PART I. DESIGN OF COMPUTER SUPPORT FOR COLLABORATION

Introduction to Part I: Studies of Technology Design

The 21 chapters of this book were written over a number of years, while I was finding my way toward a conception of group cognition that could be useful for CSCL and CSCW. Only near the end of that period, in editing the essays into a unified book, did the coherence of the undertaking become clear to me. In presenting these writings together, I think it is important to provide some guidance to the readers. Therefore, I will provide brief introductions to the parts and the chapters, designed to re-situate the essays in the book's mission.

Theoretical Background to Part I

The fact that the theory presented in this book comes at the end, emanating out of the design studies and the empirical analysis of collaboration, does not mean that the work described in the design studies of the first section had no theoretical framing. On the contrary, in the early 1990's when I turned my full-time attention to issues of CSCL, my academic training in computer science, artificial intelligence (AI) and cognitive science, which immediately preceded these studies, was particularly influenced by two theoretical orientations: situated cognition and domain-oriented design environments.

Situated cognition. As a graduate student, I met with a small reading group of fellow students for several years, discussing the then recent works of situated cognition (Brown & Duguid, 1991; Donald, 1991; Dreyfus, 1991; Ehn, 1988; Lave & Wenger, 1991; Schön, 1983; Suchman, 1987; Winograd & Flores, 1986), which challenged the assumptions of traditional AI. These writings proposed the centrality of tacit knowledge, implicitly arguing that AI's reliance on capturing explicit knowledge was inadequate for modeling or replacing human understanding. They showed that people act based on their being situated in specific settings with particular activities, artifacts, histories and colleagues. Shared knowledge is not a stockpile of fixed facts that can be represented in a database and queried on all occasions, but an on-going accomplishment of concrete groups of people engaged in continuing communication and negotiation. Furthermore, knowing is fundamentally perspectival and interpretive.

Domain-oriented design environments. I was at that time associated with the research lab of the Center for Life-Long Learning & Design (L^3D) directed by Gerhard Fischer, which developed the DODE (domain-oriented design environment) approach to software systems for designers (Fischer *et al.*, 1993; Fischer, 1994; Fischer *et al.*, 1998). The idea was that one could build a software system to support designers in a given domain—say, kitchen design—by integrating such components as a drawing sketchpad, a palette of icons representing items from the domain (stovetops, tables, walls), a set of critiquing rules (sink under a window, dishwasher to the right), a hypertext of design rationale, a catalog of previous designs or templates, a searching mechanism, and a facility for adding new palette items, among others. My dissertation system, `Hermes`, was a system that allowed one to put together a DODE for a given domain, and structure

different professional perspectives on the knowledge in the system. I adapted Hermes to create a DODE for lunar habitat designers. Software designs contained in the studies of part I more or less start from this approach: TCA was a DODE for teachers designing curriculum and CIE was a DODE for computer network designers.

This theoretical background is presented primarily in chapter 4. Before presenting that, however, I wanted to give a feel for the problematic nature of CSCL and CSCW by providing examples of designing software to support constructivist education (chapter 1), computational support for learning (chapter 2) or algorithms for selecting group members (chapter 3).

The Studies in Part I

The eight case studies included in part I provide little windows upon illustrative experiences of designing software for collaborative knowledge building. They are not controlled experiments with rigorous conclusions. These studies hang together rather like the years of a modern-day life, darting off in unexpected directions, but without ever losing the connectedness of one's identity, one's evolving, yet enduring personal perspective on the world.

Each study contains a parable: a brief, idiosyncratic and inscrutable tale whose moral is open to—indeed begs for—interpretation and debate. They describe fragmentary experiments that pose questions and that, in their specificity and materiality, allow the feedback of reality to be experienced and pondered.

Some of the studies include technical details that may not be interesting or particularly meaningful to all readers. Indeed, it is hard to imagine many readers with proper backgrounds for easily following in detail all the chapters of this book. This is an unavoidable problem for interdisciplinary topics. The original papers for part I were written for specialists in computer science, and their details remain integral to the argumentation of the specific study, but not necessarily essential to the larger implications of the book.

The book is structured so that readers can feel free to skip around. There is an intended flow to the argument of the book—summarized in these introductions to the three parts—but the chapters are each self-contained essays that can largely stand on their own or be visited in accordance with each reader's particular needs.

Part I explores, in particular ways, some of the major forms of computer support that seem desirable for collaborative knowledge building, shared meaning making and group cognition. The first three chapters address the needs of individual teachers, students and group members, respectively, as they interact with shared resources and activities. The individual perspective is then systematically matched with group perspectives in the next three chapters. The final chapters of part I develop a mechanism for moving knowledge among perspectives. Along the way, issues of individual, small-group and community levels are increasingly distinguished and supported. Support for group formation, perspectives and negotiation is prototyped and tested.

