H-0220 (H0403-003) March 27, 2004
Computer Science

IBM Research Report

Towards a Comprehensive Taxonomy for Computer-Assisted Collaboration

Uri Shani
IBM Research Division
Haifa Research Laboratory
Haifa 31905
Israel

TOWARDS A COMPREHENSIVE TAXONOMY FOR COMPUTER-ASSISTED COLLABORATION

Uri Shani
IBM, Haifa Research Lab, Haifa, Israel, ***shani@il.ibm.com***

Abstract: Computer supported cooperative work covers a very large application domain ranging from mail systems, to chat programs, and on to desktop sharing, multimedia conferencing, and virtual communities. we distinguish between the applications, which implement all these user-accessible services, and the underlying technology that supports and enables them. In all cases, these are applications, which provide collaboration among people who use end-user clients. Nevertheless, collaboration may also involve computer programs, or "agents", which participate just like people. We are not interested in collaboration among computer programs, which is geared towards solving computational or business problems in a distributive manner. Our goal is to seek a comprehensive set of terms that can serve as an acceptable taxonomy for this application area. As common in advanced software design methodologies such as UML, this taxonomy can serve as a basic foundation analysis of the problem domain. The architecture that can be thus defined will support a family of applications to span the corporate boundaries and on to intra-organizational collaboration.

Key words: CSCW, computer assisted collaboration, taxonomy, intra-organizational collaboration.

1. INTRODUCTIONS AND STATEMENT OF PURPOSE

This work collects and attempts to properly organize common terms in the area of computer-assisted collaboration. Each section briefly discusses a collection of related terms. The work is not complete in the sense that besides the informal semantics of each group of terms, there is a mere narrative description of how groups are related to each other. In the well accepted software design methodology UML (Unified Modeling Language) [11], this phase represents an essential part of the problem domain analysis. A proper completion of this analysis will lead to a well defined model into which various solutions and technologies can be applied and integrated.

1.1 The domain

Computer-supported cooperative work covers a very large application domain, including mail systems and chat programs, desktop sharing, multimedia conferencing, and virtual communities. In this paper, we distinguish between the applications, which implement all these user-accessible services, and the underlying technology that supports and enables them. In all cases, these are applications, which provide collaboration among people who use end-user clients. Nevertheless, collaboration may also involve computer programs, or "agents", which participate just like people. Collaboration among computer programs, which is geared towards solving computational or business problems in a distributive way, is beyond the scope of this paper.

1.2 Domain Definitions

Baecker ([1], p. 141) defines Computer-supported cooperative work, or CSCW, as "... a computer-assisted coordinated activity carried out by groups of collaborating individuals." Obviously, this is a heterogeneous system combining software and people.

The supporting software is further distinguished as "groupware" by Lynch ([4], p. 160) and Baecker ([1], p.141):

" Groupware is distinguished from normal software by the basic assumption it makes: Groupware makes the user aware that he/she is part of a group, while most other software seeks to hide and protect users from each other ... Groupware ... is software that accentuates the multiple user environment, coordinating and orchestrating things so that users can "see" each other, yet do not conflict with each other."

1.3 Purpose

Our purpose is to define a set of abstractions for the fundamental building blocks, components and elements used by such applications. A good understanding of these terms and building blocks will lead to an accepted model of the domain, to an acceptable architecture, and thus will extend the boundaries of collaboration.

Different collaboration applications often portray different collaboration philosophies – different user interfaces; varying levels of sophistication of built-in intelligence and of GUI and UI paradigms; and many different levels of integration of all of the above. As a result of these abstractions, we will define terminologies and semantics that can be applied to each and every application in this field.

1.4 The Structure of Collaboration Systems

Collaboration applications are also known as groupware (Johansen, 1988 [3]). Most systems have a monolithic design that provides a single total solution, including all components, interaction protocols, data-storage, UI, etc. We view a groupware application architecture as being composed of several tiers.

