

An Application Framework for Developing Collaborative Handheld Decision-Making Tools

Pedro Antunes (corresponding author)

Victoria University of Wellington,

School of Information Management,

PO Box 600, Wellington, 6140, New Zealand

Phone +64-4-4635525, Fax +64-4-4635446

pedro.antunes@vuw.ac.nz

Gustavo Zurita

University of Chile,

Department of Information System and Management of the Economy and Business

School, Diagonal Paraguay 257, Santiago, Chile

gnzurita@facea.uchile.cl

Nelson Baloian

University of Chile,

Department of Computer Science of the Engineering School,

Blanco Encalada 2120, Santiago, Chile

nbaloian@dcc.uchile.cl

Abstract. This paper describes an application framework supporting Collaborative Handheld Decision-Making (CHDM). The main characteristics of the framework are: (1) extensive usage of visual elements and gestures; and (2) independence from specific decision-making methods, processes and tasks. The research departed from the analysis and systematization of several CHDM scenarios, highlighting repeatable behavior across multiple decision-making contexts. From these scenarios we distilled a coherent set of common functional requirements organized in three main categories: process, macro and micro functionality. The proposed framework has been validated at length through the development of several CHDM tools. Six different tools are described in the paper. The main contribution of this work is a common foundation for developing CHDM tools.

Keywords: Collaborative Handheld Decision-Making; Group Support Systems; Mobile Devices.

1. Introduction

Collaborative Handheld Decision-Making (CHDM) concerns the use of handheld tools to support collaborative and decision-making activities in varied application areas such as managing clinical patient information in hospital environments (Lapinsky, 2006; Yu, Houston, Ray, Garner, & Berner, 2007), supporting learning activities in the classroom (Zurita, Antunes, Baloian, & Baytelman, 2007) and controlling production processes in the field (Bange, Deutscher, Larsen, Linsley, & Whiteside, 2004). The main

contributions of CHDM include increased convenience, extended information access, improved knowledge management, up-to-date information distribution, and collaboration independently of time and space (Srivastava, 2005).

This technology has already achieved a noteworthy penetration in complex work environments. For instance, Miller et al (2004) found out that 26.2% physicians in USA were already using handheld tools; and 66.8% were expecting to rely more on this technology in the future. The decreasing costs, improved dependability and increasing user-interface friendliness will certainly contribute to boost the acceptance of CHDM tools.

CHDM presents a wide scope of concerns and perspectives, bringing together researchers from diverse fields like decision sciences, organizational sciences, cognitive sciences, small groups research, computer supported collaborative work and ubiquitous computing (Bragge, Merisalo-Rantanen, Nurmi, & Tanner, 2006). Most of the research in this area is grounded in the pioneering work of Davies and colleagues (Davies, Blair, Cheverst, & Firday, 1994; Davies, Mitchell, Cheverst, & Blair, 1998). Their initial studies explored the challenges associated with building the technological infrastructure necessary to simultaneously support mobility and collaboration, emphasizing technical issues such as connectivity, distribution and interoperability. This area of concern remains highly relevant today, although the requirements have become more challenging regarding networks' reliability, users' reachability, event propagation and information pushing (Messeguer, et al., 2009; Sousa, Preguiça, & Baquero, 2009).

Several studies have been trying to understand the major problems associated with mobility and collaboration in specific applications. These studies contribute to

understand the main intervening factors in various CHDM contexts. For instance, CHDM in geocollaboration is highly correlated with geographical relationships (Antunes & André, 2006; Cai, et al., 2005; MacEachren, 2005; MacEachren & Brewer, 2004; Nyerges, Montejano, Oshiro, & Dadswell, 1997), situation modeling is affected by feedback (Škraba, Kljajić, & Borštnar, 2007), strategic visualization is highly influenced by the human-machine interface (Monzani, Bendahan, & Pigneur, 2004), and emergency management depends on situation awareness (Bergstrand & Landgren, 2009; Sapateiro & Antunes 2009).

But the focus on specific applications has lead research towards a fragmented body of knowledge. The main consequence is the lack of a comprehensive framework identifying and bringing together CHDM features across multiple applications. This paper describes our attempt at building and consolidating a comprehensive CHDM framework, while offering a common foundation for developing CHDM tools through reusability and composition. This framework is positioned above the infrastructure level and below the application level, thus targeting a middle layer that has been somewhat neglected by research (Laso-Ballesteros & Salmelin, 2005).

The adopted research methodology is founded on “design as a valid and valuable research methodology because the engineering research culture places explicit value on incrementally effective applicable problem solutions” (Peffer, Tuunanen, Trothenberger, & Chatterjee, 2007). The underpinning of the framework started from the systematic study of a collection of representative scenarios of use: real-world situations where handheld tools have been used to support collaboration and decision-making. The study revealed a set of functional requirements covering CHDM across the

selected scenarios of use. From these functional requirements, we developed a set of reusable software components running in the Windows Mobile platform. These components have been used to implement a number of CHDM tools. Six such tools are described in the paper: NOMAD, MCSKETCHER, SENSEMAKING, MCSUPPORTER, MCKC and MCSHELL. The proposed framework was therefore validated through its systematic use in application development.

The remaining sections of this paper are organized in the following way. The next section describes the scenarios that motivated the application framework. Section three derives a set of functional requirements from the discussed scenarios. Section four describes the application framework. Section five describes the tools that have been developed to validate the framework. Section six reviews the related work. Finally, in section seven we discuss the major outcomes of this research.

2. Considered Scenarios

Application frameworks serve to build applications in consistent, efficient and practical ways. But they may only accomplish these goals if they are deeply entrenched in the real-world context of use. They must therefore supply the conceptual elements necessary to identify and resolve repeatable design problems as well as a set of system components that may be pulled together in varied contexts for different purposes, easing the trajectory from conception to implementation. To accomplish these fundamental goals, we rooted our framework on a collection of scenarios that have for long been our focus of research and development. The following paragraphs briefly describe the

scenarios (see (Zurita, Antunes, Baloian, Carriço, & Baytelman, 2008) for complementary information).

Deliberation: This scenario is mostly related with rational decision making. The fundamental purpose of deliberation is to reach a decision by systematically going through a set of stages (Antunes, Zurita, Baloian, & Sapateiro, 2013). One remarkable example is the intelligence-design-choice approach described by Simon (1977).

Deliberation requires a considerable preoccupation with the decision process, because failure may have severe organizational impact and the problems and solutions tend to be complex. Deliberations often endure a lifecycle with several consecutive decision steps. The process participants are carefully selected and previously briefed about the process and intended goals. The role of the leader/facilitator (Kolfshoten, Hengst-Bruggeling, & Vreede, 2006) is fundamental to focus the participants on the process while avoiding typical negative factors such as repetitions, digressions and destructive conflicts.

Regarding the process structure, we observe that deliberation integrates asynchronous activities such as agenda preparation, preliminary discussion and decision wrap-up, with synchronous activities typically held in face-to-face meetings. Considering the need to orchestrate all these activities, two important requirements emerge from the deliberation scenario: (1) orchestrating the process steps necessary to reach a deliberation; and (2) sharing the problems, agenda topics, discussions, opinions and decisions produced throughout the various process steps. Specifically regarding the role of handheld tools in this context, we observe their main focus should be supporting

synchronous and asynchronous access to task-related outcomes. As described above, the process is strongly dependent on the leader/facilitator, which diminishes the possible contributions of handheld tools.

Consultation: The consultation scenario is associated with an ill-defined or unexpected reality. The most significant differences to deliberation are that the expected outcomes may be unclear and the decision process may depend on various contingencies. The fundamental purpose of a consultation is mobilizing various stakeholders having diverse information and competencies towards the identification of the best strategy to tackle a problem.

The consultation scenario may be regarded as an aggregate of several working groups, each one having different goals and participants. From the outset, it may resemble an organized chaos, where the participants flexibly move across working groups while contributing with their expertise to resolve a wide variety of problems. This behavior has for instance been observed in collaboratories (Mark, 2002). Regarding the role of handheld tools in this process, we highlight the critical support to group management. Handheld tools may help the participants dealing with chaos: setting up working groups, defining tasks, sub-tasks and to-do lists, and integrating task-related outcomes from different working groups, thus supporting context and situation awareness (Sapateiro & Antunes 2009).

