# Epilogue: How to Make Use of the Book

This epilogue focuses on two ways of applying the overall ideas and the concrete recommendations included in this book. First we provide an example and some of our general experiences in working with IT practitioners on disseminating the MUST method in ways that allow them to incorporate parts of the method into their repertoire for actions. Then we give an example of how we use it as a textbook for a university course.

## Including the MUST Method in Practitioners' Repertoire for Action

Studies have shown that the introduction of new methods often fails. The example in this section builds on our positive experience from a dissemination endeavor that we carried out to introduce the MUST method in an internal IT department. We end this section by offering three lessons learned from this example and other dissemination projects in which we have been involved.

This example demonstrates that a combination of seminars, reflections on current and emerging practices, apprenticeship relations, and supervision of technical as well as personal skills was an instrumental approach. It helped a group of IT professionals to adapt and integrate relevant parts of the MUST method into their repertoire for action, to take on new roles, and to become more creative in fulfilling these roles.

We were invited to teach the MUST method to a group of IT professionals of an internal IT department. A project was formed in order for the IT professionals to adapt and integrate our method into their repertoire for action. The project was conceived to be important as the department strove to implement the organization's strategic plans. They included

a call for major changes related to the IT department's responsibilities and its relations to the user departments. Among the strategic plans were four suggestions:

- The technological platforms for office work as well as for the production of the company's services and products should be merged.
- Business Process Reengineering (BPR), introduction of standard systems, and selective outsourcing of areas formerly taken care of by the IT department should be considered, some suggestions include development of applications not available at the market, networks, installation of PCs, support, and the running of installations.
- Future IT professionals should be trained in business analysis, technical issues, and personal management issues.
- The lack of knowledge and skills in the IT department calls for new employees, further education, or partnerships with external suppliers.

We approached method dissemination mainly from a learning perspective. Introducing the MUST method, we applied a combination of strategies. In doing so, we were inspired by Schön's (1983) studies of reflective practitioners. He provides a conceptual framework for the analysis and establishment of contexts for learning. He advocates that students and coaches conduct a reflective conversation with the materials of design situations in order to facilitate the acquisition of new skills. The work of Lave and Wenger (1991) was also an inspiration. They reconsider apprenticeship and situated learning and suggest legitimate peripheral participation as a conceptual framework for understanding how learning takes place in various trades. Nonaka's (1991) concept of learning organizations made us reflect on how to be instrumental in helping to institutionalize practices where learning becomes an ongoing concern.

First, we introduced the MUST method through seminars based on this book. Then two IT professionals started applying the method in a large IT design project. It was the first time they were part of a project of this scale and with such a close cooperation among IT professionals, managers, and users. We followed the project closely—participated in meetings, observed interviews, and reviewed plans and

various intermediary results. This allowed us to observe the IT professionals while working, which made it possible for us to develop an appreciation of their strengths and weaknesses. Thus, when teaching and acting as coaches, we were able to relate to specific instances where the IT professionals' previous practices worked and where they fell short.

The next step was to introduce the MUST method to all of the IT department's employees. This included three types of activities:

1. Using the initial IT design project described in the previous paragraphs as an example, the method was introduced through four half-day seminars and supplemented by written materials.

2. For each of the IT professionals' ongoing projects, we worked as coaches. Together with the participating IT professionals, we started out evaluating the state of their current projects and related them to earlier experiences, focusing on problems perceived by the IT professionals. Then we selected parts of the MUST method that were appropriate in addressing the problems.

Next, we introduced an ongoing review process, where the IT professionals on a biweekly basis presented and discussed their project and received feedback. Sometimes further teaching on elements of the method was required. This went on until the IT professionals had adopted parts of the method. In addition, we conducted a future workshop. The themes of the workshop were the roles of the IT department in relation to user departments and external consultants, and the problems and possibilities internal to the department. The results of the future workshop were documented in a report, which was used to prioritize the areas where the IT department should strive to improve.

3. The MUST method includes a rather different style of working with users and managers than the IT professionals had practiced before. Therefore, new technical as well as personal skills needed to be developed. The technical skills were taken care of in activities described in points 1 and 2 of this list.

