CS 290: Collaboration Systems [1]
Section 3: Taxonomies of Collaborative Applications
Prasun Dewan

1. WHY TAXONOMIES

In the previous section, we saw a large variety of collaborative applications, which correspond to different points in the design space of collaborative applications. It is useful to develop taxonomies to classify these points, for several related reasons:

—*Economic Expression*: Typically, there are fewer nodes in taxonomies than there are potential collaborative applications. Thus, it is possible to develop an economic method for expressing the collaboration design space.

—*Compare/Contrast*: Taxonomies also provide a way to compare and contrast collaborative applications. Two applications that are in the same class have common features described by the class and two that are not differ in the properties described by their classes.

—*New Applications*: Taxonomies are often created by dividing the design space into orthogonal dimensions. By combining points along the various dimensions, it is possible to come up with new useful collaborative applications.

—*Design/Evaluation of Collaboration Systems*: It is possible to design collaboration systems to cover specific classes described by taxonomies. In particular, a taxonomy consisting of a hierarchy of application classes can be used to identify the object classes in an object-oriented collaboration system. Conversely, taxonomies can be used to evaluate/describe collaboration systems by enumerating the classes of applications they can support.

We have already seen one taxonomy - sess .... In this section, we will look at some of other taxonomies proposed in the literature. These taxonomies are not detailed.

2. THE TIME/SPACE TAXONOMY

Collaboration can be classified according to whether the participants are interacting at a) the same or different "times" and b) the same or different "place". [Ellis CACM] The time and space dimensions together define four classes of collaborations: same time-same place, same time-different place, different time-same place, and different time-different place.

What does "same place" and "same time" collaboration mean exactly? Defining "same place" collaboration is not difficult - it is collaboration among a set of users that is enabled only when they go to some physical room.

Defining "same time" collaboration is more difficult. Here are three possible definitions:

Same time collaboration is WYSIWIS collaboration, that is, all users see an identical image and updates to it are sent immediately to all users. This is the strictest definition but disallows near-WYSIWIS collaborations such as those supported by

Grove (it allows users to scroll independently), which one might consider as same time collaborations.

We do not require users to see the same image but assume only that they share some underlying state with possibly different visual manifestations. We do require that changes made by a user to this state are sent instantly to other users, that is, the application does not buffer changes before sending them to others. This is less strict than the previous one but maybe still too strict since one might want to allow some buffering to support "think time". Collaborations supported by Ensemble (which buffers changes until the item is unlocked) would not be considered as same time collaborations by this definition, yet Ensemble is used for "meetings".

Another definition may be that in same time collaboration, information is sent implicitly as a side effect of some other action, whereas in different time collaboration, explicit send and receive commands have to be executed. By this definition, an editor that implicitly sends hourly updates supports same time collaboration, which we may not agree with.

Same time collaboration occurs between users if each of them has explicitly executed a `join` command to join the conference. This definition excludes Ensemble, which allows users to implicitly join a conference by concurrently editing the same file.

We can say that two users are interacting at the same time if a user's communication to the other is immediately processed by the other as soon as it is received by the latter. In other words, the two users are in a "meeting" where each user is paying immediate attention to the other's communication. (A user is free to take time in processing a message and do other tasks while the other user is thinking or messages are in transit - an in real meetings.) In technical terms, same time collaboration implies that users receive responses to their messages within some threshold "attention-keeping" time, which may depend on how patient the users are. To make our discussion simple, let us assume that there is a some universal number that applies to all groups.

Assuming we choose one of these definitions of same time collaboration, we can classify collaborative applications according to the class of collaboration they can support. A collaborative application may support more than one of these classes. In particular, an application used for different place collaboration can also be used for same place collaboration, though it may have redundant features such as displays of participants. Moreover, an application that supports same time collaboration also supports different time collaboration if we use the last definition of same time,

more than the "attention keeping" time before they respond to each other. Furthermore, an application that supports same place collaboration may also support different place collaboration if we assume they can talk to each other over the telephone.

To make our discussion simple, we will say an application supports same place/different time collaboration if it can only support same place/different time collaboration. Moreover, we will say that an application supports different time collaboration if it always imposes a delay of more than attention keeping time in delivering messages. Furthermore, we will assume that users cannot use any communication media except those supported by face to face interaction or explicitly by the application. In particular, they cannot use an audio or video telephone. This is consistent with what others assume when they use this taxonomy. Finally, we will say an application supports different place collaboration if it has some feature such as discussion windows and displays of lists of participants that would be "redundant" in a same-place collaboration.

So let us classify our tools according to this taxonomy.