The first (topmost) tier represents the user interaction programs, implementing some GUI or other type of UI, designed along some human-computer interaction (HCI) guidelines (Dix'98 [2]). These programs use various communication protocols to communicate with one or more servers. These second-tier protocols use a middleware, which supports the access of end-user applications to the third tier – the servers. The servers may also interact among themselves using additional communication protocols, some of which are standards, and some of which are proprietary. The network on which these servers and clients communicate may be the Internet, employing TCP/IP or UDP/IP, or other digital or analog networks, including POTS (Plain Old Telephony System). The data exchanged within these protocols may be analog and/or digital, media poor, or multimedia rich data. Figure 1 displays a schematic representation of this structure.

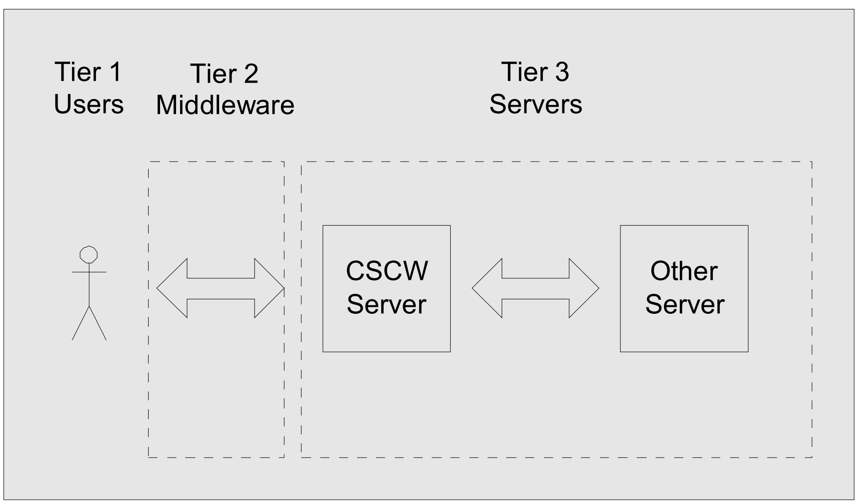

Figure 1: *General schematic of a 3 tier CSCW service implementation.*

A particular collaboration system is usually built as an integrated system, tightly coupling the clients with the servers. As servers are developed with well-published interaction protocols and services, additional clients can work with them and implement new user interactions, collaboration philosophies, and the like. As more and more servers are bound to published protocols, clients can provide an integration platform for different services, thus enabling a rich set of capabilities to support complex collaboration philosophies.

1.5 Goals

Our far-reaching goal is to define a general architecture for computer-assisted collaborative systems, implemented as distributed applications over the network. A target domain – for instance – is the computational GRID [12] on which collaboration services must be offered using a well-defined standard, which clients can employ. A proper architecture can provide standards by which services will integrate and contribute their added value to the larger grid community of services and clients. CSCW applications will be able to implement various collaboration philosophies through new service-providing servers, utilizing existing servers, and building new clients. Part of this architecture will be a set of fundamental collaboration entities, and a standard for communication (protocol) between service providers and clients, as well as among servers. The architecture will not inhibit the existence of additional protocols – standard or proprietary – as deemed essential by implementers of clients and servers. In fact, the level of "standardization" of this architecture depends on the level of adoption by the development community of CSCW tools.

A variety of CSCW tools exist. For instance, the UNITE [7] (5th EU Framework) project employs a middleware layer called JCCUM [8], through which a rich set of services can be accessed. The JCCUM middleware has been implemented as an open and extensible framework via a plug-in mechanism. Eventually, we would like to capitalize on this experience and take it further. Another example is ReachOut [9], which implements a new collaboration philosophy extending the Lotus® SameTime® chat environment, while using the same server via its proprietary API and protocols. Microsoft® MSN-Messenger®, Yahoo® Messenger®, and other chat applications are also relevant examples.

1.6 Document Plan

The rest of this document defines a taxonomy that can serve our goals. We start with the users of CSCW or groupware applications, through entities that are managed, means by which these entities are managed, axes along which they are measured, limitations and so forth. In future work, we plan to map existing technologies and products to the terms we define. Finally, we hope to define the domains where our target standards can be developed and make an initial proposition.

2. TAXONOMIES FOR COLLABORATION

2.1 Users of a Collaboration

Our first questions are “Who are the users of collaboration applications?” and “What distinguishes a computerized collaboration activity?”

