

Using A Mobile Collaborative Approach To Improve Healthcare Tasks Planning

Fouzi Lezzar, Abdelmadjid Zidani and Atef Chorfi

LaSTIC laboratory, University of Batna, Algeria

lezzar.fouzi@gmail.com abdelmadjid.zidani@univ-batna.dz atefchorfi@yahoo.fr

Abstract

Health care task planning process is fundamentally a collaborative task that requires the participation of many health care professionals with different skills working simultaneously on same cases. However, most recent healthcare research has focused on methods mainly based on communication, rather than collaboration supports. We followed then a collaborative-based planning approach which constitutes an evolution of planning environments toward new shared workspaces supporting collaboration. Our work led us first, to analyse the related tasks in an Algerian maternity ward in order to highlight the vital collaborative medical tasks that need to be modelled. The following sections outline basic design concepts of our mobile collaborative planning system, which is designed to provide a flexible group interaction support for care coordination and continuity.

Keywords: *Healthcare process modelling, Collaborative planning, Synchronous/Asynchronous interaction, Computer Supported Collaborative Work*

1. Introduction

In healthcare planning, the need of collaboration is well recognized [1] and it seems obvious now that such task remains hard to be individually completed. We started therefore by analysing the planning process and attempted to understand how health care professionals may elaborate plans, and what happens when this work is done with a team of collaborators. It is also important to well identify which users are implied in such process. Such understanding will undoubtedly allow us to provide the adequate design by addressing the following interrogations:

- How medical staff members' collaborate?
- What means are required to improve the care process?
- Which artefacts are used to coordinate work? And how?
- From a collaboration point of view, what are the specific characteristics of collaborative medical activities?
- Which computer tools may provide the required assistance for the medical staff members and get them to work collaboratively?

It is practically impossible to develop a computer tool addressing all users' needs. Nevertheless, group work experiences provide us pertinent information to clarify some useful development ideas about suitable support tools. The experimentation of these tools thereafter, will unveil obstacles to overcome as well as perspectives to follow. Our approach is drawn in a direction which aims to favour collective thought and expertise. Therefore, as we will show it in the following sections, the planning task analysis will bring us an understanding to concretely increase the commitment of participants that may have a great impact on the whole related process.

The main objective of this paper is to present the mobile collaborative system MCPlan supporting healthcare planning activities. In the following sections we will first introduce the notion of collaborative planning and review the main works achieved in this area. In section 3 we will present Android technology which is used to develop our collaborative system. Sections 4 and 5 expose our conceptual methodology and discuss the choices made as well as the software architecture designed for MCPlan. We consider then details of the different architecture levels as well as the main functionalities supported. We will explicitly attempt to show that MCPlan design is mainly focused on concepts of data sharing and exchange to favour collaboration between participants. Section 6 explains how our system is implemented. Finally, perspectives of the accomplished work are presented in the conclusion of the paper.

2. Collaborative Healthcare Tasks Planning

Hospital emergency wards such as Gynaecology and obstetrics maternities are extremely complex to manage and pose serious health risks to patients. Related tasks which are mainly focused around patient management are basically achieved through a cooperative way that involves several health care professionals.

Our observation study [4] has explicitly showed that most of the medical activities we supervised were group-based. Likewise the main deficiencies in patients monitoring precisely arose from the lack of coordination between the various members of the involved medical team, which thus constitutes a key factor as it has been so well confirmed several numerous studies carried out on this issue [2, 14].

Based on this observation, we believe that the task management should require from us a special attention. We must therefore address the issue of the targeted maternity ward under a new perspective, that of the medical staff needs, taking into account the economic and performance constraints as well as socio-health hospitals mission objectives of providing optimal care and well-being of patients. Providing a technological answer through cooperation, coordination and communication facilities seems to be the most appropriate initiative. However the past experiences reported in this area that work in-situ [4, 5] should be first carefully analysed from a social point of view [6], and through a structurally opened cooperation vision that enables users to build their cooperation workspace structure in order to interact within it [7]. Consequently, we focused our interest on collaborative practices of patient care teams [8] as well as their organization [9] to better understand the usual manner with which tasks are actually performed.

