Journal of Universal Computer Science, vol. 25, no. 8 (2019), 988-1017
submitted: 26/12/18, accepted: 15/6/19, appeared: 28/8/19 © J.UCS

Identifying Groupware Requirements in People-Driven Mobile Collaborative Processes

Valeria Herskovic
(Department of Computer Science, Pontificia Universidad Católica de Chile, Santiago, Chile
vherskov@ing.puc.cl)

Sergio F. Ochoa, José A. Pino
(Department of Computer Science, Universidad de Chile, Santiago, Chile
sochoa@dcc.uchile.cl, jpino@dcc.uchile.cl)

Abstract: People-driven mobile collaborative processes are increasingly mediated by technology due to the ubiquity, efficiency and flexibility that modern groupware systems provide their users. However, identifying groupware requirements to be considered in their development is a challenging task, since the processes being supported by them do not have a clear workflow coordinating the activities performed by the participants. Thus, software developers must usually guess these requirements based on their own experience, and so the elicitation process becomes a creative activity instead of an engineering process. Trying to reduce this uncertainty about groupware requirements identification, and thus helping developers improve their capability to predict the suitability of a collaborative system, this paper presents a visual notation to represent user interaction scenarios through models. These models are processed to automatically determine a set of potentially required groupware services. Thus, this proposal reduces the uncertainty about the groupware requirements to be considered in the development of a system supporting a particular people-driven mobile collaborative process. The usability and usefulness of the visual notation and the method to derive the groupware requirements are illustrated with a running example, and also through its application to a case study. The results are encouraging and consistent, allowing us to augur potential adoption in research and industrial settings.

Keywords: People-driven mobile collaborative processes, groupware requirements, visual modeling notation, mobile ad hoc collaborative systems, software development.
Categories: H.5.3, D.2.1, J.3, J.7

1 Introduction

Before the year 2000, collaboration among people that bridged physical distances was thought to be "unlikely to succeed", except in specific situations [Bjorn 2016]. However, the increasing availability and affordability of mobile phones, wearable computers, and other mobile technologies have opened up opportunities to improve the support to activities conducted by mobile workers. This computer-mediated support allows users to become more autonomous and multitasking, and it also reduces the dependency of activities performed by them, opening several opportunities to assist people-driven mobile collaborative processes.

P*eople-driven mobile collaborative processes*, which have also been referred to as with low ceremony [Fowler 2000], collaborative ad-hoc [Huth 2001], loosely-coupled [Pinelle 2005], or ad-hoc mixed-focus [Kuhn 2018], are those in which there is no preestablished workflow coordinating the activities of the participants [Huth 2001, Antunes 2018]. Instead, mobile workers act autonomously and decide when to collaborate with others according to their own needs, interaction opportunities, process status and awareness of their work context [Huth 2001, Erickson 2016]. In the CSCW field, ethnographers researching single-sited studies have migrated to studying situations that transcend temporal and space constraints, including boundary crossings, and nomadic work [Blomberg 2013]. Examples of people-driven mobile collaborative processes are agile software development [Fowler 2000], emergency response [Rodriguez-Covili 2011], maintenance work [Antunes 2018], and computer-mediated social processes [Gutierrez 2017].

Scenarios like the one described above are complex, and several approaches have been proposed to help describe how collaboration or interaction actually happens. Qualitative studies may be used to understand complex scenarios, but their results may be lengthy and difficult to communicate among stakeholders. Other approaches require the collaborative workflow to be already well-known or with a system under use, to analyze generated data (e.g., log files [Conca 2018]).

The software systems supporting these processes are known as mobile ad hoc collaborative or socio-technical applications. Typically, the functionality of these systems (and also their functional requirements) is grouped in two layers: *business* and *collaboration support*. The first one is mainly represented by the user interface, which provides access to the business services pertaining to the application domain and the collaboration services. This latter functionality is known as *groupware services* (e.g., awareness, notifications, floor control) and they are implemented in the layers below the user interface.

It is difficult for developers to envision these groupware services during the first stages of the development process. On the one hand, most services do not have a direct representation on the user interface; so they cannot be identified using elicitation techniques such as rapid prototyping [Herskovic 2011]. On the other hand, people-driven processes do not have a workflow informing the requirements elicitation; particularly, the identification of potential collaboration instances among the participants that will require the support of the system. Thus, the developers must guess these requirements instead of eliciting them [Poltrock 2009]. The lack of an engineering approach to obtain the groupware requirements may lead to uncertain or non-contextualized collaborative systems. Moreover, the ad-hoc requirement elicitation processes are not replicable and dependent on the developers' skills.

