

The Denver Model

for Groupware

Design

Yoooooooo Haaaaaaa

Tony Salvador, Jean Scholtz, James Larson*

with

Kathleen Aubrey, Alain Derycke, Govert deVries, Rodney Fuller, John Gintell,

Susan McDaniels, Andrew Monk, Saila Ovaska, Paul Resnick,

* contact: Jim Larson, Intel Corporation, 2111 NE 25th Ave. Hillsboro, OR 97229, USA

jim_larson@ccm.jf.intel.com

The Denver Model For Groupware Design

Tony Salvador, Jean Scholtz, James Larson

with

Kathleen Aubrey, Alain Derycke, Govert deVries, Rodney Fuller, John Gintell,

Susan McDaniels, Andrew Monk, Saila Ovaska, Paul Resnick,

Peter Rowley, Daniel Salber, Daniel Suthers

Abstract

The Denver Model is offered as a framework with which to plan or evaluate the capabilities associated with a particular groupware application. This model was the output of 14 participants at the two day workshop on Designing and Evaluating Groupware, held at CHI'95, Denver Colorado. The Denver Model consists of three submodels: goals & requirements, design and technology. A description of the framework is provided and evaluation strategies are described in this paper.

Introduction

There were two purposes of this workshop, the results of which are reported here. First, the Denver Model attempts to organize into a single, coherent framework, the parameters associated with groupware design. In describing this model, we have striven to be both coherent and brief given the available space and time; thus certain details have been omitted from this paper. Second, issues related to the evaluation of groupware applications are described. The last section offers a brief description of the workshop format with some strengths and limitations for others who are considering organizing workshops.

The Denver Model

The Denver Model is a nested collection of models describing the generic elements of any groupware application. The first model consists of three submodels describing three aspects of constructing and reviewing groupware applications: requirements, design and technology. These are illustrated as planes in Figure 1. Within each submodel are second level submodels (called categories) that describe the major aspects of each first level submodel. Categories are represented as areas within the planes of Figure 1. This paper presents several candidate categories. Other candidates are certainly possible; we encourage the groupware community to investigate alternative categories.

When using the constructs of the Denver Model to describe an existing groupware system, the following is relevant. The uppermost plane represents the system goals and requirements for the groupware application either provided explicitly by the product developer, or inferred implicitly by the people describing the system. The central plane represents the actual functional design of the system and the lower plane represents the implementation of the system using available technology. The lowest plane represents the technology available to implement the capabilities in the central plane.

