Where and with whom do you wanna meet?: Session-based collaborative work

Flávia Linhalis Arantes*
NIED
UNICAMP
flavialin@gmail.com

Claudia Roberta de Moraes
Salomão Henrique Pereira Silva
Renata Pontin M. Fortes
Maria da Graça C. Pimentel
Universidade de São Paulo
renata,mgp@icmc.usp.br

ABSTRACT
In this paper we revisit the concept of session applied to collaborative work and define a set of operators to manipulate session information related to workspaces and tools scheduling in collaborative environments. By means of a proof-of-concept implementation which uses web services and context-aware tools, we show an example of how session information provided by the operators can be used to leverage the ubiquitous capabilities of collaborative applications.

Categories and Subject Descriptors
H.5.3 [**Information Interfaces & Presentation**]: Group and Organization Interfaces—*Computer-supported cooperative work*

General Terms
Design, Documentation, Human Factors.

Keywords
CSCW, Context information, Ubiquitous computing.

1. INTRODUCTION
Users tend structure their environments when working collectively. They configure specialized places (or workplaces) to support a variety of activities demanding communication and information exchange. Web collaborative workspaces share many common features and needs. The workspace, as being the place where interest groups meet to accomplish their collaborative activities, must be easily accessible to all participants, must support communication among selected team members, and must allow access to collaborative tools. Considering the need to support such a broad scenario, the literature has investigated alternatives for designing (e.g. [13, 32]), building (e.g. [5, 15]) and evaluating (e.g. [8, 30]) collaborative applications not only in the workplace (e.g. [1, 4, 5, 7, 17]) but also with children [23] and teenagers [22]. Technologies and concepts deployed include virtual reality [16], agents [27], middlewares [19], Web Services (e.g. [6, 12, 29]) as well as context information (e.g.[3, 7, 9, 11, 28]).

The software elements of shared workspaces frequently include collaborative tools, such as users and groups management, wikis, chats, whiteboards, instant messengers, audio and video capture tools, and so on. The hardware may include cameras, audio capturing systems, electronic whiteboards, tablets, RFID readers, etc.

In accordance with the results discussed in the literature, our experience with shared workspaces has identified several software requirements that need to be considered towards meeting user needs in collaborative activities. For example, collaborative session scheduling, participants and tools selection, capturing and reuse of generated information per session, instrumented rooms selection, and easy access to scheduled session workspaces. The characteristics about workspaces and participants are particularly important to coordinate the work between geographically distributed teams. We can group these characteristics in what we call **session**. So, users can schedule their session and define, for example, where a meeting will take place, which participants can join a meeting, when it is going to begin, what is the motivation, and so on.

Some authors propose supporting sessions directly [13, 24] while other authors propose the concept of activity-based computing [3]. We define a **session** as being *the set of context information that is necessary to coordinate collaborative work between co-located teams.*

The session support provided by a collaborative workspace gives rise to what we call **session-based collaborative work**. With the concept of session introduced in workspaces, some control actions can be performed by the collaborative workspace and tools. For instance: selected participants can easily and automatically log into a scheduled session in a specific workspace, tools and material can be available only to selected participants, a collaborative environment itself can join a participant in a session, it can also startup the participant tools and workspace, and so on. The list of possibilities is wide, and limited only by the pervasive restrictions

*This work was carried out while the first author was at the ICMC-Universidade de São Paulo, funded by FAPESP.

imposed by users and workspaces.

We argue that the "session paradigm" has the potential to change users behavior whilst using collaborative workspaces because it has the power to increase the ubiquity capabilities of a collaborative environment by using session context information. For example, imagine a scenario in which a user accessed a collaborative web portal to carry out a meeting with his co-located team. In a traditional web-based paradigm, users must enter the portal in a predefined date and time, provide login and password, remember in which workspace they should enter, and possibly in which tool or tools. With what we call session paradigm, that user have only to schedule the session meeting providing some context information. Then, session-based applications can use that context information to take several actions. In the described scenario, an application could, for example, automatically log users in the meeting session. So, users do not need to remember their login, workspace, tools, and even the portal address. We demonstrate the feasibility of this idea with an implementation of an instrumented workspace.

Although session is such a useful concept to collaboration, it is not included in several shared workspaces and tools available in current collaborative environments — in particular those based on spontaneous conversations. This fact motivated us to formalize this concept and to define a set of session operators. We implemented and encapsulated the operators in such a way that they can be incorporated in shared workspaces and tools.

After the definition and implementation of the operators, we have integrated them in a set of tools, developed to a well-known collaborative environment. Web technologies, such as web services and context-aware applications, bring the desired flexibility and ease of use to our session-based environment, allowing users to schedule and personalize their collaborative meetings and to join them using a single-click on an icon on their desktops – this shows an example of ubiquity reached using session-based information.