Trying to make the identification of groupware requirements more predictable, as well as accessible to developers not necessarily experienced in mobile ad hoc collaborative applications, we propose the use of a visual notation allowing specifying interaction scenarios for participants in people-driven collaborative processes. The notation, named Mobile Collaboration Modelling (MCM), allows developers to model interaction scenarios that are represented by graphs specifying user roles and potential interactions among them. This work formalizes the MCM notation previously presented in [Herskovic 2009], also allowing the created models to be algorithmically analyzed to check their consistency. Additionally, this article presents a strategy for developers to derive a set of groupware requirements (collaboration services) [Herskovic 2011] potentially required by the participants in a particular collaborative process from the created graph. The models specified in MCM play a role similar to the workflows for structured processes, particularly, as input artifacts to derive potential user interaction requirements of a collaborative system.

Next section discusses previous work on collaborative process representation and modeling. Section 3 presents our strategy to analyze a mobile collaboration process. Section 4 describes the MCM notation, while Section 5 uses the proposed strategy to analyze hospital work. Section 6 presents the evaluation results of the proposal. Section 7 discusses the results, and finally, Section 8 presents the conclusions.

2 Related Work

Collaborative processes may be represented through an *ostensive* (how it is supposed to work) or *performative* view (how collaboration actually happens) [Feldman 2003]. Considering these perspectives, Poltrock and Handel [Poltrock 2009] studied the collaborative process structure, with the goal of deriving lessons for each type of view. For representing the ostensive view, we can use UML models, BPM models, coordination theory, and GOMS models, whereas for specifying the performative view we can use grounded theory, temporal models, social network models, and notation action perspective. Complementing these process perspectives, Huth et al. [Huth 2001] define types of collaborative processes based on their structure; these types range in a continuum from structured to ad hoc processes. In the former, the workflow (i.e., the underlying coordination of process activities) is clearly defined, and therefore these processes can be properly specified using an ostensive view. Contrarily, in ad hoc (or people-driven) collaborative processes there is not an underlying workflow to follow (even considering variability [Gottanka 2012]), therefore the structure of these processes only could be represented through a performative view. However, the number of alternative workflow paths in these processes is usually so high, that the information given by the performative views is not very useful to derive collaborative software requirements from them.

Typical mechanisms used to create a performative view are to examine log files, e.g., resulting in inferred interactions and other influences between participants, and building graphs that summarize patterns of interaction and mediation by digital artifacts [Suthers 2011]. These mechanisms are done a posteriori, allowing an analysis of how the collaboration actually took place, therefore they cannot be used to identify groupware requirements in green field development projects (i.e., projects conceived from scratch), but they do for conceiving or selecting systems that are going to replace an application that is currently into production.

There are other mechanisms that allow obtaining performative models; e.g., the influence model in which there are entities that have a state, and the entity's state is influenced by its network neighbors' states and changes accordingly. Each entity in the network has a specifically defined strength of influence over every other entity in the network. The influence model enables researchers to infer interactions and dynamics even when the network structure is unknown [Pan 2012]. An application example of the influence model is to infer functional roles in meetings [Dong 2007]. Although these mechanisms are useful for doing process mining, they do not help identify requirements in green field development projects due to the same previously explained reasons. The creation of models of the collaboration has also been used to communicate cooperative work practices within organizations, e.g., the DCR Portal modeling tool can be used to specify work practices by creating a graph of activities and their relationships [Christensen 2017]. However, this type of tool is only appropriate when the collaborative workflow is well known.

Due to these limitations to determine the structure of a people-driven collaborative process and then derive the groupware requirements, some initiatives report the use of observation mechanisms regularly used in social science, like ethnography or shadow studies [Gideon 2012]. However, the result of applying these mechanisms is a collection of unstructured information, which is not easy to process by developers to derive actual software requirements. A recent approach used the Model of Regulation theory as a descriptive tool for collaboration, finding it to also have potential as a prescriptive tool [Arciniegas-Mendez 2017], but without testing it as such. Trying to tackle the same challenge, the authors proposed the MCM notation (briefly introduced in [Herskovic 2009]). In this extended version, the paper proposes a way to analyze these collaboration scenarios and specify them through a graph. Moreover, the article also presents the foundation of the notation, its formalization, the way to use it and the evaluation results.

With the same purpose, Canche and Ochoa [Canche 2018] have recently proposed a new visual notation, named Computer-Supported Interaction Modeling Notation (CIMoN), which is based on MCM. Although promising, that notation is still a preliminary proposal.

Next section describes, using a running example, the phases recommended to identify the key aspects of a people-driven collaborative process until deriving the groupware requirements of the future system.