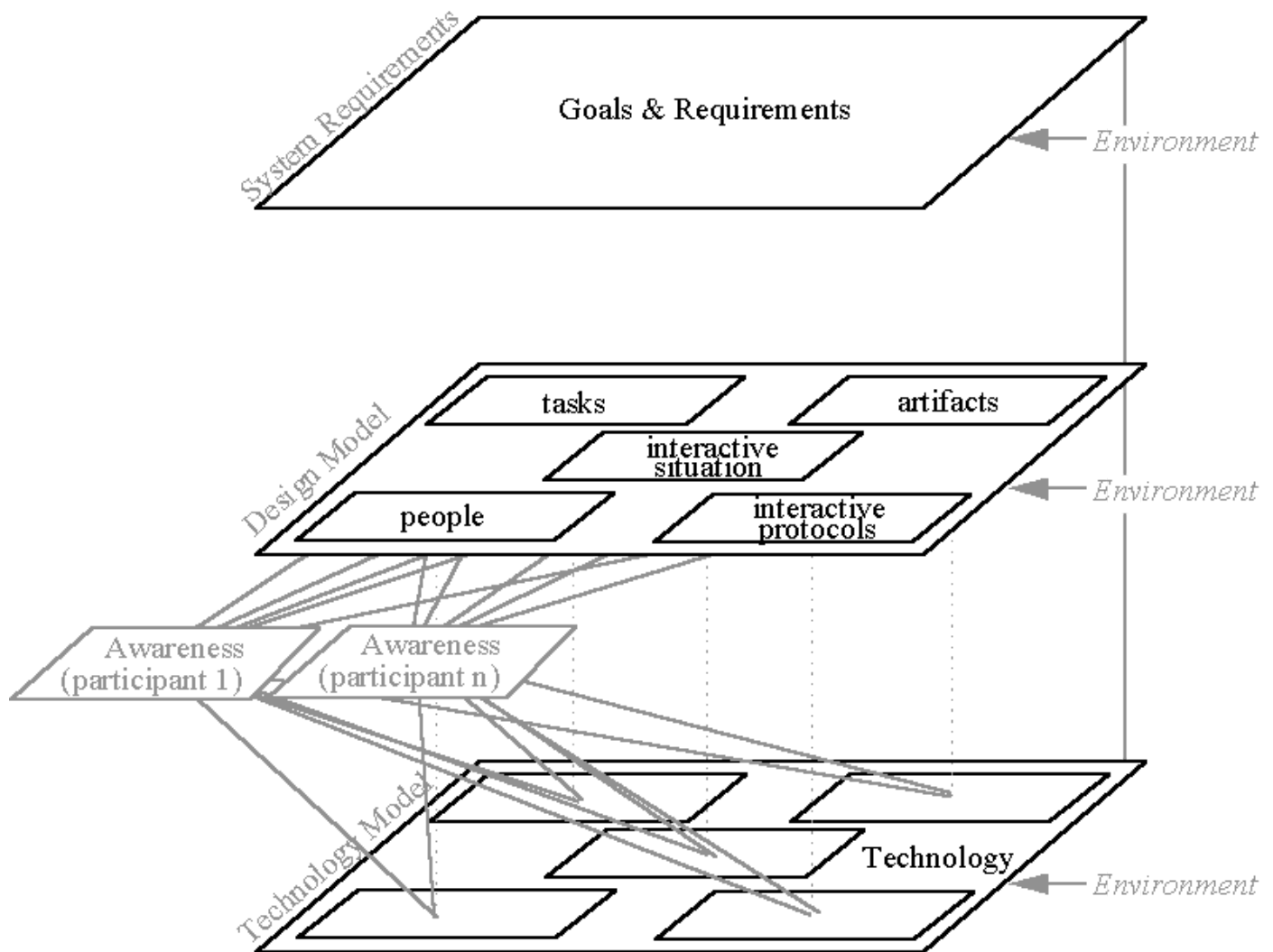


Figure 1. The Denver Model for Groupware Design

Finally, the specific content of each plane can be influenced by external environmental factors such as cost and due date (requirements), external communication, eagerness of customers and/or users to adopt the application, the possibility of interaction with other software applications, and other activities (functional design) and communications infrastructure such as networks, telephony networks, internet connectivity, processor power, etc. (technology/implementation).

The Denver Model: Description of the Design Submodel

The design submodel illustrates a framework for groupware as well as identifying those characteristics that uniquely define groupware applications as separate from single use, single user applications.

Groupware applications can be characterized by 5 categories related to: people, artifacts, tasks & activities, interactive situations and social protocols. Two of these categories, artifacts and tasks, are well known from single user, single use applications. The following five subsections systematically describe each of the five information categories.

People

The characteristics of people include items such as their names, appearance, voice, addresses, phone numbers, primary language, signature, their culture, their business, their interests, etc.

The roles people play in a group include their business status, cultural status, their technical expertise, their operational expertise with respect to the groupware application, etc. As pertains to design, users interact with each other and with the system depending upon their role in the group. More importantly, people assume different roles in different groups.

The characteristics of groups include the following, which range from stable and homogenous to unstable (changing) and heterogeneous: represented disciplines, group versus individual goals, stage in project/project cohesion, company(ies) represented, culture(s) and language(s) represented, group size, order and status in the group. The exact techniques to represent people, groups, and the role(s) of people in groups in a groupware application is an ongoing research question.

Artifacts

Artifacts refers to those objects produced or consumed during the interaction. There are five generic artifact types: text, sound, temporal image, static image and computational elements. Text refers to written language and the range of written languages. Sound refers to any information presented aurally; this can include real time voice, stored voice as in a voice mail system, music, etc. Temporal images refer to images that change over time without active human intervention, e.g., video clips, animations, etc. Static images refer to non-text graphic images that remain stable over time, e.g., a bitmaps and object drawings. Computational elements refer to artifacts that have computational capabilities, e.g., a spreadsheet or interactive electronic form. Compound artifacts may be constructed by combining elemental artifacts. Artifacts may also exhibit certain generic attributes, such as: cotemporality, revisability, reviewability and quality.

