ELSEVIER

Automation in Construction 7 (1998) 465–473

AUTOMATION IN CONSTRUCTION

Collaboration and coordination in architectural design: approaches to computer mediated team work

Mark D. Gross [a,d,*], Ellen Yi-Luen Do [a,c], Raymond J. McCall [a,d], Wayne V. Citrin [b,d], Paul Hamill [b], Adrienne Warmack [a], Kyle S. Kuczun [a]

[a] Sundance Laboratory for Computing in Design and Planning, College of Architecture and Planning, University of Colorado at Denver, Boulder, CO, USA

[b] Department of Electrical and Computer Engineering, University of Colorado, Boulder, CO, USA

[c] College of Architecture, Georgia Institute of Technology, Atlanta, GA, USA

[d] Institute of Cognitive Science, University of Colorado, Boulder, CO, USA

Abstract

The paper reports on three projects at our laboratory that deal respectively with synchronous collaborative design, asynchronous collaborative design, and design coordination. The Electronic Cocktail Napkin and its mobile extension that runs on hand-held computers supports synchronous design with shared freehand drawing environments. The PHIDIAS hypermedia system supports long-term, asynchronous collaboration by enabling designers of large complex artifacts to store and retrieve rationale about design decisions and the Construction Kit Builder (CKB) supports team design by supporting *a priori* agreements among team members to avoid conflicts. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Synchronous collaboration; Asynchronous collaboration; Coordination

1. Introduction: collaboration in architectural design

In 1993 and 1994, instructors and students of architecture at several universities around the world [1] collaborated briefly on two ‘virtual design studio’ projects. Using off-the-shelf technology of the time —e-mail, CU-See-Me internet video, international conference calls, and exchange of CAD drawings, images, and Quicktime animations—this ambitious project explored the possibility of bringing together diverse members of an international design team together to collaborate on a short term (2-week) project. Central to the ‘Virtual Design Studio’ was a ‘digital pin-up board’, an area where participating designers could post and view drawings and textual comments; video links and e-mail exchange provided the media for direct communication media about designs.

A report on the project [1] makes clear that the process was not without technical difficulties: a significant amount of communication concerned scheduling and coordinating file formats; disappointingly, little was devoted to discussions of design

* Corresponding author.

[1] Hong Kong University, Escola Técnica Superior d’Arquitectura de Barcelona, University of British Columbia, Washington University, Harvard Graduate School of Design, and Massachusetts Institute of Technology.

PII S0926-5805(98)00055-7

issues. Although it is clear that many of the minor technical problems that inevitably plague a forward-looking effort like the Virtual Design Studio will be solved in the near term, the project also reveals the need for research on software and design practices to make computer-mediated design collaboration realize its attractive promise.

William J. Mitchell, dean of the MIT School of Architecture and Planning, and a participant in the Virtual Design Studio experiments, identified four developing technologies that underlie the computer-mediated collaboration that the Virtual Design Studio experiment heralds. These are: pervasive computer networking, digital video, the integration of video with computation, and hand-held wireless digital communication [2]. Mitchell sees these components changing the prevailing paradigm in computer-aided design from traditional computer graphics involving a single designer interacting with a machine to construct CAD drawings and models, to a process of computer-mediated negotiation among multiple players.

Mitchell correctly identifies key hardware technologies that are making international design collaboration possible; most projects (including ours) in collaborative CAD do employ one or more of these technologies. However, the Virtual Design Studio project also makes it clear that effective design collaboration demands more than merely connecting the members of a team with the highest possible bandwidth. Successful collaborative design also requires attention to the organization of design process and product, that is, to the methods and representations used in design. In short, computer-mediated collaboration will not succeed on the back of technology alone.

Therefore, to Mitchell's four technologies, we add three observations about their effective use in collaborative design. First, the interface that a collaborative design system presents to its users is a critical determinant of its potential for success. Designers lack extensive and sophisticated computing experience; their productivity will correlate directly with the usability of the interface. Ideally, a collaborative design system should present as simple an interface as the designer's familiar pencil and paper. However, it must also make it easy for a designer to access the extended capabilities that computer support can provide: constraint checking, simultaneous work with (possibly remote) other designers, filing and indexing, iterative changes and versioning, etc. As drawing remains a primary means of communication among designers, we have begun there. In our discussion of synchronous collaborative work, we discuss drawing as an interface to collaborative design.