3 Analyzing a Mobile Ad Hoc Collaboration Scenario

Considering the related work and the authors' previous experiences and research, we conceived the mobile ad hoc collaboration process as a graph of interactions among people who play particular roles while performing specific activities. People can do two types of activities: *individual* or *collective,* as in mixed-focus collaboration, in which people "shift their attention continuously between group and individual activity" [Harr 2002]. Individual activities are those performed locally by the users (e.g., on their own mobile devices) without requiring interaction with other users or remote components, like a server or service provider. Collective activities cannot be accomplished without interaction among the actors (including remote resources). For example, in collaborative writing, users write asynchronously and then exchange the document, or use an online word processor such as Google Docs. In the first situation, users switch between writing (an individual activity) and sending the document when they require others' interaction (a collective activity). We do not consider individual activities in the specification of the interaction process, since we focus on the collective activities because they may help us determine the interaction support required by the roles in the collaborative process. This classification of activities is aligned with the definition of loosely-coupled mobile work [Pinelle 2005].

Fig. 1 shows the general process the developers must follow to derive the final set of groupware requirements, based on the observations made of the people-driven collaborative process they want to support with technology. The rectangles represent the activities that should be done to obtain the interim and final product (the products are shown with ellipses). The numbers and the dashed arrows indicate the suggested sequence. As stated in Section 2, the observation process can be done utilizing regular observation mechanisms used in social science, e.g. ethnography, shadow studies or a mix of several instruments. The key aspect here is to realize which aspects of the collaboration should be observed to create the MCM model using the MCM notation. After this activity, we propose an algorithm to validate the completeness and coherence of the MCM model, and if the model structure is correct, to generate an initial set of groupware requirements for the supporting system. This set is a list of suggested collaborative services potentially useful to support the activities of the participants; however, the algorithm may propose more than one service to support the interactions. Thus, the developers must analyze the suggestions and decide which of them will be implemented. After this review, it is possible to have a definitive list of groupware requirements, which will be used as input to the development process.

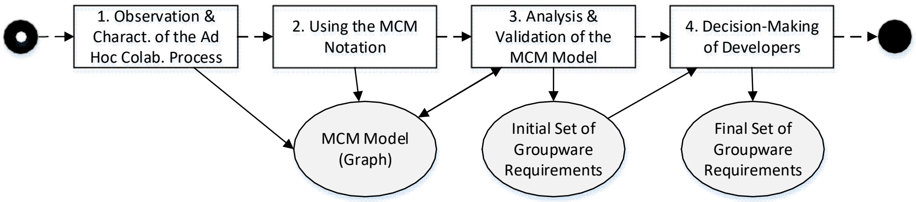


Figure 1: A strategy to derive groupware requirements from people-driven collaborative processes

Next subsections describe, using the urban search-and-rescue (USAR) process conducted in an emergency training scenario, the first activity of the process shown in Fig. 1. There, we describe the observation done on that mobile ad hoc collaborative process, and then we characterize the interaction scenarios involved in that setting. In Section 4, and after introducing the MCM nomenclature, we explain how to create the MCM model using the previous information. This model is the output of the first stage of the process. Based on it, the article shows how to derive the groupware requirements to be considered in the development of USAR supporting applications.

3.1 Observing the Ad Hoc Collaborative Process

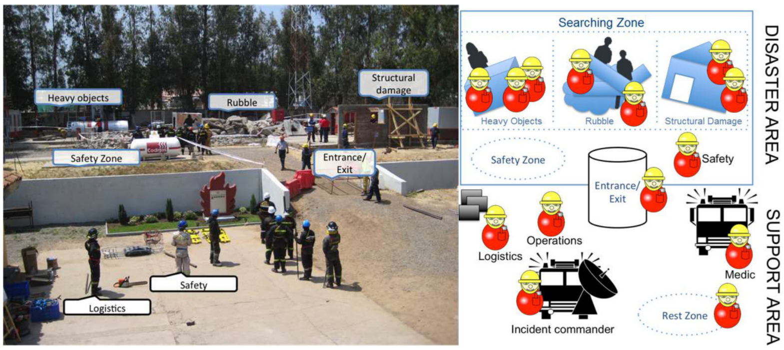
USAR processes are done by emergency response organizations (e.g., firefighters) with the goal of reducing the consequences of a catastrophic event, e.g. an earthquake, or terrorist attack, affecting an urban area. These processes differ from everyday firefighting operations in that they may require a larger number of workers, they are highly unpredictable and they last a long period. The focus of the process is to rescue trapped victims, while minimizing risks for victims and first responders.

In order to identify the interaction requirements that participants in an ad hoc collaborative process could have, it is required to understand and model it. The observation can be done by analyzing process traces, shadow studies or similar approaches. In our case we used a mixed approach that included ethnography and process trace analysis. The ethnographic work was conducted by an observer, who recorded the process to identify the main actors and interaction requirements among them, and interviewed the participants to confirm their roles and that of others.