This paper is organized as follows: Section 2 reviews some related work. Section 3 presents the session operators and the context information associated with them. Section 4 describes the Session Manager component we developed to encapsulate the operators, details how the Session Manager is used by a set of tools to incorporate session-based information, and how it is used with web services to automatically startup a session in a specific workspace. Section 5 presents the evaluation of a session-based tool and automatic workspace startup. Section 6 presents our final remarks.

2. RELATED WORK

Some collaborative tools and environments work with the session concept or similar, but are not aware of it. For example, chat tools uses the "room" concept, which usually has a set of context information, such as subject, objective and history. Someone creates a room and people join it if they share common interests about its subject [14, 10]. This tool usually have a specific set of session operators, that are not easily extensible to other kind of application.

NetMeeting and SunForum are commercial and representative generic application sharing environments to support face-to-face meetings. These systems support "view-sharing" in a manner referred to as WYSIWIS (What You See Is What I See), in which users have exactly the same view of the shared application at the same time [26]. In order to provide this kind of collaboration, such applications must share at least the same communication channel between the participants of a meeting. We can say this is a limited use of the session paradigm, because it identifies basically context information relative to participants.

LiteMinutes [7] was projected to capture annotations, slides, audio and video in planned meetings. The SmartClassroom [25] enables the teacher to control the presentation of lectures in a context aware smart room. Filochat [30] combines an audio recorder with a tablet for taking notes to construct a meeting record. In these systems, the session paradigm is present and provides various context information. The main difference between these results and the work described in this paper is that we propose a general purpose session-based set of operators, aiming to serve as a basis for session related applications.

In his research, Bardram proposes the concept of activity-based computing to support human activities a ubiquitous computing environment [3]. His framework, which addresses mobility and cooperation in human work activities, has been extensively experimented, in particular in pervasive health-care environments [4]. In the framework there is a one-to-one relationship between an activity and a session and, as in our work, sessions exist even if there are no active users: session management is implicit because users join a session by activating an activity. In Bardram's framework, separate modules provides interfaces to create, delete, manage the runtime behavior of activities: our approach treats sessions directly, detailed by the operators we have defined and used in our applications.

Traditional groupware environments such as Groupkit [24] and Rendevouz [13] provide direct treatment of sessions: in our work we define operations associated with a 3-tier web-based architecture, and demonstrate their use in collaborative applications.

3. SESSION OPERATORS

Over the years we have developed several session-based software tools, aiming mainly to use their context information to investigate problems and solutions in ubiquitous and pervasive computing applications. These tools have been developed in the context of the TIDIA-AE project, which focuses on research leading to development of open source e-learning tools to be used in co-located or distance e-learning contexts. As a result of this project, a web portal[1] offers an e-learning environment with many synchronous and asynchronous tools.

The DiGaE (Distributed Gathering Environment) tool corresponds to a ubiquitous instrumented environment that takes advantage of existing TIDIA-AE tools to provide ubiquity to co-located meetings. We describe the DiGaE environment with more details in the next section.

[1] http://tidia-ae.usp.br/portal

A key contribution in this paper is the definition a general session-based infrastructure to collaborative work which shows that context information provided by sessions can leverage many possibilities to ubiquitous applications.

According to our observations using session-based tools in the TIDIA-AE and DiGaE projects, we defined the following context information as important to collaborative sessions in general:

- A brief description of the session (e.g, its title) to explain its main objective (*what* and *why*);
- The session participants (*who*);
- The start and end times of the session (*when*);
- An indication of the session status, for example, if the session has been closed (to modifications) or not (*how*);
- The workspace and associated tools in which the session will take place (*where*). In some situations, the collaborative activities take place in instrumented environments, and a brief description of the room is also important.

Using the context information, we formalized seven session operators, which can be used by session-based tools and workspaces. The first four operators are the basic ones — which we call CRUD operators. They enable users to create, to retrieve, to update, and to delete sessions.

The operators discussed here are about sessions only. Other issues, like permissions and sessions materials are not covered by the operators to keep their flexibility. Collaborative environment should have their own policies about those issues since, for example, a user with administrator permission may be able to view the sessions scheduled by all users.

It is important to explain that we call the following items "operators" because they can apply context information to modify the way people use their shared workspaces and tools. They have the potential to bring the session concept to collaborative work, to leverage context-aware application development, and consequently, change users behavior whilst using collaborative workspaces.

3.1 The Create Operator

The context information of a session is stored using this operator. In order to create a session, a user may want to add a title and a description, select participants, choose date and time, define the workspace, the tools, and (in some cases) the instrumented environments where the session will take place. It is also important to define the session status, which can be ongoing, opened and finalized.

One variation of this operator allows a user to create default sessions, which are useful when the user does not want to explicit schedule a session. We defined a default session as being the current one when there is no other session scheduled to a specific time and workspace.