Second, as designers work, they produce a lot of data and it is essential to find ways to capture, structure, and index this data so that all members of a team can use it. As one participant comments, ''In the VDS project a mountain of files were generated by participants from different universities. It was a difficult task determining which and how some of these files related to each other. (Which floor plan goes with which section or surface model?)'' [1] (Renato Garcia, p. 34). More is needed than a public area where collaborators can post their work. Designers need to be able to quickly, easily, and in some cases automatically construct links and annotations among these postings that capture the relationships among the individual drawings, photographs, and comments that comprise the collaborative design document. We take these issues up below in discussing our work on asynchronous collaboration and the construction of digital design databases of artifacts and argumentation.

Finally, a design team must coordinate its efforts explicitly: the effective functioning of a team requires designers to agree to work in certain ways. At a basic level, the team must determine protocols for communication (turn-taking, file formats, scheduling). At the level of the design artifact, the team must agree about areas of design responsibility—who will make what decisions and what rules shall govern the decisions designers make? We take up these issues below, in describing our work on design coordination.

In the following sections, we report on three projects that deal respectively with synchronous collaborative design, asynchronous collaborative design, and design coordination. In each of these sections, we provide an overview of the project, directing the interested reader to our more detailed reports on the work. We conclude with a description of several current efforts that suggest connections among these projects, and perhaps a way to incorporate them into a larger framework for collaborative design.

2. Synchronous collaborative design

Same time (synchronous) collaboration in design is perhaps the most obvious form of collaboration that computers can support and extend. In traditional architectural design settings, design team members sit together to hear a presentation, discuss design issues, and sometimes, sketch a preliminary design that can then be carried out in detail by one or more team members after the meeting. Traditionally, these meetings take place around a conference table, but recently, technologies such as video teleconferencing, fax, Liveboard, and computer-mediated meeting spaces have made it possible to hold meetings for synchronous collaboration among design team members who are geographically dispersed. The earliest efforts to support synchronous collaboration in architectural design employed video links between collaborating designers' offices [3]; several other recent projects have focused on integrating live video of distributed team members with shared drawings and text [4–6]. Like others [7,8], we have concentrated on freehand drawing because we believe it is a natural and familiar way for designers to interact, especially in the early stages of design.

The Electronic Cocktail Napkin is a program that supports collaborative freehand sketching and drawing on digitizing pads with pens [9]. It combines paint and draw features: users draw whatever they want, unrestricted by menus of graphic primitives yet marks users make can be selected, dragged, resized, and rotated, and combined into groups or configurations. The Cocktail Napkin supports simulated tracing paper and underlays, constraint based drawing, sketchbooks, and a pin-up bulletin board. Unlike earlier shared drawing programs, the Napkin also provides trainable recognition, user programmable parsing, and contextual interpretation of diagrams as input to simulation programs and visual libraries. Thus, diagrams and sketches are primary means of entering designs into 'knowledge based' tools and arguing about them (Fig. 1).

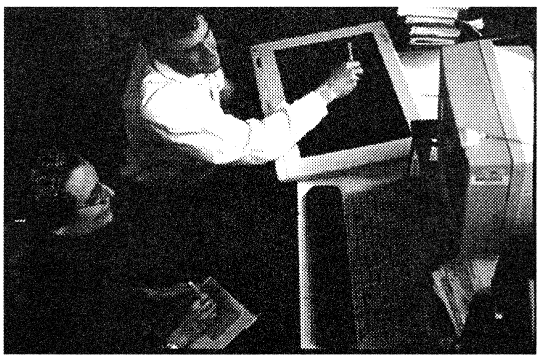

Fig. 1. Designers engaged in informal sketching with the Electronic Cocktail Napkin.

Fig. 2. Collaborative drawing with a wired Mac and a wireless PDA connected by a radio frequency modem.