Then, we confirmed the gathered information by analyzing the radio messages exchanged among participants, which allowed us to identify the main actors and the interactions required among them. Then, we created a detailed map of roles and interactions by combining the information from both sources. Studying this process is interesting since it considers the participation of several roles, several users per role, high mobility, high interaction needs among workers, and heterogeneous and dynamic work contexts. However, observing the collaborative work in a real USAR process is difficult: first, complex USAR processes happen sporadically, an authorization to participate as observer is needed, as well as capability to survey the work in several places of the affected area. Given these limitations, we observed the work performed by firefighters during a USAR training course, conducted in Peñalolén, Chile.

The training course lasted three days and finished with a simulation of a USAR process, conducted in a training field by mimicking the activities that first responders have to do after an earthquake. The instructors created a setting as realistic as possible, but keeping a balance between realism and control as recommended for urban computing research [Reilly 2009].

Several zones (or areas) were defined in the field to help coordinate the activities. Fig. 2(a) shows the zones in the training field, and Fig. 2(b) presents a conceptual diagram of such a place. Understanding the work scenario and the process that the rescuers conduct there, helps identify the people roles, their collaboration needs, and the contexts variables to consider in the design of the supporting applications.



(a) (b)

Figure 2: (a) Labeled picture of the training ground, (b) Diagram of the area

We observed two USAR processes as well as some shorter training exercises in this work scenario, and the participants were different groups of firefighters. Each USAR process lasted more than two hours, and the total observation time was around seven hours. The participants were firefighters with at least five years of experience.

An *incident commander*, who is located in the communication zone (next to the affected area), heads firefighters. This person gives orders to several teams, which have a leader and particular tasks to be done (e.g., searching, transporting victims, establishing an exit route). The incident commander interacts with the team leaders through an *operations officer*, who moves around the area trying to coordinate the teams. The *rescue team leaders* determine how to conduct team activities in the assigned zone (i.e., a portion of the searching zone); e.g., the heavy objects, the rubble or the structural damage.

There is a *safety officer* who leads a team that evaluates the safety conditions of the places where search and rescue should be done, defining the entry and evacuation routes, controlling the entry and exit of people from the area, and establishing safety zones. The *logistics officer* is located outside the affected area and close to an entry/exit point. This person manages the resources available to support the search and rescue process. There is also a first-aid zone (in the training course, this was a designated place to leave the rescued victims without further involvement). Therefore, the medical personnel were not considered in the training course.

Radio communication is typically used for communications with other leaders, and face-to-face communication is used within the teams. During the training exercises, we observed and recorded the activities and radio communication of seven participants (incident commander, operations officer, logistics officer, safety officer, and a leader for each search-and-rescue team) who had radios and performed the collaboration activities that could be supported with technology.

During the first USAR course, we observed and recorded a complete simulation, to understand how work and interactions were performed in the field. During the second course, we observed another complete simulation, recording all radio messages, coding them to study times, the involved people, their roles and the purpose of the messages. A total of 934 radio messages were exchanged; six messages were explicitly sent to several recipients trying to do a kind of broadcast. The remaining messages were direct interactions between one sender and one recipient, although any person with a radio device could listen to them.

We found that, even in this synchronous and co-located collaboration, there were definite patterns to the collaboration related to the roles of each team member; e.g., rescue team leaders did not communicate among themselves and only communicated with the operations officer. Almost all communication was done between two roles and did not involve others. As expected, the operations officer was the person with highest radio use, since he/she must organize and coordinate the teams using the radio.

3.2 Characterizing the Interaction Scenarios Among Participants

The work style of the teams was loosely coupled, i.e., the firemen work autonomously most of the time (individual activities) and carry out sporadic on-demand interactions through their leaders (collective activities) in order to exchange/report information. The work of a few participants was tightly coupled; for instance, the activities performed by the incident commander and the operations officer, and between the work of this latter one and the safety leader activities. Although these leaders must be able to find the required collaborators and communicate with them when needed, the interaction support is in higher demand between people doing tightly coupled work than between those doing loosely coupled activities.

Considering these situations, we defined two perspectives (or dimensions) to classify the interaction scenarios between two roles. These dimensions are *simultaneity* and *reachability*, which are related to the potential for interaction between two participants. These dimensions were introduced in [Herskovic 2009] and we explain them below.

Simultaneity between two actors occurs in time periods when they are potentially available to interact with each other, i.e., they are working at the same time. However, one of them may be working on an urgent task and not available to be interrupted, so they may choose to interact through synchronous or asynchronous technology. When two actors are non-simultaneous, they can just interact asynchronously.

Reachability between two actors is defined as the availability of an interaction channel (physical or virtual) between them, combined with their availability to interact with each other. Two reachable actors are able to communicate in a highly predictable way, i.e., a message sender can expect a reply within a given period [Grudin 1994]. For instance, two actors may be reachable through e-mail, with the expectation of receiving an answer within a few days, or by WhatsApp, with the expectation of receiving an answer within a few hours or even within a few minutes. If an actor is unavailable or if there is no communication channel between them, then they are unreachable. The features of the physical environment where the collaboration takes place and the communication media used to support the interactions directly influence the reachability of the collaborators; i.e., if the collaboration scenario or the supporting technology changes, the reachability of the participants could also change.