Aligning: Information sharing is the major requirement characterizing this scenario. Information sharing is essential to maintain organizational structures: supervisors need

to communicate with their subordinates to define goals and activities, and obtain the feedback necessary to adjust plans and schedules; and coworkers also must share information to adjust task dependencies and work flows. The aligning scenario may thus be decentralized (co-workers) or centrally moderated (supervisors). The role of handheld tools in this scenario is to overcome physical constraints to communication. One example is allowing the participants tracking shared information while moving around the physical workspace (Antunes & André, 2006).

Ritualization: The ritualization process is mostly focused on building social ties (The 3M Meeting Management Team, 1994). The reasons underlying this process are varied and include, for instance, conveying organizational culture, norms, practices and rituals, as well as fostering teambuilding (Webne-Behrman, 1998). The process is usually simple and critically centered on the leader/facilitator, who is responsible for conveying cultural signals, stimulating assimilation and developing team culture. The outcomes of ritualization are mostly intangible, e.g. motivation, satisfaction and sense of belonging. Therefore, information management requirements are quite low. The role of handheld tools in this scenario is most probably sporadic, centered on information diffusion and restricted to few participants.

Ideation: The fundamental goal of ideation is supporting a divergent decision-making process where the participants seek to increase the amount of information available about a problem (Reinig & Briggs, 2006). In general, most ideation techniques rely on the brainstorming principles defined by Osborn (1963): free-wheeling is welcomed,

quantity is wanted, criticism is avoided and combination and improvement are sought. The critical role supported by handheld tools in this scenario is stimulating parallel production while giving awareness about the others' contributions.

Exploration: The exploration process is characterized by the need to develop novel solutions to demanding problems. This usually requires building group stories (Kankainen, Vaajakallio, Kantola, & Mattelmäki, 2012) and structuring information in creative ways (Shneiderman, 2002). Some particular characteristics of handheld tools may significantly contribute to this process. For instance, mobility and unobtrusiveness both contribute to explore ideas within the physical context. Furthermore, these tools support sketching, visual symbols and spatial relationships, thus contributing to explore solutions rapidly and efficiently (Forbus, Ferguson, & Usher, 2001). As some other scenarios, this one brings forward the requirement to integrate the public and private dimensions of collaborative work: in many circumstances some work pieces have to be initiated, explored and detailed in privacy by the author, to be made public only when sufficiently matured.

Learning: This scenario is focused on knowledge structuring and reflection with support and guidance from a knowledgeable person. It emphasizes the role of technology supporting the learning goals and strategies. In this respect, handheld tools may help guiding the participants throughout learning activities and orchestrating divergent and convergent working modes, which are often found in learning strategies. According to Tyran and Sherpherd (2001), the degree of anonymity supported by these

tools helps reducing evaluation apprehension by allowing group members to contribute without exposing themselves; and parallelism may help reducing domination effects, since more persons may express their ideas at the same time. In Table 1 we summarize the main characteristics of the considered scenarios.

Table 1. Summary table of CHDM scenarios.

Scenarios	Main goal	Main process structure	Possible roles of handheld tools
Deliberation	Rational decision making	Going through stages	Managing decision steps and sharing task-related outcomes
Consultation	Mobilizing stakeholders	Multiple groups of decision makers	Group management
Aligning	Information sharing	Centrally moderated	Remote communication
Ritualization	Building social ties	Centered on leader/facilitator	Information diffusion
Ideation	Increasing the amount of available information	Brainstorming	Parallel production
Exploration	Develop novel solutions	Building group stories	Sketching
Learning	Knowledge structuring and reflection	Guidance from a knowledgeable person	Guidance and setting working modes

3. Functional Requirements

3.1. Taxonomy adopted to classify functional requirements

The scenarios described in the previous section constitute a collection of insightful “stories” about decision-making processes and their goals, contexts, roles, and potential functionality brought by handheld tools. From this collection we have to construct a

coherent list of requirements, stressing common functionality and repeatable behavior across all scenarios. To accomplish this goal, we need a taxonomy that may be consistently applied. Several taxonomies identifying the decision-making elements relevant to our discussion have been proposed in the research literature. One of the earliest and mostly cited ones is the task-process taxonomy (Fjermestad & Hiltz, 1999; Nunamaker, Dennis, Valacich, Vogel, & George, 1991), which differentiates between task structure, focused on the specific group conditions in focal situations such as brainstorming, and process structure, addressing the more general conditions under which the group organizes activities and accomplishes goals.

Other taxonomies highlight the distinctions between hardware, software and people (Kraemer & King, 1988), coordination modes (Malone & Crowston, 1994), collaborative services (Bafoutsou & Mentzas, 2002), facilitation support (Antunes & Ho, 2001) and other even more specific requirements. In our research we adopted the general purpose of the task-process taxonomy, however separating the task dimension in two categories:

- Task dimension
 - Macro level – Regards the task from the perspective of the group, i.e. the conditions under which the participants collaborate to accomplish their goals.
 - Micro level – Regards the task from the perspective of the individual participants, addressing the conditions under which they contribute to the task.
- Process dimension

- Adopts a broad perspective over the decision-making process, including the assumption that a collection of tasks may have to be managed to improve the group's performance.

Using this taxonomy, we analyzed our collection of scenarios to come up with a list of the most relevant requirements. In Table 2 we summarize the several requirements that were captured this way. These requirements are discussed in more detail in the following section.

Table 2. Main requirements extracted from the scenarios. Details are discussed in section 3.2.

Scenario	Process functionality	Macro functionality	Micro functionality
Deliberation	<ul style="list-style-type: none"> ② Managing process steps ③ Managing synchronous and asynchronous modes 	⑥ Sharing task-related outcomes	<ul style="list-style-type: none"> ⑨ Sketching, writing and editing ⑩ Listing, relating and structuring
Consultation	① Managing working groups	⑦ Moving between working groups and process steps	
Aligning	④ Managing centralized and decentralized modes		
Ritualization		⑧ Managing private and public information	
Ideation			
Exploration			
Learning	⑤ Managing divergent and convergent modes		

3.2. *Elicited requirements*

We start with the process functionality. The first requirement we identify is **managing working groups (❶)**. This functionality is necessary to distribute the participants through multiple concurrent activities, which were posited by the consultation scenario. The facilitator should be the person responsible for creating groups and assigning participants to groups.

Another requirement is **managing process steps (❷)**. The purpose is extending the activity decomposition introduced by the previous requirement with a more fine-grained specification of the assigned activities. Typically, such decomposition is either task-oriented or goal-oriented. In the former case, the decision process might be structured as a list of consecutive tasks. This is the type of decomposition suggested by the deliberation scenario. The later case concerns a list of issues or topics that should be addressed by the group. The learning scenario suggests this type of decomposition. The type of decomposition more adequate to a specific case is left to the facilitator to decide.

Process steps may be conceptually defined as a list where each element corresponds to a particular problem, task or goal, although establishing loose relationships between them. This looseness allows adapting the process to the specific contingencies and complexity of the problem at hand. It should be noted that we currently restrict a working group to have one single list of process steps.

Another process functionality emerging from the scenarios concerns **managing synchronous and asynchronous modes (❸)**. The synchronous mode engages a group in same-time collaboration, while the asynchronous mode allows the group members to determine individually when to collaborate. For example, the aligning case suggests a

synchronous mode, while the consultation scenario is clearly implying the adoption of an asynchronous mode.

The synchronous/asynchronous modes may be defined for each process step. At any moment, the facilitator may select a process step and specify that the collaboration mode will be synchronous or asynchronous. In the former case, all group members will immediately share the information associated with the process step, while in the later case information sharing is deferred until a member makes an explicit request. This type of functionality supports the active role of the facilitator in determining which collaboration modes are more appropriate to specific tasks and goals.

Managing centralized and decentralized modes (④) complements the synchronous/ asynchronous modes with the capacity to specify who may actively manage the task outcomes. In the centralized case, only the facilitator can modify the information associated with a process step, while the remaining participants act as passive information consumers. The aligning scenario mandates this type of functionality.

In the decentralized case, all assigned participants may modify the information associated with a process step. The ideation and exploration scenarios, which usually are based on parallel production, are a good fit for this mode.

The **managing divergent and convergent modes (⑤)** functionality complements the centralized/ decentralized modes with the capacity to control the participants' focus of attention. In the convergent case, only one process step is made available to the participants. This mode is adequate for collaborative situations where the group should be kept strictly focused on the same task or topic, e.g., the deliberation scenario.

In the divergent mode, all process steps are simultaneously available to the participants. This mode allows the participants to wander through the steps independently from each other, working in parallel and developing different strategies to contribute to the group, as suggested by the consultation and exploration scenarios.