General form:
createSession(sessionId, beginDate, endDate, title, description, objective, toolId, workspaceId, participantsId, status)

The operator accepts variations with information about the instrumented room (*instrumentedRoomId*) and about the existence of *default* sessions.

3.2 The Update Operator

The update operator allows a user to change the context information of a session. Like the create operator, it also accepts instrumented rooms as a variation.

General form:
updateSession(sessionId, beginDate, endDate, title, description, objective, toolId, workspaceId, participantsId, status)

3.3 The Retrieve Operator

When a user enters his/her workspace, only his/her scheduled sessions to that workspace and tool are retrieved. If the tool or workspace implements the default session concept, then the default session is also retrieved. Another feature of this operator is that it can retrieve sessions according to its status (ongoing, opened or finished).

General form:
retrieveSession(workspaceId, toolId, participantId, status)

3.4 The Remove Operator

This operator deletes a session. In order to perform this action, it only needs the session ID.

There are some important issues with respect to the deletion of a session in a collaborative environment. For example, if a session is deleted should all its contents (generated during the session) also be deleted? We defined the operators as simple as possible. Questions like that will depend on the collaborative environment policy.

General form:
removeSession(sessionId)

3.5 The Join Operator

This operator joins a user into a session. We can say that this operator is the one which is most dependent of specific parameters associated with the tool. For example, when a user joins a chat room, the following information are important: does the chat room allow reserved messages? Does it support emoticons? Does it allow to write-block participants?, etc.

All the operators can be overloaded or extended to satisfy the needs of specific tools. We kept the *joinSession* operator simple, which facilitates its extension.

General form:
joinSession(sessionId, participantId)

3.6 The Finalize Operator

In collaborative environments some tools and workspaces allows users to access the content produced during the session. This operator indicates that a session was ended (meaning that it will no longer be available for joining): this indicates applications that its contents can be processed, if needed, before being made available to participants.

General form:
finalizeSession(sessionId)

3.7 The Check Conflict Operator

When a session is scheduled, it is important to check if a specific user is already a participant in another session. In the case of sessions that are going to take place is instrumented rooms, this operator indicates that a participant cannot be part of two instrumented room sessions at the same time.

General form:
hasSessionConflict(beginDate, endDate, selectedRoomsId, participantId)

4. SESSION OPERATORS IN USE

We developed the Session Manager as a software component that implements the session operators (Figure 1).

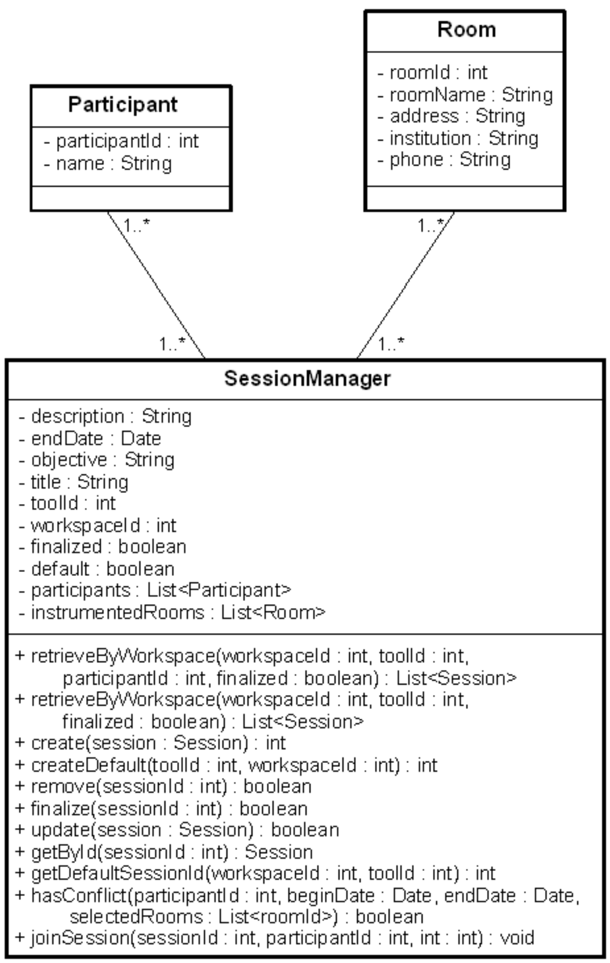

Figure 1: The Session Manager class diagram showing session context information as attributes and session operators as methods.

The Session Manager attributes are related to the session context information identified in Section 3. All attributes have getters and setters methods that are not shown in the diagram to keep it cleaner. The Session Manager methods are an implementation of the operators general forms and some variations. The Participant and Room classes were created to represent participants and instrumented rooms specific information. They are manipulate by the Session Manager methods.