Several collaborative medical care needs have been identified by a wide body of researches in informatics and medical science fields. There are common processes that are more difficult and complex in collaborative situations, because they need to integrate many parties. Such as decision making process that needs involvement of several persons to arrive to a decision, which can take long time [2]. In [10] authors showed that the collaborative nature of the executed process, determines the type of information management necessary for this process. Though, a poor structure of information can lead to coordination and communication breakdowns.

Maternity services are highly risky and still very hard to manage. They require coordination among several teams whose tasks achievement most of the time confront them to conflict situations [1]. The exploitation of information and communication technologies proves that is an effective approach if it is appropriately used [11]. It will enable us reduce the effects generated by the coordination problems that directly disrupt the patient's care chain and degrade their quality as noted Scupelli [12] through his study. Consequently, coordination breakdowns among the medical staff members inevitably that have an impact on the quality of care provided to patients and put them in a potentially vulnerable and dangerous situations should be significantly reduced with the availability of a medium of communication, cooperation, and coordination. Such an approach will provide collaborative tools that may effectively address medical staff vital needs and improve the quality of patients' care. Our research work falls then within the CSCW area (Computer Supported Cooperative Work). Thus, with a CSCW-based management strategy [3] we wish to provide an effective support of these activities enabling by the way finer planning features of the related tasks as well as providing real time mutual awareness around the occurring events within the maternity unit which constitutes a priority of our work.

MCPlan (for Mobile Collaborative Planning) design is intended with many pertinent objectives in mind such as to take in account healthcare professionals availability and mobility. MCPlan supports two main interaction modes through which users may individually or collectively achieve their tasks.

According to these observations, we claim that a planning task should be meticulously analysed from a groupware perspective. Such work will probably provide us with a better understanding of the collaborative healthcare task planning process. Indeed, the complexity generated by scheduling process modelling comes from the fact that there are unexpected events. This kind of events plus a lack of coordination can generate undesirable results.

Furthermore, the complexity is increased with a team of healthcare professionals working together. Thus collaborative maintenance projects still difficult, even when sophisticated communication tools are used. The main reason is the lack of social contact due to the geographical separation of experts [14, 15]. Besides, it is necessary to provide the adequate supports to maintain data consistency, concurrent accesses, group interaction, and coordination. Groupware concepts may help us to effectively reduce these problems, such as group awareness that allows experts to be aware of the others work and be warned about events that may arise within the hospital and within the shared workspace on the common handled artefacts.

3. Android

Android is a mobile phone platform developed by the Google-led Open Handset Alliance (OHA)¹. Because of the open source platform and adoption by telecommunication providers, Android quickly became popular amongst the developers community.

¹ <http://www.openhandsetalliance.com/>

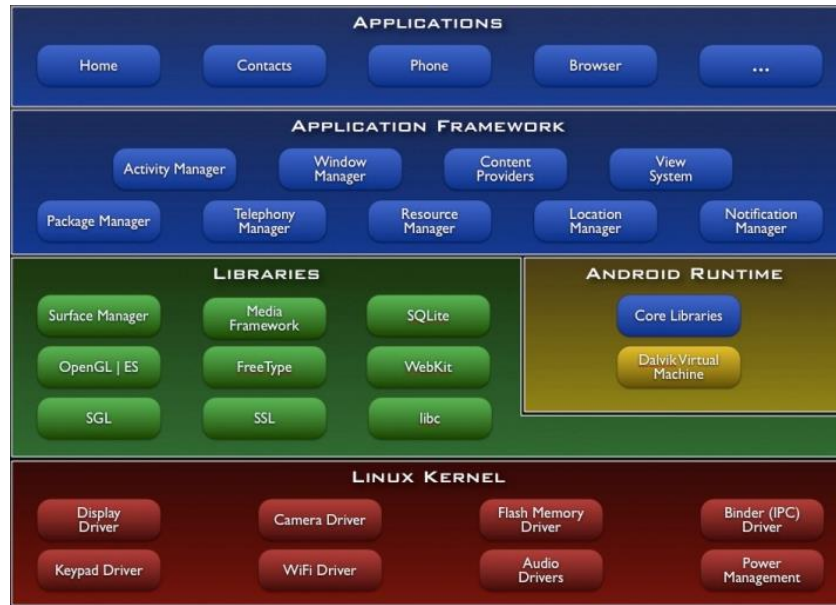


Figure 1. Android architecture [16]

Android is based on Linux (Figure 1), what hides traditional OS abstractions because of to the middleware presented to application developers. Much of the platform functionality is implemented as applications in the same way used by third-party developers. Android relies on Linux version 2.6 for core system services: memory management, network stack and security. The kernel acts also as an intermediate between the hardware and the rest of the software stack.