Regarding the task-macro functionality, the first requirement we consider is **sharing task-related outcomes (6)**. Each process step is expected to produce various data. For instance, typical deliberation scenarios lead to the creation of agenda items, problems, ideas, solutions, actions and priorities. Independently of the working modes that may be considered (synchronous/ asynchronous, centralized/ decentralized, divergent/ convergent), the group members must have a shared view over these data elements. We thus define that each process step must have an associated shared workspace. The shared workspace inherits all the properties of the process step (working group and working modes) and confers these properties to the resident data elements.

The main purpose of the **moving between working groups and process steps (7)** functionality is allowing group members to switch their focus of attention to different working groups and process steps. This may only be possible if both the departing and arriving process steps have divergent modes assigned by the facilitator. The effect is switching the participant's current workspace.

The purpose of **managing private and public information (8)** is covering a personal dimension of work that has not been considered so far: often activities require managing information with discretion before exposing them to the group. In many other cases it is more productive to avoid annoying the group with frequent data modifications. These situations may clearly occur in the exploration and ritualization

scenarios. We may thus define that each process step, besides having a shared workspace, may also have one private workspace for each working group member.

We now move on to the task-micro functionality. The first requirement to consider is **sketching, writing and editing (9)**. Sketching is particularly important because of the considerable opportunities it gives to organize task-based outcomes. For instance, sketches may serve to visually integrate ideas and comments into high-level concepts, especially in exploration scenarios. In a broad perspective, sketching may be used to establish visual relationships between the information present in a workspace. Editing accomplishes a set of standard actions (copy, paste, insert, edit, delete, etc.) that may be executed over the data elements present in a workspace. We consider four data types: sketches, cursive text, typed text, and pictures. Cursive text is a kind of sketch that is translated into text by humans.

The **listing, relating and structuring (10)** requirement expands the notion of workspace previously defined. Listing highlights the need to implement a very common type of task-related outcomes: the list. Lists are highly pervasive in decision-making, appearing recurrently in the form of to-do lists, wrap-ups, goals, solutions, votes, priorities, etc. For instance, the ideation scenario fundamentally aims to build a list of ideas. The relating and structuring features support complex task-related outcomes such as action plans and decision trees. These structures may be implemented with hierarchical workspaces. We may therefore define that a workspace (private or shared) may have child workspaces that inherit the properties of their parents. The child workspaces may be navigated by using anchors available in the parent's workspace.

3.3. *Summary*

Overall, the identified requirements bring forward several functional elements necessary to manage CHDM. This includes defining working groups, process steps and hierarchical workspaces. By adopting this generic structure, we seek to organize work processes independently from the specific decision-making constraints that may be imposed by the application domain.

Working groups support concurrent work, while process steps serve to decompose the decision-making process. The various working modes that may be assigned to process steps support facilitators when organizing the tasks necessary to accomplish the designated goals with flexibility and adaptation to the circumstances. The task-related outcomes reside in workspaces. Workspaces may be private or shared, hierarchically decomposed, and inherit the working modes of the parent's process steps. We highlighted one important type of task-related outcomes: lists. Beyond lists, we considered cursive text, typed text, sketches and pictures. As we show in the next section, these are the main building blocks of the proposed CHDM framework.

4. Application Framework

The developed application framework is illustrated in Figure 1. It defines three functional layers: communications, data model and user interface. These layers, which were derived from patterns for generating layered architectures (Coplien & Schmidt, 1995), will be discussed in detail in the following sections.

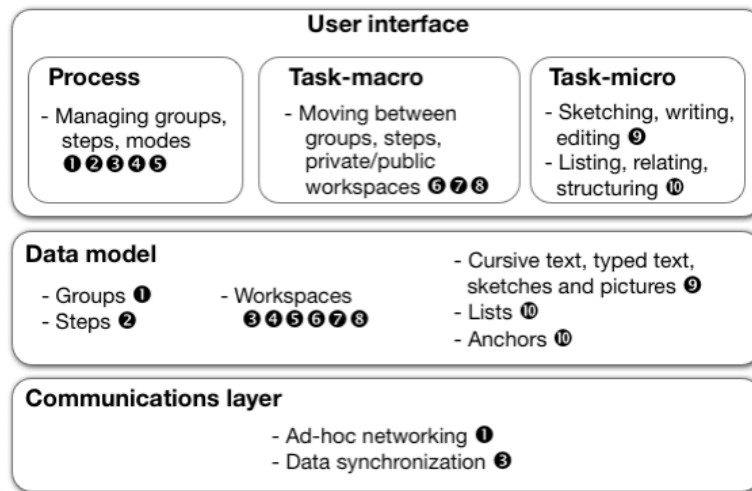


Figure 1. Application framework.

4.1. Communications layer

Mobile computing imposes some specific requirements to the communication layer when compared with those imposed by desktop computing:

Ad-hoc networking: In many mobile scenarios, the only available network may be an ad-hoc network provided by mobile devices, which can roam around at their own will (Neyem, Ochoa, & Pino, 2008). Since the connectivity may be intermittent, the application framework must dynamically manage groups according with their availability (①). This also means that the communications architecture must follow a peer-to-peer schema.

Data synchronization: In a peer-to-peer architecture there is no central server keeping a master copy of the data. Every mobile device must replicate and synchronize data (③). There are mainly two ways to do it in a peer-to-peer architecture: by event or by state. By event means that all mobile devices start with exactly the same state and propagate events

in a way that preserves the same state on all devices. If one data object is concurrently changed in one device, that change must be serialized and propagated to the other ones. Synchronizing by state means that from time to time the whole data structure must be propagated and reconciled. If the data structure is very complex, or the object changes are highly concurrent, the state-based synchronization will cause more network traffic than the event-based. But in an ad-hoc environment where mobile devices may not be constantly reachable, the state-based synchronization can actually be more effective, as it allows work to proceed without serialization. Thus the framework data synchronization scheme adopts the state-based approach.

The communications layer consists of a set of classes implementing an application programming interface that the programmer may use to share data objects. These classes are available in Java and C# and implement the necessary mechanisms for creating and replicating object instances. They also encapsulate the functionality necessary to discover which mobile devices are present in the ad-hoc network, convert data objects from their internal representations into XML (Extensible Markup Language) and transmit data between the connected replicas. And finally, they also encapsulate the data synchronization mechanisms necessary to reconcile data among the connected replicas. The communications layer sends multicast messages at regular intervals to detect the presence of mobile devices and to reconcile data.

4.2. Data model

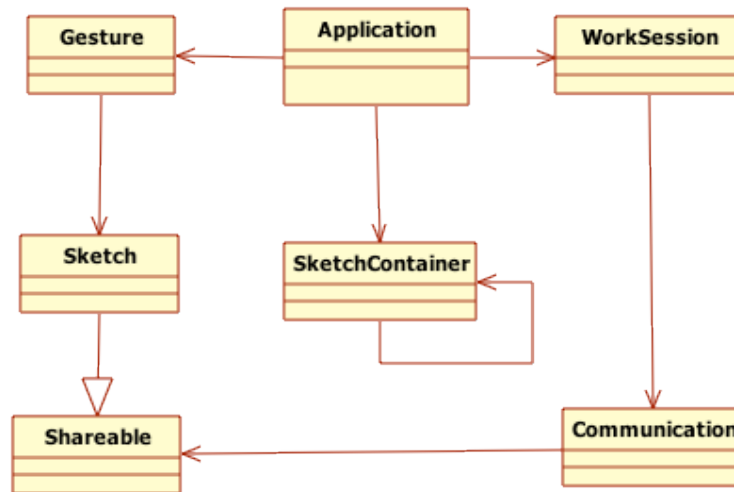


Figure 2: A simplified class diagram describing the data model.

Figure 1 identifies the main concepts managed by the data model, while Figure 2 shows the most important class diagrams implementing these concepts. These classes are common to most of the CHDM tools we have developed so far and may be grouped in three categories: session management, workspace support and gesturing support. The **Application** class is the main class of the application being developed which instantiates the necessary objects required to implement its functionality.

The **WorkSession** class implements session management. It allows the facilitator to define working groups (❶), which are dynamically updated with information coming from the communications layer (**Communication** class) indicating who is actually connected to the ad-hoc network. The facilitator also specifies the process steps (❷) by defining a list of **SketchContainer** objects.