Artifacts may be described using any of the popular data modeling techniques including entity relationship diagrams, relational schemes, etc. Normally, there is also a set of generic attributes describing these artifacts, such as name, file size, file type, etc. In addition, artifacts can possess attributes associated with ownership, e.g., an static image annotation can be described by the relationship of it to it's owner and to it's text artifact.

Tasks & Activities

Four levels of Tasks & Activities represent this category: Goals, Tasks/Scenarios, Activities and Operations.

Goals refer to high level work goals that guide all behaviors in the workplace and to a large extent can determine culture and specific characteristics of products that might work well in that environment. For example, a standing goal might be to control costs by reducing the onsite workforce. In this case, products that support the "stay at home" employee may be particularly useful.

Tasks represent the high level representation of the type of work that occurs in that environment. For example, the tasks below represent a sampling of work settings.

- air traffic control, process control
- emergency communications
- negotiate an agreement
- teach a course
- author a paper
- specify a design
- analyze and interpret data
- instrument control decisions (advising)
- provide advice and expertise remotely
- purchase goods or services
- publish a report

Activities, on the other hand represent the basic communication interaction unit, which in combination will accomplish a task. Activities are listed below.

- monitoring
- orchestrating
- gathering information and resources
- generating ideas
- evaluating ideas/information

- organizing ideas/information
- share/teach ideas/information
- discuss and come to agreement
- plan a project
- produce a project.

Operations are the basic interface manipulation unit to manipulate audio, video and data during the communication act.

Interactive Situations

Figure 2 illustrates our first attempt at modeling interactive situations. Each axis in the model initiates from the center (maximum entropy) and radiates toward the perimeter (minimum entropy). Additional research is needed to identify additional axes and to identify additional points along any one axis.

An interactive situation is defined as the relationship of participants to themselves, time and space. Participants can be in a large or a small group and they can be dependent on one another or not. The dependency of one participant on another refers to the granularity of the working interactions, that is, how much work can one participant do before having to interact with another. Thus, participants can be loosely coupled, where relatively few interactions are required to make relatively significant progress, or tightly coupled, where participants need to interact frequently relative to the amount of work that needs to get done. In space, participants can also be collocated or in different physical locations. Finally, participants can participant synchronously, that is, in real time with the potential for immediate or pre-immediate feedback (through non verbal signals, interruptions, etc.), or asynchronously, where individuals interact without the possibility of immediate or pre-immediate feedback. Participants also can initiate interactions spontaneously, i.e., in an ad-hoc fashion, or in a more pre-planned fashion, where group interactions scheduled in advance.

Figure 2 shows two lines. One is dotted and one dashed. These lines connect different points on each axis. The polygonal shape created by intersecting each axis represents one interactive situation. The set of "shapes" supported by a particular product then represents the set of interactive situations supported by the product.