The definition of these dimensions (i.e., *simultaneity* and *reachability*) implies that at a given moment, whenever two actors need to interact, they will be in one out of four states (or quadrants): simultaneous/reachable (SR), simultaneous/unreachable (SU), non-simultaneous/reachable (NR) and non-simultaneous/unreachable (NU), as shown in Fig. 3. Identifying in which quadrant the interaction is going to be performed is mandatory for software developers, since the interaction support to be provided to the end-users will depend on it.

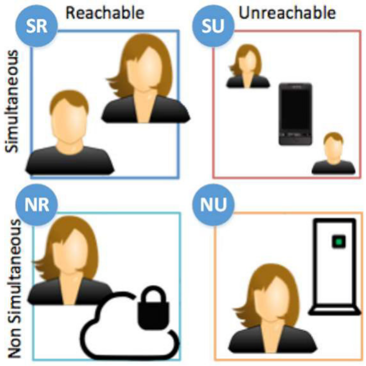


Figure 3: Classification of interaction scenarios

In the ***SR*** scenario, two actors work at the same time and may interact directly. If they are willing, they will be able to communicate and work together during one unbroken period. We can expect the communication support be highly predictable in this interaction scenario. This scenario includes face-to-face and remote interactions through a network.

Actors in the ***SU*** scenario work synchronously, but they are unreachable (i.e., without a communication link between them), and thus they cannot interact in a predictable way. For instance, in a scenario with no network infrastructure, two users may be working within the same building on related tasks, but in an area uncovered by the communication range of their devices. In this case, actors could choose to collaborate by leaving files on a shared intermediary device. Even if there is communication service, one of the collaborators may be unavailable (unreachable).

The ***NR*** scenario considers actors working in different periods, but there is an infrastructure allowing them to communicate asynchronously. For example, two users working in different shifts may use an intermediary resource to leave notes or messages (e.g., a cloud server, as in Fig. 3, or a message board) to make the other person aware of some situation. Both users are aware of this resource, therefore the first user knows the other user will receive the information they will share, but with some delay.

Finally, the ***NU*** scenario represents interactions between actors that work asynchronously, and there is no permanent intermediary allowing them to communicate asynchronously. This scenario is infrequent, but it may happen, e.g., when collaboration is being established or between actors who are unpredictable in their communication. The work in this scenario is practically autonomous; however, provision of technological support may promote the collaboration process.

4 The Mobile Collaboration Modeling Notation

Considering the previous analyses as well as the design decisions made to represent these interaction scenarios, we conceived the MCM notation based on a directed graph, in which the nodes represent the roles of the actors participating in the people-driven mobile collaboration process, and the arcs are labeled with a notation that illustrates the interaction needs among them. We named these representations as Role Interaction (RI) graphs, and they represent models of the interaction scenarios potentially present in the process to be supported with technology.

The nodes of the RI graphs may be active users (i.e., people) or intermediaries in the interaction scenario (Fig. 4 – left side). Intermediaries are non-human participants that enable others to interact, e.g., a server allowing users to synchronize the results of their activities. An arc represents interaction between two roles. A directed arc from A to B indicates that role A may send information to role B (Fig. 4 – right side); if no interaction is considered then there will be no line between them. Although the direction of each arc captures relevant information about the interaction process, in the analysis we consider the graphs as undirected, since one interaction requirement in each direction is enough to consider this support for both roles.

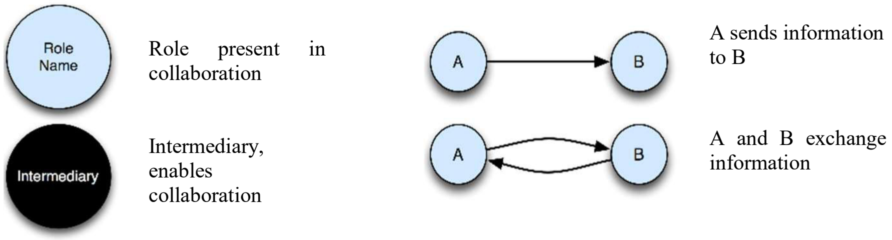


Figure 4: Elements of the MCM notation: nodes (left) and arcs (right)

The label of each arc represents the possible scenarios in which interaction would take place. The scenarios correspond to the previously described classification scheme. Each arc has a simple label (SR, SU, NR or NU), as represented in Table 1, or a composite label, which is an OR composition of simple labels and it is represented by a square with the corresponding colored quadrants. For instance, a representation showing a colored top half means a SR or SU composite label; i.e., whenever one of the actors requires to interact with the other, their relationship will be simultaneous and either reachable or unreachable. Considering these combinations, there are 15 possible labels to be used on an arc. In order to characterize the arcs, we have to consider the physical environment where the interactions take place and the technology that will support these interactions.