The Cocktail Napkin program provides several collaborative drawing modes. First, two designers can share a single drawing surface—a digitizing tablet. The designers use digitizing pens that can be distinguished by the software and the program tracks authorship of different parts of the drawing. This mode is similar to two designers sketching together on a cocktail napkin or the back of an envelope in an informal brainstorming session. In a second mode for collaborative drawing, the designers still share a drawing space, but each holds their own tablet and pen. The third mode involves two programs running on separate machines, connected (point–point) over a local area network. In this mode, both designers run a version of the program on their own machines and the programs exchange drawing and editing commands using a protocol we devised for graphics interchange. We have also built a version of the collaborative drawing program that runs between a hand-held wireless PDA (personal digital assistant: an Apple Newton) and a host, or between two PDAs, providing a portable digital sketchbook, connected with a central database or with other designers (Fig. 2).

3. Asynchronous collaborative design

A rather different kind of support for collaborative design involves asynchronous work. Short-term asynchrony enables people in different time zones or on different schedules to participate at their own convenience, rather than coordinating schedules to hold an on-line design meeting at a specific time. Long-term asynchrony enables the collaboration of designers over an extended design calendar and over the life cycle of a product, allowing designers who join the team later to gain access to design rationale of designers who participated earlier, but who have left the project or moved on to other issues. To support long-term asynchronous collaboration, it is useful to provide an archive or repository for design rationale and the means for designers to record the reasons for certain decisions and to retrieve rationale stored previously by others.

The PHIDIAS hypermedia system [10–12] is an example of support for long-term asynchronous collaboration. PHIDIAS—Procedural Hierarchy of Issues Design Intelligence Augmentation System—enables designers of large complex artifacts to store and retrieve rationale about design decisions. Storing and retrieving design rationale enables designers engaged later in the design life cycle to understand the reasons for what might otherwise be obscure decisions taken by earlier designers.

Commercial products such as Microstation's TeamMate™ and Autodesk's WorkCenter for the Web® offer document and work flow management, including shared use of drawings, version and update control. TeamMate™ uses ODBC (Open Data Base Connectivity) to support check-in and check-out of documents, and change notification of text files and drawings, at the document level. Likewise, AutoDesk's WorkCenter helps design teams manage the access, cataloguing, tracking, sharing, and distribution of project documents. By contrast, PHIDIAS (like all hypermedia systems) is concerned with links in a collection of design rationale, and has no concept of a document: PHIDIAS operates at the level of words, sentences, and paragraphs. PHIDIAS is designed to handle discussion about design projects, not just to manage the design documents. In addition, PHIDIAS does not attempt to manage workflow at all.

The PHIDIAS system is organized around Horst Rittel's scheme of an Issue Based Information System (IBIS) [13]: information about an artifact is organized as a hierarchy of issues, answers, and arguments (Fig. 3). PHIDIAS differs from other issue based information systems such as gIBIS [14] and SYBL [15] in its granularity—it is a fine grained system—and that it stores not only textual information, but also drawings, photographs, and sequences of digital audio and video as parts of the design argument. Earlier work on PHIDIAS has resulted in a single-user system that can be used to enter new design rationale in the form of issues, answers, and arguments. It included a structured editor for textual design rationale, facilities for producing three-dimensional models and entering audio and video as nodes in the hypermedia design argument structure. The current PHIDIAS system has been extended to respond dynamically to queries received over the worldwide web, providing design rationale on-the-fly to Java-enabled clients.

PHIDIAS has been used to construct a large issue base of design rationale about space-based habitation, providing NASA's Man Space Information System documents in an electronic format, structured as an issue base. To assist users in locating relevant information in this enormous issue base, the system supports 'argumentative agents' that detect overlaps in the concerns of different participants in a design process, notify these participants of overlapping con-