Android contains some basic Java libraries along with Android specific libraries and the Dalvik Virtual Machine (DVM). DVM allows the device to run multiple virtual machines (VMs), which allows every Android application to run in its own process. The Dalvik VM relies on the Linux kernel for functionalities such as low-level memory management and threading. For IPC (Inter Process Communication) Android provides a sophisticated message passing system, in which Intents are used to link applications. Intents are used in applications use for both inter and intra-application communication. Intentions allow applications to have notifications on events. These can be sent by the terminal (incoming call, SMS, internet connection lost) or by another application wants to send information. The framework is located above the Android runtime and libraries.

Android provides APIs that enable developers to create rich applications. Android applications include an email client, SMS (Short Message Service) programs, a calendar, maps, a web browser, contacts and other programs written in the Java programming Language. There are four types of components used to construct applications; each type has a specific purpose:

- **Activity:** An Activity is an application component that provides a screen with which users can interact in order to do something, such as dial the phone, take a photo, send an email, or view a map. Each activity is given as a window in which the user interface is drawn. The window typically fills the screen, but may be smaller than it and float on top of other windows.
- **Content Provider:** manages and controls the accessibility data for a certain

application. The various applications on a device do not need to be independent. Indeed, they may need information stored in other applications (recovery of SMS, phone settings, *etc...*)[16].

- **Services:** A Service is an application component that can perform long-running operations in the background and does not provide a user interface. Another application component can start a service and it will continue to run in the background even if the user switches to another application. Additionally, a component can bind to a service to interact with it and even perform interprocess communication (IPC). For example, a service might handle network transactions, play music, perform file I/O, or interact with a content provider, and all that from the background [16].
- **Broadcast receiver:** provide a generalized mechanism for asynchronous event notifications. So it acts as mailboxes for messages from other applications.

4. The Mobile Collaborative System

In this paper we developed a synchronous Android mobile groupware that enables real-time collaboration among collocated or geographically separated group members.

The first part of our software architecture presented in Figure 2 contains the system database which is mainly characterized by its capacity to provide concurrency control management, reliable data for long time, security capabilities and data storage. The developed system database contains all the necessary data for management and for the collaborative planning process as the list of available and occupied resources and list of future tasks to accomplish... We used a relational and centralised database for many reasons: a reduced complexity during the development process and also because the data volume is not important.

The second part contains all the software components. In the case of a real-time groupware, sharing data and events constitutes the most important aspect. Thus enabling data sharing requires that any event or data generated by one user has to be immediately delivered to all the other collaborators (in real-time). For better workspace awareness, fault tolerance, responsiveness, and replication of shared data objects are often used together with other operations on them like creation, updating, deletion and reading.

There are many techniques to ensure awareness in a mobile collaborative system:

- **Poll:** The idea here is to periodically poll the server for new messages from the background. The more often the application poll, the most the application data will be in time. This kind of applications is very easy to implement, but obviously we will never be actually in real-time. If the interval polling is 10 minutes, we can get a message that is late of 9 minutes and 59 seconds. Polling more often than every 15 min will kill the battery life pretty quickly.