The SketchContainer class is the most complex data element supported by the framework, as it implements the workspace concept. It constitutes a repository for sketches (Sketch class) and other data elements. Sketches are created with the support from the Gesture class, which implements various types of user inputs using hand gestures (⑨). Some of these gestures allow creating lists of SketchContainer instances and links to child SketchContainer instances (⑩). The Sketch class inherits the properties of the Shareable class, which allows sharing task-related outcomes (⑥).

The SketchContainer class controls and enforces a set of behavioral properties that have been already discussed: synchronous/ asynchronous mode, determining when data synchronization is enforced (③); centralized/ decentralized mode, controlling what replicas of the SketchContainer are allowed to receive sketches from the users (④); divergent/ convergent mode, controlling when the workspace is presented to the user (⑤); public/ private mode, controlling which Sketch instances are replicated to the group (⑧) and which are not; and also controlling how the users may navigate the SketchContainer objects (⑦).

4.3. *User interface*

As expected, the user interface is organized according with process, task-macro and task-micro functionality. Whenever possible, the supported interactions are pen-based.

Only the facilitator has access to process functionality: managing working groups, process steps and working modes. Groups are defined by dragging the users' icons to working-group icons, as shown in Figure 3. Process steps are managed by manipulating a list with names identifying the tasks/ goals. Figure 4 shows how the list is displayed to

the facilitator. The working modes are selected in menu options displayed to the facilitator.



Figure 3 - Assigning participants to working groups: The working groups are represented by rectangles (Groups 1-3). Icons labeled with the corresponding name visually represent the participants.

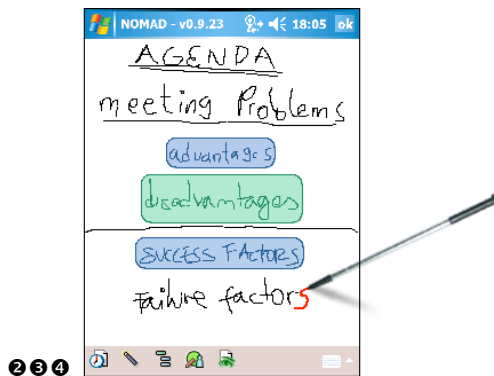


Figure 4 – Defining process steps for a meeting agenda. The three list elements shown at the center (advantages, disadvantages and success factors) define the meeting activities.

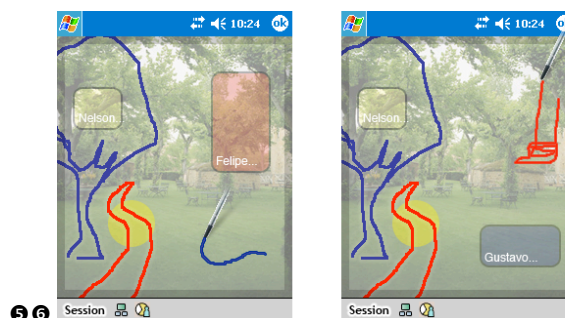


Figure 5 – Convergent mode. Only one process step is active allowing, in this example, two participants drawing over the same picture at the same time.

Considering the task-macro functionality, the user-interface is mostly focused on managing workspaces. Workspaces serve to place and visualize sketches and other data elements, and may extend beyond the display size through drag movements with the pen. The typical scrollbars were omitted to maximize display space, which is very limited in many mobile devices. Figure 5 shows a shared workspace where the participants have drawn ideas for new elements to improve the layout of a park. A photograph of the park has been used as background of the workspace. There are two types of data elements in this workspace: sketches (tree, road) and anchors to other workspaces (yellow spots). The data elements resident in a workspace are displayed according with the inherited synchronous/ asynchronous working modes: the synchronous mode implies that all data elements in the workspace are displayed almost simultaneously, while the asynchronous mode implies the user must explicitly request synchronization. This request is accomplished through a menu option. Figure 5 shows activity in a synchronous mode: all users see the same workspace. Awareness elements in the form of rectangles are displayed in order to inform that other users are sketching on that area. Figure 6 shows an activity in divergent mode, as users are working on different workspaces.

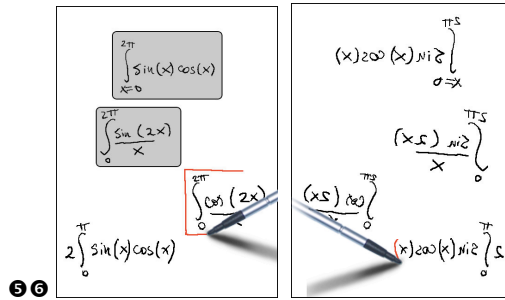


Figure 6 – Divergent mode. Several process steps are active, allowing concurrent viewing. The participant at the left is creating process steps while the participant shown at the right is already working on a step.

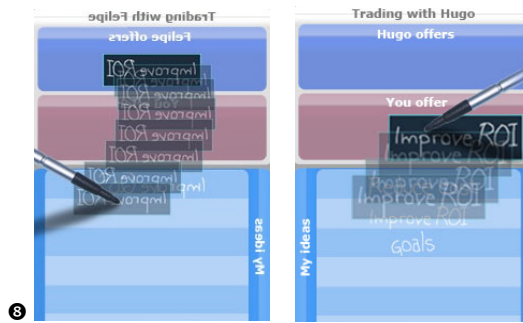


Figure 7 – The “offer area”. At the left, the participant exports an item to the offer area. At the right, the receiving participant imports the item to the private workspace.

As we mentioned before, every process step, besides having a shared workspace, has also a private workspace for each participant. Visually, the private and shared workspaces are shown the same way. The option to view the public or shared workspace is selected in a menu.

To extend the value of private workspaces, we support exchanging private data elements between several members of the working group. This functionality was implemented with a visual interface designated “offer area” (Figure 7), which serves to export data elements (sketches, text, etc.) to selected group members. These members

will be presented with the offer area and may import the information to their private workspaces.

Moving between working groups and process steps required the implementation of the overview window shown in Figure 8. This window shows the available groups, corresponding process steps and workspace hierarchy. The participants may select one of these elements by clicking with the pen. We note however this functionality depends on the specified working mode: the participants do not have access to the overview window if they are in convergent mode; in that case only the facilitator has access to the overview window to select a different step and guide the participants. The synchronous/asynchronous and centralized/decentralized modes have no impact on this functionality.

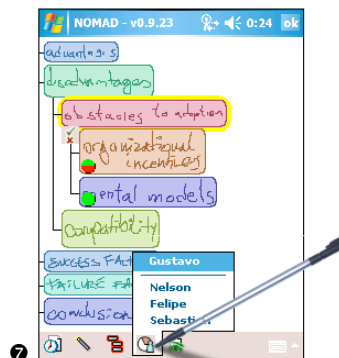


Figure 8 – Moving between working groups and process steps through the overview window.

Let us now describe the task-micro functionality. Sketching and writing deal with cursive text, typed text and sketches. Cursive text allows writing down pieces of text using the pen, while typed text allows writing text using a virtual keyboard. The interactions with these data elements are based on pen-based gestures.

Sketching also serves to organize task-based outcomes and visually integrate information into high-level concepts. For instance, in Figure 9 we illustrate how several list items are visually organized in a typical SWOT (Strengths, Weaknesses, Opportunities, and Threats) diagram. Sketches may also be done over pictures (Figure 10).

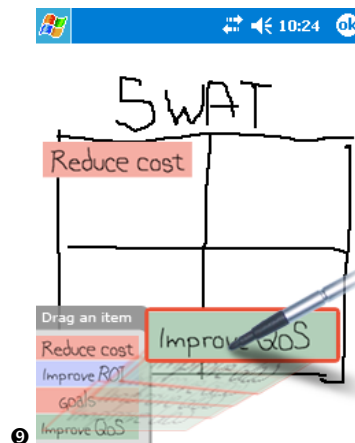


Figure 9 – Using sketches to relate information.

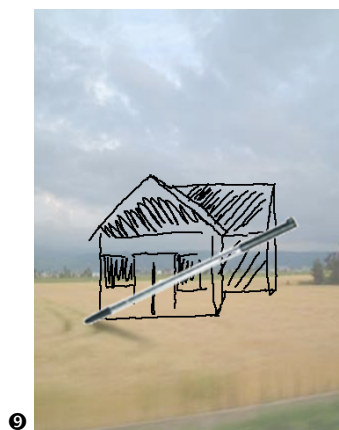


Figure 10 – Sketches over pictures.