We used the Session Manager in some tools in the context of TIDIA-AE project. When logged into TIDIA-AE Portal, users can access several e-learning tools, such as the Schedule, Announcements, Portfolio, Chat, Whiteboard and Videoconference tools – users with the proper privileges compose workspaces with their tools of choice. In the following, we describe how the sessions operators have been exploited by a set of tools using the Session Manager. We also describe an application as an example of how users can benefit from the ubiquity provided by applications that use session-based context information.

4.1 Session-based Tools

The TIDIA-AE project involves a partnership with the Sakai project[2], and adopts its framework for the development of e-learning tools. We used the Session Manager to implement sessions in three tools, as detailed next.

Whiteboard. This tool provides features to capture pen-based electronic ink annotations, which are useful in several application domains [21]. It provides a rich set of operations which allow a user to manipulate different ink colors and width, use predefined geometric forms, operations involving the copy, cut, paste, move and erase of objects; redo and undo, create new slides, duplicate existing slides or navigate through slides. Whiteboard enables users to capture annotations during a collaborative meeting, to record these annotations and access them in the future. Users can interact with the Whiteboard software using tablet PCs, Interactive Whiteboards or a personal computer.

Chat. The chat tool has some features useful to e-learning context. For example, mediation, interview, write-block a specific user, and reserved messages [10]. These features, configurable by means of a session management service, are demanded when casual message exchange, as the one discussed by Herbsleb et al. [17] for instance, is not appropriated.

DiGaE (Distributed Gathering Environment) DiGaE is an instrumented environment used for general purpose collaborative meetings. An DiGaE environment (a meeting room, for example), is equipped with a video camera, an audio capturing system, an electronic whiteboard to capture pen-based interactions (possibly with a projector), and a RFID reader to user authentication. These are the DiGaE physical elements. The software elements currently used to support DiGaE meetings are the ones corresponding to Whiteboard and the Chat tools described in the previous items, and a Conference tool with offers both audio and video-based communication. The idea is to have DiGaE en-

[2]http://sakaiproject.org/portal

vironments connected through a high speed Internet to enable distributed and collaborative meetings. We developed the DiGaE Session tool specifically to schedule DiGaE sessions. We also developed DiGaE Home tool that integrates the Whiteboard, Chat and Conference tools in a single tool. The DiGaE Home tool can be used in desktop or laptop computers, and enables users to participate in DiGaE sessions even if they are not in the instrumented rooms.

The Sakai platform has a 3-tier application architecture, according to the model-view-controller (MVC) design pattern. The presentation layer contains GUIs used for user interaction vua a web browser. The middle tier contains the set of rules for processing business information and a set of web services to facilitate the access to some Sakai services, while the storage layer manages access for database or file system persistence. Figure 2 shows the Session Manager and our tools applied to the Sakai 3-tier architecture. Sakai modules are presented in black, our modules are shown in white, and the common modules are presented in gray.

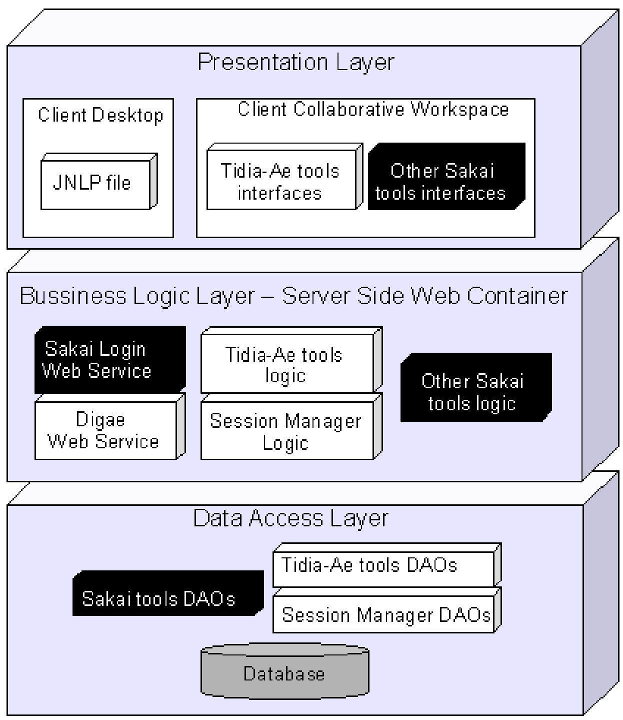

Figure 2: Session Manager and session-based tools applied to Sakai 3-tier architecture.

In the logic tier, the tools access the Session Manager component APIs. The Session Manager is divided into two main modules – logic and DAOs (Data Access Objects). The logic module provides an API to access the generic session operators implementation (which interface is shown in Figure 1). The tools use the Inversion of Control (IoC) design pattern, provided by a Sakai container, to access the Session Manager API. This allows the implementation of the architecture to be easily exchanged by another, as long as interfaces are respected.

