Journal of Universal Computer Science, vol. 14, no. 1 (2008), 123-147
submitted: 6/11/06, accepted: 11/10/07, appeared: 1/1/08

Using PDAs in Meetings: Patterns, Architecture and Components

Gustavo Zurita
(University of Chile, Chile
gnzurita@facea.uchile.cl)

Pedro Antunes
(University of Lisboa, Portugal
paa@di.fc.ul.pt)

Nelson Baloian
(University of Chile, Chile
nbaloian@dcc.uchile.cl)

Luís Carriço
(University of Lisboa, Portugal
lmc@di.fc.ul.pt)

Felipe Baytelman
(University of Chile, Chile
felipe@baytex.net)

Marco Sá
(University of Lisboa, Portugal
marcosa@di.fc.ul.pt)

Abstract: This paper addresses the role of Personal Digital Assistants (PDAs) in electronic meetings. Several real-world scenarios of PDA usage in meetings are defined using a pattern language. Anchored on these scenarios, we propose an upper-layer meeting middleware which addresses three major goals: defining a common architecture and set of components for meeting systems; standardizing the meeting memory and process data structures commonly managed by electronic meetings; and supporting XML-based interoperability between these components. The patterns, architecture and components were validated through their adoption in three applications, developed by different teams and covering quite different domains. The applications, encompassing several meeting patterns and adopting multifaceted combinations of the upper-layer components, demonstrate the high level of interoperability supported by the proposed upper-layer middleware.

Keywords: Electronic Meetings, Meeting Patterns, Upper-Layer Middleware
Category: H.1

1 Introduction

PDAs have been recently regarded as powerful CSCW devices combining several well-known characteristics such as autonomy, mobility, pervasiveness, transportability and unobtrusiveness with shared information support. Two good examples are the mediation between healthcare personnel in hospital environments [Muñoz, 2003] and the relationships between therapists and patients [Sá, 2007], where mobility and flexibility are extremely necessary.

PDAs may also assume a paramount role in the meeting support scenario. PDAs simplify the way people bring information into and out of meetings and serve as a dissemination tool for meeting-related information throughout the organization [Costa, 2001]. Another important role is reducing the footprint of technology in meetings, since people are used to highly resilient, non-intrusive and low-tech solutions such as paper and pencil. Moreover, meeting processes are governed by complex and subtle procedures, which in many circumstances require an expert facilitator, who may also benefit from PDA support [Antunes, 2001].

Besides these potential benefits we should also account for a number of drawbacks. The complexity associated to meetings has always challenged information technology [Briggs, 2003, Nunamaker, 1997]. Meetings may be distributed in time and space, posing significant restrictions to shared context awareness. They also bring together people with very distinct abilities, making it difficult to specify the interaction requirements. Many times people are forced to plan the meeting process in advance, while other times such advance planning is impossible (e.g., emergency management), making technology configuration and management highly contextual dependent. To complicate even further these matters, meeting activities must be constantly adapted to the varying perceptions of problems and goals.

One recent attempt to resolve many of these problems evolves around the notion of Collaboration Engineering (CE): the capability to deploy meeting technology in a systematic, repeatable, measurable and sustainable way, based on reusable collaboration components and the best practices in the field [Briggs, 2003, Santanen, 2006, Vreede, 2006]. CE relies on well known and repeatable solutions and, most importantly in a long-term perspective, the opportunity to transfer tailorability to the end-users, so that no human expert is necessary to configure and operate meeting technology.

Unfortunately, CE is still in its infancy. In particular, we observe that CE has not yet been accompanied by the middleware infrastructure necessary to supply the envisioned levels of flexibility, tailorability and interoperability. This research work aims to study and develop such middleware infrastructure. More precisely, our objective is to conceptualize and build an "upper-layer" meeting middleware on top of currently available basic collaboration services.

The distinction between the upper-layer meeting middleware and the basic collaboration services is important to avoid addressing yet again well known problems with collaboration support, e.g., information sharing, group management, multi-user interaction, persistency, messaging, etc. The upper-layer meeting middleware is focused on supporting the user activities through various contexts and work modes, while at the same time providing collaboration awareness and orchestrating the various underlying services.

A natural consequence of this view is that the upper-layer middleware is more influenced by the applications than by the technological constraints of collaborative technology. This explains why we decided to start our research by analyzing a collection of representative application scenarios on which the upper-layer middleware could subsequently be founded.

This paper also benefits from a key opportunity: it consolidates research work performed by three independent groups studying PDA usage in quite different scenarios. One such group was working in face-to-face electronic meetings at the University of Lisboa, while another, working in the same location, was using PDA technology to support psychotherapy. The third group, working at the University of Chile, was developing PDA-based brainsketching and brainwriting tools for learning environments. The fortunate intersection of these interests highlighted many common problems and solutions, and led to a common understanding consolidated in the proposed upper-layer meeting middleware.

Our contributions to the state of the art are the following ones:

- We define a collection of representative meeting scenarios where PDAs may be used to best advantage as meeting tools, either because they support information management, simplify the meeting process or increase contextual awareness.
- We model the information structure required by the application scenarios. This includes standardizing the memory and process elements managed by meeting systems.
- Based on the defined scenarios and information structure, we propose and validate a set of integrated services supporting meeting participation throughout various contexts and work modes. This collection of integrated services constitutes the upper-layer middleware.

The paper is organized as follows. In section two we present the research context. The section three is dedicated to introduce the developed applications. Section four describes the meeting patterns inspired by our experience with the applications. In section five we present the upper-layer meeting middleware. Finally, in sections six and seven we discuss the obtained results and present our research conclusions.

2 Research Context

PDAs attracted the interest of researchers in electronic meetings mostly because of their non-obtrusiveness, user-interface capabilities (e.g. freehand input [Davis, 1998]), ubiquity [Baldonado, 2000, Costa, 2001] and mobility [Wiberg, 2001]. Naturally, some research studies explored the integration of PDAs with already existing meeting components and systems. This was the case of large shared displays known as Single Display Groupware (SDG) [Greenberg, 1999, Myers, 1998, Stewart, 1999], which have for long been considered fundamental to focus the participants on convergent tasks.

The combination of PDAs and SDG highlighted two potential benefits of componentizing electronic meetings: interoperability and flexibility. For instance, Pebbles [Myers, 1998] supports the meeting participants developing their individual ideas on a PDA and, whenever necessary, interconnecting the PDA and SDG to share ideas with the group. Pebbles even allows PDAs to remotely control the SDG presentation as if they were PC mice and keyboards.

Further developments componentizing electronic meetings highlighted the important role of another meeting component: the Shared Repository (SR). The SR allows meeting participants to share meeting information before, during and after meetings, thus expanding the application context of electronic meetings beyond the traditional face-to-face and remote settings. The SR is essential to maintain organizational knowledge. For instance, NotePals [Davis, 1999] and ShareNote [Greenberg, 1999] support creating personal notes in PDA anytime before meetings, publicizing notes when people meet and managing the shared notes after people leave meetings. Another example of the use of the SR is given by the Notable [Baldonado, 2000] system, which uses the Post-It metaphor to enrich the notes taken in meetings with other relevant information.

More recently, the componentization of electronic meetings has pursued an increased flexibility and mobility in information sharing, offering ways to share notes across multiple contexts, including face-to-face meetings in mobile situations. For instance, FieldWise [Fagrell, 2000] and RoamWare [Wiberg, 2001] support knowledge sharing amongst mobile and distributed meeting participants, including face-to-face meetings held in such places as corridors, relying on wireless technology to detect the group members and support information sharing. Meeting Space [Neyem, 2006] supports fully decentralized ad-hoc meetings for highly mobile situations, such as disaster relief, relying on no additional infrastructure than the one provided by the mobile workers, who may share comments and documents in an opportunistic way. Some other relevant approaches consider the adaptive distribution of content based on the resources available to users, integrating also information from the surrounding devices [Roman, 2005].