Figure 2. Software Architecture

- **SMS:** Android allows interception of SMS messages. The server sends a special SMS to our phone, whenever there is something new. The application intercepts all messages, looks for the ones from the server, and then displays the notification. This idea is easy to implement, and the system reminds fully real-time updated. The problem with this solution is that it costs a lot, especially if the notification number is important.
- **Persistent TCP/IP:** The phone initiates a long-lived mostly idle TCP/IP connection with the server and maintains it by occasionally sending keep-alive messages. Whenever there is something new on the server, it sends a message to the phone over the TCP connection. This technique provides a fully real-time updates. The Android OS is known to be able to kill services when it's running low on memory, so the notifications service can easily disappear. Another problem of battery life can appear that is related to maintaining an active connection.
- **Android Cloud to Device Messaging (C2DM):** C2DM² is a Google service that helps developers sending data from servers to their applications on Android devices. In the C2DM architecture, three parts are involved: the android application on an Android device. This latter must has at least an Android OS 2.2 version (in 2012 40.5% of Android devices have an OS version above than 2.2 [17]), has Android Market installed, and it must have at least one logged in Google account. An application server that developers set up as part of implementing C2DM in their applications. This part sends data to an Android application on the device via the

² <https://developers.google.com/android/c2dm/>

C2DM server. And finally the Google C2DM servers that are involved in taking messages from the application server and sending them to the device. The message size limit is 1024 bytes, and Google limits the number of messages a sender sends in aggregate, and the number of messages a sender sends to a specific device.



Figure 3. Notification of an event

The technique we need to develop a synchronous and real time collaborative system is called Push. The server contacts the mobile application once new data is available (notification), without pre-sending of a request from the application for an update which optimizes to minimum bandwidth and ensures real time update in the collaborative workspace. Our system MCPlan combines the SMS solution and persistent TCP/IP to produce a relatively stable way of implementing push.

In our approach an SMS is sent only to inform a participant about a new event (in general when the schedule is changed; the system notifies all the concerned persons to keep them constantly and automatically informed). The server extracts the required information from the database and uses an SMS gateway to send messages to the staff members.

When a participant logs in the workspace, an android application that uses TCP/IP protocol takes over the responsibility of notifications and ensuring real time awareness. When a participant add, delete, edit a task in the schedule, this modification appears on the devices of

all participants connected to MCPlan in real time, persons who are not connected and involved in a new task for example, are notified by an SMS as we mentioned it before.

The second part defines two servers: Mobile Server and Displaying-Server [18]. The mobile server contains the server that notifies and sends (push) data to mobile devices. The displaying server is intended to display the schedule on eWhiteBoards (screens disposed on the appropriate locations in the hospital). After every modification of the schedule, all the eWhiteBoards are automatically updated. Also to significantly reduce users' cognitive overload such as nurses, surgeon, the eWhiteBoard(s) can be configured to restrict the display only for pertinent information needed by each group [1] and decrease the amount of data on screens. Finally, the third part consists of the client devices.

The developed system allows an authorized physician to access at any location to the electronic patient record data, using a hand held device or even a desktop. Another version of the system for desktops has been developed for users who prefer working on big screens, or who are already in front of a computer. Using the mobile version, allow direct access of participants to consult, or modify the schedule.

5. Software Components

The components shown on the Figure 4 are launched in the mobile device. Our architecture is composed of several modules which are important for the collaborative scheduling task:

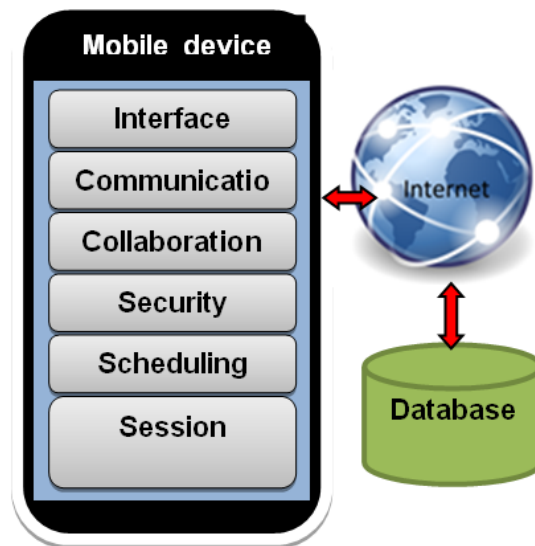


Figure 4. Components of the developed mobile architecture

- **Interface module:** this module plays the role of a medium between the user and the system. The Interface module intercepts the local users' actions and sends them to the server via internet.
- **Scheduling module:** This module is the most important component; it provides the necessary tools to manage tasks and resources. A list of tasks is displayed in a table to users with their related information (priority, starting time...). Once a task is created, it will appear in real time on the other users (participants) mobile screens. So the authorized participants can modify it while the others get the latest information on

their patients' state, and the cooperation process may naturally take place among the medical staff. When a user joins a work session a set of information is displayed on his screen, the shown information concerns only that user. After a creation or an update of a task, an SMS is sent to persons assigned to that task.