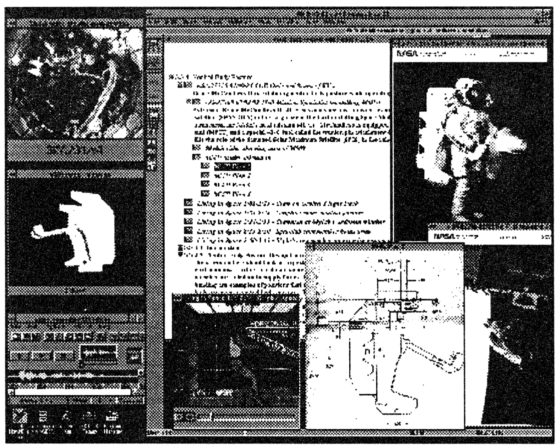

Fig. 3. The PHIDIAS HyperCAD system structures a database of design artifacts and argumentation.

cerns, and enable and support sustained communication among these people to deal collaboratively with the overlaps. Three kinds of argumentative agents have been devised: advocates, knowledge-based critics that examine and critique partially-formed solutions; scouts, which watch for and report on new information entered anywhere in the issue base; and reporters, which report on activity in certain parts of the issue base (for example, for attempts by other designers to change parts of a design).

4. Design coordination

Orthogonal to the synchronous–asynchronous distinction of collaborative design outlined above is the approach of design coordination. Our Construction Kit Builder (CKB) project explores this way of supporting team design under the premise that *a priori* agreements among team members can avoid many conflicts that otherwise might arise and need to be resolved during designing.

Complex artifacts, e.g., buildings, are assemblies of many different systems, each with its own particular characteristics, and each the responsibility of different design team members. For example, one designer might be in charge of laying out partition walls, while another is to design the electrical distribution system, and a third will be responsible for laying out the system of heating and ventilating ducts. In a conventional CAD process, each subsystem layout will be done separately, then combined together (for example, on separate layers of a CAD drawing) and checked for interference conflicts either by hand or using a 3D interference checker built into the CAD program. This process postpones the discovery of conflicts until most of the design work is complete, and therefore, resolutions and repairs are likely to be ad hoc and/or costly. Even if the designers were networked, sharing a single CAD database, physical interference problems could not be detected until the second component is placed, and resolutions would still be negotiated in a one off manner.

To avoid this sort of conflict, the design team members must reach agreements about the placement of physical components into the design in the first place. Before beginning to lay out the design, the team must agree on allowable locations for the components of each system. As many of the systems are pervasive—ventilating ducts must reach all parts of the building—it is useful to establish a scheme of three-dimensional zones that can be allocated to the components of the various systems. For example, a zone that occurs every 12 ft is allocated to carry the ventilating ducts. Many of the ad hoc interference conflicts can be avoided by using dedicated spatial channels for each of the systems that must be laid out in the building. Any interference conflicts that do occur will happen at zone intersections, where the condition can be anticipated and a standard resolution designed.

To support this scheme, each designer's CAD editor must be programmed with the rules for placement and assembly of their particular system. The HVAC designer who is laying out a system of ducts, fans, and vents works with a standard catalog of components that go together in certain ways, and that (according to the team's agreed-upon rules) can be placed only in certain zones in the building. By programming the CAD editor with these rules, the HVAC designer can proceed fairly independently of other system designers, knowing that interference conflicts will be minimized and will only occur in certain locations.

We have built a program to demonstrate these ideas. Construction Kit Builder (CKB) [16,9,17] is a CAD program that operates at two levels. At the lower (layout) level, it is simply a design editor in which a designer can select components from a catalog and lay them out to make a design. However, the designer is restricted to placing elements only in certain locations (zones) and assembling them in certain ways. At the higher (coordination) level, Construction Kit Builder enables the design team coordinator to assign placement and assembly rules to the components of each system (for example, restricting ventilating components to one set of zones, and electrical components to another) (Fig. 4).