Several pen-based gestures have been developed to edit data elements. A single tap allows selecting one element. Drawing a “double closed shape” allows selecting several elements present in the workspace (Figure 11). Drawing a “connected cross” over one or several elements implements the erase function. Rotation, resizing and other

advanced editing features were also implemented. The cut, copy, paste and delete functions are available through contextual menus.

The creation of lists is also based on pen gestures. To create a list, the user writes down some few words using the pen (e.g., “reduce cost”) and then turns those words into a list item by drawing a special gesture recognized by the framework: an horizontal line between two sketches (Figure 12). After creating a list, the user may then manage the list items either individually (e.g., moving an item up or down the list, see Figure 12, bottom) or as group (e.g., moving the list within the workspace).

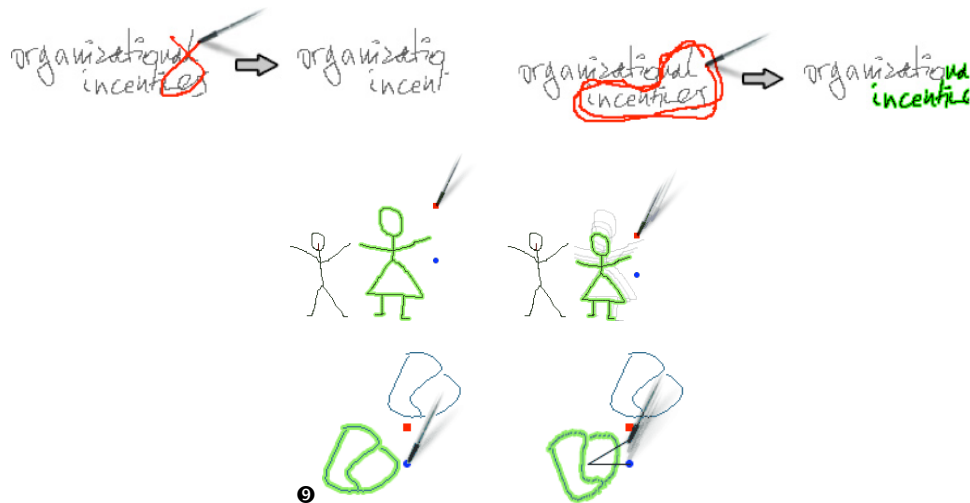


Figure 11 – a) The “connected cross” deletes data elements; b) the “double closed shape” serves to select data elements; c) the “select and move dot” resizes objects; and d) the “select and rotate dot” rotates the data element.

The relating and structuring functionality is based on hierarchical workspaces and anchors. A child workspace may be created through the specification of an anchor. The anchor is a visual object placed in the workspace to indicate that there is the possibility to move down the hierarchy. To create an anchor, the user first has to write some words

and then turn those words into an anchor by drawing a special gesture: a line bending from the vertical to the horizontal. Double-clicking the anchor leads the user towards the sibling workspace. Moving up the workspace hierarchy requires clicking on a special icon, as shown in Figure 13.



Figure 12 – Managing lists. Top: creating list item. Bottom: moving list item.

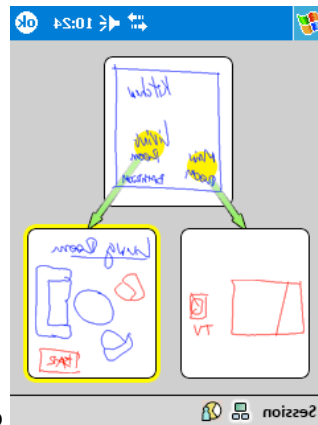


Figure 13 – Moving up and down the workspace hierarchy with anchors.

5. Validation

The framework described in the previous section has been validated through the development of several applications. Some parts of the framework were developed for the first application and reused in almost all subsequent ones, while other parts were developed to resolve more specific design problems. A platform implementing the

required low-level functionality, including ad-hoc networking, data sharing, synchronization, and use-interface elements, has been developed for the first application and then reused for all the other applications. The platform is described in detail in another paper (Baloian, Zurita, Antunes, & Baytelman, 2007). In this paper we will instead focus on demonstrating how the framework elements effectively support CHDM in the selected meeting scenarios.

NOMAD: This was the first developed application. It runs on mobile devices and supports collaboration in the field, where participants generate, discuss, refine and prioritize ideas for further discussion in the office (Zurita & Baloian, 2005). The mobile nature of the application makes it possible to start a decision process anytime and anyplace. It uses pen-based gestures to allow users creating a discussion agenda and other common meeting elements (Figure 2). Each list item can be expanded to a whole-screen page by double-clicking on it, in order to annotate the outcomes of the discussion. Freehand sketching and writing add contents to these pages. Gesture-based edition commands are also available (Figure 9). New lists may be generated inside a page, thus creating a hierarchical tree-like structure, which can be viewed and managed using the overview page (Figure 6). A voting functionality was implemented to rank the generated ideas (list items, and/ or sub items). CHDM using NOMAD typically starts with the creation of the first page of a new project. The subsequent activities (such as creating groups, linking pages, etc.), as well as the order they are performed, depend on what the users may need or want to do. Overall, we observed very significant flexibility implementing most collaboration arrangements falling within the limits of the scenarios

described in Section 2. We have also observed very significant flexibility relative to the presence or absences of the leader/ facilitator.

MCSKETCHER: This application was conceived to support a group of designers generating ideas while visiting an intervention area (G. Zurita, N. Baloian, & F. Baytelman, 2008). It allows sketching design ideas over photographs taken on site (see Figures 3 and 8). Each photograph takes a workspace, but may be copied to other private or shared spaces for refining ideas. These refinements are accessed through anchors. This application implements the convergent and divergent modes, which are illustrated in Figures 3 and 4. MCSKETCHER also implements awareness mechanisms, which have revealed very helpful showing users what the others are doing. It also uses the editing features developed for the previous application, supporting the creation of rough design sketches, which might be refined in future sessions using more powerful design tools.

SENSEMAKING: This application addresses the classroom scenario where a teacher assigns the task of analyzing a collection of papers to a group of students (Zurita, et al., 2007). The papers are related with some topic, but it is part of the assigned task to find the topic and the relationships through exploration and collaboration. Each student receives one paper and must relate it with the others. Students are encouraged to engage in parallel negotiations with multiple parties to reach a common view.

In this application, lists were used to organize the students' notes about the papers. The spatial, visual and conceptual relationships (e.g., Figure 7) were also extensively

used. The infrared communication mechanism available on the mobile devices allowed students to engage in face-to-face collaborations with selected people. The offer area (see Figure 5) is activated when the infrared signal of another device is detected, and allows students to exchange their notes with others in close physical proximity.

MCSUPPORTER: This tool was developed to promote some pedagogical practices in the classroom using mobile devices (G. Zurita, N. Baloian, & F. Baytelman, 2008). It implements teacher's and students' modules. The teacher's module allows assigning students to groups, creating problems, assigning problems to groups and assessing the students' outcomes. The students' module allows working collaboratively on assigned problems. The teacher creates groups by dragging the students' icons to groups, as shown in figure 1. Problems are created with pen-based sketches. Open problems require students to write or draw their solutions. Closed problems require students to choose alternatives from a list. Both questions and answers are managed as lists.

Problems are distributed to the groups through workspaces. The teacher monitors the work being done by navigating the workspaces. Closed problems are managed in a centralized mode, while open problems may be decentralized. The teacher may control the focus of attention by setting a certain workspace to be seen by the whole class, for example, when an interesting answer is worth to be discussed with the whole class.

MOBILE COLLABORATIVE KNOWLEDGE CREATION (MCKC): This application supports knowledge creation. It runs wirelessly on mobile devices, using ad-hoc networks, but it is also able to synchronize data with a central repository.

Knowledge is managed through conceptual maps. Gestures are used to create and manage concepts: create, link, move, drag, etc. Users may associate concepts with external documents like text files and images. MCKC facilitates collaborative knowledge creation based on Nonaka's SECI model (Nonaka & Takeuchi, 1995). According to this model, the application affords knowledge externalization, socialization, combination, and internalization using handheld tools. The participants exchange their notes and concept maps using the offer area (Figure 5).

MCSHELL: This tool supports mobile analysis of complex socio-technical environments (Antunes, et al., 2008; Antunes, Carriço, & Bandeira, 2011). It adopts the SHELL model to characterize relationships between humans, called liveware (L), and four other elements of the working environment (Edwards, 1972): Hardware (H); Software (S), including rules, regulations, procedures and practices; Environment (E); and liveware (L), i.e. other humans. The interfaces between SHELL elements define major areas of analysis: liveware-liveware (L-L), liveware-hardware (L-H), liveware-environment (L-E) and liveware-software (L-S). These interfaces offer a structured approach to assess human factors and system requirements.