- **Collaboration module:** to allow users to simultaneously work together, we designed several appropriate tools, such as the tasks shared table. Users share the same display which instantly shows any event that may occur in consequence of a user action. Such way provides users with real time awareness capabilities and enhances coordination among them. Furthermore, we take into account the ergonomic considerations while implicitly informing users about: "*Who does what*" through coloration use. Thus when a user starts editing a task, this latter is coloured by his specific colour (assigned once a user is registered). So users can easily identify the task author but also locate him.
- **Communication module:** In order to coordinate their work, users must exchange their ideas. MCPlan supports both synchronous and asynchronous communications. Through asynchronous communication, users can post comments, new messages, express their opinions, give valuable suggestions to their colleagues. We wish through asynchronous communication improve the efficiency of the shared schedule. However, when two users or more are working at the same time, they use the instant messaging tool to communicate synchronously.
- **Collaborative diagnostic module:** Is a component intended for diagnostics elaboration of a given case under a collaborative way. Participants who are generally doctors can propose different diagnostics for a specific case and share their points of view with peers. Working under a collaborative way increases chances for faster diagnostic elaboration. The shared diagnostic is displayed as a tree expressing the whole doctors' points of view around the examined case.
- **Session Management module:** This component is intended to manage users' work sessions. During the registration of a user, the administrator can specifies his access rights on the schedule list (read/write). The Session Manager also manages users join/leave within the shared workspace, and any related event will be immediately notified the others such as latecomers...
- **Security module:** this module assures security of electronic medical records during collaboration process, by encrypting and decrypting data. We used elliptic curve cryptography technique [19, 20]. This technique is a very efficient asymmetric algorithm and it is more adequate to mobiles, in which energy and calculus capacities are limited.

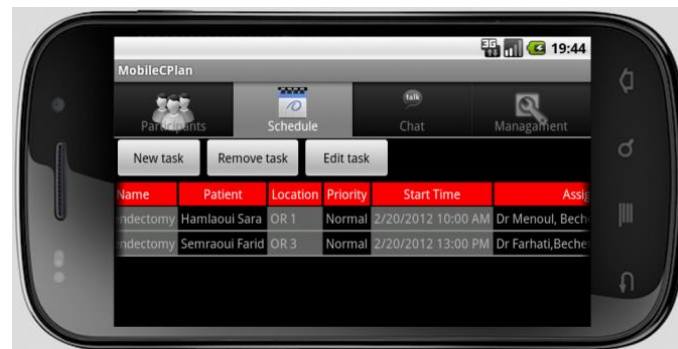
6. Implementation

Figure 5 shows a snapshot of the developed application. Part (a) shows the participant list, it is divided into 3 distinct areas, which explicitly shows the participants who are offline, the participants who are available in the current work session and can edit the schedule, and finally the participants who are connected and can only consult the schedule without modifying it. Part (b) of Figure 5, shows the collaborative task table. It is a shared workspace displaying the list of common tasks. Therefore users share the common view and those whom are authorized may act on it while the others are synchronously following the schedule

evolution. This prototype contains other functionalities such as a conversational tool, resource management tool.



(a)



(b)

Figure 5. A snapshot of the developed mobile groupware

7. Testing

The next step is to test our system in a real medical context which is the maternity clinic where our study is done, to have users' opinions and to improve the collaborative process in MCPlan. But before giving our application to users for tests, we made first another kind of test, to let users focus only on functionalities limitations and not on application bugs. For this purpose we used a set of tools available in the Android environment to make the following tests:

- **Activities test:** to test activities. JUnit is a tool integrated with the Android development environment. JUnit generate several class tests based on the developed

activities. We have first to identify activities of MCPlan, and generate a testing class for each activity.

- **Event test:** Monkey is a tool used for verifying GUI application. Monkey tool generates random or deterministic event sequences such as touches, clicks, or gestures, as well as another number of events. Monkey can be used also for stress-test applications.
- **Type tests:** type bug are exception that appear in a log file.

