailoring software; after the test subjects completed a questionnaire in which they were asked, for instance, to draw a map of where they thought certain artifacts were located at different times in the process. Both of these additional techniques proved useful. The test subjects needed fewer introductory explanations and were able to understand the rather complex task because they had already been acquainted with the topic. The questionnaire complemented the test results and supplied us with insights that the CITeCS method could not have provided.

Prospects

Our experience with constructive interaction has encouraged us in several ways. In times of increased computer-supported collaboration over distance, CITeCS offers possibilities for testing over distance with two or more test takers connected by audio or video, or both, using a collaborative system and performing a set of collaborative tasks. Furthermore, constructive interaction is not limited to testing purposes; it can also be used for a hybrid that combines training users with fine-tuning a system to the users' specific needs. In the PoliTeam project [9], we customized a system to the needs of a group, introduced it, and trained the users. Using constructive interaction by pairing or grouping persons in the training sessions and having them perform tasks in such a situation could serve two purposes. First, it would be an appropriate way of teaching them the



basics of the system that has been customized to their needs to the best of our knowledge and helping them to understand the specific aspects of collaborative work and the interrelations of the actions that group members perform with the system. Second, we, as system designers or people who customize systems for others, could learn both about characteristics of the group that we may not have foreseen or fully understood and about the specific requirements for fine tuning the system.

We are convinced that there is still more potential in CITeCS, and we will continue to improve and use it to design, introduce, use, and evaluate collaborative systems.

References

1. Cicourel, A.V. *Method and Measurement in Sociology.* Free Press, New York, 1964.
2. Kahler, H. and Stiemerling, O. "Pass me the toolbar, please!": Cooperative Tailoring of a Word Processor. Position paper. The 1999 International Joint Conference on Work Activities Coordination and Collaboration (WACC '99), San Francisco, CA, 1999, available at http://www11.informatik.tu-muenchen.de/workshops/wacc99-ws-impltailor/
3. Kennedy, S. Using video in the BNR usability lab. *SIGCHI Bulletin 21*, 2 (Oct. 1989), pp. 92–95.
4. Mayes, T., Kibby, M., and Anderson, T. Learning about learning from hypertext. In *Designing Hypertext/ Hypermedia for Learning* (Jonassen, D.H. and Mandl, H., eds.), Berlin, Springer, pp. 227–250.
5. Miyake, N. Constructive interaction and the iterative process of understanding. *Cognitive Science 10*, 2 (1986), pp. 151–177.
6. Miyake, N. Constructive Interaction. Technical Report 113, Center for Human Information Processing, San Diego, University of California, 1982.
7. Nielsen, J. *Usability Engineering.* Boston, Academic Press, 1993.
8. O'Malley, C.E., Draper, S.W., and Riley, M.S. Constructive Interaction: A Method for Studying Human–Computer Interaction. In *Proceedings of Human-Computer Interaction* (INTERACT '84), Elsevier, 1985, pp. 269–274.
9. Pipek, V. and Wulf, V. A groupware's life. In *Proceedings of the Sixth European Conference on Computer-Supported Cooperative Work*, Kluwer, 1999, pp. 199–218.
10. Sasse, A. Eliciting and Describing Users' Models of Computer Systems. Unpublished doctoral dissertation. School of Computer Science, The University of Birmingham, U.K. 1996.
11. Westerink, J.H.D.M., Rankin, P.J., Majoor, G.M.M., and Moore, P.S. A new technique for early user evaluation of entertainment product interfaces. In Volume 2, *Proceedings of Human Factors and Ergonomics Society*, 38th Annual Meeting, 1994, p. 992.
12. Wildman, D. Getting the most from paired-user testing. *interactions* 2, 3 (1995), pp. 21–27.
13. Wilson, C. Pros and cons of co-participation in usability studies. *Usability Interface 4*, 4 (April 1998).