There are many distributed applications in which programs interact for specific purpose, to serve requests and answer problems posted by users of the system. As stated in the introduction, when the activity goal is to enable individuals to act and work as a group, the interaction goes beyond the realm of a mere distributed application.

This answers the second question. The first question seeks to define the boundaries of the collaboration system in the sense that is defined in the “use case” terminology of UML (Unified Modeling Language) [11]. While collaboration in our context is computer-mediated interaction among people, we recognize the fact that programs can also act on behalf of real people. Such programs are usually termed “agents”. The collaborative system boundary is plotted between what we will collectively term “peers” for the entire group of users involved in a collaboration activity, and the supportive system, which consists only of computerized machinery. The role of a peer may be of a “participant” with several flavors, or of an authority that carries additional or other functions.

The participants’ role may be permanent or transitional. For instance, there may be one participant at a time that makes a presentation, otherwise known as the “speaker”. This role rotates among the participants. The role of “chairperson” is a permanent role of a participant assigned throughout a collaboration activity. However, there are additional roles of peers woh are not merely participants and carry some level of authority. For instance, a “controller” is a peer whose role is to control participation in an activity. The controller may allow participation or limit it, may assign participation roles to participants, and may synchronize the communication channels used in the collaboration activity (i.e., whiteboard or desktop sharing, multimedia channels, etc.). Another authority role is that of the “monitor” that can create a log, excerpt a version of the activity, provide feedback for the controller, and so forth.

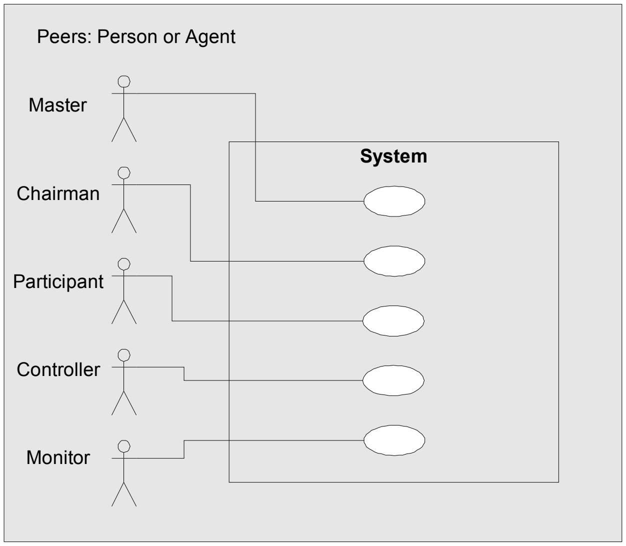

Figure 2: *Collaboration typical use cases.*

A person peer, or a program agent may take on one or more of the roles we described. A peer (person or agent) may also perform more than one role at a time.

Terminology for users of a collaboration activity:

1.1. Peer – user of a collaboration activity.
1.2. Person – human peer.
1.3. Peer ID – identification of the peer within a certain name space. A name space may be as large as the entire set of potential peers for session creation by a certain groupware. For instance, that of all employees of certain organization, within an intranet network, or as large as the entire WWW.
1.4. Peer address – identification of a location where the peer can be reached when interacting within a collaboration activity.
1.5. Agent – program acting as a peer.
1.6. Participant – peer participating in a collaboration activity.

1.7. Chairperson or Chairman – peer participating in an activity and recognized as the initiator and leader of the activity. May also be combined with the *controller* role.
1.8. Controller – peer that has a certain level of authority to perform various control and synchronization of the activity.
1.9. Monitor – peer that has the authority to monitor the activity and create various data based on it, such as logs, excerpts, etc.
1.10. Role – set of privileges and responsibilities attributed to a peer (cf. Ellis'91, p.46). In UML, roles are assigned to associate classes in a diagram, such as the users of the system. In both cases, the above roles are an incomplete list of the roles users of CSCW may have.

2.2 Collaboration Activity in General

The term CSCW (Computer-Supported Cooperative Work) was coined in 1984 in a workshop organized by Irene Greif of MIT, and Paul Cashman of DEC Inc. Computer-assisted collaborative work in its many manifestations are widely accepted as instances of CSCW. Another widely used term is GDSS (Group Decision Support Systems). This term comes from the IS and business communities, and mostly relates to group activities that take place in the same location and time (see below on place/time categorization). The name for computer support software for CSCW work is groupware.