The personal skills involved handling unstructured, open-ended situations and much more oral and written communication with users and managers than the IT professionals were used to. This made these IT professionals somewhat uneasy applying the method. Therefore, in addition to the technically oriented review process described in point 2 of this list, we offered—on a biweekly basis—individual supervision addressing personal issues. We discussed the problems each perceived in fulfilling the role of an IT designer, and set up small-scale experiments for them to practice on in their ongoing projects. The results of the experiments were discussed in the following supervision session and new initiatives were taken.

These activities continued for a period of six months, until the IT professionals felt more secure in applying the method. Major elements of the MUST method became part of the organization's practices for conducting projects.

## Lessons Learned

Our approach to method dissemination is based on two basic premises:

1. Introduction of a new method should be coupled with actual challenges in real design projects.
2. Traditional teaching cannot stand alone in method dissemination.

These premises have emerged from earlier projects. Drawing on our previous teaching experiences, we applied a combination of lectures, reflections on current and emerging practices, apprenticeship relations, and supervision of technical as well as personal skills. The central point is to get beyond a mode of detached reflection in the interaction between the IT professionals and the person responsible for the dissemination endeavor (from now on referred to as "the consultant").

Practitioners who are simply given a general presentation of a new technique are left on their own when trying to integrate the technique into their work practices. And a consultant who is simply told about events and changes in a recent project is left with the question about what really happened. So, to get beyond the say/do problem, we advocate that the consultant get involved in the work of the IT professionals

through observations or ultimately through working together on a project. This makes it possible for the consultant to relate to problems in the practitioners' current practices when presenting a new technique or proposing changes in their design practice.

Finally, we turn to a closer inspection of lessons learned about method dissemination activities. We do this under three headings (commitment to change, observation led to a breakthrough in the dialogue, and accepting the stranger) that capture important issues in relation to the research question that guided our work: What are appropriate ways of introducing the method supporting IT designers in an industrial context? In the previous paragraphs, we drew on our experiences from an internal IT department. In the next sections, we also include our experiences at an IT consulting company and a university hospital—two additional partners who worked with us on research projects, addressing this central research question.

## Commitment to Change

A commitment to change is an important factor in any change process, including method dissemination. The commitment should be established at the management level as well as among those whose practices are the subject of change—the IT professionals. It is general wisdom that management commitment to change is pivotal. In the IT consulting company that worked with us, for example, it was the department manager who originally took the initiative to try out the MUST method in a project. The manager was highly engaged in the project and served as its project manager. In the internal IT department that worked with us, the IT manager also took the initiative. When he left, however, two levels of management in the business unit with whom we also worked sustained his commitment.

Both companies knew beforehand that they could do things better and had decided to spend resources in trying to improve their design processes. Experimentation with the new method in the IT consulting company took place in a commercial project with an important customer. The project in the internal IT department had the attention of the highest level of management since it was part of a major invest-

ment. This encouraged the companies to take a serious and critical approach in learning, using, and evaluating the different activities and techniques proposed by the method.

In the university hospital, a large number of projects had been carried out following the outsourcing strategy. There was a very positive attitude to the need for changing work practices in design—the IT professionals openly discussed their problems in the projects at lunch and at weekly meetings. However, they did not have the time and resources to investigate these problems thoroughly and identify similarities across projects. For these reasons, we chose to approach the dissemination project at the hospital in two steps. Step one focused on identifying problems, investigating similarities, and proposing improvements. In the second step, the IT department would choose which of the proposals they wanted to implement.

The very organization of the project into these two steps reflects our understanding of another aspect of the importance of commitment to change: Changes should address areas where improvement is appreciated by the people involved. In order to locate such areas, we had to spend time identifying common problems and their nature across the projects. The projects chosen for detailed investigation were carefully selected to reflect the diversity of IT projects at the hospital. Great care was taken to present preliminary findings to the whole group of IT professionals before reporting to the management. Such presentations were done on a regular basis throughout the project; it was arranged as a meeting or a part of the IT professionals' weekly meeting, where our findings documented in a report had been distributed to the participants in advance. The discussion often resulted in changes and additions to our report.

### Observation Led to a Breakthrough in the Dialogue

In order for a consultant to communicate effectively with practitioners, he or she benefits from observing the practitioners experimenting with the new work practices.