Design rules that enforce the placement of different systems' components in different zones, and the assembly of the components of each system are implemented using algebraic constraints, which apply locally when the designer lays out components. Placement constraints are attached to systems and inherited by individual components and assembly

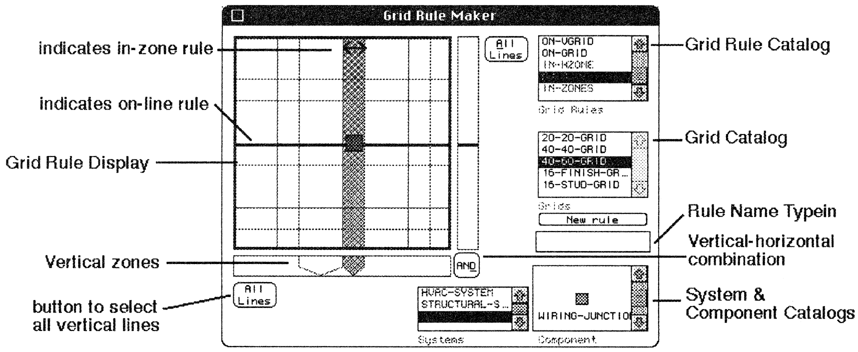

Fig. 4. Construction Kit Builder provides a design team the means to express and enforce rules to coordinate the placement and assembly of the components of different systems.

constraints apply to specific pairs of components. The constraint machinery is simple propagation, though more sophisticated techniques naturally could be applied. But sophisticated constraint solving is not the point. Rather, we aim to show how the layout concerns of a design team can be partitioned based on simple a priori agreements about the placement of components, thereby sidestepping what otherwise might later become thorny conflicts.

To be sure, other kinds of conflicts than spatial ones occur in a building design process, for example originating from trade-offs among various functional requirements. Construction Kit Builder makes no attempt to represent functional requirements or indeed any requirements other than rules about placement and assembly. In that sense, the program leaves all management of design goals and objectives in the hands of the layout designer. The management of spatial layout rules could be augmented by tools that support other aspects of designing. For example, a pipe routing system could be used to suggest optimal paths, or a structural simulation could ensure that positions and dimensions chosen for columns and beams meet load requirements. But the aim of CKB is merely to manage the spatial constraints, and leave functional decision-making to the designer.

5. Current and future work

Several interesting connections between these originally independent projects have emerged and we are exploring the potential for combinations, some described below. For example, in HyperSketch, we explore the use of freehand sketches as nodes in a hyper document. In Retrieving Cases with Diagrams, we look at how drawings can be used as queries to large design databases. And in Digital Design Sketchbooks, we combine our work on collaborative drawing work a design archive to support asynchronous collaboration.

5.1. Hypersketch—drawings as nodes in a CAD hyper document

We have observed that in the course of a design session, a single designer may produce as many as a hundred sketches, generated sequentially. In team work, several designers may work at this activity more or less in parallel, and come together from time to time to compare notes and integrate their designs. Sketches are linked conceptually: often each successive sketch made by a single designer responds to some perceived problem or opportunity in the previous drawing. In McCall's HyperSketch prototype, sketches are stored as nodes in a hyper document, with links that indicate the sequence in which they were made (Fig. 5). Additional links can also indicate relationships among sketches such as ‘design alternative’, ‘fixes problem in’, ‘elaboration of’, and ‘abstraction of’. HyperSketch makes it easy for the designer to specify these and other relationships among sketches. Members of a design team working separately can browse the sketches others have made and establish labeled links between these sketches and their own. The sketches themselves can be anno-