8. Discussion

Table 1, shows a comparison between our system MCPlan, and other techniques that can be used to develop a synchronous mobile collaborative system. We can notice that MCPlan has many positive aspects comparing to the other techniques. The other technologies like C2DM are not totally useless or bad, but they do not fit with our objectives:

- **C2DM:** Google developed a very interesting approach, but in a medical context we need some times to send a whole file to a device, such as a medical image. The size limitation of 1 Kilo bytes in C2DM hinders that.
- **Persisting TCP/IP:** using a permanent TCP/IP connection ensure real time awareness, but do not save battery life, which is a big problem in a medical context where the staff is generally busy and do not have time to recharge their devices batteries several times a day.
- **SMS:** using an SMS for each notification especially during a work session where there is a great number of modifications that need to be notified in real time, is not practical and costly.
- **Poll:** the polling technique does not ensure real time awareness, and do not save battery life for long time, so it is not useful in a real time context.

Our approach can be used to develop other mobile collaborative systems in medical field, or in other fields like collaborative learning.

Table 1. Comparison between MCPlan and the other techniques

	MCPlan	C2DM	Persistent TCP/IP	Poll	SMS
Saving Battery life	Yes	Yes	No	No (if the polling interval is small)	Yes
Costly	No	No	No	No	Yes
Works on all Android versions	Yes	No	Yes	Yes	Yes
Real time update	Yes	Yes	Yes	No	Yes
Message size limitations	No	Yes	No	No	Yes

9. Conclusion

This paper presents a study led in an Algeria maternity, to identify main coordination problems between medical staff members. In this study we found that coordination breakdowns can be solved by using the adequate communication technologies to improve coordination capabilities. We developed a mobile application which supports collaboration through Internet between several users.

To objectively measure the efficiency of MCPlan, we have developed and evaluate a first prototype. Testing the current version on a local network give rich ideas on the provided collaboration and coordination opportunities.

We have discussed basic design concepts of our groupware mobile application MCPlan. We have also showed that it enables collaboration between several participants within a shared workspace and allows users' individual and collective actions on a common patient case as the care planning elaboration.

Care scheduling sharing has been widely discussed; its efficiency is increased over transparency while our approach allows us to concretely inform participants on their mutual actions. At the visual level, the interface shows explicitly the shared plan phases and significantly reduces participants' cognitive view and enables them to intuitively understand what is currently going on and get knowledge about their patients' states evolution as well as the next actions that should be achieved.

During a work session, the medical staff members may act on the shared care plan under a specific role, dynamically exchange messages and interact through a natural cooperation way. Such flexibility is motivated by the necessity to enable MCPlan to support the dynamics implied by the care process.

Being conscious of the great interest of MCPlan experimentation in effective context situations, we plan in the next step of our research work to collect more information about our targeted context with the medical staff members. It is of an extreme importance for us and represents a double objective. First, we can validate or forsake some technical choices among those we made during the implementation. Second, we will be able to determine with more precisions the appropriated adaptations we should apply to the supports provided in MCPlan. To this end, such as any software project we designed modular and extendable software architecture, in the sense that it allows design and integration of new components through an incremental way.

Acknowledgements

This work is being carried out in the context of a project financed by the Algerian Ministry of the Higher Education and Scientific Research. We are grateful to all people who had brought their contribution to the first version of this paper.

References

- [1] Y. Ren, S. Kiesler, S. Fussell and P. Scupelli, "Multiple Group Coordination in Complex and Dynamic Task Environments: Interruptions, Coping Mechanisms, and Technology Recommendations", *Journal of Management Information Systems* / Summer, vol. 25, no. 1, (2008), pp. 105-130.