Same time-same place: rIBIS, Cognoter, Tivoli.

Same time-different place: RTCAL, CES, GROVE, Groupsketch, Groupdraw, Ensemble (the prev three are in this class because of gesturing support), GroupDesign, Mercury, MUD, Video Walls, Shared Awareness Space (is it fair to call it same time?), and 3-D Aura.

Different time-same place: Physical bulletin board.

Different time-different place: Information Lens, Coordinator, ActionWorkflow, Mercury, Chronicle, and MUDs with mail.

3. COMMON TASK/SHARED ENVIRONMENT DIMENSIONS

Ellis et al [Ellis CACM] also define the two dimensions of "common task" and "shared environment". Increasing points along the "common task" dimensions represent increasing commonality in the tasks carried out by the collaborators and increasing points along the shared environment dimension represent increasing amounts of shared information. Ellis et al put time sharing systems low in the common task dimension, a software review system high in the common task dimension, electronic mail system low in the shared environment dimension, and finally, an electronic classroom high in the shared environment dimension.

It is difficult to precisely define these dimensions and the points along it, and so it is difficult to use them to classify applications. It may be possible to order applications along the shared environment spectrum – an application that allows sharing of more kinds of objects is higher on the spectrum. The kinds of objects could include shared artifacts such as group drawings and documents, annotations and comments about them of the kind provided by Quilt and PREP, static images of the participants, dynamic videos of the participants, and notification mail messages of the kind provided by Mercury, Quilt, Coordinator, and Information Lens. However, it is difficult to precisely define the common task spectrum. One definition, which is consistent with Ellis et al's usage, can be based on synchrony – an application that allows users to view and respond quicker to other user's input would be higher on the common task spectrum.

So one ordering of our example applications in the Shared Environment Spectrum may be:

Coordinator ¡ Quilt

since Coordinator supports only notification messages while Quilt also supports shared documents and annotations. Similarly, one ordering of our example application in the Common Task/Synchrony dimension may be:

InformationLens ¡ Ensemble ¡ Grove

Ensemble sends changes after they are committed, and Grove sends changes incrementally. since InformationLens requires explicit sends and receives, A definition of common task based on synchrony is not entirely satisfying since it is possible for users of synchronous applications to not have much commonality in tasks - consider a user (e.g. manager) simply monitoring the activities of people in a shared awareness space or users of MUD. Perhaps another definition is that an application that provides more facilities to support conference discovery is low in the common task dimension. Yet another definition is to use the interdependencies among the activities of of different users as a measure of common task. We can use Malone's work described later to categorize interdependencies. The challenge is to identify tool features for supporting them, so that we can use the common task dimension to categorize tools.

4. FUNCTION CLASSES

Ellis et al [Ellis CACM] also describe a taxonomy based on application-functionality consisting of several (non-orthogonal) function classes:
Message Systems – allow users to exchange textual messages among themselves. Examples: Information Lens, Coordinator, PREP, and Quilt.
Multiuser Editors – allow users to edit a shared textual document. Examples: CES, Quilt, PREP, and Grove.
Group Decision Support Systems and Electronic Meeting Rooms – allow users to explore "unstructured" problems involving decision making. Examples: RTCAL, IBIS, PREP.
Real-Time Computer Conferencing Applications – support same time interaction among the collaborators. Examples (depend on definition of same time): GROVE, Groupsketch.
Computer Teleconferencing Applications – support conference calls and video conferencing. Examples: Video walls.
Desktop Conferencing Applications – support real-time conferencing and teleconferencing. Examples: Video walls + Grove.
Intelligent Agents – include non-human participants. Example: The LIZA toolkit [Gibbs LIZA CHI] allows a special application called Liza to join a conference and displays the image of an android. When Liza participates, a set of rules become activated, which allow it to respond to other user's input. Another example is computer programs playing with humans in MUD or even the Elisa AI program.
Coordination systems–allow users to "coordinate" their activities. These are further classified into form-oriented workflow systems, which route forms among users, (e.g. ActionWorkflow), procedure-oriented systems, which impose fixed procedures among the collaborators (e.g. Cognoter and process programming systems in software engineering), conversation-oriented systems, which are based on social theories of conversation (e.g. Coordinator, ActionWorkflow), and communication structure-oriented systems, which are based on role relationships (Example: ITT, [Holt Coordination] which organizes a person's work environment into several centers representing different functions performed by the person. Centers are associated with roles and connected to other centers by a Petri Net).