Reflecting over the functionality provided by the systems mentioned above, we observe that what is currently missing is a scheme to make the various architectural solutions and components interoperable. We also observe that although many systems identify some common meeting information types (e.g. personal and meeting notes), they do not aim to standardize and integrate meeting information. Nevertheless, this standardization is necessary to support interoperability. The support to flexibility and interoperability based on a collection of standardized and reusable components is exactly the type of functionality implied by the upper-layer meeting middleware proposed in this paper.

The definition of upper-layer middleware was proposed by a group of experts mandated by the European Commission to define key challenges for the Next-generation Working Environments (NWE) [Collaboration @ Work, 2004]. According to the proponents, an upper-layer collaboration platform resides on top of existing middleware platforms, currently offering basic collaboration services such as person-to-person communication, web services, remote object invocation, persistency, reliability, and security [Laso-Ballesteros, 2005]. This upper-layer would allow co-workers to orchestrate their activities while changing work contexts, places and work modes; the underlying services would be used in a flexible way and customized by the users. The vision is that a large suite of collaborative activities will seamlessly utilize the upper-layer collaboration platform. These are expected to include the e-business, e-commerce, e-manufacturing, e-government, e-health and e-learning application domains.

The IST European research program has been supporting several research projects on NWE [AMI, 2006], but they do not specifically address electronic meetings. We thus have here the opportunity to further develop this concept in the context of electronic meetings, also profiting from the many ideas being experimented with the previously cited systems.

3 Applications

The following applications were developed with the purpose of investigating different meeting scenarios, architectures and components. The upper-layer middleware proposed in this paper is based on the experience gained while developing and using these applications. The details provided below introduce the reader to the aims and generic functionality of the several applications. Later on we relate these prototypes with our design decisions and considerations about the upper-layer middleware.

3.1 LightMeet

The major goal of LightMeet is to explore the use of PDAs and SDG in face-to-face meetings. Considering the importance given to the SDG, LightMeet is mostly adequate to decision convergence:

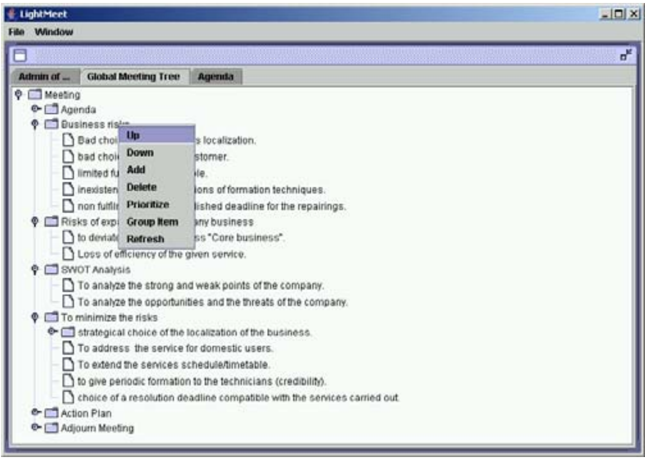

Figure 1: LightMeet SDG component

Decision convergence. The generic group decision-making process is usually divided in two major phases: divergence and convergence. While the first phase is mostly concerned with finding and exploring the problem context, which can be done individually, the second one is focused on the group and its need to reduce possibilities, find possible solutions and ultimately adopt one single solution. Convergence requires sophisticated and complex group abilities, since the participants have to interact, negotiate and persuade in order to build consensus.

LightMeet includes SDG, SR and PDA components. The central component is the SDG, focusing the meeting participants in one common, visible and interactive artifact, a principal requirement associated to convergence. In this scenario, the PDAs play a secondary role, serving as individual scratchpads, where information may be managed before being delivered to the SDG.

LightMeet adopts a simplified information model: all meeting data is organized in a tree. This not only includes the typical meeting data, such as proposed solutions, comments and priorities; but also the agenda and report, which are treated by the system as “special” nodes. Our principal reason for adopting this simplified approach was that we aimed to develop a very simple mental model of meeting systems, so that any participant not familiarized with them could nevertheless participate and interact with the system.

As shown in Fig. 1, the SDG allows the meeting participants to create and manage the shared meeting information using the Drag & Drop and Explorer metaphors that are now very common in many technological devices. Observe also that the SDG has tabs displaying specific portions of the tree, such as the agenda. The admin tab is dedicated to manage the connections to PDAs.

Figure 2: LightMeet PDA component

The PDAs operate as dynamic cursors over the tree available in the SDG, i.e. only single nodes and the immediate sub-nodes are displayed in the PDAs (Fig. 2). When a user moves around the tree, the PDA requests the corresponding node and sub-nodes to the SDG. The participants may freely create, delete, move and modify the nodes. Since the agenda and report are special nodes (e.g. they cannot be deleted), the users have shortcuts to easily move there. The PDAs have private spaces where users can privately edit nodes. There is also support for publishing these private nodes in the SDG.

The SDG adopts an optimistic approach to concurrency control, allowing the participants to freely manipulate the nodes, and relying on the face-to-face interactions to resolve any occurring problems. As we observed in our experiments, the information management is sometimes chaotic but the meeting participants can easily define a social protocol. Sometimes, when there is a definite need to control the information management, the interaction is restricted to the SDG.

Additional comments. The LightMeet system was evaluated in a laboratory experiment with two groups of users with different levels of proficiency with computers. One group had five participants with low computer skills, including five persons with degrees in different fields, such as economics and management. The other group had six participants highly proficient with computers, mostly with degrees in informatics and mathematics.

The experiments were conducted with short briefings about the meeting technology, followed by face-to-face electronic meetings, and concluded with a questionnaire with 25 questions based on SUMI (Software Usability Measurement Inventory). The meetings involved the discussion of the risks of underpinning a home-based business, using pre-defined agendas resembling SWAT analysis.

The obtained results indicate that the affective criteria (user friendliness and emotional reaction) were the most positively evaluated, followed by the ease of learning, control, and efficiency. The most negative evaluated item was the system usefulness. More details about the system and evaluation results can be found in [Pereira, 2006].

Chronologically, LightMeet was our first application exploring the major concepts of an upper-layer middleware. It decisively inspired the later applications in several ways: identifying requirements (flexibility, interoperability, tailorability); giving ideas about the upper-layer functionality (e.g. managing meeting data with nodes and dynamic cursors); and giving preliminary indications about usefulness and usability.

3.2 Nomad

The major goal of Nomad is supporting the collaborative generation and manipulation of sketches using PDAs. The development of Nomad has been focused on two major scenarios: collaborative design and learning.

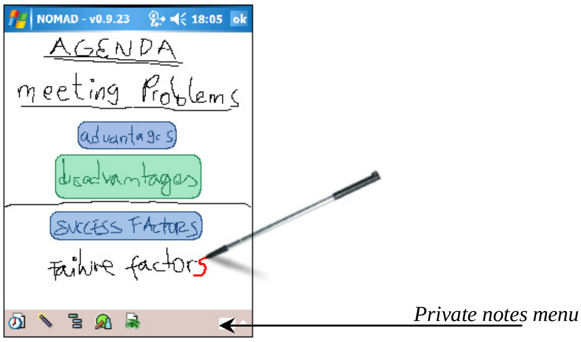

Figure 3: Nomad screenshot

Collaborative design. The collaborative design scenario emphasizes the use of inspiration, creativity and collaboration to produce innovative ideas, which may be expressed through sketching and writing. This emphasizes the ability to use PDAs as collaborative scratchpads (unlike LightMeet, which used PDAs as individual scratchpads), profiting from the native support to pen-based interactions. Fig. 3 shows a screenshot of Nomad during a typical design meeting.

Nomad implements a set of special gestures to facilitate design activities. Two examples are the select and cut functions illustrated in Fig. 4. The select function utilizes a gesture designated double lasso. The cut function utilizes a cross gesture to remove strokes from the screen.