Some methods were overloaded in the logic layer, according to specific tools needs. For example, the DiGaE Session tool allows users to choose if they want Chat, Whiteboard, and Conference during a session. Because of this feature, the *createSession*, *updateSession*, and *joinSession* operators were overloaded to support tools selection.

In order to retrieve or store persistent data, the logic layer methods call the data access layer. In the data access layer, the Session Manager DAOs module, through the Sakai Framework, maps the persistent session-based context information, using an integration of Spring and Hibernate. This eliminates lots of code to interact with the database.

In the presentation layer, when users access their workspace tools, the web container processes a JSF (Java Server Faces) page. Our session-based tools (Whiteboard, Chat and DiGaE Session) presents JSF pages with options to create, delete, update and list sessions, according to Sakai's users permission. The JSF pages communicate with the logic layer through a session bean (or backing bean).

The interfaces in the presentation layer were developed considering a user-centered design approach. It consists of iterating design and evaluation phases in the development of an application. By obtaining user feedback at each development phase, an improved new version of the application is build, achieving then a better usability level. The design and evaluation of DiGaE environment and tools interfaces is out of the scope of this paper. Interested readers can find more information in previous work [20].

The presentation layer has no integration with the Session Manager. This is the most laborious part when integrating a tool with the operators, because the Session Manager does not provide any support to session-based interface components. We plan to define a set customized interfaces as a workaround to this problem. We say "define" in the sense of suggest as a good practice, but not implement because the presentation layer is very technology dependent. For example, we used JSF with RichFaces in our tools interfaces to the Sakai platform. One could use JSP (Java Server Pages) to the same platform. This would invalidate the interface components implemented for that technology.

Using this flexible architecture, the TIDIA-AE tools were integrated with the Session Manager modules, in the business logic and data access layers (see Figure 2). The tools and workspaces can now use session context information to enrich ubiquitous applications capabilities, as we describe in the next section with an example.

4.2 Session Automatic Startup

Let's now remember the scenario described earlier, in which a user enters a collaborative portal, such as Sakai, to schedule a work session with his co-located team. Instead of access the Sakai portal and enter the scheduled session in the traditional way, users can access it only with a click in their desktops. In order to provide this ubiquitous functionality, we used session context information together with web services and a JNLP file to automatically log a user in DiGaE Home session.

As previously explained, users can use the DiGaE Session tool to schedule a DiGaE session, with may include the

Whiteboard, Chat, Audio and Video tools.

In order to automatically start a DiGaE Home session, an user must provide his login information, beyond the context information necessary to configure a session. With the context and login information, DiGaE Session tool generates a JNLP (Java Network Launching Protocol) file that can be downloaded to a user's desktop.

When executed, the JNLP file calls the Startup class in DiGaE Session component. This class calls the SakaiLogin web service to authenticate the user and to log him/her into the Sakai portal (see Figure 2). Next, the Startup class calls the DiGaE web service to obtain context information regarding scheduled sessions. Using the user session information, the DiGaE Session tool automatically logs the user into his/her configured and ready to use session.

With this approach, users can take their JNLP file to wherever they want and join DiGaE sessions only with the click of a mouse. The tests with users in collaborative environments verified the convenience of this solution, as described next session. This application shows how session-based context information can increase the ubiquitous possibilities of a system.

5. EVALUATION

Our main objective with the evaluation was to capture users' impressions about using sessions in a collaborative tool and about accessing a session using the automatic startup.

In order to evaluate the automatic startup and sessions we used the DiGaE Home tool, illustrated in Figure 3(a); it integrates the Whiteboard, Chat and Conference tools. The sessions were scheduled with the DiGaE Session tool, shown in Figure 3(b) and (c). Figure 3(b) shows a session scheduling, with objective description, date, time, participants, and tools selection. Figure 3(c) illustrates all the scheduled sessions for a specific user.

The evaluation was performed with 7 users, who were familiar with collaborative activities and tools in general. About 45% of them were already familiar with Sakai and our tools (without sessions), the others were familiar with other collaborative tools such as Skype [3] and Gmail/Gtalk [4].

The evaluators were asked:

1. To enter the DiGaE Home tool without scheduled sessions – i.e., directly access the DiGaE Home integrated tools.

2. To schedule a session, specifying title, the date and time to begin/end, the participants, and the tools (Whiteboard, Chat, Audio and Video).

3. To enter the scheduled session (manually, not automatically).

4. To register the login information and download the shortcut for the automatic session startup functionality (JNLP file).