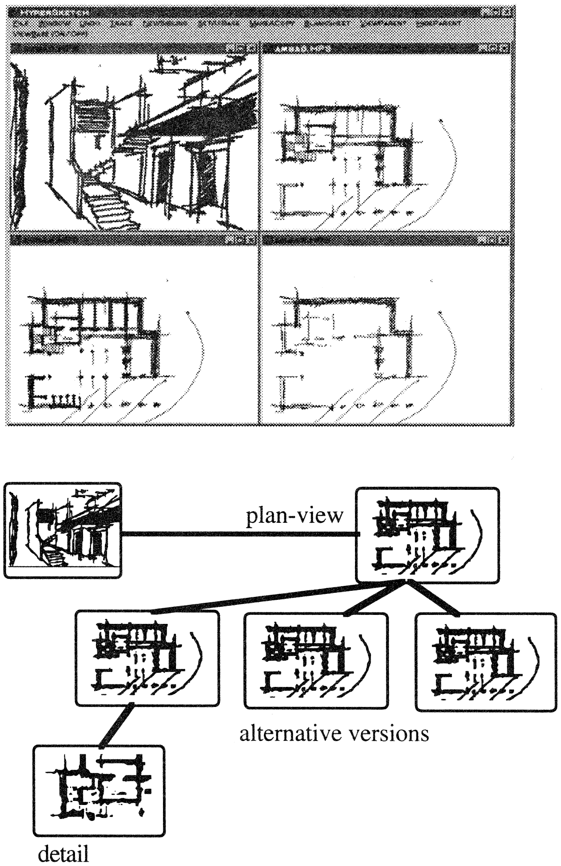

Fig. 5. In HyperSketch, designers' freehand drawings are linked as nodes in a hyperdocument graph. (Above: designers' sketches made in HyperSketch; Below: hypermedia structure).

tated with text (e.g., arguments about and comments on the designs), and the resulting structure of linked and labeled sketches then shared as a repository of information about the design.

Commercial redlining products (e.g., Autodesk's WorkCenter, Radian Systems ImageDOCS, and Intergraph's DM/Redline) enable a designer to add text and graphic annotations to drawing documents without modifying the actual file. Some products (e.g., Microstation Review) support pen based interface to allow quick freehand annotations. The functionality prototyped in Hypersketch would add to these products' functionality full, typed link service between hand-drawn annotations on various drawings, as well as links from a sketch drawing to supporting design rationale documents.

5.2. Retrieving cases with diagrams

We have used the Electronic Cocktail Napkin to build a query-by-diagram retrieval scheme for databases of designs and we have used the scheme to index case bases of architectural post-occupancy evaluation studies [18] and on-line libraries of technical data about heating, ventilating, and air-conditioning (HVAC) [19]. For data in which spatial relations or physical form is salient, designers may prefer to construct queries by drawing diagrams, rather than constructing a textual query from key-

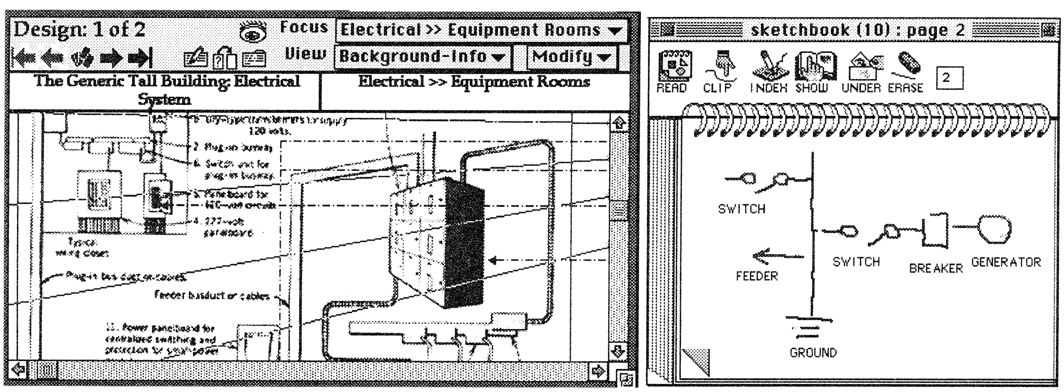

Fig. 6. In Retrieving Cases with Diagrams, sketches and diagrams are used to index relevant information from on-line design databases.

words. Our scheme employs simple visual bookmarking: A designer first draws diagrams to index items in the case library. Later, those cases may be retrieved by drawing similar diagrams (Fig. 6).

In a large collaborative design database, sketches, diagrams, drawings, and photographs will comprise a significant fraction of the data. We can apply the visual bookmarking scheme we developed for querying case libraries to design databases that are constructed collaboratively by a design team. We see query by sketch as a valuable addition to text based search and retrieval for designers participating in a large collaborative project to find and retrieve design information that others have stored in the database.