Terminology for general collaboration activity

2.1. CSCW – computer supported cooperative work.
2.2. GDSS – group decision support system.
2.3. Groupware – computer support technology for CSCW.

2.3 Collaboration Activity: The Time/Place Taxonomy

Collaboration activities have traditional time/space categorization of the basic four categories of collaboration (Dix'98 [2]) in Figure 3, the four categories of CSCW activities (Johansen'88 [3], Baecker'95 [1]) in Figure 4, or nine (Grudin'94 [5]) classes in Figure 5:

	Time		
Place		Same time	Different time
	Same place	Face-to-face conversation	Post-it note
	Different places	Telephone	Letter

Figure 3*: Taxonomy of the time/place for collaboration: A - basic 4 space.*

	Time			
Place		Same	Different but predictable	Different and unpredictable
	Same	Meeting facilitation	Work shifts	Team rooms
	Different but predictable	Tele/video desktop conferencing	Electronic mail	Collaborative writing
	Different and unpredictable	Interactive multicast seminars	Computer bulletin boards	Workflow

Figure 4*: Taxonomy of the time/place for collaboration: B – CSCW 9 space.*

The time/place categorization refines terms like synchronization, real-time, off-line, and asynchronized communication. In all cases, we are interested in the CSCW case, using groupware. Therefore, the same-place category is not of interest, and certainly the same-place/same-time category, which means that all participants are in the same physical room, socializing and working together is irrelevant. The subject of our work is same virtual place, the equivalent of same place via groupware, which bridges the geographical distances using computerized means.

<table>
<tr><td></td><td colspan="3">Time</td></tr>
<tr><td rowspan="3">Place</td><td></td><td>Synchronous communication (same time)</td><td>Asynchronous communication (different time)</td></tr>
<tr><td>One meeting site (same places)</td><td>Face-to-face interactions
o Public computer displays
o Electronic meeting rooms
o Group decision support systems</td><td>Ongoing tasks
o Team rooms
o Group displays
o Shift work groupware
o Project management</td></tr>
<tr><td>Multiple meeting sites (different places)</td><td>Remote interactions
o Shared view desktop conferencing systems
o Desktop conferencing with collaborative editors
o Video conferencing
o Media spaces</td><td>Communication and Coordination
o Vanilla email
o Asynchronous conferencing bulletin boards
o Structured messaging systems
o Workflow management
o Version control
o Meeting schedulers
o Cooperative hypertext & organizational memory</td></tr>
</table>

Figure 5: *Taxonomy of the time/place for collaboration: C – CSCW four space.*

Types of Synchronization

In a CSCW groupware, synchronization means that each party or component of the activity waits to receive responses to requests or notifications it sent out, and does not continue without some response. This is not to be confused with social and psychological consideration of a collaborative activity of the involved peers. When shifted to the social and psychological domain of CSCW, synchronization is mixed and confused with the terms “real-time” and “not real-time,” or “off-line.”

Likewise, asynchronous activity of a groupware system means that components are not awaiting response to their outgoing events. When receiving events, components produce outgoing events of their own. In an asynchronous activity, the responses will not necessarily be sent to the source of the incoming triggering event. The participating components, expect a proper handling of the events, whether it is an asynchronous activity or not, or whether there is a centralized controlling facility, or a distributed implementation of its equivalent. The result should be a consistent overall activity in an integrated system.

In (Ellis'91 [6]), real-time conversations are considered synchronous (such as telephone calls), whereas the alternative are interactions that happen over an extended period of time are considered not real-time, and not synchronous. We will denote non real-time interaction as "off-line," such as is typical of mailing systems.

Yet, synchronization in CSCW activities deserves a deeper discussion offered later in this paper, to distinguish several levels of synchronization; the level discussed here is the lowest of its rank.