In our project with the IT consulting company, we made general presentations of the method before parts of it were used in a project with an important customer. In retrospect,

we realized that these presentations were basically an account of abstract knowledge that the practitioners had to relate to their individual experiences. We could only relate to, and give examples from, our individual experiences from projects and situations in which the practitioner had not taken part. We often struggled to understand each other since both parties were interpreting the abstract method descriptions from the perspective of different practical and situated experiences.

This changed dramatically once a common ground was established. During the project at the customer site, we participated as an observer and a shared base of experience was developed. This led to a breakthrough in the mutual dialogue: Different aspects of the method (and its general guidelines) could now be related to common and situated project conditions. This shared base of experience established possibilities for discussing how the method could be applied in specific situations. Discussing the method based on shared experience also allowed the practitioners to develop faith in our ability to understand their work situation. In other words, this contributed to a confidence in us, which is another element that facilitates practitioners' commitment to change.

The same was the case in our work with the internal IT department. When presenting the MUST method at seminars to this group, we were able to relate to an earlier project carried out by some of the practitioners in which the method was used. From our observations in that earlier project, we had even learned about the products and services produced by the company. This established some kind of a common ground for the communication.

## Accepting the Stranger

Accepting the stranger denotes that the company's IT designers have to accept that a stranger—the consultant—comes very close to their work practice. It is primarily a message for the participating industrial company—but also a lesson that presents food for thought for the consultant. Observing the activities of IT professionals is vital for the consultant in order to be able to communicate his knowledge and method to the practitioners (as discussed in the

previous section). At the outset, project members from all three companies agreed to this condition. But this issue later became a reason for concern for some of the participants.

In the IT consulting company, the practitioners were all senior consultants with highly established and well experienced work practices. They felt concerned when the consultant, through his observations came close to their work practice and organization. One practitioner explained during the project that they in fact had already "written" 80 percent of the final report for the customer even before they had the first visit at the customer site. The consultant immediately recorded this, and later it turned out that the practitioner in question had felt very annoyed by this. He was concerned how the consultant would interpret this "work practice" and how it would be presented to other colleagues and managers. The "80 percent rule" could be explained in a very positive way: The practitioner was a highly experienced and knowledgeable domain expert and in general 80 percent of his findings had been experienced before with other customers. However, it could also be explained in a less positive manner: This practitioner had a tendency to jump to conclusions and recommend IT solutions to a customer based on his knowledge of the company's IT portfolio rather than on the needs and problems observed at the customer site.

The diagnostic analysis performed at the IT consulting company was based on a series of interviews and led to a report pointing out four problematic issues suitable for experiments involving the MUST method. One of these issues exposed an internal conflict within the company. The conflict was rooted in a dilemma of prioritizing the IT platform. On the one hand, they might prioritize the IT solution as a generic system where new releases could be offered to all customers. On the other hand, individual customer's specific needs might be prioritized in a way that could lead to different tailored systems, which might be hard to maintain through new versions of the generic system modules. The manager did not appreciate that this conflict was identified in the report and suggested that his employees had not read the report—a suggestion that did not align with the dissemination approach.

The university hospital's concern about accepting the stranger was also important. Our approach stipulated that changes in their current work practice had to be based on a common understanding of the areas for improvements. That clearly involved some kind of evaluation of past and present performance, which implied evaluations of specific individuals since the projects were often staffed with very few people. Strangers (i.e., the consultants) took part in this evaluation. Since our dissemination endeavor depended on a constructive dialogue with all parties involved, we needed to handle the evaluation with delicacy. Therefore, for a considerable part of the project period we spent full days in the IT department—which meant that we took part in lunch and various meetings, and thus became less of a stranger. Second, we always sent out interview summaries to the interviewed persons in order to allow them to correct mistakes in what would become the project's record. Third, we stressed again and again that the purpose was to identify general problems and not to identify success stories or failures. And finally, we took great care not to name individuals in reports or oral presentations.

The lesson seemed to be that establishing and maintaining a positive attitude toward dissemination projects requires considerable attention to confidentiality and personal integrity issues.

## Using the Textbook for a University Course

In a course for graduate students in participatory IT design, the Danish version of this book (as well as earlier versions of it) has been used as the primary textbook in courses for graduate and undergraduate students in information systems and computer science programs. The example that follows is taken from a course offered to students as part of their master's program. Students earlier in their career may need more direct instructions and exercises than indicated here.