- [2] C. E. Kuziemsky and L. Varpio, "A Model of Awareness to Enhance Our Understanding of Interprofessional Collaborative Care Delivery and Health Information System Design to Support it", *International Journal of Medical Informatics*, doi:10.1016/j.ijmedinf.2011.01.009, forthcoming, (2011).
- [3] K. Schmidt and C. Simone, "Coordination mechanisms: Towards a conceptual foundation of CSCW systems design", *Journal of Computer Supported Cooperative Work: The Journal of Collaborative Computing*, vol. 5, no. 2-3, (1996), pp. 155-200.
- [4] F. Lezzar, A. Zidani and A. Chor, "A web application for supporting health care tasks with a groupware planning approach; International Conference on Information Technology and e-Services ICITeS', (2012).
- [5] F. J. Seagull, C. Plasters, Y. Xiao and C. F. Mackenzie, "Collaborative management of complex coordination systems: Operating room schedule coordination", *Proc. of HFES'03*, (2003).
- [6] J. Cummings and S. Kiesler, "Coordination and success in multidisciplinary scientific collaborations", *International Conference on Information Systems (ICIS)*, Seattle, WA: Association for Information Systems, (2003).
- [7] M. Zacklad, "Communities of Action: a Cognitive and Social Approach to the Design of CSCW Systems", *GROUP'03*, Sanibel Island, Florida, USA, (2003) November 9-12, pp.190-197.
- [8] A. Seffah, P. Forbrig and H. Javahery, "Multi-devices "multiple" user interfaces: development models and research opportunities", *J SystSoftw*, vol. 73, (n 2), (2004), pp. 287-3001.
- [9] S. R. Barley, W. H. Dutton, S. Kiesler, P. Resnick, R. E. Kraut and J. A. Yates, "Does CSCW Need Organization Theory ?", *Proceedings of the 2004 ACM conference on Computer supported cooperative work (CSCW'04)*, ACM Press, Chicago, Illinois, USA, (2004) November 6-10, pp. 122-124.
- [10] M. S. Reddy and P. R. Spence, "Collaborative information seeking: A field study of a multidisciplinary patient care team", *Information Processing and Management*, vol. 44, (2008), pp. 242-255.
- [11] Z. Niazkhani, H. Pirnejad, H. van der Sijs and J. Aarts, "Evaluating the medication process in the context of CPOE use: The significance of working around the system", *Int. J. Med. Inform.*, (2011).
- [12] P. Scupelli, Y. Xiao, S. R. Fussell, S. Kiesler and M. Gross, "Supporting coordination in surgical suites: Physical aspects of common information spaces", *Proceedings of the Conference on Human Factors in Computing Systems CHI10*, NY: ACM Press, (2010).
- [13] K. Schmidt and C. Simone, "Coordination mechanisms: Towards a conceptual foundation of CSCW systems design", *Journal of Computer Supported Cooperative Work: The Journal of Collaborative Computing*, vol. 5, no. 2-3, (1996), pp. 155-200.
- [14] J. E. Bardram and T. R. Hansen, "Context-based workplace awareness concepts and technologies for supporting distributed awareness in a hospital environment", *Computer Supported Cooperative Work*, vol. 19, (2010), pp. 105-138.
- [15] P. Dourish, "What we talk about when we talk about context", *Personal and Ubiquitous Computing*, vol. 8, no. 1, (2004), pp. 19-30.
- [16] Google's Android platform, <http://developer.android.com>.
- [17] <http://developer.android.com/about/dashboards/index.html>.
- [18] M. Hertzum, "Electronic emergency-department whiteboards: A study of clinicians' expectations and experiences", *I. J. Medical Informatics*, vol. 80, no. 9, (2011), pp. 618-630.
- [19] N. Koblitz, "A Family of Jacobians Suitable for Discrete Log Cryptosystems", In Shafi Goldwasser, editor, *Advances in Cryptology - Crypto '88*, vol. 403 of *Lecture Notes in Computer Science*, Berlin, (1988), pp. 94-99.
- [20] S. Nabil and B. Mohamed, "Security Ontology for Semantic SCADA", *ICWIT'2012*, (2012), pp. 179-192.

Authors



Fouzi Lezzar is a post graduate student for doctor degree in computer science of the university of Batna-Algeria. His research interests include electronic health and telemedicine.



Abdelmadjid Zidani is a professor of computer science specialized in CSCW field. Received his Ph.D degree in computer science from the University of Batna in Algeria. He is a professor in computer science since 2011.



Atef Chorfi is a post graduate student for doctor degree in computer science of the university of Batna-Algeria. He is interested in collaborative learning.