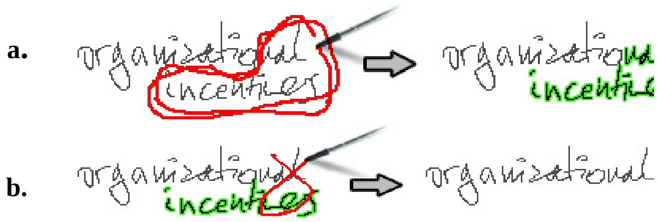

Figure 4: Double lasso gesture for complex selecting and cross gestures for cutting

Nomad organizes meeting information in hierarchical pages. Pages are also created with a special gesture, which consists of surrounding a piece of text with a partial rectangle, as shown in Fig. 5. After this gesture, a page is created and named after the piece of text that was originally surrounded.

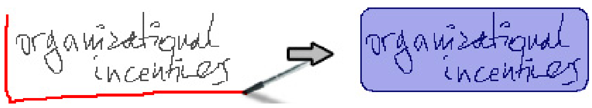

Figure 5: Surrounding a piece of text to create node

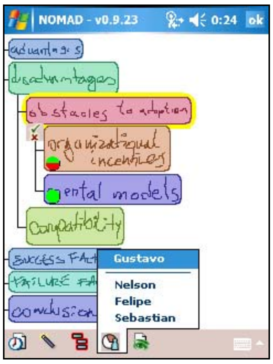

Figure 6: Overview window

Since the page hierarchy may become very large, Nomad provides an overview window allowing the user to move around and zoom in and out using predefined gestures (Fig. 6). The overview also gives awareness on who is participating in the meeting or working on a node. Another functionality allows users to vote for or against nodes. This functionality requires a designated person, the meeting facilitator, to request the participants to vote on one or several designated nodes. The participants vote using the predefined gestures shown in Fig. 7. The results are represented as pie charts, where the green portion represents positive votes, red represents negative votes and black represents users who have not voted yet.

a.

b.

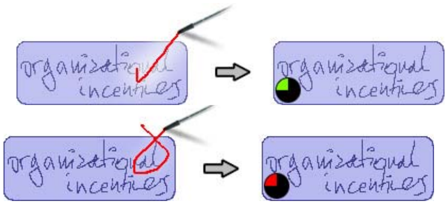

Figure 7: Voting gestures (note also the voting results)

Collaborative learning. Nomad was also developed to support collaborative problem solving and discussion in learning meetings. In this type of meeting, students are encouraged to collaboratively develop solutions to problems suggested by a teacher using PDAs. The teacher may also use a PDA to follow the discussions and give on-line feedback. Nomad attempts to increase the learning motivation [Dev, 1997] in two different ways: (1) giving assessments as soon as possible [Strong, 1995]; and (2) promoting mastery learning [Bloom, 1980], which means that "when a student completes an assignment that does not meet the expected criteria, give her or him one or more opportunities to tackle the task again, with assessment or guidelines on how to achieve the desired result [Dev, 1997]."

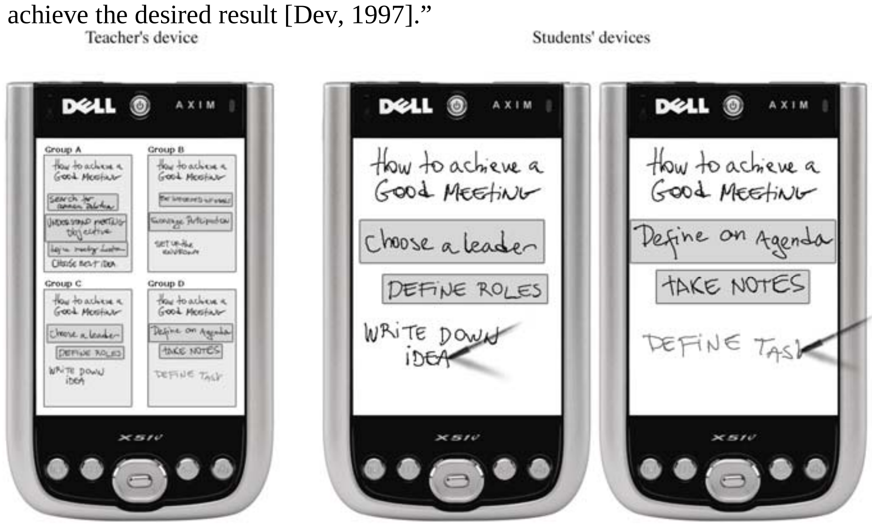

Figure 8: Feedback and assessment from the teacher to the students

Fig. 8 illustrates this scenario with a case where groups were assigned to write conditions for achieving a good meeting. Each group contributes with ideas using the handheld stylus, while the teacher remotely watches their progress. In Fig. 8, the teacher's PDA is shown to the left (displaying a summary view of four participating groups), while two students' PDAs are shown to the right. In this scenario, the teacher actively participates in the task, giving on-line feedback and cooperating with students towards an appropriate answer.

Additional comments. Nomad operates in a synchronous mode, relying on a wireless network to share meeting information. The system may support SR and SDG, although no dedicated components have been developed yet. Nomad allows users to generate information locally and, when necessary or convenient, export local information to a meeting. The system provides two export options: one where the individual and meeting trees are merged, another where the individual tree becomes a new version of the meeting tree.

Chronologically, Nomad was developed after LightMeet and served as our major test bed for consolidating the upper-layer middleware architecture, components and services. Building upon the generic architectural issues studied with LightMeet, our efforts were centered on two endeavors: (1) making the upper-layer independent from the underlying infrastructure (e.g. network topology, connectivity), characteristics of physical devices (e.g. PDA, SDG) and data sharing services (e.g. persistency, synchronization); and (2) addressing the generation and manipulation of data at the user level (e.g. gestures, sketches, notes, pages, zooming) and the group level (e.g. sharing, awareness, feedback).

3.3 JoinTS

The major goal of the JoinTS (Joint psychological Therapy Support [Sá, 2007]) system is exploring PDA support to psychotherapy. The psychotherapy processes occur in several scenarios, two of them currently covered by JoinTS: individual and group therapy.

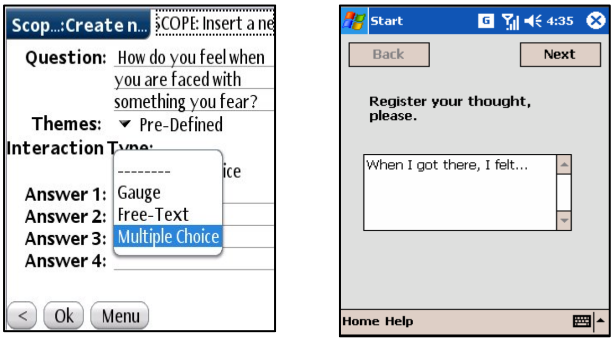

Figure 9: Screenshots of two JoinTS tools developed for different PDAs. Letft: The questionnaire customization tool developed for Palm. Right: The therapy tool developed for Pocket PC

Individual therapy. Individual psychotherapy meetings are complex and demanding. They require numerous activities performed by a pair of participants, constituted by a therapist and a patient. The therapy meetings are primarily centered on face-to-face interactions held in the office but are also inherently related to individual activities performed between meetings.

On the therapist's side, we account for the preparation and follow-up of face-to-face meetings, while on the patient's side we include multiple activities prescribed by the therapist, such as responding to questionnaires, planning daily life and registering thoughts throughout the day. JoinTS supports all the information management, including the customization of questionnaires and forms, form filling, note taking during face-to-face meetings, registering thoughts, visualizing and analyzing the accomplishments (Fig. 9).

The initial therapy steps are informal, because the therapist is setting up the stage for addressing the patient's problems. Most of these activities are conversation-based and collaborative. After the main problems are defined, the therapist adopts a more structured approach, therapy meetings become more carefully planned and start following a strict agenda. Homework is given to the patient and the results are analyzed and discussed in subsequent meetings.