Label	Meaning	Symbol	Label	Meaning	Symbol
SR	Simultaneous Reachable	▘	NR	Non-simultaneous Reachable	▖
SU	Simultaneous Unreachable	▝	NU	Non-simultaneous Unreachable	▗

Table 1: Label representations - simple labels (from [Herskovic 2011])

In order to illustrate the specification of a RI graph using the proposed notation, next we show the graph defined for the USAR exercise described in section 3 (Fig. 5). Although this graph is simple, it conveys important knowledge about the collaboration, e.g., it identifies all roles that need interaction support of an eventual collaborative system, distinguishing which roles receive and send information.

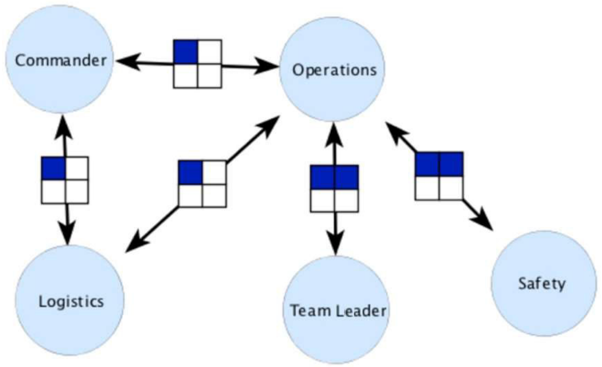


Figure 5: RI graph for the USAR scenario (from [Herskovic 2011])

Given the hierarchical structure of firefighters' interactions, e.g., a commander will not communicate directly with rescue team leaders, only through an operations officer. In this scenario, all actors work simultaneously; but at some point in time, some users (e.g., the safety officer) may become unreachable for short time periods, therefore, the supporting applications must consider synchronous and asynchronous interaction services depending on people's reachability. This may affect the design of the system, because mechanisms to provide connection flexibility and offline awareness should be embedded in the supporting system. This example shows how RI graphs can be used to inform the design of applications that support a particular collaboration process. Provided the RI graph represents the interaction capabilities and limitations among the participants in the collaboration process, system developers can identify requirements (i.e., services to be provided) and potential interaction problems or limitations that should be addressed in the design of these applications.

4.1 Analysis of the Roles Interaction Graph

This section shows how to analyze RI graphs to determine their consistency from an interaction capability point of view, which is part of the second stage of the process indicated in Fig. 1. We consider three cases to do that: (1) graphs consisting of two nodes, (2) graphs containing cycles of exactly 3 nodes, and (3) graphs with cycles of more than 3 nodes. Any graph can be decomposed using these interaction structures. Moreover, we consider relationships between two nodes (i.e., 1-1 relationships) and not relationships between a node and any number of nodes (i.e., 1-m relationships), since we may instead choose to view this last situation as m 1-1 relationships.

When two roles r_1 and r_2 interact, they do so in a given collaboration scenario (i.e., a quadrant of the matrix – Fig. 3). The interaction scenarios that two roles pass through while one of them is active are called the possible interaction scenarios between roles r_i and r_j. We denote as $s_{i,j}$ (Fig. 6(left)) to the interaction scenario that is present when two roles decide to collaborate. Such a scenario can be characterized as explained in section 3.2 and represented using MCM.

When three roles are involved in collaboration, we may analyze the specification of the interaction scenarios to ensure consistency among them. In Fig. 6(right), roles r_1, r_2 and r_3 interact, forming the cycle $r_1r_2r_3r_1$. We call this a *triangle* and denote it as r_{123}.

If we know $s_{1,2}$ and $s_{1,3}$ then we may infer some information about the scenario governing the interactions between r_1 and r_3.

Lemma 1. *Considering triangle* r_{123}*: if both* $s_{1,2}$ *and* $s_{2,3}$ *are simultaneous, then* $s_{1,3}$ *is also simultaneous.*

Proof. If r_1 and r_2 work simultaneously, then during <the time in which r_1 is active, r_2 is also active. If r_2 and r_3 are simultaneous when r_2 is active, then r_3 is also active, i.e., while r_2 is active, both r_1 and r_3 must also be active. Therefore, $s_{1,3}$ is simultaneous.

Figure 6: (left) Interactions between two roles, and (right) among three roles

Lemma 2. *Considering the triangle* r_{123}*: if* $s_{1,2}$ *is simultaneous, and* $s_{2,3}$ *is non-simultaneous, then* $s_{1,3}$ *is non-simultaneous.*

Proof. Since r_1 and r_2 are simultaneous, if r_1 and r_3 were simultaneous, through Lemma 1 we would conclude that r_2 and r_3 must also be simultaneous. This is a contradiction because r_2 and r_3 are non-simultaneous. Therefore, r_1 and r_3 must also be non-simultaneous.