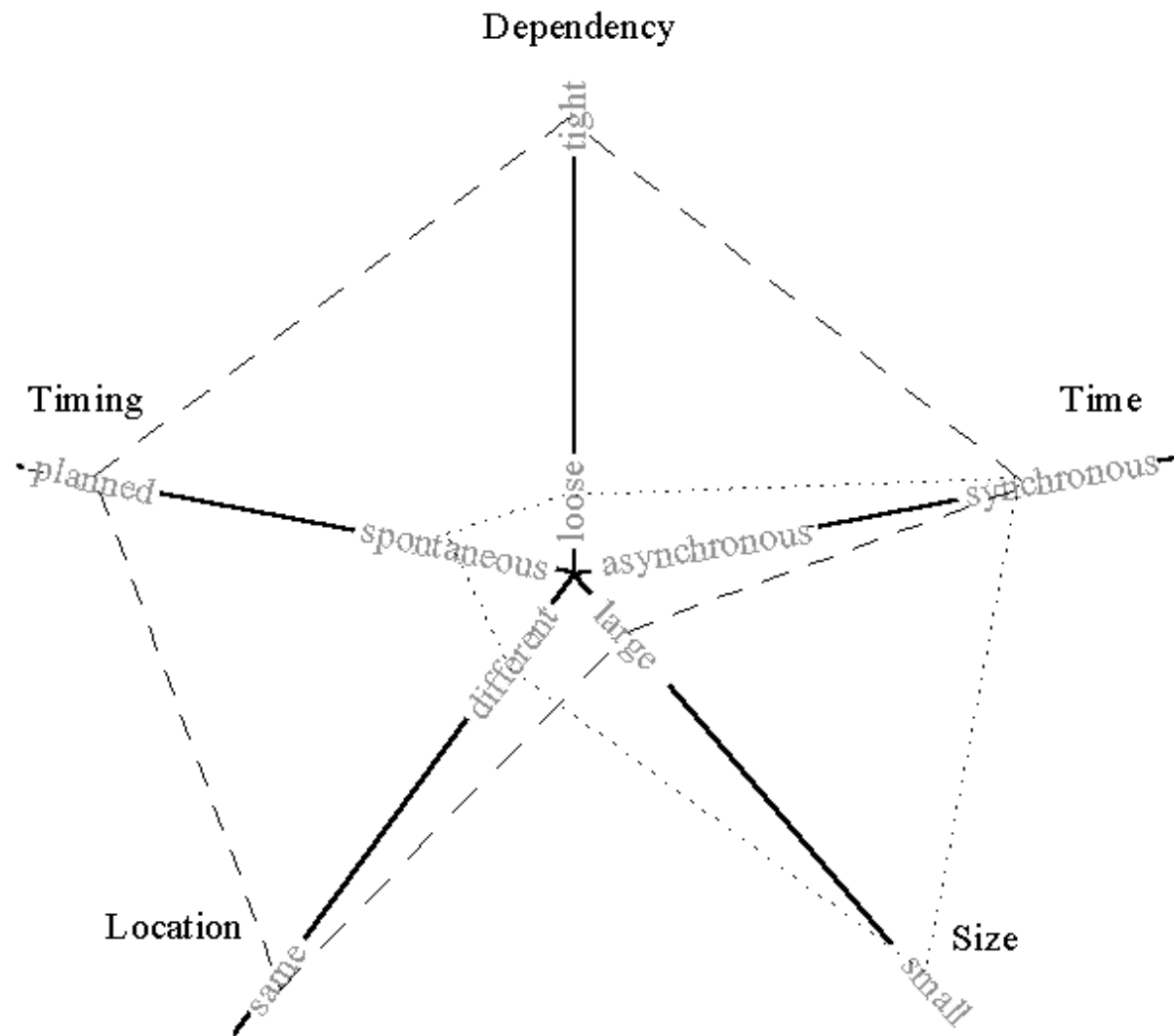


Figure 2. Interactive Situation Model

Interactive Social Protocols

The interactive social protocol model is analogous to the interactive situation category of the previous section. A social protocol refers to the allowable sequence of exchanges of signals and information that determine and identify resolutions and conflicts. Social protocol includes the concept of heterogeneity, which refers to the malleability of structure (organization of roles and resources) and the function (use of tools and objects) as well as to how to choose which protocol. Also included are explicit and/or implicit rules governing interaction repair, construction, change and modification, as well as how to recover from failures in social protocol.

Environmental factors particularly salient to social protocols include the following:

- (who) and (how) to repair,
- how to signal progress or any inflection point within a social protocol
- how to recover from failures in social protocol
- protocol language: how does technology, organization, culture and the task at hand define, set, constrain protocols

Figure 3 illustrates the social protocol model. The format of this exchange may be defined by the styles associated with the group size (small to large), the style of the meeting, the level of contention detection and resolution, the floor control style and the formality of address.

Meeting style refers to unidirectional, or mostly unidirectional exchanges, e.g., presentations with questions held until the end, and multidirectional exchanges where, on the average, each person participates an equal amount, e.g., a brainstorming session.