The role of PDAs supporting this process occurs in multiple ways: (1) supporting the therapist's note taking during meetings without obstructing the face-to-face interactions; (2) supporting the patient's individual tasks between sessions; (3) supporting the exchange of questionnaires between the therapist and patient; and (4) functioning as a SDG, focusing the attention on specific issues raised by the therapist during sessions. We specifically emphasize the important role of PDAs assisting the patient's individual tasks between sessions, presenting hints and suggestions whenever an abnormal behavior is detected, according to rules specified by the therapist.

Group therapy. Individual therapy is frequently complemented with group therapy. Here, meetings involve several patients sharing the same pathology. Although the main activities are similar to those described for individual therapy, goals and procedures diverge for each scenario. Generally, questionnaires are filled cooperatively and thought registration is many times subject to consensus. Overall, every activity requires the intervention and participation of all patients, always guided by the therapist. On occasions, therapists work in pairs to accommodate the various parallel tasks that have to be accomplished during meetings. Consequently, two groups emerge. The first one is composed of the therapists, who exchange specific information between them, whilst the second group is composed of patients.

As with the individual therapy, initial meetings focus on the adaptation to the procedures and group. Meetings having more clear objectives and schedules then follow these informal meetings. Therapists control the topics and subjects addressed during these meetings, communicating simultaneously with several patients. This scenario requires a SDG to focus all the participants on the objectives and facilitate collaboration. PDAs support the various underlying activities, including control of the SDG (see Fig. 10). Also in this scenario, the JoinTS system uses a SR to preserve annotations, patient records and questionnaires. The therapists, using their PDAs, may retrieve relevant information from the repository and publish it in the SDG.

Another possibility that has been explored is supporting private communication between the two therapists using PDAs (see Fig. 10). This subgroup is particularly important when critical issues emerge. In many cases, resolving these critical issues becomes a major goal, turning the therapists particularly active. The communication between therapists is considered uncomfortable for patients in normal meetings but, using PDAs, these private conversations are less conspicuous.

Additional comments. JoinTS relies on a wireless network and a centralized server to synchronize and control the information flows among the PDAs, SR and SDG. There is one communication channel available for sending information from PDAs to the SR and then either to the SDG or the therapists' PDAs. There is a second channel available for receiving information directly from the therapists' PDAs.

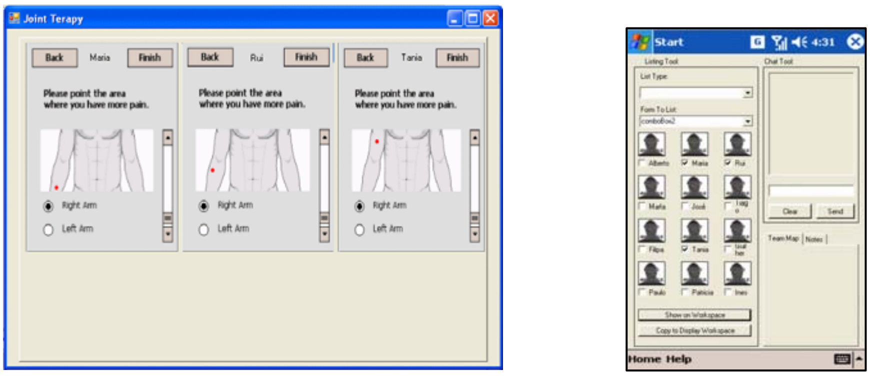

Figure 10: Left: Screenshot of the JoinTS SDG, showing the selected patient's PDAs. Right: The list of patients is selected on the therapists' PDAs. You may also observe the chat space allowing therapists to share notes

The SR preserves all the information exchanged by the system as well as descriptions of the meeting activities. Each intervention is stored with its validity, author and time. A log for each participant is generated, as well as a log for the whole meeting. Therapists may trace the patients and groups' evolution from meeting to meeting.

JoinTS was the last application being developed and is currently under further developments. Its architecture fully adopted the upper-layer middleware developed along with the other applications. The various supported scenarios (mobile, home, office, group session) illustrate the importance of the upper-layer middleware flexibility and interoperability.

4 Meeting Patterns

The central purpose of middleware technology is to hide low-level services and implementation details related to data and network management into a virtualized set of services. Unfortunately, as it has been shown in the CSCW field, this virtualization is somewhat problematic [Begole, 1999, Buszko, 2001]. On the one hand, virtualization introduces what is known as transparency problem [Lauwers, 1990]: hidden lower-level constraints have side effects that are not detected or controlled by the level above. The consequence of the transparency problem is that applications become highly dependent on unknown behavior related to messaging protocols, network delays and partitions, synchronization issues, etc. On the other hand, collaborative applications are usually regarded as highly specialized and context dependent, making it difficult to generalize their behavior into a comprehensive set of virtualized services [Blair, 1993].

From our point of view, the "upper-layer" notion implies that middleware should be more proximate to the application than to the infrastructure layer, and thus some effort must be applied to focus on the applications. Furthermore, since we are restricting our context to the usage of PDA in meetings, we do not have to virtualize middleware services for the whole diversity of CSCW applications.

Based on the above principles, our strategy was to depart from the applications and develop a set of patterns reflecting the following key issues:

- The infrastructural setting, including the major hardware and software components relevant to the applications;
- How meeting participants operate the major hardware and software components with the purpose of accomplishing their goals;
- How the usage of this infrastructure evolves according to time, i.e. before, during and after meetings.

The patterns presented below were obtained by reflecting about our experience developing meeting applications (see also [Antunes, 1999, Antunes, 2006, Antunes, 2001]) and were structured according to the GammaForm pattern format [Gamma, 1995] and recommendations from [Evitts, 2000]**.**

4.1 Name: Deliberate meeting

Intent: The deliberate meeting is mostly related to group problem solving and decision-making. The principal purpose of the deliberate meeting is to apply structured and rational procedures to systematically reduce the distance to set goals.

Also known as: Work meeting, decision meeting, formal meeting.

Motivation: Organizations often must bring people together in order to analyze and discuss problems, identify and evaluate possible solutions and, ultimately, make decisions. Deliberate meetings are a very common approach to this problem.

Applicability: The particular structure of deliberate meetings is strongly related to the problem complexity and group stage. However, it is mostly applicable to situations demanding a rational approach, where the problem and decision process are complex, and potential failure has severe organizational impact.

Structure: Deliberate meetings require advance preparation, integrating various asynchronous activities accomplished prior to meetings, such as agenda preparation, document sharing or even preliminary discussion [Borges, 1999]. Often, this structured approach also includes post-meeting activities such as wrap-up or defining subsequent activities. A SR is necessary to maintain pre and post meeting information. Deliberate meetings also require a SDG, necessary to focus the participants on the decision process and often used to regulate their interventions.

The role of PDAs in deliberate meetings is most probably sporadic and restricted to few users. PDAs may be used to interact with the SR, updating information either synchronously during the meeting or asynchronously before and after the meeting. PDAs may also be used to manage the SDG, acting as remote commanders [Myers, 2000].

Participants: This type of meeting requires a leader/facilitator to prepare and conduct the decision process. The leader selects the participants prior to the meeting.

Examples: Meetings with LightMeet adopt the deliberate meeting pattern. After the initial steps, therapy meetings with JoinTS also adopt this pattern.

4.2 Name: Meeting ecosystem

Intent: The meeting ecosystem is associated to an ill-defined or unexpected reality. The most significant difference to the deliberate meeting is that advance planning is compromised. The central purpose of the meeting ecosystem is thus to mobilize a group towards the identification of the best strategy to achieve the intended goals (which may also be compromised [Rosenhead, 1989]).

Also known as: Strategy meeting, collaboratory meeting.