5. To use the JNLP file to automatically start a session.

6. To answer a questionnaire, which was prepared according to ISO 9126 standard [18], which establishes six characteristics applicable to any type of software. We concluded that four of them fit the features we wanted to evaluate in our tools: functionality, usability, efficiency and reliability.

[3] www.skype.com
[4] www.google.com

First, the DiGaE Home tool was compared with and without sessions (activities 1, 2 and 3). When evaluating the tool without sessions, users noticed it was faster than the alternative. However, most of them considered the sessions a very important feature in collaborative tools. Participants selection, date/time definition, and other session context information are important to collaborative work coordination. About the session scheduling process, all of them classified it as being easy and succeeded after a few attempts. Most of them thought the process to manually enter the session is relatively fast and with an average number of steps – although some users have reported a little amount of errors.

About the use of the automatic session startup (activities 4 and 5), the evaluators classified it as very easy and performed the activities in the first attempt. Most of them thought the feature is important because it simplifies and speeds the entry of a user in a session. According to 72% of the users, the method is very fast. No one reported errors.

We present the questionnaire results in Table 1 (activity 6). The scores were divided into three satisfaction levels (Excellent, Good and Average) and one for unsatisfactory score (Weak). The positive results suggest that participants satisfied with the functionality, usability and reliability of the DiGaE tool, and their main concern was the its efficiency. Future evaluations should clarify if the concern relative to efficiency is regarding the management of sessions or results from the use of the individual tools which compose the DiGaE tool.

Functionality				
	Excellent	Good	Average	Weak
Session	57%	43%		
Startup	50%	50%		
Usability				
	Excellent	Good	Average	Weak
Session	50%	50%		
Startup	58%	36%	6%	
Efficiency				
	Excellent	Good	Average	Weak
Session		36%	64%	
Startup	72%	14%	14%	
Reliability				
	Excellent	Good	Average	Weak
Session	71%	29%		
Startup	100%			

Table 1: Questionnaire evaluation results.