Terminology for time/place in collaboration activity

3.1. Synchronous – relates to the method by which collaborating components interact with close cooperation and exchange of messages.
3.2. Asynchronous – collaborating components respond to incoming messages and events in their own time, not tightly coupled with other messages or events in the system.
3.3. Real-time – relates to the peers, interacting in the collaboration, who respond and exchange events in a "synchronous" way fashioned after the synchronous interaction of collaborating components. Responses to incoming events or messages are done within the "same time," or the same "thinking time."
3.4. Off-line – another term for interactions among peers which is not real-time, and therefore involves asynchronous interaction among the involved components of the supporting groupware.
3.5. WISIWIS – What You See Is What I See (or What You Sense Is What I Sense) to characterize a same-time collaboration, typical to desktop and application sharing, phone conversations, or multimedia conferences.

2.4 Collaboration Activity: Unit of Activity

A collaboration activity unit is a "session." It may be a single episode lasting a limited time, or a chain of episodes that share some common context. An off-line collaboration session such as a news group may be considered a very long session on a common subject area. There may be many internal "threads" of subtopics, concerning more focused issues within the larger subject domain of the specific news group. Such threads can be formed from a posted message and the chain of responses and counter-responses to it. Such a thread or episode may spawn its own new news group for a new subject. In general, we can say that collaboration sessions may have embedded and nested sub-sessions, as well as related sessions that are not nested within them. Sessions, sub-sessions and related sessions may be of different time/place types. For instance, a real-time synchronous phone conversation may be spawned as a result of a customer discussing a specific issue (or thread) in a news group for a particular product.

A session may start or come to exist for various reasons, and in various ways. It can be planned, scheduled, or be spontaneous. Two hand-held devices, equipped with "Bluetooth" devices, may "discover" each other and start a collaboration that was not planned or scheduled ahead of time. This could occur in an entertainment, social situation, a cellular advertisement, or a professional meeting conference.

A related group-collaboration problem is "workflow," where many activities need to be synchronized among themselves and project an integrated activity towards a common goal. Workflows may be thought of as a special kind of collaborative session, which employ inter-session synchronization (unlike synchronized sessions). This kind of synchronization should be addressed in more depth, more laboriously, perhaps in a follow up work. A workflow session will always be composed of internal subsessions, which will be related to each other as related sessions.

Terminology for units of collaboration activities

4.1. Session – unit of collaboration which spans some length of time.

4.2. Session ID – unique identification of a session within the name space for a certain groupware.

4.3. Spontaneous session – session that has not been planned or scheduled ahead of time.

4.4. Planned session – session that has been planned and scheduled ahead of time, with target time, alternatives, list of participants and designated roles.

4.5. Sub-session – session started within another session. Sub-sessions may be spontaneous or planned, and in general do not need to be of the same type as the parent session in which they began.

4.6. Related session – session that started in relation to another session. We'd say that subsessions are related by sharing the same higher-level subject of the parent session. Another reason for being related is being part of a more complex session such as a workflow session.
4.7. Thread session – a certain type of a sub-session representing a thread of ideas or a subtopic to the parent session, thread, or topic.
4.8. Episode session – another name for a subsession.
4.9. Workflow session – a relatively complex session that includes many subsessions. The workflow session synchronizes between the subsessions and thus is a kind of a synchronized session, but not necessarily a real-time session. Synchronizing among sessions is a macro-level synchronization, unlike synchronization within a session as defined above, relative to the activities of components in the groupware system.

2.5 Collaboration Messages, Events and Responses

Collaboration sessions are carried our through the exchange of messages. Each message is a notification of some sort, or a response to a notification, and may carry lots of information. One can distinguish different classes of messages, and different delivery methods for those messages. A message which signals a state change of a session is usually termed an "event." A message that comes as a response to another message may be termed "response." However, in a lengthy exchange of messages back and forth, how should one classify them? Or is that important at all? We can simply treat all messages as events notifying of some change of status, and that may also cause yet another change of status. Events that come as a result of other events may be categorized as "responses" by associating them with the causal event that triggered them.

Events are sent as messages via asynchronous messaging systems, or via synchronous mechanisms such as remote method invocations (RMI), remote procedure calls (RPC), inter process communication (IPC) and so forth. The delivery method depends heavily on the implementation and type of collaboration session. For instance, a voice conversation will use the proper standard protocols for audio streaming, which dictate the format of messages being exchanged, as well as the entire end-to-end synchronization protocols, policies and means. Other types of events are used to synchronize among several parallel and related sessions, which make up an integrated parent session. These events may have to abide to other kinds of rules and protocols that are dictated by the underlying servers which support the session and its sub-sessions. For instance, mail sessions will use the standard SMTP (Simple Mail Transfer Protocol – SMTP [10]), and send/receive events accordingly.