Motivation: In face of ill-defined problems, organizations must bring people with diverse knowledge together in order to creatively define and plan new solutions. Meeting ecosystems afford the required levels of information exchange, creativity and lateral thinking, multiple views and flexibility necessary to approach this type of problems.

Applicability: The flexible and dynamic structure of meeting ecosystem is strongly related to ill-defined problems.

Structure: The meeting ecosystem may be regarded as an aggregate of sub-meetings with different goals. From the outset, the structure resembles an organized chaos, where participants flexibly move across different sub-meetings while contributing with their expertise to resolve a wide variety of problems. This type of behavior has been observed in collaboratories [Mark, 2002].

The meeting ecosystem may utilize a SDG to support situation awareness. The participants may rely on PDAs to facilitate dealing with this organized chaos, e.g. setting up sub-groups, defining tasks, sub-tasks and to-do lists, and exchanging information between different contexts.

The SR may be required only in certain cases. For instance, the SR is necessary if the participants have to integrate information produced by the sub-groups, or preserve that information for usage in future meetings.

Participants: This type of meeting does not require a leader to prepare or conduct the whole process. Several facilitators may emerge during meetings to manage sub-processes.

Examples: Nomad supports an ecosystem of multiple simultaneous design meetings.

4.3 Name: Creative/design meeting

Intent: This type of meeting is associated to the collaborative generation of ideas or plans.

Also known as: Brainstorming meeting.

Motivation: Organizations are often required to creatively overcome existing business barriers, find innovative ideas or develop new designs and plans. Creative/design meetings support this important organizational process.

Applicability: The particular structure of creative/design meetings is only adequate to the "generate" quadrant defined by McGrath [McGrath, 1984], which includes generating ideas and generating plans in a collaborative context.

Structure: The most common structure supporting creativity and design relies on the several principles attributed to the brainstorming technique [Osborn, 1963]: free-wheeling is welcomed, quantity is wanted, criticism is avoided and combination and improvement are sought. Considering this fairly simple structure, PDAs may assume a very important role as input devices for text and sketches. Sketching, in particular, affords the visual symbols and spatial relationships necessary to express ideas in a rapid and efficient way during design activities [Forbus, 2001]. Also, PDAs afford meeting participants to explore the physical context along with the creativity/design process, thus enhancing creativity and productivity. For instance, a group of architects may work jointly on a sketch at a construction site [May, 2005].

This type of scenario does not require a SR because, in most situations, the obtained outcomes do not require further work by the group. The SDG is also dispensable, since parallel work should not only be possible but encouraged, to increase the group productivity.

Participants: This type of meeting usually does not distinguish roles among participants.

Examples: LightMeet supports text-based brainstorming. Nomad supports brainsketching.

4.4 Name: Ad-hoc meeting

Intent: There is one major intention behind ad-hoc meetings: information sharing.

Also known as: Unstructured meeting.

Motivation: Most meetings in organizations are ad-hoc: unscheduled, spontaneous, lacking an agenda, and with an opportunistic selection of participants [Romano, 2001]. In spite of an apparent informality, we may define two different motivations based on work relationships: the need to share important information among coworkers, which is related to a horizontal type of relationship; and the need to exert management control, which is associated to a vertical type of relationship.

Applicability: The ad-hoc meeting happens anywhere and anytime, whenever there is a need to share important information.

Structure: During an ad-hoc meeting, the participants are focused on information sharing, which may be centrally moderated. PDAs may support the social protocols necessary to moderate information sharing. PDA synchronization may be beneficial to offer the group an overall perception of the work carried out in the meeting.

This scenario also emphasizes the opportunities of PDAs to overcome several restrictions imposed by the environment, e.g. the lack of a whiteboard, table, paper, etc. Furthermore, PDAs may automatically obtain information about the meeting location and other PDAs in the vicinity, thus preserving the meeting context.

Once the ad-hoc meeting has ended, it has produced various types of outcomes, consisting of private and public data such as agreements, to-do lists, deadlines and schedules. A SR is key to preserve a coherent view of what happened in the meeting. A SDG could also be beneficial in this scenario, but its usage depends on local availability, which emphasizes the structural flexibility of ad-hoc meetings.

Participants: This type of meeting requires a leader in situations where the focus is on vertical information exchange. Facilitation is unnecessary, as both the meeting preparation and management are quite simple.

Examples: The initial therapy steps supported by JoinTS adopt this pattern. Nomad also supports ad-hoc meetings.

4.5 Name: Learning meeting

Intent: This type of meeting is focused on the group exploration and structuration of knowledge with the support and guidance from a knowledgeable person.

Also known as: Classroom meeting.

Motivation: The general motivation is to support teachers involving students in the learning process trough structured and focused activities, which may include problem solving, generation and organization of ideas, analysis, group decision-making, group writing and action planning.

Applicability: Learning meetings are very different from other types of meetings, as they are more focused on the use of the technology to enhance the learning process [Tyran, 2001].

Structure: Learning meetings emphasize the role of technology supporting the teachers' goals and strategies. In this respect, SDG may help focusing the students on the information conveyed by the teacher, while PDAs facilitate the set up and conduction of parallel activities. According to [Tyran, 2001], the degree of anonymity supported by PDAs in this scenario helps reducing evaluation apprehension by allowing group members to execute their activities without having to expose themselves in front of the group; and parallelism aids reducing domination, since more persons may express their ideas at the same time. A SR may be beneficial to support pre- and post-meeting activities by the teacher, such as advance preparation of contents and assessment.

Participants: Teachers and students. Teachers are responsible for setting up and conducting the learning process.

Examples: Nomad has been used to implement learning meetings, when configured to support the teachers' roles.

In Table 1 we present an overview of our meeting patterns. The occurrence of a meeting pattern is associated to a goal, problem or motivation set explicitly or implicitly by the meeting participants. The other three attributes (inputs, process and outputs) characterize the meeting patterns according to a common production model.

These attributes do not have a direct impact on the meeting system architecture and components, but definitely tie the architecture and components to the application context, especially in what concerns information management.

In Table 2 we present a complementary overview of the meeting patterns, now focusing in more detail on the architectural setting. Several topologies using three major meeting components (PDA, SDG and SR) have clearly emerged from these patterns. Furthermore, Table 2 highlights the major features of these components as well as the information managed by the participants to accomplish their goals. We have thus set the stage to specify the upper-layer middleware based on user goals and application requirements.

Meeting	**Major goals**	**Meeting Inputs**	**Meeting process**	**Meeting Outputs**
Deliberate	Problem solving and decision making	Attached documents, agenda, attendees list	Structured, conducted by the facilitator	Reports and other formal meeting elements
Meetings ecosystem	Unknown problem solving	Ill defined problem	Organized chaos	New solutions
Creative/design	Brainstorming and collaborative design	On-site material (e.g. building snapshot)	Unstructured with free collaboration	Ideas and sketches
Ad-hoc	Information sharing	Individual contributions	Simple	Individual notes
Learning	Collaborative activities set by teacher	Pedagogical materials (e.g. lectures)	Structured, conducted by the teacher	Pedagogical achievements

Table 1: Overview of meeting patterns

Meeting	**PDA**	**SDG**	**SR**
Deliberate	Manage meeting data, meeting process and SDG	Focus the participants' attention	Preserve pre- and post-meeting information
Meetings ecosystem	Move data across groups, manage groups	Situation awareness	Integrate information
Creative/design	Input device	None	None
Ad-hoc	Share notes and substitute SDG	None	Preserve outcomes
Learning	Input device	Focus the students' attention	Preserve pre- and post-meeting information

Table 2: Meeting patterns, major components and features

5 Upper-Layer Meeting Middleware

We will now describe the upper-layer middleware enabling the features and topologies suggested by the meeting patterns described above. The description is organized according to three increasing levels of detail: (1) middleware architecture; (2) components' internal structure; and (3) meeting management. References to the applications are given in order to illustrate or emphasize key design ideas.