In any meeting, contention detection and resolution is valuable for maintaining forward progress. Sometimes, contention is important to discover and resolve immediately before further progress can be made, e.g., gathering information and making decisions. Thus, there would be a low threshold for detecting contention. However, at other times, it is valuable sometimes to ignore contention as it may be tangential to the point of the meeting, e.g., during brainstorming, which would translated to a relatively hi threshold for contention detection and resolution.

Floor control refers to the competition among participants for the ability to control the interaction. Maximum floor control may be indicated by something analogous to parliamentary procedures, whereas minimum floor control might be a free for all where nothing more tangible than social convention.

Formality of address refers to just that, how formally participants are to be addressed - either verbally, or in how they themselves are represented in the user interface on other participant's machines. For example, is it: Dr. Salk, Jonas or Joe.

As with interactive situations, the polygonal shape determined by intersecting the five axes represents the interactive protocol. The set of shapes supported by the application represents the full set of interactive protocols supported by the product.

Groupware applications suggest the development of new social protocols that can be built within the application layer of the ISO/OSI communication protocol architecture. The enumeration & relationship of new social protocols in an open research issue.

- who is work on artifact
- where in artifact are people working
- what artifacts are being worked on
- what type of protocols exist for working
- what is the current protocol
- what roles are there
- who is playing what role
- what's the emotional state of each individual
- what are related roles
- who's gesturing,
- who's voice is this
- who's video image is this
- who's cursor is this
- who's pointing
- who's signaling
- who's annotating

Evaluation of Groupware

Evaluating groupware also presents some unique challenges to the evaluator. Unlike single user software, the use of and output from groupware is dependent on multiple users, their interactions with the software and their interactions with each other. Furthermore, as a medium for communication, the groupware application itself may influence the group dynamics as well as the type and/or quality of the output. Thus, evaluating groupware may involve testing not only the a single user’s interaction with the user interface, but the effects of users interacting with each other using the software as a communications medium.

As a result of this expanded role, the logistics associated with groupware evaluations are far more complex than for laboratory usability tests of single user products. This increased complexity is evident not only in the laboratory but also in the field. Additionally, typical methods for software evaluation, e.g., paper prototypes, are considerably less useful and considerably more difficult to orchestrate as many features of groupware software are invisible until multiple users are present doing work.

The following section identifies the problems we identified, but only touches on methods and techniques to overcome the issues specifically related to groupware evaluation.

Evaluation Space

The full "space" of a groupware evaluation includes the following: the requirements of the groupware, the issues in the design space which can hinder or facilitate these requirements, the procedure that can be used to evaluate these issues and the useful evidence which should be collected. The table below summarizes this information. The succeeding paragraphs explain the table in more detail.

Requirement	Design Space	Procedure	Useful Evidence
1. Task focus	· anonymous or attributed contributions · locking and update · number and flexibility of representations, what you see is what I see · indexing - types of links between information	· video tape realistic exercises in lab · comparative mini experiment	· nature of discussions · problems encountered · percent of time off task

	· distribution of input devices (sharing/individual) · proximity of participants · forced agreement or individual		
2. Accurate predictions Trust usefulness of system	· explicit/implicit models · accurate feedback · access to features · number of features	· Field trials with and without · use by developers · mini comparative experiments	· do users use features · do users find features useful · questionnaires · ratings
3. Low cognitive demand Casual use	· white board and/or shared apps · linked scrolling · automatic setup for groups - turn taking, color coding, user names on objects, etc. · number of different features · models for features	· Single user walkthrough · lab experiment with real work/groups · lab experiments repeated over time	· errors in doing a task · features used · retention · predictions by users
4. Awareness	· settings · additional channels · bandwidth	· lab experiment	· number of questions that users ask · who did that · where are you · number of collisions
5. Acceptable	· task flow · support for group dynamics	· use by developers · beta use · field studies	· interaction log · questionnaires · interview · % of users adopting

-

Task Focus

This section corresponds to section 1 in Table 1. Groupware should be designed to facilitate the users' focus on the task rather than on the groupware, per se, making the task more explicit and pointing out steps along the way to achieving the task. This, in fact, is the opportunity for groupware to be better than current face-to-face methods. Examples of design issues that can affect task focus include

the following: whether the group must agree on something before moving on to the next step, whether all users are aware of the contributions of others, and whether all users can work independently or whether they have to take turns at all design decisions.