Terminology for events in collaboration sessions

5.1. Event – collaboration unit of communication that is exchanged among components of a groupware, representing an abstraction of all types of information exchanges. The end-to-end events exchange abides by rules of one of more protocols from standard or proprietary sources.
5.2. Message – event which carries information as part of a lengthy "stream" of messages.
5.3. Synchronization event – event which coordinates among sub-sessions of a parent session.
5.4. Response – event which can be distinguished as being triggered in response to another event.
5.5. Request – event which can be distinguished as the initiator of some activity within a certain session, or which may start a session or a subsession.
5.6. Low-level event – abstraction of all proprietary (or standard) low-level events as defined in standard or proprietary protocols such as for multimedia streaming, etc.
5.7. Control event – event which serves to control a session.
5.8. Monitor event – event which serves to monitor a session.

2.6 Collaboration Tasks and Artifacts

Collaborations are intended to execute some piece of work, or a task, which is the subject of the collaboration. The task contents may be completely embedded within the exchange of messages and events, such as in the e-mail messages. The subject may also be the authoring of some artifact, which is a common shared data. The artifacts may be in a database, of a document, such as the news group archive of messages. There are many types of artifacts. For instance, news group archives are built by new pieces of data that are appended or inserted. Complicated collaboration may involve the collaborative authoring of a document, or the alteration of database records. Therefore, we can distinguish the access types to the shared data repository of a collaboration session: *read-only*, *append-only*, *update* and so on. The task may also be distinguished according to the level of collaboration among its peers for each of the access types identified above. Ellis [6] distinguishes sessions along two separate dimensions: the "common task dimension," which represents how tight the cooperation is in executing the common task; and the "shared environment dimension," which refers to the artifacts which the collaboration generates. Another dimension indicates the media content level of a session task. These dimensions are all illustrated in the following figure (based on Ellis'91, p. 41 [6]) in Figure 6.

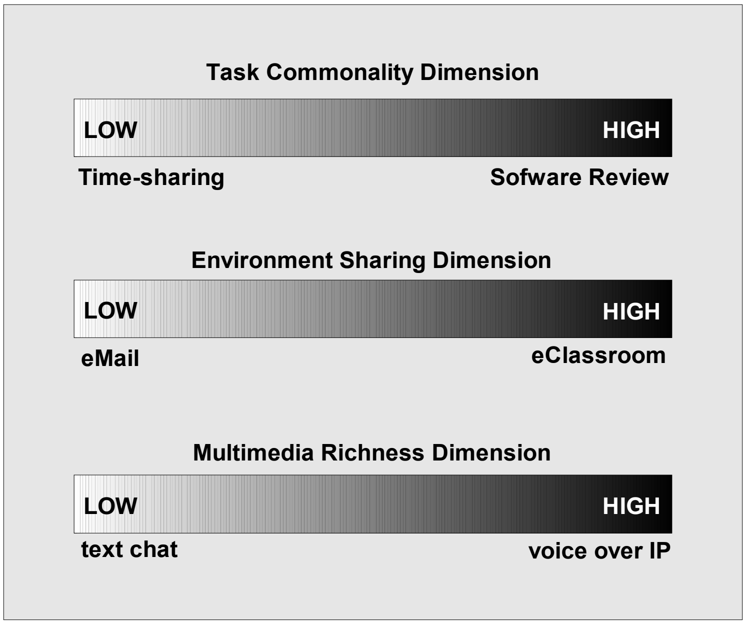

Figure 6: *Collaboration dimensions*

Each participant of the session has some visual representation of the session, or even something more basic than that depending on the interaction device used (see "Terminals" below). These views make up the environment in which the session operates. There are many issues concerning how these views are used to render sharing of the collaboration environment.

Terminology for collaboration tasks

6.1. Session task – identified piece of work that is associated with a specific session, and which dictates some derived properties for the session.
6.2. Task commonality level – indicates how tightly session peers are related to each other, based on how much the task is common to them as a group.