Middleware architecture. The adopted middleware architecture is naturally based on the architectural setting summarized in Table 2, relying upon PDA, SDG and SR components. Our perspective is that these components should be interoperable but highly independent, considering that the meeting patterns use various topologies. For instance, creative/design meetings do not use the SDG and SR, while deliberate meetings require a SDG.

In our middleware architecture, every instance of PDA, SDG and SR components share a common "loose virtual bus" illustrated in Fig. 1. The semantics associated to the operation of the loose virtual bus is the following one: (1) no reliable information about the number or type of components currently connected to the bus; (2) no guarantees that messages are delivered to any specific components; and (3) no status information available. This type of semantics assumes the upper-layer perspective and is adequate to the flexible and dynamic use of PDA in meetings. Furthermore, by requiring that applications explicitly take care of availability and synchronization issues, it avoids the transparency problem previously stated. This semantics was adopted with success in our three applications.

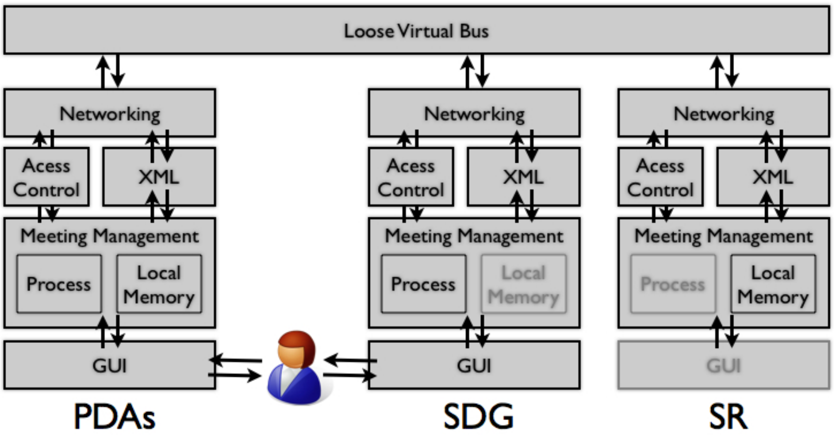

Figure 11: Upper-layer middleware: Architecture, components and components' internal structure

Components' internal structure. We now focus on the internal structure of the upper-layer middleware components. There are five classes (see Fig. 11). The first one to consider is the Networking class.

Although a complete discussion of the Networking class is out of the scope of this paper, we note it relies on wireless networks to automatically detect other components and to exchange messages with them. Internally, the Networking class may operate in Peer-To-Peer and Client-Server modes. For instance, the LightMeet prototype uses a Client-Server mode to synchronize the SDG with multiple PDAs. When a user interacts with a PDA, e.g. to write an idea, the PDA sends the information to the SDG, which then disseminates it to the other PDA. However, the PDA components are unaware of this specific implementation, as they simply send ideas to the Networking class. We use several variations of the Networking class in our prototypes, based on Java and .NET. LightMeet uses a Java Networking class, while Nomad and JoinTS use .NET variations.

The Graphical User Interface (GUI) is responsible for supporting the user interactions with the system. The user interacts with the GUI through pen-based gestures, mouse or keyboard. This class is not present in the SR. The Meeting Management class manages the whole meeting information and process (more details below). The XML class is responsible for coding meeting relevant information into a common format and exchanging that information with the Networking class. Finally, the Access Control class is responsible for applying application-level access control policies to components and information resources.

The operation of a component is as follows. The user manages meeting information through the GUI. Whenever an individual interaction requires collaboration support, the action itself is registered by the Meeting Management, encoded in XML and delivered to the Networking class. Of course, XML messages also arrive from other components through the Networking class, which are delivered to the XLM class and then passed to the Meeting Management. If the Meeting Management decides that the user should be aware of the consequences of the incoming message, that information is delivered to the GUI.

Meeting management. As hinted by the previous discussion, a lot of the upper-layer functionality resides in the Meeting Management class. The main purpose of this class is to support the user goals and application requirements by managing meeting relevant data.

This goal is relatively difficult because of the high degree of informality associated to meeting patterns. Too much formalization reduces the middleware flexibility, while the lack of formalization results in less support to organizational memory, contextualization, and decision-making. Nevertheless, based on our previous research on meeting structures [Antunes, 2006], we attempt a characterization departing from a separation of concerns in Meeting Memory and Meeting Process classes.

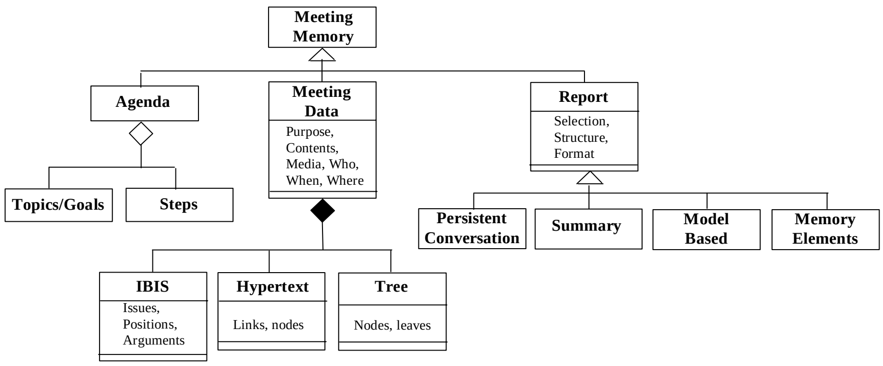

Figure 12: Memory elements

The Meeting Memory class organizes the data elements produced or manipulated in relation to meetings. For instance, the Meeting Memory class in LightMeet and Nomad consists of a repository for ideas, comments and voting results.

We identify three principal meeting memory elements (Fig. 12): agenda, meeting data, and meeting report [Costa, 2001]. The agenda is a critical element to successfully manage some meeting patterns, such as the deliberate meeting, since meetings tend to crystallize their actions around it [Niederman, 1996]. The agenda is mandatory in LightMeet. The prototypical agenda has two different types of information: a list of topics or goals that the group must deal with; or a series of steps that the group should execute to accomplish their goals.

The meeting data concerns all the raw data distilled during meetings. The major attributes associated to the meeting data are [Antunes, 2006]: purpose; contents; media used; who is involved in producing the data item; when was the data item produced and where should the data item be produced or used. We adopted three existing information models to structure meeting data, which cover various levels of complexity: the simple tree structure, the more complex hypertext model, and the even more complex IBIS model [Kunz, 1970]. Both Nomad and LightMeet utilize trees, while JointTS uses the hypertext model.

The meeting report aggregates the tangible outcomes of meetings and is characterized by the particular selection, structure and format of the report items. We may identify four different report types associated to meeting systems:

- Persistent conversation – Transcripts of all the information exchanged in meetings, most often automatically produced by the application. It is adopted in JoinTS.
- Summary of the outcomes – Referencing only the pieces of information considered most important, like voting results. Humans in general generate these summaries. It is adopted in LightMeet.
- Model based information structures – When the underlying information model is applied to automatically produce meeting reports. It is used in JoinTS.
- A collection of group memory elements generated and structured during meetings – This includes selected elements like action plans and calendaring information commonly produced in meetings. It is used in Nomad.

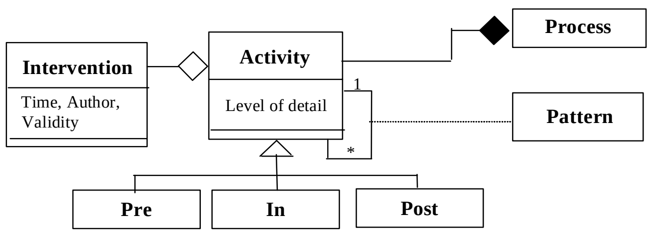

Figure 13: Process elements

The Meeting Process class organizes descriptive information about the activities executed by the meeting participants (Fig. 13). The nature of these activities changes as the participants, along time, move forward towards their goals. Therefore, we typify activities in accordance to that progression and consider three increasing levels of detail:

- Level 1 – The meeting as a whole, i.e. one single activity;
- Level 2 – The partition of the meeting process as a sequence of activities; and the decomposition of these activities in sub-activities;

- Level 3 – The fragmentation of the meeting in an intricate collection of elementary individual interventions.