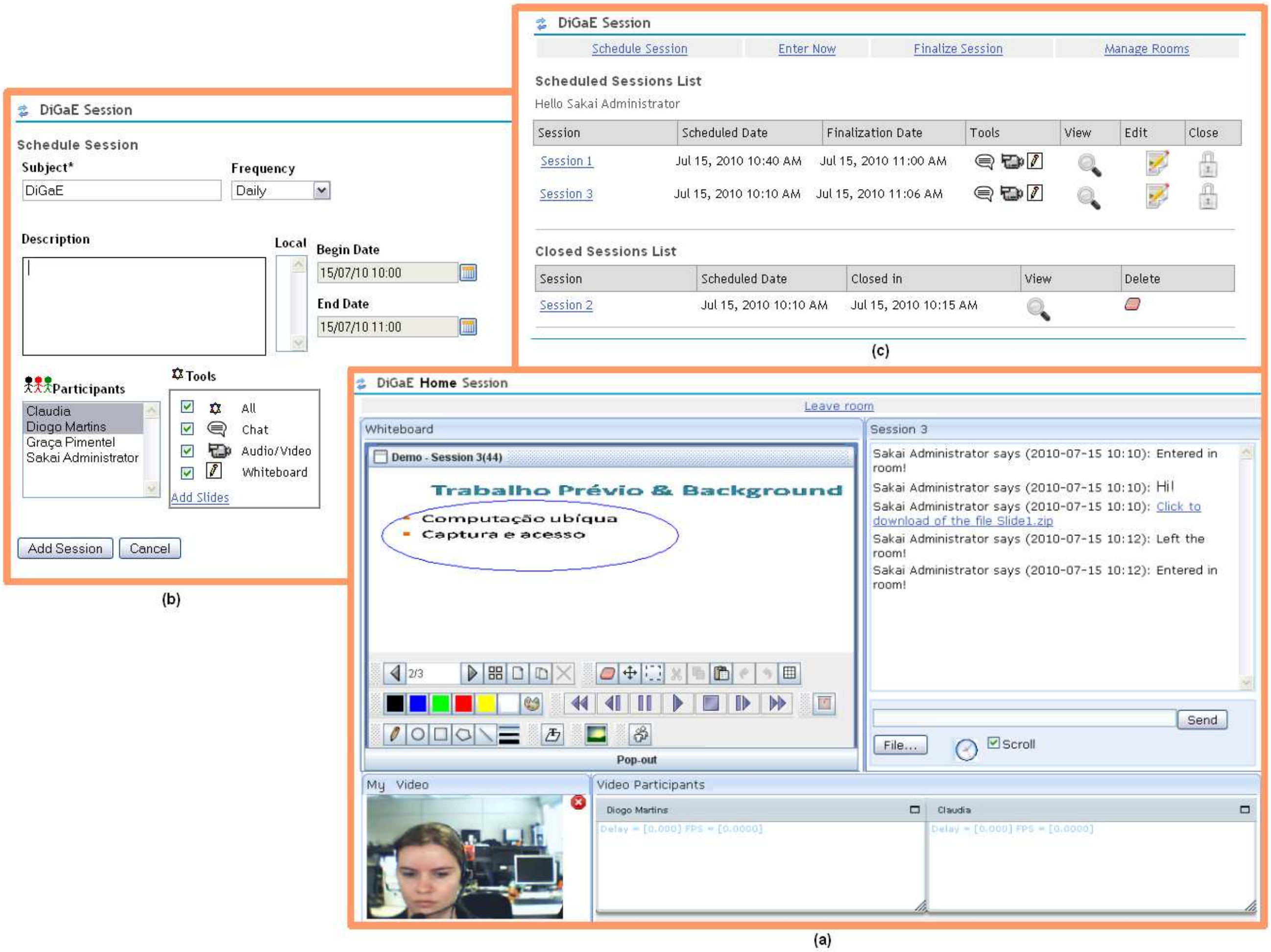

Figure 3: (a) DiGaE Home tool showing integrated Whiteboard, Chat and Conference. (b) DiGaE Session tool showing session scheduling. (c) DiGaE Session tool illustrating all scheduled sessions for a specific user.

6. FINAL REMARKS

In this paper, we have revisited advantages that sessions can bring to collaborative work by means of adding and managing context information to facilitate scheduling and coordination of shared activities. We defined and implemented a set of session operators that can be integrated into shared workspaces and tools. We discussed the operators implementation in some tools in a well known collaborative environment and we showed how session context information can bring ubiquitous possibilities to collaborative teams, demonstrated with the automatic session startup application.

We evaluated one of our session-based tools (DiGaE Session) and the automatic session startup in DiGaE Home with collaborative environments users. The evaluation results were considered positive, which motivated the integration of the Session Manager with another tool our Platform: the Conference tool.

The evaluation showed that our worst result is related to efficiency. We plan to work around this problem in future work in order to maintain the performance required in terms of speed and resource usage. We should include, in this effort, support to manage availability by means of lightweight negotiations as an alternative to handle interruptions, as introduced by and Wiberg and Whittaker [31].

As another future work we plan to implement the Session Manager as a web service to allow its use by collaborative tools other then the Sakai Platform ones. This approach should allow us to integrate in our infrastructure augmented reality features, as discussed by Barakonyi et al. [2]

Finally, we plan to associate semantics to each session operator in the future. An ontology should be developed to the session model in order to better formalize how context information is related to the session concept and operators.

Acknowledgments. Our thanks to the evaluators. We thank CAPES, CNPq and FINEP for funding part of our our. Special thanks to FAPESP for funding the TIDIA-AE Project and our participation in WebMedia 2010.

7. REFERENCES

[1] A. Albin-Clark. Virtual chat in an enquiry-based team project. In *Proc. Conf. Innov. and Techn. in Computer Science Education*, pages 153–157. ACM, 2008.

[2] I. Barakonyi, T. Fahmy, and D. Schmalstieg. Remote collaboration using augmented reality video-conferencing. In *Proc. Graphics Interface 2004*, pages 89–96, 2004.

[3] J. E. Bardram. Activity-based computing: support for mobility and collaboration in ubiquitous computing. *Personal and Ubiquitous Comput.*, 9(5):312–322, 2005.

[4] J. E. Bardram. Activity-based computing for medical work in hospitals. *ACM TOCHI*, 16(2), 2009.

[5] V. Bharadwaj and Y. V. R. Reddy. A framework to support collaboration in heterogeneous environments. *SIGGROUP Bull.*, 24(3):103–116, 2003.

[6] R. Bulcão-Neto, C. H. O. Jardim, J. A. C. Guerrero, and M. da Graça Campos Pimentel. A web service approach for providing context information to cscw applications. In *Joint Conf. WebMedia & LA-Web 2004*, pages 46–53, 2004.

[7] P. Chiu, J. S. Boreczky, A. Girgensohn, and D. Kimber. Liteminutes: an internet-based system for multimedia meeting minutes. In *World Wide Web Conf.*, pages 140–149, 2001.

[8] L. Damianos, L. Hirschman, R. Kozierok, J. Kurtz, A. Greenberg, K. Walls, S. Laskowski, and J. Scholtz. Evaluation for collaborative systems. *ACM Comput. Surv.*, page 15, 1999.

[9] M. Danninger and R. Stiefelhagen. A context-aware virtual secretary in a smart office environment. In *Proc. ACM Intl. Conf. Multimedia*, pages 529–538. ACM, 2008.

[10] C. R. de Moraes, L. da Silva Santos, F. Linhalis, and M. da Graça C. Pimentel. Uma ferramenta de chat com ajax e spring para um ambiente de aprendizado eletrônico. In *Work. Tools and Applications held in conjunction with Proc. Brazilian Symposium on Multimedia and the Web (Webmedia)*, pages 176–178, 2008.