In order to evaluate these issues, real tasks accomplished in realistic settings is preferable. However, experiments could also be done in a laboratory setting. Data collected with either method would include video tapes of the dialogues, times, and problems encountered in doing tasks. In analyzing the data, one would measure the time and content of the time spent off task, e.g., on issues related to coordination associated with the software. Qualitative data about the types of problems users encountered when trying to achieve the tasks would also be useful. In addition, several groups performing several different types of tasks would allow us to compare tasks across groups.

Accurate predictions, trust, usefulness

This section corresponds to section 2 in Table 1. This category includes issues of functional utility as well as the ability of the groupware system to provide adequate communication mediation cues. The first issue derives from the fact that making implicit activities, e.g., communication cues that implicitly occur in face to face meetings, explicit may not be necessary or useful for engendering trust or quality communications in a groupware product. For example, can users find the features they need and do the features do what the users want them to do?

The second issue refers directly to the level of trust that any user might have in the groupware system regarding the representation of the current state of the participants in the meeting. For example, do the users trust the system? If the system indicates that three users are present, do I necessarily agree? If the system indicates that Joe made the current annotation, do all users agree with that? Are users able to accurately predict what will happen next and is that what they really want to happen?

Three means of evaluation were identified. First, developers can try out different models for features and see which users are able to understand and use. Use by developers is a reasonable way to begin in these cases as there will often be many different permutations that need trying out. Second, if the software is in a reasonably stable state, different versions (at least of the user interface) can be implemented and placed for use in the field; the resulting data can be compared to decide which of several models are more appropriate. Third, mini-experiments could also be done in the lab to see if users can predict what is going to happen.

Low cognitive demand

This section corresponds to section 3 in Table 1. Users should be able to use the software while maintaining their focus on the task, not the technology. It is assumed that for most group interactions, the focus is on the content of the interaction and not on the groupware, per se. Therefore, there should be a low cognitive demand on users related to the operation of the groupware.

This issue is probably more important for groupware than for single user software. First, groupware users must manipulate the groupware application while simultaneously interacting with the other participants. Second, in a typical single user application, one user controls switching from user interface manipulation to content manipulation; in groupware, if one user must interact with the groupware, there is a potential to derail the group interaction to do so.

Evaluating the level of cognitive overhead is a tough issue. It appears to be related to the amount of automated support for particular activities in the groupware application. For example, such items as automatic support for turn taking, automatically providing all the shared documents to a late comer, automatic support for meeting minutes, etc. are possible groupware capabilities considerations that require evaluation. Data from single users walking through the software can be used to predict areas of possible concern. The methods identified in the last paragraph of the previous section apply here.

Acceptability

This section corresponds to section 4 in Table 1. This section considers the perceived utility of the groupware application as related to current practices. Some specific issues are: comparing the performance and utility of the groupware application to the current practice, evaluating how well the users will be able to adopt the software, evaluating whether the groupware application diminishes the users' ability to do other jobs. Evaluating this issue requires field studies of real people doing real work with a real product. Further, if developers were to genuinely use the software as early as possible in the development, there would be some early indication surrounding these issues.

Awareness

This section corresponds to section 5 in Table 1. This issue concerns the feedback related to the users' awareness of others and other events in the meeting. For example, ensuring that users are aware of each other and current activities such as who is doing what, who did what, who is speaking, who is being seen, what is being shown, etc. Most of these issues can be solved in the lab with small, controlled experiments. Data collection should focus on users questions about who is where, and collisions with other users in time or space.

Issues in choosing procedures and evidence

The group also discussed some issues involving the choice of data collection. These issues include:

- privacy of data
- cost effectiveness
- richness of feedback
- group outcome
- validity of data.

Privacy of data: In field work there are two important privacy issues. First, many companies may not be willing to have their decision making meetings collected by an outside company, thereby ensuring difficulty in observing real work by real users. Second, from a data collection perspective, many companies or employees may not be willing to have particular meetings audio or video recorded, making observations more fleeting.