MCSHELL facilitates the elicitation of SHELL elements using mobile devices. For instance, analysts have used this tool during interviews and field observations in a hospital. The tool offers features not found in pen/paper approaches: the manipulation of SHELL elements is more immediate and flexible; and a shared workspace allows sharing details about the subjects (liveware) while in the field. The SHELL elements are

specified inside hierarchical workspaces. This way a father node and four son nodes define the L-L, L-S, L-E, L-H relationships.

In Table 3 we present a summary view of the applications described above, highlighting the scenarios of use they address and the framework elements that were used.

Table 3. Validation of framework elements. (❶ - Managing working groups; ❷ - Managing process steps; ❸ - Managing synchronous and asynchronous modes; ❹ - Managing centralized and decentralized modes; ❺ - Managing divergent and convergent modes; ❻ - Sharing task-related outcomes; ❼ - Moving between working groups and process steps; ❽ - Managing private and public information; ❾ - Sketching, writing, editing; ❿ - Listing, relating, structuring).

Applications	Scenarios	Process	Macro	Micro
NOMAD	Deliberation, Consultation, Ideation, Exploration	❶ ❷ ❸ ❹ ❺	❻	❾ ❿
MCSKETCHER	Ideation	❸ ❹	❻ ❽	❾ ❿
SENSEMAKING	Learning, Ritualization, Exploration	❶ ❸	❻ ❼ ❽	❾ ❿
MCSUPPORTER	Learning	❶ ❷ ❸ ❹ ❺	❻ ❼	❾ ❿
MCKC	Aligning, Consultation, Ritualization, Exploration	❸ ❹ ❺	❻ ❽	❾ ❿
MCSHELL	Exploration	❷ ❸ ❺	❻ ❼	❾ ❿

6. Related Work

In this section we discuss several CHDM frameworks recently developed (since 2006). All selected works propose some kind of application framework and involve handheld tools and collaboration support. In Table 4 we highlight the similarities and distinctions with the framework proposed in this paper.

David et al. (2006) defined a framework for developing mobile collaborative applications. The main research goal was developing a graphical formalism that could transform collaborative use cases into work process specifications, which would then be executed by several underlying components implementing data distribution, sharing and collaboration. The proposed formalism addresses synchronous/ asynchronous working modes (⑤) and shared/ private activities (③). It also considers session management (②), user and group management (①), data sharing (⑥) and concurrency control (④).

Neyem et al. (2006) developed a framework supporting ad-hoc data sharing for mobile collaboration (⑥). It holds upon the view that in many situations, especially those involving emergency response, had-hoc networks may compensate the collapse of structured communications networks. In these circumstances, work may change from synchronous to asynchronous on conditions extraneous to the group and task (③). This framework also supports centralized/ decentralized work modes (④) and shared/private activities (③).

Bollen et al. (2008) developed an application framework supporting learning activities where mobile technology serve to decentralize data inputs and integrates with a shared whiteboard facilitating group visualization and discussion (⑥③). The framework focus is on the integration of heterogeneous data and devices, encapsulating

the specific details of data communication, synchronization and persistency. Furthermore, the framework also supports the specification of action patterns, which serve to manage the collaboration process (②).

Table 4. Comparative overview of several CHDM systems (❶ - Managing working groups; ❷ - Managing process steps; ❸ - Managing synchronous and asynchronous modes; ❹ - Managing centralized and decentralized modes; ❺ - Managing divergent and convergent modes; ❻ - Sharing task-related outcomes; ❼ - Moving between working groups and process steps; ❽ - Managing private and public information; ❾ - Sketching, writing, editing; ❿ - Listing, relating, structuring).

Reference	Main goal	Process					Macro			Micro	
		❶	❷	❸	❹	❺	❻	❼	❽	❾	❿
(David, et al., 2006)	Collaboration modeling	X	X	X	X		X		X		
(Neyem, et al., 2006)	Ad-hoc data sharing			X	X		X		X		
(Bollen, et al., 2008)	Mobile note taking		X				X		X		
(van der Heijden, 2006)	On-the-go information access										X
(Cowie & Burstein, 2007)	Mobile decision making										X
(Bastéa-Forte & Yen, 2007)	Shared sketching						X			X	
(Lee, 2009)	Mobile meeting services									X	

Van der Heijden (2006) developed a framework supporting on-the-go access to information assets like the prices of goods displayed in a store (❿). The main research focus was set on developing the ability to automatically obtain contextualized information (using Radio Frequency Identification) and match that information with decision criteria. This way, while the customer wanders through the store, the handheld tool displays a dynamic matrix with pricing information. The tool helps comparing similar goods and deciding which ones to buy.

Cowie and Burstein (2007) developed a mobile decision-making tool. Departing from a typical multi-criteria decision-making model, the authors extended it with parameters that depend on the mobile context. Since the main research focus is restricted to the interaction between the user and the decision-making model, we only check the listing, relating and structuring functionality in Table 3 (10).

Bastéa-Forte and Yen (2007) developed an application supporting the use of mobile devices in face-to-face brainstorming meetings. Ideas are individually sketched in mobile devices (9) and shared in a large public whiteboard (6), which promotes parallel contributions and at the same time allows perceiving the group progress. Experiments highlight the application promoted even contributions and more collaboration from the group members.

Lee (2009) reports the development of an application framework based on Web-based technologies to integrate information access and communication into a series of coherent services supporting mobile meetings. Various types of mobile devices may be used, with some emphasis on mobile phones. Within the list of services we find: (1) mobile information access; (2) simultaneous voice conversations; and (3) content navigation (9).

Overall, we observe two functions not covered by these works: managing divergent/convergent modes and moving between working groups and process steps. This lack of support impacts the flexibility of CHDM tools, as they afford more decentralized and loose work structures. We also observe a dichotomy between support to process/macro functionality and support to micro functionality, which indicates that one potential

benefit of the framework proposed in this paper is the integrated management of such distinct functionality.

Finally, we note that the approach proposed by David et al. (2006) has strong affinities with the one developed in this paper. However, the main focus of the cited research was the definition of a graphical specification language, instead of an application development framework. We nevertheless emphasize the significant complementarity and potential synergies between the two approaches.

7. Discussion and Conclusions

One interesting characteristic of the proposed framework is that it reconsiders and repurposes different traditions found in the decision-making body of research, which based on the research by Morton et al. (2003) we classify as tool-driven, process-driven and model-driven. The central preoccupation of tool-driven approaches is providing a set of configurable and interoperable tools supporting typical decision-making activities such as brainstorming, categorizing and voting (Nunamaker, Briggs, Mittleman, Vogel, & Balthazard, 1997). Process-driven approaches are mainly concerned with the procedural aspects of decision-making. The focus is on partitioning decision processes using the engineering paradigm, analyzing the organizational context and designing a decision-making process that may be successfully adopted by a group (Kolfshoten & de Vreede, 2009; Kolfshoten, et al., 2006).

Finally, model-driven approaches regard decision modeling as the gist of decision making (Morton, et al., 2003). The emphasis is on the articulation of a common view

about the problem and possible solutions, usually with support from an expert facilitator that helps eliciting and organizing data, resolving conflicts, and building consensus.

We observe that the proposed framework combines all facets mentioned above. In the top layer, task-micro functionality addresses decision modeling, while task-macro and process functionality tackle tool interoperability and process structuring. In the middle layer, workspaces and sketches cover the task and decision structures posited by process-driven and model-driven approaches. Finally, the bottom-layer supplies communication functionality shared by the tool, process and model-driven approaches. Therefore a distinctive characteristic of the proposed framework is specifying a comprehensive application platform that embraces functions found in other decision-making approaches but that have not been integrated until now.

Furthermore, the proposed framework extends decision-making support to handheld/mobile contexts. This extension seems particularly adequate to the new types of users and professional engagements that have been recently encouraged by mobile devices and networks. In particular, we note the close relationships with “offroaders”, i.e. people who have highly developed skills and dedication to their tasks, and have the “capability to deliver solutions to problems in circumstances that others would find difficult” (Harmer & Pauleen, 2012). One interesting characteristic of “offroaders” is that, while being inseparable from mobile technology, their main focus of attention is the task and not the technology. The proposed framework contributes to enable technology support to decision making with such level of transparency.