Lemma 3. *Considering the triangle* r_{123}*: if* $s_{1,2}$ *is non-simultaneous, and* $s_{2,3}$ *is non-simultaneous, then* $s_{1,3}$ *may either be simultaneous or non-simultaneous.*

Proof. If r_1 and r_2 are non-simultaneous, then both are active at different times. The same is true for r_2 and r_3. There are two possibilities for r_1 and r_3: they may be simultaneous, i.e. they work at the same time, but at a different time from r_2, or they may be non-simultaneous, which means r_1, r_2 and r_3 all work at different times.

Lemma 4. *Considering the triangle* r_{123}*: if both* $s_{1,2}$ *and* $s_{2,3}$ *are reachable, then* $s_{1,3}$ *is also a reachable connection.*

Proof. If $s_{1,2}$ is reachable, then r_1 may communicate with r_2 in a highly predictable way. The same is true for r_2 and r_3. Therefore, r_1 and r_3 are available and there is a communication channel that can connect them, possibly through r_2.

Lemma 5. *Considering the triangle* r_{123}*: if* $s_{1,2}$ *is reachable, and* $s_{2,3}$ *is unreachable, then* $s_{1,3}$ *is unreachable.*

Proof. If $s_{1,2}$ is reachable, then r_1 may communicate in a predictable way with r_2. If $s_{1,3}$ were reachable, because of Lemma 4, then $s_{2,3}$ would be reachable. Since $s_{2,3}$ is an unreachable connection, $s_{1,3}$ must be unreachable as well.

Lemma 6. *Considering the triangle* r_{123}*: if* $s_{1,2}$ *is unreachable, and* $s_{2,3}$ *is unreachable, then* $s_{1,3}$ *may be either reachable or unreachable.*

Proof. If both $s_{1,2}$ and $s_{2,3}$ are unreachable, then there are two possibilities: $s_{1,3}$ may be reachable, e.g., in case r_2 is unavailable but r_1 and r_3 are not, or $s_{1,3}$ may be unreachable e.g., if no communication channel exists.

Lemma 7. *Considering the triangle* r_{123}*: we may infer the possible scenarios of interaction* $s_{1,3}$ *if* $s_{1,2}$ *and* $s_{2,3}$ *are known and each consists of only one possible scenario* (Table 2).

Proof. Each scenario $s_{1,2}$ and $s_{2,3}$ consists of two dimensions: reachability and simultaneity. Given this and the fact that $s_{1,2}$ and $s_{2,3}$ must consist of only one interaction scenario, then $s_{1,2}$ and $s_{2,3}$ each have four possible configurations. Thus, we have ten distinct combinations of $s_{1,2}$ and $s_{2,3}$. We use lemmas 1–6 to find $s_{1,3}$.

Theorem 1 (Triangle Inference). *Considering the triangle* r_{123}*: we may infer the possible scenarios of interaction between* r_1 *and* r_3 *if* $s_{1,2}$ *and* $s_{2,3}$ *are known.*

Proof. There are two cases. First, if $s_{1,2}$ and $s_{2,3}$ each consists of only one possible interaction scenario, then $s_{1,3}$ is determined by Lemma 7. Otherwise, $s_{1,2}$ and/or $s_{2,3}$ consist of several possible interaction scenarios. In this case, at a given point in time r_1 and r_2 may be at any scenario of the set $s_{1,2}$, and r_2 and r_3 may be at any scenario of the set $s_{2,3}$. We may find all possible scenarios by combining all single interaction scenarios of $s_{1,2}$ and all single interaction scenarios of $s_{2,3}$. This results in a maximum of 10 combinations of two scenarios (the case where $s_{1,2}$ and $s_{2,3}$ each include all 4 interaction scenarios). The resulting possible interaction scenarios for each combination is the union of all possible $s_{1,3}$, determined by Lemma 7.

When several roles are present in a collaboration scenario, we may study the scenarios of interaction in cycles of length greater than 3 to ensure consistency. In a cycle of length N, if there is a path $r_1r_2r_3...r_{N-1}r_Nr_1$, if we know the interaction scenarios of the first N-1 arcs, then we may infer some information about the interaction scenarios of arc r_Nr_1.

Lemma 8. *Assume there is a path* $r_1r_2r_3...r_{N-1}r_N$ *in which* $s_{i,i+1}$ *for i* ∈ *(1,...,N-1) arc is simultaneous. If arc* r_Nr_1 *exists, then it must also be simultaneous.*

Proof. If r_1 is simultaneous with r_2, and r_2 is simultaneous with r_3, then r_1 must be simultaneous with r_3. Then, if we added arc $s_{1,3}$, it would be simultaneous. Since $s_{3,4}$ is simultaneous, if we added $s_{1,4}$, it would also be simultaneous. By induction on i ∈ (2,...,N), each r_i will be simultaneous to r_1. So finally, $s_{1,N}$ (which is equivalent to $s_{N,1}$) will also be simultaneous.