Study 1, TCA. The book starts with a gentle introduction to a typical application of designing computer support for collaboration. The application is the Teachers

Curriculum Assistant, a system for helping teachers to share curriculum that responds to educational research's recommendation of constructivist learning. It is a CSCW system in that it supports communities of professional teachers cooperating in their work. At the same time, it is a CSCL system that can help to generate, refine and propagate curriculum for collaborative learning by students, either online or otherwise. The study is an attempt to design an integrated knowledge-based system that supports five key functions associated with the development of innovative curriculum by communities of teachers. Interfaces for the five functions are illustrated.

Study 2, Essence. The next study turns to computer support for students, either in groups or singly. The application, State the Essence, is a program that gives students feedback on summaries they compose from brief essays. Significantly increasing students' or groups' time-on-task and encouraging them to create multiple drafts of their essays before submitting them to a teacher, the software uses a statistical analysis of natural language semantics to evaluate and compare texts. Rather than focusing on student outcomes, the study describes some of the complexity of adapting an algorithmic technique to a classroom educational tool.

Study 3, CREW. The question in this study is: how can software predict the behavior of a group of people working together under special conditions? Developed for the American space agency to help them select groups of astronauts for the international space station, the Crew software modeled a set of psychological factors for subjects participating in a prolonged space mission. Crew was designed to take advantage of psychological data being collected on outer-space, under-sea and Antarctic winter-over missions confining small groups of people in restricted spaces for prolonged periods. The software combined a number of statistical and AI techniques.

Study 4, Hermes. This study was actually written earlier than the preceding ones, but it is probably best read following them. It describes at an abstract level the theoretical framework behind the design of the systems discussed in the other studies—it is perhaps also critical of some assumptions underlying their mechanisms. It develops a concept of situated interpretation that arises from design theories and writings on situated cognition. These sources raised fundamental questions about traditional AI, based as it was on assumptions of explicit, objective, universal and rational knowledge. Hermes tried to capture and represent tacit, interpretive, situated knowledge. It was a hypermedia framework for creating domain-oriented design environments. It provided design and software elements for interpretive perspectives, end-user programming languages and adaptive displays, all built upon a shared knowledge base.

Study 5, CIE. A critical transition occurs in this study, away from software that is designed to amplify human intelligence with AI techniques. It turns instead toward the goal of software designed to support group interaction by providing structured media of communication, sharing and collaboration. While TCA attempted to use an early version of the Internet to allow communities to share educational artifacts, CIE aimed to turn the Web into a shared workspace for a community of practice. The specific community supported by the CIE prototype was the group of people who design and maintain local area computer networks (LANs), for instance at university departments.

Study 6, WebGuide. WebGuide was a several-year effort to design support for interpretive perspectives, focusing on the key idea proposed by Hermes, computational perspectives, and trying to adapt the perspectivity concept to asynchronous threaded discussions. The design study was situated within the task of providing a shared guide to the Web for small workgroups and whole classrooms of students, including the classroom where Essence was developed. Insights gained from adoption hurdles with this system motivated a push to better understand collaboration and computer-mediated communication, resulting in a WebGuide-supported seminar on mediation, which is discussed in this study. This seminar began the theoretical reflections that percolate through part II and then dominate in part III. The WebGuide system was a good example of trying to harness computational power to support the dynamic selection and presentation of information in accordance with different user perspectives.

Study 7, Synergeia. Several limitations of WebGuide led to the Synergeia design undertaking. The WebGuide perspectives mechanism was too complicated for users, and additional collaboration supports were needed, in particular support for group negotiation. An established CSCW system was re-designed for classroom usage, including a simplified system of class, group and individual perspectives, and a mechanism for groups to negotiate agreement on shared knowledge-building artifacts. The text of this study began as a design scenario that guided development of Synergeia and then morphed into its training manual for teachers.

Study 8, BSCL. This study takes a closer look at the design rationale for the negotiation mechanism of the previous study. The BSCL system illustrates designs for several important functions of collaborative learning: formation of groups (by the teacher); perspectives for the class, small work groups and individuals; and negotiation of shared knowledge artifacts. These functions are integrated into the mature BSCW software system, with support for synchronous chat and shared whiteboard, asynchronous threaded discussion with note types, social awareness features, and shared workspaces (folder hierarchies for documents). The central point of this study is that negotiation is not just a matter of individuals voting based on their preconceived ideas; it is a group process of constructing knowledge artifacts and then establishing a consensus that the group has reached a shared understanding of this knowledge, and that it is ready to display it for others.

The chapters of part I demonstrate a progression that was not uncommon in CSCL and CSCW around the turn of the century. A twentieth century fascination with technological solutions reached its denouement in AI systems that required more effort than expected and provided less help than promised. In the twenty-first century, researchers acknowledged that systems needed to be user-centric and should concentrate on taking the best advantage of human and group intelligence. In this new context, the important thing for groupware was to optimize the formation of effective groups, help them to articulate and synthesize different knowledge-building perspectives, and support the negotiation of shared group knowledge. This shift should become apparent in the progression of software studies in part I.