Cost effectiveness: Setting up lab studies with real groups doing real work takes a considerable effort both to organize and to evaluate. Both the cost to do this and the time to analyze the data must be factored in when deciding on a particular evaluation method.

Richness of feedback: The richness of the data must be balanced against the time to collect and evaluate and the particular data needed for any particular design decision. For example, field observations are much richer in data than questionnaires sent around to participants, yet one may be viable in one circumstance, and one viable in another.

Group outcome: The material generated by the group as well as the group's process can be evaluated and used as a basis for groupware evaluation. However, the normal working dynamics of the group must be factored into this data. For example, if the group typically disagrees, and disagrees after using the groupware, is this evidence that the groupware is lacking? Thus comparisons to some baseline condition may be warranted.

Generalizability of data: Groupware evaluations done with only one task and one type of group should in general, not be used to make general design decisions. It is often difficult to separate out evidence about mechanical use of features in a groupware session where group dynamics were a disaster. Were the group dynamics a result of the features of the application, of the group itself or of the task?

Future Work

We emphasize that this is a draft. We would like to have designers and evaluators of groupware review this information and give us feedback. In particular, we would like to have case studies about particular types of groupware and about the evaluations that were done. We would like readers to attempt to use this framework when describing their evaluations, pointing out additions, deletions, improvements to the framework we have set forth.

Workshop Format

This full two day workshop focused on identifying design and evaluation issues specifically related to groupware. There were 15 participants from 5 nations, representing university and industry.

Prior to the workshop, participants provided proposals indicating their top 5-10 design and evaluation issues associated with groupware, as opposed to single use, single user software. Based on these data, the organizers constructed two models capturing the basic qualities of the data: one for design issues and one for evaluations issues.

Day 1: After some short introductions, the group divided into sub-groups each assigned to a particular component of the design model. The mission was to more fully flesh out that section, or if desirable to reorganize that section of the model into something making more sense. In other words, the model was a prop to initiate the real work and was hardly sacrosanct. Several times joined together for reports and by the evening, we had the basic framework of the Denver Model in place. One of the organizers wrote a summary of the day's work that evening which was distributed for all participants the next day.

Day 2: One small group continued work on the Denver Model for design. The others considered evaluation issues. Again, we broke into subgroups and rejoined several times for coordination.

Workshop evaluations submitted at the end of Day 2 revealed one central theme: The extensive preparation allowed the group to accomplish a lot of work. However, given the enthusiastic and capable nature of the participants, we were all extremely tired by about 3pm rather than 5:30 or 6 whenever we were supposed to end. Nevertheless we pushed on until 4pm each day. One clever suggestion was to offer a three day, 5 hours per day, workshop rather than the 2 day, 8 hours per day.

Biographies

Jean Scholtz co-manages the Human Factors Engineering group at Intel Corporation's Business Communications Laboratory. She is currently responsible for work practices requirements, user interface design and usability testing for a range of multi-media communications products. She is active in the CHI community, both locally and internationally. She was the industry liaison for CHI'95, co-chair of the organizational overviews program, and a co-program chair for the industry track of INTERACT '95. She has participated in several workshops at CHI in the last several years, including the 94 workshop on taxonomies for usability evaluations.

Tony Salvador co-manages the Human Factors Engineering group at Intel Corporation's Personal Conferencing Division. Dr. Salvador is currently responsible for work practices requirements analysis, user interface design and usability testing for Intel's desk-top conferencing products. Prior to this, he worked at GTE Laboratories, Inc. where he performed requirements analysis and user interface design for GTE's centralized network management system.

Jim Larson is the manager of a team of collaborative software developers at Intel's Architecture Development Lab. He is the author of "Interactive Software: Tools for Building Interactive User Interfaces" (Prentice Hall, '92) and co-editor of "IFIP Transactions: Engineering for Human Computer Interactions" (North Holland, 1992). Larson is also an adjunct professor at Portland State University where he teaches courses in User Interfaces and CSCW software, and teaches professional development courses in databases for Learning Tree International. Dr. Larson is an active member of IFIP working Group 2.7 (Interactive User Interfaces) and conducted a workshop on CSCW software at their 1992 working conference in Finland.