[11] D. Dearman, K. M. Inkpen, and K. N. Truong. Mobile map interactions during a rendezvous: exploring the implications of automation. *Personal and Ubiquitous Comput.*, 14(1):1–13, 2010.

[12] S. Dustdar, H. Gall, and R. Schmidt. Web services for groupware in distributed and mobile collaboration. In *Euromicro Conf. Parallel, Distributed and Network-Based Processing*, pages 241–247. IEEE, 2004.

[13] W. K. Edwards. Session management for collaborative applications. In *Proc. ACM Conf. CSCW*, pages 323–330, 1994.

[14] D. Fono and R. Baecker. Structuring and supporting persistent chat conversations. In *Proc. ACM Conf. CSCW*, pages 455–458, 2006.

[15] S. Greenberg and D. Marwood. Real time groupware as a distributed system: concurrency control and its effect on the interface. In *Proc. ACM Conf. CSCW*, pages 207–217, 1994.

[16] C. Greenhalgh and S. Benford. Massive: a collaborative virtual environment for teleconferencing. *ACM Trans. Comput.-Hum. Interact.*, 2(3):239–261, 1995.

[17] J. D. Herbsleb, D. L. Atkins, D. G. Boyer, M. Handel, and T. A. Finholt. Introducing instant messaging and chat in the workplace. In *Proc. ACM CHI*, pages 171–178, 2002.

[18] ISO/IEC. Software engineering – product quality – part 1: Quality model. Available at: http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749, 2001.

[19] M. A. S. Mangan, M. R. S. Borges, and C. M. L. Werner. A middleware to increase awareness in distributed software development workspaces. In *Joint Conf. WebMedia & LA-Web 2004*, pages 62–64, 2004.

[20] V. G. Motti. Design centrado no usuário de um ambiente de reunião instrumentado. Master Thesis. Universidade de São Paulo, available at: http://www.teses.usp.br/, 2009.

[21] M. G. C. Pimentel, R. G. Cattelan, and L. A. Baldochi Junior. Prototyping applications to document human experiences. *IEEE Pervasive Computing*, 6(2):93–100, 2007.

[22] A. M. Piper, E. O'Brien, M. R. Morris, and T. Winograd. Sides: a cooperative tabletop computer game for social skills development. In *Proc. ACM Conf. CSCW*, pages 1–10, 2006.

[23] J. Rick, A. Harris, P. Marshall, R. Fleck, N. Yuill, and Y. Rogers. Children designing together on a multi-touch tabletop: an analysis of spatial orientation and user interactions. In *Proc. Intl. Conf. Interaction Design and Children*, pages 106–114. ACM, 2009.

[24] M. Roseman and S. Greenberg. Building real-time groupware with groupkit, a groupware toolkit. *ACM TOCHI*, 3(1):66–106, 1996.

[25] Y. Shi, W. Xie, G. Xu, R. Shi, E. Chen, Y. Mao, and F. Liu. The smart classroom: Merging technologies for seamless tele-education. *IEEE Pervasive Computing*, 2(2):47–55, 2003.

[26] M. Stefik, D. G. Bobrow, G. Foster, S. Lanning, and D. Tatar. Wysiwis revised: early experiences with multiuser interfaces. *ACM TOIS*, 5(2):147–167, 1987.

[27] T. A. Tavares, S. A. Oliveira, A. M. P. Canuto, L. M. Gonçalves, and G. S. Filho. An infrastructure for providing communication among users of virtual cultural spaces. In *Joint Conf. WebMedia & LA-Web 2004*, pages 54–61, 2004.

[28] K. N. Truong, G. D. Abowd, and J. A. Brotherton. Personalizing the capture of public experiences. In *ACM UIST*, pages 121–130, 1999.

[29] V. H. Vieira, D. G. Sante, A. P. Freire, and R. P. M. Fortes. A web service for CSCW applications. In *Proc. Brazilian Symposium on Multimedia and the Web (Webmedia)*, pages 1–3. ACM, 2005.

[30] S. Whittaker, S. Tucker, K. Swampillai, and R. Laban. Design and evaluation of systems to support interaction capture and retrieval. *Personal Ubiquitous Comput.*, 12(3):197–221, 2008.

[31] M. Wiberg and S. Whittaker. Managing availability: Supporting lightweight negotiations to handle interruptions. *ACM TOCHI*, 12(4):356–387, 2005.

[32] N. Yankelovich, W. Walker, P. Roberts, M. Wessler, J. Kaplan, and J. Provino. Meeting central: making distributed meetings more effective. In *Proc. ACM Conf. CSCW*, pages 419–428, 2004.