## Motivation

Most companies and organizations rely on standard software rather than developing their own systems from scratch. Therefore, some IT designers focus on design of generic systems for the market. Others design for a specific customer's needs and opportunities, delivering what we refer to as "coherent visions for change." Thus, the latter group works on standard software as well as custom-made software in their overall design that also comprises changes in users' qualifications and in the organization of work. This course aims at developing students' competence within the latter type of jobs.

For many years, information systems and computer science programs have offered courses in software engineering and systems development focusing on technical and project management issues. However, we have found a lack of courses that develop students' competencies in designing coherent visions for change for a specific customer. This type of training also requires attending to organizational issues.

## Objectives

Students in this course learn to plan and conduct the initial part of an IT project. After the course, successful students have the theoretical and practical qualifications to do several things:

- Explore and document needs and opportunities for a specific company
- Design one or more coherent visions for change
- Evaluate potential consequences of a realization of the visions
- Plan an IT design project as well as a subsequent IT realization project

Thus, students learn to design two things: a coherent vision for change, as well as the processes through which project participants get from initial ideas to actual IT applications in use. However, instructions on how to conduct an IT realization project is outside the scope of this course.

## Size

The course is planned to take up one-third of the students' time during a term of twelve weeks, which in the European Credit Transfer System equals 7.5 ECTS. Generally speaking, this equals one-eighth of a student's work in a year, which we see as a necessary minimum. If students are able to spend more time on this subject, the list of literature later in this epilogue has proven to be a valuable resource in the past.

## Prerequisites

Students are expected to have taken courses in programming and systems development or software engineering. A general understanding of organizations and of their IT usage is considered an advantage.

## Contents

This course deals with theories of and methods for IT design, as well as IT design practices. Various accounts of IT design and how it may be approached are studied. The students plan and conduct a small IT design project according to the guidelines of the primary textbook.

The focus of this course is on design as a process: what are the main elements of design and their dynamics; adequate tools and techniques; results of a design process and their relations to other activities in systems and organizational development; and improving one's skills as an IT designer.

## Format

The format of the course includes lectures by the professor and presentations by the students (see figure E.1). In addition, students work in groups of three to five people on an IT design project (see figure E.2). Each group finds a small company or a department within a larger company for which they produce a design report. The professor or a teaching assistant offers supervision to each group and advice on how to plan and conduct the project including which tools and techniques to apply.

Week One: An Introduction to the Course

Readings: Chapter 1 of this book and an example from the professors' own repertoire of IT design projects. If the professors do not have such examples, Kensing, Simonsen, and Bødker (1998) (see figure E.3) may prove helpful. Students should start forming groups and investigate potential companies for the IT design project.

Week Two: Concepts and Principles

Reading: Chapter 2 (except section 2.3) and chapter 3 of this book.

The groups for the IT design project should now be formed and the project companies chosen.

Week Three: Project Establishment and Strategic Alignment Analysis

Reading: Chapters 4 and 5 of this book, plus the tools and techniques described in sections 9.1–9.7.

The groups should work on their project charters and focus on the work domains that will be the subjects for further study.

Week Four: Ethnographically Inspired Analysis
Reading: Chapter 6 and section 2.3 of this book plus the tools and techniques described in sections 9.8–9.13.

The groups' first deliverable: a combined project charter and strategic alignment report.

Week Five: Vision Development

Reading: Chapter 7 of this book plus the tools and techniques described in sections 9.10.7 and 9.14–9.16.

The groups work on the in-depth analysis phase.

Weeks Six through Eleven: Student Presentations

Reading: See the list of recommended books and papers in figure E.3.

The students can present individually or in groups. They are encouraged to present the main points of the chosen readings and to relate these points to the textbook and their experiences so far.

We have experienced classes where students, in groups of two to three people, were comfortable preparing a talk based on one of the recommended books, while for other classes one to two papers per student or group were a challenge. In case the paper model is chosen, all papers are included in the curriculum. When students have presented whole books, we selected papers of the same author(s) or parts of the books for the curriculum.

Week Seven: The Groups' Second Deliverable: The Analysis Report

Weeks Eight through Eleven: The Groups Work on Developing Coherent Visions for Change.