The level 1 describes the meeting process while maintaining the perspective of the whole. This allows typifying meetings according to simple patterns. For instance, Nomad adopts a level 1 approach to support creative/design meetings and thus has a Meeting Process class supporting one single task: sketching.

The second level addresses the decision structure. This follows a logical view over decision making that is recurrent in literature [Ho, 1999], where the goal is divided in partial goals that can be accomplished in a systematic way. Each goal may be decomposable in multiple levels of detail, with goals and sub-goals, and in three different stages: (1) pre-meeting, considering activities that have to be executed before the meeting; (2) in-meeting, with activities accomplished during the meeting; and (3) post-meeting, considering activities that may be required afterwards. LightMeet adopted a level 2 approach with pre, in and post meeting stages.

Finally, in level 3 the meeting process is characterized according to the flows of individual interventions produced by the participants. The generic attributes associated to these interventions are:

- Time – The moment when the intervention is produced. Based on this attribute, we can characterize meetings as synchronous or asynchronous.
- Author – The person that produces an intervention may be identified or not. This factor can have an important role in the process results. Several researchers have reported the positive effects of anonymity in the interaction process (e.g. [Connolly, 1990]).
- Validity –The validity corresponds to the time during which the intervention can be accessible. The validity has repercussions on the organizational memory.

6 Discussion

In Table 3 we summarize the major relationships between the developed prototypes and the upper-layer meeting middleware proposed in this paper. This table highlights the breadth of cases, designs and tests that were accomplished in this research, which could only be possible having several teams involved in the process. Many insights, observations and lessons naturally emerged from this work, and most of them were built into the final middleware specification reported in this paper. Nonetheless, we would like to bring forward the following noteworthy points for discussion:

- **The PDA, SDG and SR are common denominators**. Although they may be absent from some applications or used in different ways, according to the meeting patterns and also to the applications, the importance of these components seems very significant. Future meeting technology should consider supporting these three components in an independent, flexible, tailorable and interoperable way.
- **XML communication is also a common denominator**. This observation per se may not seem very significant, because XML is nowadays a key technology to support interoperability and reusability in many different domains. However, it is important to emphasize that current meeting systems

do not support interoperability, mostly because, besides XML support, a common information structure must also be supported, which was previously missing. This paper proposes an information structure that is at the same time reusable and detailed, allowing future standards in this area to emerge.

- **The information structures seem highly compatible**. The agenda, meeting report and hierarchical objects (pages, comments, sketches, voting results) consistently appeared in our prototypes. From our point of view, this supports our suggested memory and process classes, as common meeting denominators, thus resolving the problem of a lacking common information structure necessary to support interoperability.
- **All proposed patterns were exercised by the developed prototypes**. The deliberate, ad-hoc, meeting ecosystem, creative/design and learning patterns showed to be generally consistent with the multiple uses of the several applications under discussion. Although this list of patterns may not be complete, at least it seems to be useful and representative.

	NOMAD	**LIGHTMEET**	**JOINTS**
Components	PDA	PDA, SR, SDG	PDA, SR, SDG (group meetings)
Communication	XML	XML	XML
Meeting data	Agenda Hierarchy of pages	Agenda Hierarchy of comments Report	Agenda Therapy sheets Questionnaires
Process	Sequence of meetings	Sequence of meetings	Sequence of meetings Collection of elementary events
Patterns	Creative/design meetings Ad-hoc meetings Learning meetings	Deliberate meetings	Deliberate meetings Meeting ecosystem (with several therapists)

Table 3: Upper-layer middleware perspective over the developed prototypes

7 Conclusions

In this paper we provide an integrated perspective over meeting systems and how users may utilize PDAs in meeting environments supported by an upper-layer middleware. Regarding the architecture of the upper-layer middleware, we identify three major components: PDA, SDG and SR.

Regarding the individual components, we organized their functionality in five classes: Networking, Access Control, XML, Meeting Management and GUI. The Meeting Management class is further characterized with several additional classes related to the meeting memory and the meeting process.

The principal implications drawn from this research result from the opportunity to make meeting systems interoperable to a level that has not been achieved before, allowing different devices (in our case, PDA, SDG and SR, but the list can be extended in the future) to exchange, share and manipulate meeting-related information in an integrated way. Such level of interoperability is possible because of two fundamental reasons: (1) we support information exchange using the XML standard, thus allowing very different devices to plug in the meeting system; and (2) we standardized the meeting management around common memory and process elements, such as agenda, report, and other meeting data structures, like trees.

Furthermore, the upper-layer meeting middleware is anchored on a collection of patterns covering several types of meetings commonly used by organizations to disseminate information, coordinate activities, generate ideas, make decisions and learn.

This research emerged from three independent research groups, working in different fields and developing their own prototypes, but nevertheless sharing a common understanding about the nature of meetings, the important roles that PDAs may assume in meetings and the requirements to make interoperable meeting systems. We strongly believe the proposed architecture benefits from such varied and complementary perspectives, as well as parallel development efforts.

Acknowledgements

This paper was partially supported by Fondecyt 1050601, DI-Universidad de Chile Nro. I2 04/01-2, and the Portuguese Foundation for Science and technology, Projects POSI/EIA/62473/2004 and CONC-REEQ/643/2001.

References

[AMI, 2006] http://www.ami-communities.eu/

[Antunes, 1999] Antunes, P.: "On the Design of Group Decision Processes for Electronic Meeting Rooms"; CLEI Electronic Journal, 2(1), 1999.

[Antunes, 2006] Antunes, P., Costa, C., Pino, J.: "The Use of Genre Analysis in the Design of Electronic Meeting Systems"; Information Research, 11(3), 2006.

[Antunes, 2001] Antunes, P., Ho, T.: "The Design of a GDSS Meeting Preparation Tool"; Group Decision and Negotiation, 10(1), 5-25, 2001.

[Baldonado, 2000] Baldonado, M., Cousins, S., Gwizdka, J., Paepcke, A.: "Notable: At the Intersection of Annotations and Handheld Technology"; Lecture Notes in Computer Science, v. 1927. Heidelberg: Springer-Verlag, 100-113, 2000.

[Begole, 1999] Begole, J., Rosson, M., Shaffer, C.: "Flexible Collaboration Transparency: Supporting Worker Independence in Replicated Application-Sharing Systems"; ACM Transactions on Computer-Human Interaction, 6(2), 95-132, 1999.

[Blair, 1993] Blair, G., Rodden, T.: "The Challenges of CSCW for Open Distributed Processing"; Proceedings of the IFIP TC6/WG6.1 International Conference on Open Distributed Processing II, Berlin, Germany, 127-140, 1993.

[Bloom, 1980] Bloom, B.: "All Our Children Learning". New York: McGraw-Hill, 1980.

[Briggs, 2003] Briggs, R., Vreede, G., Nunamaker, J.: "Collaboration Engineering with Thinklets to Pursue Sustained Success with Group Support Systems"; Journal of Management Information Systems, 19(4), 31-64, 2003.

[Buszko, 2001] Buszko, D., Lee, W., Helal, A.: "Decentralized Ad-Hoc Groupware API and Framework for Mobile Collaboration"; Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, Boulder, Colorado, USA, 5-14, 2001.

[Collaboration @ Work, 2004] Collaboration @ Work: "Next Generation Collaborative Working Environments 2005-2010"; European Commission, Information Society Directorate-General, Brussels, 2004.