Discussing the framework elements in more detail, we note that the most distinct component is the workspace, which manages several important attributes related to data

sharing and decision-making: managing the groups' focus of attention, defining interaction modes, allowing the users to move between different tasks and contents, structuring complex information, and integrating shared and private information. Such functional richness has been considered essential to support decision-making in complex settings such as distributed medical team meetings (Kane, Groth, & Randall, 2011; Li & Robertson, 2011).

Another important framework component is the sketch. Sketches serve not only to jot text but also to establish conceptual relationships and structure workspaces. Again, very rich functionality is available to work with sketches, such as selecting and editing sketches using specific gestures. Sketching is particularly relevant to support handheld decision-making, considering they support offroaders managing information while on the move.

In this paper we describe six applications developed with the framework. These applications implement different decision-making activities in quite different working contexts. Overall, with these implementations we have come to the following conclusions:

- Many different decision-making tasks and processes may be developed using the proposed framework;
- The framework offers flexible support to CHDM based on a relatively small number of user-interface, data model and communications elements;
- The framework is adequate to develop decision-making applications for mobile contexts;

- The burden associated with group and task management is quite low. Although the framework supports the facilitator role, it does not depend on that role and indeed allows any user to assume that role with no specific training;
- The application development on top of the framework has been surprisingly effortless. For instance, the SHELL application was developed in one week.

Of course the proposed framework also brings significant challenges to several stakeholders. Let us start with application developers. One fundamental problem that application developers have to consider is structuring the application functionality using workspace hierarchies, which may be considered too low-level and too restrictive (currently, we only support text, sketches, pictures, lists and anchors).

Considering decision makers, we observe that the proposed framework brings about social and cultural implications. One problem is that decision makers may find themselves in the uncomfortable position of designers, i.e. having to use a graphical tool to design a decision process. To illustrate the case, we note that the developed technology provides the user-interface elements necessary to execute SWOT analysis, but the users will have to make several decisions regarding how to do it, e.g. by defining a sequence of four tasks (Strengths, Weaknesses, Opportunities, and Threats) or by populating four workspaces with the four types of assessments. Another significant problem to consider is that decision-makers may find it uncomfortable to use sketches. Studies with storyboarding show that non-experts often feel they lack the drawing skills necessary to use creative tools (Truong, Hayes, & Abowd, 2006), something that may be extrapolated to our case. This problem may be aggravated by mobile devices, which

have been considered to create several mobile impairments related with information processing and visualization (Harper, Yesilada, & Chen, 2011).

We finally consider some important challenges to researchers. Gray and Mandiwalla (1999) noted that decision-making tools have been hardly used in organizations, often because they lack a strong link to the organizational practices of the users they are trying to support. The process-driven approach addresses this problem by having collaboration engineers fine-tuning decision processes. However such an approach introduces other issues, such as the dependence on external collaboration engineers, and dealing with constant internal and external changes. The proposed application framework embraces the collaboration engineering approach and contributes to its realization by supplying visual elements, functionality and structure necessary to implement process designs with mobile devices.

Another interesting research challenge is related with the current lack of standardization in the decision-making field. The proposed framework responds to this lack of standardization tackling aspects of language (e.g., macro and micro distinctions), functionality (process, task-macro and task-micro functions) and data model (session management, workspace and gesturing support).

Acknowledgments. This research was funded by Fondecyt, project nr. 1085010.

References

Antunes, P., & André, P. (2006). A Conceptual Framework for the Design of Geo-Collaborative Systems. *Group Decision and Negotiation*, 15, 273-295.

- Antunes, P., Bandeira, R., Carriço, L., Zurita, G., Baloian, N., & Vogt, R. (2008). Risk Assessment in Healthcare Collaborative Settings: A Case Study Using Shell Groupware: Design, Implementation, and Use. *14th International Workshop, CRIWG 2008, Omaha, Us, September 2008 Proceedings* (Vol. 5411). Heidelberg: Springer.
- Antunes, P., Carriço, L., & Bandeira, R. (2011). Assessing Risk in Healthcare Collaborative Settings. In S. Kabene (Ed.), *Human Resources in Healthcare, Health Informatics and Health Systems*. Hershey, NY: Medical Information Science Reference.
- Antunes, P., & Ho, T. (2001). The Design of a Gdss Meeting Preparation Tool. *Group Decision and Negotiation*, 10(1), 5-25.
- Antunes, P., Zurita, G., Baloian, N., & Sapateiro, C. (2013). Integrating Decision-Making Support in Geocollaboration Tools. *Group Decision and Negotiation*, online first.
- Bafoutsou, G., & Mentzas, G. (2002). Review and Functional Classification of Collaborative Systems. *International Journal of Information Management*, 22, 281-305.
- Baloian, N., Zurita, G., Antunes, P., & Baytelman, F. (2007). *A Flexible, Lightweight Middleware Supporting the Development of Distributed Applications across Platforms*. The 11th International Conference on CSCW in Design, Melbourne, Australia.
- Bange, M., Deutscher, S., Larsen, D., Linsley, D., & Whiteside, S. (2004). A Handheld Decision Support System to Facilitate Improved Insect Pest Management in

- Australian Cotton Systems. *Computers and Electronics in Agriculture*, 43(2), 131-147.
- Bastéa-Forte, M., & Yen, C. (2007). *Encouraging Contribution to Shared Sketches in Brainstorming Meetings*. CHI '07 extended abstracts on Human factors in computing systems, San Jose, CA.
- Bergstrand, F., & Landgren, J. (2009). *Information Sharing Using Live Video in Emergency Response Work*. Conference on Information Systems for Crisis Response and Management - ISCRAM '09, Gothenburg, Sweden.
- Bollen, L., Giemza, A., & Hoppe, U. (2008). Flexible Analysis of User Actions in Heterogeneous Distributed Learning Environments *Times of Convergence. Technologies across Learning Contexts* (Vol. 5192). Heidelberg.
- Bragge, J., Merisalo-Rantanen, H., Nurmi, A., & Tanner, L. (2006). A Repeatable E-Collaboration Process Based on Thinklets for Multi-Organization Strategy Development. *Group Decision and Negotiation*, 16(4), 363-379.
- Cai, G., MacEachren, A., Brewer, I., McNeese, M., Sharma, R., & Fuhrmann, S. (2005). Map-Mediated Geocollaborative Crisis Management *Intelligence and Security Informatics* (Vol. 3495, pp. 429-435). Heidelberg: Springer Verlag.
- Coplien, J., & Schmidt, D. (1995). *Pattern Language of Program Design*. Reading, Massachussets: Addison-Wesley.
- Cowie, J., & Burstein, F. (2007). Quality of Data Model for Supporting Mobile Decision Making. *Decision Support Systems*, 43(4), 1675-1683.

- David, B., Chalon, R., Delotte, O., Masserey, G., & Imbert, M. (2006). Orchestra: Formalism to Express Mobile Cooperative Applications *Groupware: Design, Implementation, and Use* (Vol. 4154). Heidelberg: Springer.
- Davies, N., Blair, G., Cheverst, K., & Firday, A. (1994). *Supporting Adaptive Services in a Heterogeneous Mobile Environment*. Proceedings of the Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA.
- Davies, N., Mitchell, K., Cheverst, K., & Blair, G. (1998). *Developing a Context Sensitive Tourist Guide*. Lancaster, UK: University of Lancaster.
- Edwards, E. (1972). *Man and Machine: Systems for Safety*. proceedings of British airline pilots Association Technical Symposium, London.
- Fjermestad, J., & Hiltz, S. (1999). An Assessment of Group Support Systems Experimental Research: Methodology and Results. *Journal of Management Information Systems*, 15(3), 7-149.
- Forbus, K., Ferguson, R., & Usher, J. (2001). *Towards a Computational Model of Sketching*. IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces, Santa Fe, New Mexico.
- Gray, P., & Mandiwalla, M. (1999). New Directions for Gdss. *Group Decision and Negotiation*, 8, 77-83.
- Harmer, B., & Pauleen, D. (2012). Attitude, Aptitude, Ability and Autonomy: The Emergence of 'Offroaders', a Special Class of Nomadic Worker. *Behaviour & Information Technology*, 31(5), 439-451.