6.3. Task sharing level – indicates how much the session task is shared among its peers. A highly shared task dictates a high level of synchronization among the peers and their access to the task common data repository.
6.4. Task multimedia level – indicates how rich in multimedia the session is, ranging from simple text, to real-time voice, sound and video.
6.5. Data repository – storage where the shared data of a session is stored and managed.
6.6. Session view – a representation of the session as sensed by a certain peer. This can be a visual representation on a screen, or a control panel in a simpler device.
6.7. Session task state – state of task of the session, which may be described as a combination of the states of the data shared in the session, and the session progression itself.
6.8. Session data state – part of the representation of the state of the session task. This may represent the status of the common data repository,
6.9. Session Progression State – part of the representation of the state of the task of the session, pertaining to the status of the collection of peers participating in the session, and their view of the session.

2.7 Session Life Span

Session life span may be limited, or unlimited. The limitation on the lifespan of a session may be implied, or explicit. For instance, a conference call, which represents a virtual meeting, may have a scheduled start time, and a designated end time. These limitations may represent resource allocation scheduling much like reservation of physical meeting rooms. This later case represents explicit session lifespan planning. Implicit limitations may come from the understanding that although there is no planned termination for a session, it will terminate in a short while, such as a telephone call for consulting on a certain product installation to a customer. Sessions may also have no limit and may terminate eventually due to lack of interest, or failure of resources, such as news groups where sessions may last as long as the specific news group is still defined and accessible on the network.

Terminology for session life-span

7.1. Limited Session – session which has predefined termination time or lifetime period limitation.

7.2. Unlimited session – session which can last for indefinite period of time, such as an Internet new group, or a thread in an email exchange.

2.8 Controlling and Monitoring a Session

A session is always controlled. This is done by one of the peers, who carries out controller's role. It does not have to be a person. For instance, a scheduling system may initialize a phone conference among peers and act as the controlling agent for the session. In the same domain, a conferencing software may control access to the session by interested peers via a password prompt, either by telephone dialing, or WEB browsing. Controlling a session answers problems of access permission of peers to the session, to visualization of its task, and to the data repository used for the session. In a complicated session with many related subsessions, the session acts as the controller of the subsessions as an "agent" playing the role of a controller. For instance, control events can be used to synchronize the activities among the constituent subsessions in a workflow session.

Monitoring of sessions provides a passive reflection of the activities in the session for recording, logging, and reporting. A recording of a session can be used for playback, and may be archived. A log is a simpler kind of a recording, which may pertain only to activities, and not contain contents as recording do.

Terminology for controlling and monitoring a session

8.1. Controlling – activity of a peer executing the role of a controller of a session, having authority over access rights and permissions to other peers in the session.
8.2. Monitoring – activity of a peer executing the role of a monitor for a session, doing authorized eavesdropping on session contents and progress, creating a record for these in some form, stored in some media, and under some rules.

2.9 Event Attributes

Events are a very common entity in computer communications, and are used in combination of all sorts of protocols. The following are a set of what seems to be essential attributes, although it can be argues as to how essential, critical, or otherwise these are. A different issue is the format and "packaging" of these events when exchanged among collaboration components and peers. For instance, an XML [13] format is most popular and generally used form presently.

Terminology for event attributes

9.1. Start time – time when event was created.
9.2. End time – time when event is no longer valid.
9.3. Originating peer – peer which created the event.
9.4. Target peer – peer to which event is intended, or a group or class of such peers (groups and classes of peers were not discussed in this paper). It is possible for target to be “everyone”. If target identifies more than one peer, the event is multicasted, hopefully in an efficient way.
9.5. Event id – identifying the event in some context, which can be the related session, or some uniquely generated id in a wider context.
9.6. Related session – session in which the event was created.
9.7. Message – content of the event, having some pre-identifiable type (e.g. mime-types [14]).
9.8. Message code – code name identifying the event type within some coding scheme as identified in the next attribute.
9.9. Message coding system – coding system where semantics of the message code can be found.