5.3. Digital Design Sketchbooks and mobile, wireless, graphical communication

We have experimented with a wireless mobile sketchpad communicating with a wired host computer to support a team of distributed, collaborating designers [20,21]. Each design team member works with a Digital Design Sketchbook, a hand-held Personal Digital Assistant (PDA). We are currently using the Apple Newton Message Pad 130 with a wireless modem. The Newton and a Macintosh running the Cocktail Napkin program communicate using a wireless (radio frequency) modem connection speaking Appletalk. A web site for the design team serves as a shared repository and design history for drawings, written comments, and photographs, contributed by each design team member Fig. 7(a). Drawings and text from the web site can be downloaded to the PDA, annotated locally, and the designer's marked-up version uploaded to the web site for others to consider (Fig. 7a,b and c).

In our prototype, the mobile PDAs running our SmartPad program communicate with the Electronic Cocktail Napkin software running on a 'host' Macintosh. When the Napkin program receives a drawing from one of the mobile PDAs, it saves the drawing as a GIF file Fig. 7(b), and copies the file to a special directory on the web server. A CGI on the web server polls this directory and whenever a new file appears it adds the file to the design team's web page. A simple web browser running on the PDA would enable a user to retrieve any drawings, photographs, or text that other team members have posted; the present version can only return drawings originally made on the PDA or the Napkin. In a future version, we plan to replace the generic web serving software with a PHIDIAS-like server that can help structure the emerging design as the team works.

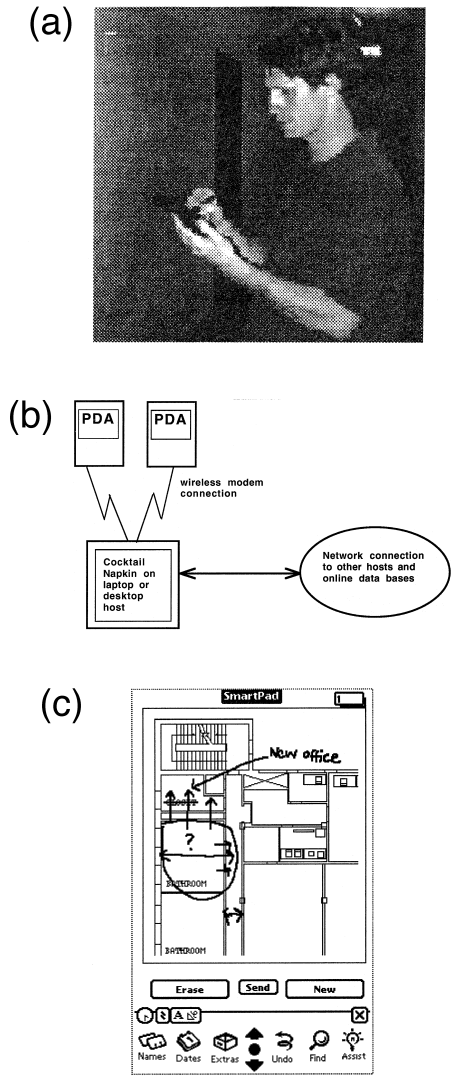

Fig. 7. (a) Documenting on-site conditions; (b) Web server for the design team stores drawings, comments, and other data the PDA provides; (c) Designer's marked-up drawing on the PDA, uploaded to the server.

5.4. Summary

We have outlined three approaches to computer support for collaborative design that we are exploring—shared drawing, an archive of rationale, and coordination of decision-making. Correspondingly, we believe—drawing is a primary medium in many design domains, design tools should support not only construction of the artifact but also the argument about the artifact, and that computer tools should help design teams, as well as manage and work within explicit agreements about the design. The interplay between these three approaches, and their execution with various hardware and software technologies—mobile sketchpads, Java clients and backend servers, promise a fascinating, if fast-changing, research program in collaborative CAD.

Acknowledgements

We gratefully acknowledge the following support for the projects described in this report: NSF grant, DMII 93-1316, 'Avoiding interference conflicts in architectural subsystem layout—a constraint based approach'; a NASA SBIR contract with Johnson Engineering, Boulder, 'Using hypermedia and the information superhighway to improve design of spacecraft interiors'; NSF grant IRI 96-19856, 'Back of an envelope'; a grant from the Colorado Advanced Software Institute and US West Advanced Technologies, 'PDA-Based architectures for graphical interchange'; A University of Colorado President's Educational Technology Grant, 'Toolkit for technology enhanced education'.