[Connolly, 1990] Connolly, T., Jessup, L., Valacich, J.: "Effects of Anonymity and Evaluative Tone on Idea Generation in Computer-Mediated Groups"; Management Science, 36(6), 689-703, 1990.

[Costa, 2001] Costa, C., Antunes, P., Dias, J.: "EMS/PDA: Connecting Meetings with People in Organisations"; Proceedings of the 24th Information Systems Research Seminar in Scandinavia, IRIS 24, Ulvik in Hardanger, Norway, 2001.

[Davis, 1999] Davis, R., Landay, J., Chen, V., Huang, J., Lee, R., Li, F., Lin, J., Morrey III, C., Schleimer, B., Price, M., Schilit, B.: "Notepals: Lightweight Note Sharing by the Group, for the Group"; Proceedings of the CHI 99 Conference on Human Factors in Computing Systems, Pittsburg, 338-345, 1999.

[Davis, 1998] Davis, R., Lin, J., Brotherton, J., Landay, J., Price, M., Schilit, B.: "A Framework for Sharing Handwritten Notes"; Proceedings of the 11th annual ACM symposium on User interface software and technology, San Francisco, California, 119-120, 1998.

[Dev, 1997] Dev, P.: "Intrinsic Motivation and Academic Achievement: What Does Their Relationship Imply for the Classroom Teacher?" Remedial and Special Education, 18(1), 12-19, 1997.

[Evitts, 2000] Evitts, P.: "A Uml Pattern Language". Indianapolis, IN: Macmillan, 2000.

[Fagrell, 2000] Fagrell, H., Forsberg, K., Sanneblad, J.: "Fieldwise: A Mobile Knowledge Management Architecture"; Proceeding of the ACM 2000 Conference on Computer supported cooperative work, Philadelphia, 211-220, 2000.

[Forbus, 2001] Forbus, K., Ferguson, R., Usher, J.: "Towards a Computational Model of Sketching"; IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces, Santa Fe, New Mexico, 77-83, 2001.

[Gamma, 1995] Gamma, E., Helm, R., Johnson, R., Vlissides, J., Booch, G.: "Design Patterns: Elements of Reusable Object-Oriented Software". Reading, MA: Addison Wesley, 1995.

[Greenberg, 1999] Greenberg, S., Boyle, M., Laberge, J.: "PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal"; Personal Technologies, 3(1-2), 54-64, 1999.

[Ho, 1999] Ho, T., Antunes, P.: "Developing a Tool to Assist Electronic Facilitation of Decision-Making Groups"; Fifth International Workshop on Groupware, CRIWG '99, Cancun, Mexico, 243-252, 1999.

[Kunz, 1970] Kunz, W., Rittel, H.: "Issues as Elements of Information Systems"; Institute of Urban and Regional Development, University of California at Berkeley, 1970.

[Laso-Ballesteros, 2005] Laso-Ballesteros, I., Salmelin, B.: "Ami-Endowed Collaboration@Work"; Ambient Intelligence: The Evolution of Technology, Communication and Cognition Towards the Future of Human-Computer Interaction, v., G. Riva, F. Vatalaro, F. Davide, and M. Alcaniz, Eds. Amsterdam, The Netherlands: IOS Press, 237-265, 2005.

[Lauwers, 1990] Lauwers, J., Lantz, K.: "Collaboration Awareness in Support of Collaboration Transparency: Requirements for the Next Generation of Shared Window Systems"; Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, Seattle, Washington, United States, 303-311, 1990.

[Mark, 2002] Mark, G.: "Extreme Collaboration"; Communications of the ACM, 45(6), 89-93, 2002.

[May, 2005] May, A., Mitchell, V.: "Opportunities and Challenges for Location Aware Computing in the Construction Industry"; Proceedings of Mobile HCI, 255-258, 2005.

[McGrath, 1984] McGrath, J.: "Groups: Interaction and Performance". Englewood Cliffs, NJ: Prentice-Hall, 1984.

[Muñoz, 2003] Muñoz, M., Rodriguez, M., Favela, J., Martinez-Garcia, A., González, V.: "Context-Aware Mobile Communication in Hospitals"; Computer, 36(9), 38-46, 2003.

[Myers, 2000] Myers, B.: "The Pebbles Project: Using PCs and Hand-Held Computers Together; Demonstration Extended Abstract"; Adjunct Proceedings CHI'2000: Human Factors in Computing Systems, The Hague, The Netherlands, 14-15, 2000.

[Myers, 1998] Myers, B., Stiel, H., Gargiulo, R.: "Collaboration Using Multiple PDAs Connected to a PC"; Proceedings of the ACM Conference on Computer Supported Cooperative Work, Seattle, WA, 285-294, 1998.

[Neyem, 2006] Neyem, A., Ochoa, S., Pino, J.: "Supporting Mobile Collaboration with Service-Oriented Mobile Units"; Groupware: Design, Implementation and Use, 12th International Workshop, CRIWG 2006, Lecture Notes in Computer Science, v. 4154, Y. Dimitriadis, I. Zigurs, and E. Gómez-Sánchez, Eds. Medina del Campo, Spain: Springer-Verlag, 228-245, 2006.

[Niederman, 1996] Niederman, F., Volkema, R.: "Influence of Agenda Creation and Use on Meeting Activities and Outcomes: Report on Initial Results"; Proceedings of the 1996 Conference on ACM SIGCPR/SIGMIS Conference, Denver, Colorado, 192-205, 1996.

[Nunamaker, 1997] Nunamaker, J., Briggs, R., Mittleman, D., Vogel, D., Balthazard, P.: "Lessons from a Dozen Years of Group Support Systems Research: A Discussion of Lab and Field Findings"; Journal of Management Information Systems, 13(3), 163-207, 1997.

[Osborn, 1963] Osborn, A.: "Applied Imagination". New York: Charles Scribner's Sons, 1963.

[Pereira, 2006] Pereira, L.: "Electronic Meetings with PDA"; Department of Informatics, Faculty of Sciences of the University of Lisboa, Masters Thesis (in Portuguese), 2006

[Roman, 2005] Roman, M., Islam, N., Shoaib, S.: "A Wireless Web for Creating and Sharing Personal Content through Headsets"; Pervasive Computing, April-June, 67-73, 2005.

[Romano, 2001] Romano, N., Nunamaker, J.: "Meeting Analysis: Findings from Research and Practice"; Proceedings of the 34th Hawaii International Conference on Systems Science, Hawaii, 2001.

[Rosenhead, 1989] Rosenhead, J.: "Rational Analysis for a Problematic World". Chichester, England: Jonh Wiley & Sons, 1989.

[Sá, 2007] Sá, M., Carriço, L., Antunes, P.: "Ubiquitous Psychotherapy"; IEEE Pervasive Computing, 6(1), 20-27, 2007.

[Santanen, 2006] Santanen, E., Kolfschoten, G., Golla, K.: "The Collaboration Engineering Maturity Model"; Proceedings of the 39th Hawaii International Conference on System Sciences, Hawaii, 2006.

[Stewart, 1999] Stewart, J., Bederson, B., Druin, A.: "Single Display Groupware: A Model for Co-Present Collaboration"; Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, Pittsburgh, Pennsylvania, 286-293, 1999.

[Strong, 1995] Strong, R., Silver, H., Robinson, A.: "What Do Student Want?" Educational Leadership, 53(1), 8-12, 1995.

[Tyran, 2001] Tyran, G., Sherpherd, M.: "Collaborative Technology in the Classrom: A Review of the GSS Research and a Researh Framework"; Information Technology and Management, 2(4), 395-418, 2001.

[Vreede, 2006] Vreede, G., Kolfschoten, G., Briggs, B.: "Thinklets: A Collaboration Engineering Pattern"; International Journal of Computer Applications and Technology, 25(2/3), 140-154, 2006.

[Wiberg, 2001] Wiberg, M.: "Roamware: An Integrated Architecture for Seamless Interaction in between Mobile Meetings"; Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, Boulder, Colorado, 288-297, 2001.