2.10 Terminals

Terminals are devices, logical or physical, or combinations thereof on which a collaboration session is rendered for a human peer. A terminal can also be an application occupying part of the physical rendition resources of a physical device. Terminals rendition level dictates how much information and in what format, the terminal can render, and therefore need. It is possible that while the rendition capabilities of a terminal are limited, it is capable of storing richer information. The terminals capabilities can dictate to the communication system and to the session control how much information and in what format is useful for it.

For example, a simple telephone is capable of rendering voice at a 8KHz quality. A cellular can also receive multimedia data in a limited resolution and colors, as well as frame rate. Likewise, a cellular can also render structural display graphics including text, drawings and images. A computer workstation is a powerful terminal capable of a high quality rendition, and so on.

Terminology for terminals

10.1. Terminal type – classifying the terminal type such as a “hand-held device”, “simle analog telephone set”, “computer workstation”, etc.
10.2. Terminal rendition power – classification of the terminal rendition capabilities to help session control what content is most fit to be sent for the terminal to render. This also include the terminal signal capturing – voice, text, image, etc.
10.3. Terminal communication class – dictates how the terminal can communicate, its availability, and so forth.

CONCLUSION

This paper presents the initial phase of work. It is far from being complete, but will hopefully lead to a better understanding of collaboration in the electronic era. The terminology that will evolve from this paper and follow up research will be useful in eBusiness, eCommerce, eGovernment, eAcademia, eLearning, and so forth. These are all forms of collaboration which, by sharing a common terminology and taxonomy can share and reuse better the eResources available.

ACKNOWLEDGMENTS

My colleagues in the Haifa Research Labs for initial discussions of the ideas presented in this paper, suggestions and advice, and to Chani Sacharen for editing and language.

REFERENCES

1. Baecker, R. M., Grudin, J., Buxton, W. A. S., and Greenberg, S. *Readings in Human-Computer Interaction: Towards the Year 2000* (Second Edition) Morgan Kaufmann Publishers, Inc. 1995.
2. Dix, A., Finlay, J., Abowd, G., and Beale, R. *Human-Computer Interaction* (Second Edition) Prentice Hall. 1998.
3. Johansen, R. *Groupware: Computer Support for Business Teams.* 1988, The Free Press.
4. Lynch, K., Snyder, J., and Vogel, D. “The Arizona Analyst Information System: Supporting Collaborative Research on International Technology Trends.” In Gibbs, S.,

and Verrijn-Stuart, A. (Eds.) 1990. *Multiuser Interfaces and Applications* North-Holland, pp. 159-174.

5. Grudin, J. "Computer-Supported Cooperative Work: History and Focus" Computer, May 1994, pp.19-26.
6. Ellis, C.A., Gibbs, S.J., and Rein, G.L. "Groupware: Some Issues and Experiences", CACM, Vol 34, No 1, pp.38-58, Jan 1991.
7. UNITE: **U**biquitous a**n**d **I**ntegrated **T**eamwork **E**nvironment, a 5-th EU Program Framework consortium, http://www.unite-project.org/public/index.html
8. Smith, E., "JCCUM: Java Communication and Collaboration Unified Middleware", 2002, IBM Haifa Research Lab, Haifa, Israel, Internal memo (part of the UNITE project)
9. Ribak, A., Jacovi, M., and Soroka, V. "Ask before you search": peer support and community building with ReachOut," *Proceedings ACM Computer Supported Cooperative Work (CSCW 2002), New Orleans, LA, 2002.* Available in: http://www.haifa.il.ibm.com/km/ct/reachout/papers/p212-ribak.pdf(275KB)
10. IETF RFC 821, SMTP – Simple Mail Transfer Protocol, http://www.ietf.org/rfc/rfc821.txt.
11. Rumbaugh, J., Jacobson, I., and Booch, G. *The Unified Modeling Language Reference Manual*. 1999, Reading, MA.: Addison-Wesley.
12. Foster, I., and Kesselman, C., editors, *The GRID: Blueprint of a New Computing Infrastructure.* 1998, Morgan Kaufman.
13. XML – "e**X**tensible **M**arkup **L**anguage", http://www.w3c.org/XML.
14. N. Fried, and N., Borenstein, IETF RFC 2046 "Multipurpose Internet Mail Extension (MIME) Part Two: Media Types", Network Working Group, November 1996. http://www.ietf.org/rfc/rfc2046.txt