Week Twelve: Course Evaluation and Deadline for Delivering the Final IT Design Reports

*Figure E.1 Outline of a course for master's students*

Groups of three to five students are to work on an IT design project for a specific company and apply the relevant elements of the MUST method to their projects. To reduce the level of complexity, the groups are advised to look for a small company or a department within a larger company for which they produce a design report of about twenty-five pages plus appendixes.

In our experience, a weekly or biweekly supervision session of thirty minutes to one hour is sufficient. Some groups need help on how to apply specific tools and techniques, while others are more uncertain about how to structure the process or how to cooperate with people in the company.

Students may also need help on how to delimit their project. On the one hand, a project should be complex enough for the method to be relevant to apply. On the other hand, it should be simple enough for the group to finish within the given timeframe. For this reason, it is important that each group finds a company within the first couple of weeks of the course.

As a rule of thumb, a student project may be divided into three phases combining the MUST method's phase one (project establishment) and phase two (strategic alignment analysis). After the first couple of weeks of the course, the groups should be prepared to work on the project. Over a period of one to two weeks, they work on the first deliverable (the project charter and strategic alignment report) for the company to ensure that the parties agree on the scope and other relevant prerequisites for the project (see figures 4.6 and 5.4). Among the important issues that require agreement are students' access to the company, how much time specific people within the company will be able to spend on the project, confidentiality, and so forth.

Having thus focused in on the relevant parts of the company, the group is ready for the ethnographically inspired analysis. For the next three weeks, students will spend time on interviews and observations at the company, on reading acquired material, and on group sessions developing a shared understanding of the company's needs and opportunities in relation to IT. These are summed up in a report (see figure 6.5) that is presented to the company and possibly adjusted after a discussion.

Finally, the group works for approximately four weeks on the design of one or more coherent visions for change. Students should be urged to involve managers and employees at the client company in their design activities as much as possible. The resulting visions are presented to the company and are likely revised after discussion. The students may use the final week for finishing the report (see figure 7.3), including perhaps improving the earlier deliverables as appendixes.

Sometimes representatives from the companies agree to come to either a joint or a separate session of the class where each group presents its final report and receives feedback on the report as well as on the ways in which the group managed its cooperation with the company.

*Figure E.2 Project assignment*

Finally, to help students further develop their own stance and identity as IT designers, it has proven instrumental to invite three or four professional IT designers to talk about their experiences from recently finished projects. The invitation may ask them to structure their talk around the issues related to the contents and focus of the course. These pre-

sentations may be offered to just those attending the course or to all students if appropriate.

## Pedagogical Considerations

To help students develop skills as competent IT designers, we have found that a combination of the following four elements is instrumental:

1. Reading and discussing the literature on theories of and methods for IT design is essential (this should take the form of lectures and student presentations).
2. Small-scale exercises are helpful when practicing the application of specific tools and techniques (as part of the supervision for the projects or separate course exercises).
3. In order to learn how to deal with the complexities involved in real-life IT design projects, students need to experience these projects firsthand.
4. Finally, a diverse set of exemplary cases is a valuable resource when planning and conducting projects (the textbook, the students' own projects, and those offered by the invited professional IT designers can provide such cases).

If students are able to spend more than the implied one-eighth of a year on this course, we recommend that the reading list for the course be expanded (see figure E.3).

## Examination

At the oral examination, scheduled for one-half hour, each student is requested to give an in-depth reflection on how his or her own experiences from the IT design project related to the literature studied in the class. This is followed by a discussion with the examiner. Alternatively, students may hand in such reflections as a written test.

Textbook:

Bødker, K., F. Kensing, and J. Simonsen. 2004. *Participatory IT Design. Designing for Business and Workplace Realities*. Cambridge: MIT Press.

A list of recommended books and special issues of journals to choose from:

Avison, D. E., and A. T. Wood-Harper. 1990. *Multiview. An Exploration in Information Systems Development*. New York: McGraw-Hill.

Beyer, H., and K. Holtzblatt. 1998. *Contextual Design: Defining Customer-Centered Systems*. San Francisco: Morgan Kaufmann Publishers.

Blomberg, J., and F. Kensing, eds. 1998. *Computer Supported Cooperative Work—A Journal of Collaborative Computing* 7, nos. 3–4. Kluwer Academic Publishers. (Special issue on Participatory Design.)

Greenbaum, J., and M. Kyng. 1991. *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hammer, M., and J. Champy. 1993. *Reengineering the Corporation. A Manifesto for Business Revolution*. New York: Harper.