- Harper, S., Yesilada, Y., & Chen, T. (2011). Mobile Device Impairment ... Similar Problems, Similar Solutions? *Behaviour & Information Technology*, 30(5), 673-690.
- Kane, B., Groth, K., & Randall, D. (2011). Medical Team Meetings: Utilising Technology to Enhance Communication, Collaboration and Decision-Making. *Behaviour & Information Technology*, 30(4), 437-442.
- Kankainen, A., Vaajakallio, K., Kantola, V., & Mattelmäki, T. (2012). Storytelling Group – a Co- Design Method for Service Design. *Behaviour & Information Technology*, 31(3), 221-230.
- Kolfshoten, G., & de Vreede, G. (2009). A Design Approach for Collaboration Processes: A Multimethod Design Science Study in Collaboration Engineering. *Journal of Management Information Systems*, 26(1), 225-256.
- Kolfshoten, G., Hengst-Bruggeling, M., & Vreede, G. (2006). Issues in the Design of Facilitated Collaboration Processes. *Group Decision and Negotiation*, 16(4), 347-361.
- Kraemer, K., & King, J. (1988). Computer-Based Systems for Cooperative Work and Group Decision Making. *ACM Computing Surveys*, 20(2), 115-146.
- Lapinsky, S. (2006). Mobile Computing in Critical Care. *Journal of Critical Care*, 22(1), 41-44.
- Laso-Ballesteros, I., & Salmelin, B. (2005). Ami-Endowed Collaboration@Work. In G. Riva, F. Vatalaro, F. Davide & M. Alcaniz (Eds.), *Ambient Intelligence: The Evolution of Technology, Communication and Cognition Towards the Future of*

- Human-Computer Interaction* (pp. 237-265). Amsterdam, The Netherlands: IOS Press.
- Lee, A. (2009). *Mobile Web Widgets: Enabler of Enterprise Mobility Work*. Madrid, España: Nokia Research Center.
- Li, J., & Robertson, T. (2011). Physical Space and Information Space: Studies of Collaboration in Distributed Multi-Disciplinary Medical Team Meetings. *Behaviour & Information Technology*, 30(4), 443-454.
- MacEachren, A. (2005). Moving Geovisualization toward Support for Group Work. In J. Dykes, A. MacEachren & M. Kraak (Eds.), *Exploring Geovisualization* (pp. 445-462): Elsevier.
- MacEachren, A., & Brewer, I. (2004). Developing a Conceptual Framework for Visually-Enabled Geocollaboration. *International Journal of Geographical Information Science*, 18(1), 1-34.
- Malone, T., & Crowston, K. (1994). The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, 26(1), 87-119.
- Mark, G. (2002). Extreme Collaboration. *Communications of the ACM*, 45(6), 89-93.
- Messeguer, R., Ochoa, S., Pino, J., Medina, E., Navarro, L., Royo, D., et al. (2009). Building Real-World Ad-Hoc Networks to Support Mobile Collaborative Applications: Lessons Learned. In L. Carriço, N. Baloian & B. Fonseca (Eds.), *Groupware: Design, Implementation, and Use. 15th International Workshop, CRIWG 2009, Douro, Portugal* (Vol. 5784). Heidelberg: Springer.

- Miller, R., Hillman, J., & Given, R. (2004). Physician Use of It: Results from the Deloitte Research Survey. *Journal of Healthcare Information Management*, 18(1), 72-80.
- Monzani, J., Bendahan, S., & Pigneur, Y. (2004). *Decision and Visualization for Negotiation*. Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004.
- Morton, A., Ackermann, F., & Belton, V. (2003). Technology-Driven and Model-Driven Approaches to Group Decision Support: Focus, Research Philosophy, and Key Concepts. *European Journal of Information Systems*, 12(2), 110-126.
- Neyem, A., Ochoa, S., & Pino, J. (2006). Supporting Mobile Collaboration with Service-Oriented Mobile Units. In Y. Dimitriadis, I. Zigurs & E. Gómez-Sánchez (Eds.), *Groupware: Design, Implementation and Use, 12th International Workshop, CRIWG 2006* (Vol. 4154, pp. 228-245). Medina del Campo, Spain: Springer.
- Neyem, A., Ochoa, S., & Pino, J. (2008). Integrating Service-Oriented Mobile Units to Support Collaboration in Ad-Hoc Scenarios. *Journal of Universal Computer Science*, 14(1), 88-122.
- Nonaka, I., & Takeuchi, H. (1995). *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*: Oxford University Press.
- Nunamaker, J., Briggs, R., Mittleman, D., Vogel, D., & Balthazard, P. (1997). 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.

- Nunamaker, J., Dennis, A., Valacich, J., Vogel, D., & George, J. (1991). Electronic Meeting Systems to Support Group Work: Theory and Practice at Arizona. *Communications of the ACM*, 34(7), 40-61.
- Nyerges, T., Montejano, R., Oshiro, C., & Dadswell, M. (1997). Group-Based Geographic Information Systems for Transportation Site Selection. *Transportation Research C*, 5(6), 349-369.
- Osborn, A. (1963). *Applied Imagination*. New York: Charles Scribner's Sons.
- Peffer, K., Tuunanen, T., Trothenberger, M., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), 45-77.
- Reinig, B., & Briggs, R. (2006). *Measuring the Quality of Ideation Technology and Techniques*. HICSS '06. Proceedings of the 39th Annual Hawaii International Conference on System Sciences, Hawaii.
- Sapateiro, C., & Antunes, P. (2009). *An Emergency Response Model toward Situational Awareness Improvement*. International Conference on Information Systems for Crisis Response and Management, Göteborg, Sweden.
- Shneiderman, B. (2002). Creativity Support Tools. *Communications of ACM*, 45(10), 116-120.
- Simon, H. (1977). *The New Science of Management Decision*. Upper Saddle River, NJ: Prentice Hall.
- Škraba, A., Kljajić, M., & Borštnar, M. (2007). The Role of Information Feedback in the Management Group Decision-Making Process Applying System Dynamics Models. *Group Decision and Negotiation*, 16(1), 77-95.

- Sousa, P., Preguiça, N., & Baquero, C. (2009). Forby: Providing Groupware Features Relying on Distributed File System Event Dissemination. In L. Carriço, N. Baloian & B. Fonseca (Eds.), *Groupware: Design, Implementation, and Use. 15th International Workshop, CRIWG 2009, Douro, Portugal* (Vol. 5784). Heidelberg: Springer.
- Srivastava, L. (2005). Mobile Phones and the Evolution of Social Behaviour. *Behaviour & Information Technology*, 24(2), 111-129.
- The 3M Meeting Management Team (1994). *Mastering Meetings*. New York: McGraw-Hill, Inc.
- Truong, K., Hayes, G., & Abowd, G. (2006). *Storyboarding: An Empirical Determination of Best Practices and Effective Guidelines*. Proceedings of the 6th conference on Designing Interactive systems, University Park, PA, USA.
- Tyran, G., & Sherpherd, M. (2001). Collaborative Technology in the Classroom: A Review of the Gss Research and a Research Framework. *Information Technology and Management*, 2(4), 395-418.
- van der Heijden, H. (2006). Mobile Decision Support for in-Store Purchase Decisions. *Decision Support Systems*, 42(2), 656-663.
- Webne-Behrman, H. (1998). *The Practice of Facilitation*. Westport, Connecticut: Quorum Books.
- Yu, F., Houston, T., Ray, M., Garner, D., & Berner, E. (2007). Patterns of Use of Handheld Clinical Decision Support Tools in the Clinical Setting. *Medical Decision Making*, 27(6), 744-753.

- Zurita, G., Antunes, P., Baloian, N., & Baytelman, F. (2007). Mobile Sensemaking: Exploring Proximity and Mobile Applications in the Classroom. *Journal of Universal Computer Science*, 13(10), 1434-1448.
- Zurita, G., Antunes, P., Baloian, N., Carriço, L., & Baytelman, F. (2008). Using PDAs in Meetings: Patterns, Architecture and Components. *Journal of Universal Computer Science*, 14(1), 123-147.
- Zurita, G., & Baloian, N. (2005). Handheld-Based Electronic Meeting Support Groupware: Design, Implementation, and Use (Vol. 3706, pp. 341-350).
- Zurita, G., Baloian, N., & Baytelman, F. (2008). A Collaborative Face-to-Face Design Support System Based on Sketching and Gesturing. *Advanced Engineering Informatics*, 22(3), 340-349.
- Zurita, G., Baloian, N., & Baytelman, F. (2008). *Supporting Rich Interaction in the Classroom with Mobile Devices*. Proceedings of the Fifth IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education, Beijing, China.