References

[1] J. Wojtowicz (Ed.), Virtual Design Studio, Hong Kong Univ. Press, Hong Kong, 1995.
[2] W.J. Mitchell, The Future of the Virtual Design Studio, in: J. Wojtowicz (Ed.), Virtual Design Studio, Hong Kong Univ. Press, Hong Kong, 1995, pp. 51–59.
[3] S. Harrison, Computing and the social nature of design, ACADIA Q. 12 (1) (1993) 10–18.
[4] S. Bly, S. Harrison, S. Irwin, Media spaces: bringing people together in a video, audio and computing environment, Commun. ACM 36 (1) (1993) 26–45.
[5] H. Ishii, M. Kobayashi, J. Grudin, Integration of interpersonal space and shared workspace: Clearboard design and experiments, in: S. Greenberg, S. Hayne, R. Rada (Eds.), Groupware for Real-Time Drawing, McGraw-Hill, London, 1995, pp. 96–125.
[6] S.A.R. Scrivener, D. Harris, S.M. Clark, T. Rockoff, M. Smyth, Designing at a distance via real-time designer-to-designer interaction, in: S. Greenberg, S.Hayne, R. Rada (Eds.), Groupware for Real-Time Drawing, McGraw-Hill, London, 1995, pp. 5–23.
[7] S. Greenberg, S. Hayne, R. Rada, Groupware for Real-Time Drawing, McGraw-Hill, London, 1995.
[8] S.L. Minneman, S.A. Bly, Managing à trois: a study of a multi-user drawing tool in distributed design work, Conference on Human Factors in Computing Systems (CHI'91), ACM Press/Addison Wesley, New Orleans, LA, 1991, pp. 217–224.
[9] M.D. Gross, The electronic cocktail napkin—working with diagrams, Design Stud. 17 (1) (1996) 53–70.
[10] R. McCall, PHIBIS: procedurally hierarchical issue based information systems, Proceedings, International Congress on Planning and Design Theory, American Society of Mechanical Engineers, 1987.
[11] R. McCall, P. Bennett, P. d'Oronzio, J. Ostwald, F. Shipman, N. Wallace, PHIDIAS: A PHI-based design environment integrating CAD graphics into dynamic hypertext, European Conference on Hypertext, 1990.
[12] R.J. McCall, P. Bennett, E. Johnson, An Overview of the Phidias II HyperCAD System, in: A. Harfmann, M. Fraser (Eds.), ACADIA (Association for Computer Aided Design in Architecture), pp. 63–74.
[13] W. Rittel, W. Kunz, Issues as elements of information systems, Working Paper 131, Center for Planning and Development Research, University of California, Berkeley, 1970.
[14] J. Conklin, M. Begeman, gIBIS: a hypertext tool for exploratory policy discussion, Conference on Computer-Supported Cooperative Work, Association for Computing Machinery, 1988, pp. 140–152.
[15] J. Lee, SYBL: a tool for managing group decision rationale, Conference on Computer-Supported Cooperative Work, ACM, 1990, pp. 79–92.
[16] M.D. Gross, Avoiding conflicts in architectural subsystem layout, Concurrent Eng.: Res. Appl. 2 (1994) 163–171.
[17] M.D. Gross, Why can't CAD be more like Lego?, Automation Construction 5 (1996) 285–300.
[18] M.D. Gross, C. Zimring, E. Do, Using diagrams to access a case base of architectural designs, in: J. Gero (Ed.), Artificial Intelligence in Design '94, Kluwer, Lausanne, 1994, pp. 129–144.
[19] E.Y.-L. Do, M.D. Gross, Reasoning about cases with diagrams, Third Congress on Computing in Civil Engineering, American Society of Civil Engineers, 1996, pp. 314–320.
[20] W.V. Citrin, M.D. Gross, Distributed architectures for pen-based input and diagram recognition, ACM Conference on Advanced Visual Interfaces, ACM, Gubbio, Italy, 1996, pp. 132–140.
[21] W.V. Citrin, M.D. Gross, PDA-based graphical interchange for field service and repair workers, Comput. Graphics 20 (5) (1996) 641–650.