Muller, M. J., and S. Kuhn, eds. 1993. *Communications of the ACM, Special Issue: Participatory Design* 36, no. 4 (June).

Norman, D. 1990. *The Design of Everyday Things*. New York: HarperCollins.

Schön, D. A. 1983. *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books.

Schuler, D., and A. Namioka. 1993. *Participatory Design: Principles and Practices*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Simon, H. A. 1969. *The Sciences of the Artificial*. Cambridge, MA: MIT Press.

Suchman, L. 1987. *Plans and Situated Action: The Problem of Human-Machine Communication*. Cambridge: Cambridge University Press.

Trigg, R. H., and S. I. Anderson, eds. 1996. *Human-Computer Interaction* 11, no. 3. (Special issue on current perspectives on participatory design.)

Winograd, T. 1996. *Bringing Design to Software*. New York: ACM Press.

A list of recommended papers and book chapters to choose from:

Bansler, J., and K. Bødker. 1993. A reappraisal of structured analysis: Design in an organizational context. *ACM Transactions on Information Systems* 11, no. 2:165–193.

Blomberg, J., J. Giacomi, A. Mosher, and P. Swenton-Hall. 1993. Ethnographic field methods and their relation to design. In *Participatory Design: Principles and Practices,* eds. D. Schuler and A. Namioka, 123–155. London: Lawrence Erlbaum Associates.

Brown, J. S., and P. Duguid. 1994. Borderline issues: Social and material aspects of design. *Human-Computer Interaction* 9, no. 1:3–36.

Dreyfus, H. L. 1988. The Socratic and Platonic basis of cognitivism. *AI and Society* 2:99–112.

Floyd, C. 1984. A systematic look at prototyping. In *Approaches to Prototyping,* ed. R. Budde, K. Kuhlenkamp,
L. Mathiassen, and H. Zullighoven, 1–18. Berlin: Springer-Verlag.

Grudin, J. 1991. Interactive systems: Bridging the gaps between developers and users. *IEEE Computer* 24, no. 4:59–69.

Grudin, J. 1994. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM* 37, no. 1:92–105.

Grønbæk, K., M. Kyng, and P. Mogensen. 1997. Toward a cooperative experimental systems development approach. In *Computers and Design in Context*, ed. M. Kyng and L. Mathiassen, 201–238. Cambridge, MA: MIT Press.

Hammer, M. 1990. Reengineering work: Don't automate, obliterate. *Harvard Business Review* 68, no. 4:104–112.

Hughes, J. A., Randall, D., and Shapiro, D. 1993. From ethnographic record to system design: Some experiences from the field. *Computer Supported Cooperative Work* 1, no. 3:123–147.

Kensing, F., and J. Blomberg. 1998. Participatory design: Issues and concerns. *Computer Supported Cooperative Work* 7, nos. 3–4:167–185.

Kensing, F. J. Simonsen, and K. Bødker. 1998. Participatory design at a radio station. *Computer Supported Cooperative Work* 7, nos. 3–4:243–271.

Kyng, M. 1994. Scandinavian design: Users in product development. In *Proceedings of CHI'94,* 3–9. Boston: ACM Press.

Schön, D. A. 1992. Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems* 5:3–14.

Simonsen, J. 1999. How do we take care of strategic alignment? Constructing a design approach. *Scandinavian Journal of Information Systems* 11:51–72.

Simonsen, J., and F. Kensing. 1997. Using ethnography in contextual design. *Communications of the ACM* 40, no. 7:82–88.

Stolterman, E. 1992. How systems designers think about design and methods: Some reflections based on an interview study. *Scandinavian Journal of Information Systems* 4:137–150.

Suchman, L. 1983. Office procedure as practical action: Models of work and system design. *ACM Transactions on Office Information Systems* 1:320–328.

Suchman, L. 1994. Do categories have politics? The language/action perspective reconsidered. *Computer Supported Cooperative Work (CSCW): An International Journal* 2–3:177–190.

Suchman, L. 1995. Making work visible. *Communications of the ACM* 38, no. 9:56–64.

Winograd, T. 1994. Categories, disciplines, and social coordination. *Computer Supported Cooperative Work (CSCW): An International Journal* 2–3:191–197.

*Figure E.3. Course material*