5. BAECKER'S TAXONOMY

Baecker [Baecker Groupware] has edited a book of collected papers, and used a taxonomy to classify application papers into various categories:
Electronic Mail and Computer Conferencing– tools such as mail and bulletin boards

that support point to point delivery or broadcast of messages.
Structured Mail. Agents, and Workflows– applications such as Information Lens and Coordinator that add intelligence to mail systems by adding structure to it, providing filters, and automatically routing messages.
Cooperative hypertext and organizational memory– applications such as PREP and IBIS that link group information.
Desktop Conferencing– same time applications such as Cognoter, Groupsketch, and Groupdraw.
Electronic meeting and decision rooms– same place applications such as Liveboard applications often presupposing a particular room layout.
Media space– applications such as video walls providing audio and video connections to support different place collaboration.

6. GDSS TAXONOMY

Kraemer and King [Kraemer King] have divide GDSS systems into several classes:

Electronic Boardrooms– computers used in these rooms for generating and/or controlling audiovisuals. By our definition of a collaborative application, the software here is not by itself collaborative,

The Teleconferencing Facility– allows users in two or more locations to communicate with each other. The software here simply provides digital transmission of voice, data, and pictures. Examples: Video Walls.

The Information Center– allows collaborators to generate and modify reports, manipulate and analyse data, and make spontaneous inquiries. Example: regular databases.

The Decision Conference– provides software that influences the decision making process by supporting some formal, quantitative decision analytic technique. Example: University of Arizona and Minnesota tools.

Collaboration Laboratory– provides software that influences the decision making process by supporting informal, qualitative, brainstorming techniques such as writing and argumentation. Example: Colab.

The Group Network– provides support for same time use of scheduling and other applications (e.g word processors, spreadsheets) by small groups. A meeting chair is associated with the conference. Example: RTCAL with a shared window system.

This framework is useful more for classifying groups of hardware and software tools together rather than individual collaborative applications.

7. MALONE AND CROWSTON'S TAXONOMY

Malone and Crowston [Malone Crowston] define a taxonomy based on the following theory or model of coordination/collaboration. According this model collaboration/coordination occurs among actors, having interdependencies among them, doing activities to meet goals. This defines four components of collaboration/coordination:

identifying goals

mapping goals to activities

selecting actors to perform activities, and

selecting actors to perform activities, and

"managing" interdependencies among activities/actors, which can be further broken into managing resource allocation, sequencing and synchronizing, managing physical interdependencies between objects, avoiding duplicates, and managing other interdependencies.

These phases of collaboration are carried out by processes arranged in a hierarchy, where a higher-level process depends on a lower-layer process to do its task. Four levels of processes are defined:

coordination/collaboration– goal decomposition, task assignment, resource allocation, synchronizing, sequencing;

group decision making– proposing and evaluating alternatives;

communication among the collaborators– transporting and routing messages;

perception of common objects– seeing and manipulating shared information.

We can now classify the applications by which components of this model they support.

Level 1, Coordination:

Goal decomposition: An example is a system called Polymer, [Croft Lefkowitz] which not only helps in decomposition of goals into tasks, but also sequencing of these tasks.

Task assignment (agreeing on commitments): Coordinator.

Resource allocation (task prioritization): Information Lens (Filters).

Synchronization: meeting scheduling tools such as RTCAL.

Sequencing: Polymer.

Level 2, Group Decision Making:

Proposing Alternatives: gIBIS, RTCAL, Cognoter.

Evaluating Alternatives: gIBIS.

Level 3, Communication:

Transporting messages: electronic mail.

Routing: ActionWorkflow.

Level 4, Perception of common objects: Quilt, PREP, Grove, and the rest of the tools not covered above.

8. SMITH'S TAXONOMY

Like Malone, Smith [Smith Collective] proposes a model of collaboration with several dimensions/parameters:

Information type built by the group: intangible private, ephemeral, tangible instrumental, tangible target.

Time over which the collaboration occurs: few seconds, several minutes, 15 minutes, several hours, several days, several weeks, several months, several years, life's work.

Space over which collaboration occurs: office, cluster, floor, building, site, 1-hour travel for access, 1-day travel for access.

Agents collaborating: people, informal coalitions, teams, collection of teams, group as a whole.

Processes involved: cognitive, metacognitive, social interaction, mediated, organizational, collective/distributed, algorithmic.

Collaborative tools can be classified according to which points alongs these dimensions they support. For instance, CES, which does checkpointing, can support collaborations spanning all durations whereas a tool such as RTCAL supports collaborations of less duration than the time between computer failures. (We need

John's help in defining the points along these dimensions and using them to classify our tools.)