Lemma 9. *Assume there is a path* $r_1r_2r_3...r_{N-1}r_N$ *in which* $s_{i,i+1}$ *for i* ∈ *(1,...,N-1) arc is reachable. If arc* r_Nr_1 *exists, then it must also be reachable.*

N	$s_{1,2}$	$s_{2,3}$	⇒	$s_{1,3}$	**N**	$s_{1,2}$	$s_{2,3}$	⇒	$s_{1,3}$
1	▘	▘	⇒	▘	6	▝	▖	⇒	▗
2	▘	▝	⇒	▝	7	▝	▗	⇒	▄
3	▘	▖	⇒	▖	8	▖	▖	⇒	▌
4	▘	▗	⇒	▗	9	▖	▗	⇒	▐
5	▝	▝	⇒	▀	10	▗	▗	⇒	█

Table 2: Consistency rules

Proof. If $s_{1,2}$ and $s_{2,3}$ are reachable, then r_3 is reachable from r_1. Since $s_{3,4}$ is reachable, then if we added $s_{1,4}$, it would also be reachable. By induction on i ∈ (2,...,N), each r_i will be reachable from r_1. Finally, $s_{1,N}$ (which is equivalent to $s_{N,1}$) will be reachable.

Theorem 2. ***(Cycle Inference)*** *Considering a cycle $r_1r_2r_3...r_{N-1}r_Nr_1$, we may infer the possible scenarios of interaction $s_{N,1}$ in some cases if $s_{i,i+1}$ (i ∈ (1,...,N-1)) are known. The cases where $s_{N,1}$ may be inferred are shown in Table 3; they are the following:*

- ***C1:*** *If all $s_{i,i+1}$ are simultaneous and reachable, then $s_{N,1}$ will also be simultaneous and reachable.*
- ***C2:*** *If all $s_{i,i+1}$ are simultaneous, then $s_{N,1}$ will also be simultaneous.*
- ***C3:*** *If all $s_{i,i+1}$ are reachable, then $s_{N,1}$ will also be reachable.*

Proof. Each scenario $s_{i,i+1}$ has two dimensions: reachability and simultaneity. For rule C1, given Lemmas 2 and 3, if all $s_{i,i+1}$ (i ∈ (1,...,N-1)) are simultaneous and reachable, then scenario $s_{1,N}$ must also be simultaneous and reachable. For C2, if all $s_{i,i+1}$ (i ∈ (1,...,N-1)) are simultaneous, then $s_{1,N}$ must also be simultaneous, but if some $s_{i,i+1}$ are unreachable, then $s_{1,N}$ may be reachable or unreachable. Finally, for C3, we know if all $s_{i,i+1}$ (i ∈ (1,...,N-1)) are reachable, $s_{1,N}$ must also be reachable, but if some $s_{i,i+1}$ are non-simultaneous, $s_{1,N}$ may be simultaneous or non-simultaneous.

C	**∀ $s_{i,i+1}$ ∧ i ∈ (1, N-1)**	⇒	$s_{N,1}$
1	▘	⇒	▘
2	▘ or ▝	⇒	▀
3	▘ or ▖	⇒	▌

Table 3: Rules for n-nodes cycle

4.2 Validation of the Role Interaction Graphs

The analysis rules may be applied to complex RI graphs (i.e., involving many nodes and arcs) representing real-life situations. The analysis of these graphs allows software developers to identify inconsistent cycles that create ambiguity about the interaction scenarios that should be supported. This analysis does not pinpoint exactly the erroneous interaction scenarios; this is because if, e.g., we have a triangle $r_{1,2,3}$, we check if we may infer $s_{1,3}$ from $s_{1,2}$ and $s_{2,3}$ (and then do the same for $s_{1,2}$ and $s_{2,3}$). If we may not infer $s_{1,3}$ from $s_{1,2}$ and $s_{2,3}$, then it may be due to a mistake in $s_{1,3}$ or a mistake in one of the other two scenarios. Therefore, the algorithm can identify problematic cycles, but it will be up to the user to analyze the precise error and correct it. The complexity of this problem is O((N+E)C) – the complexity of finding all cycles in a graph [Mateti 1976], where N is the number of nodes, E the number of arcs, and C the number of cycles. The rest of the algorithm consists of checking the consistency rules (at most 10) for each cycle, so this only adds O(C) to the overall complexity. Once the consistency of the graph has been verified, it is possible to generate the requirements list based on the proposal presented in [Herskovic 2011].

4.3 Deriving the Groupware Requirements

The set of groupware requirements that could be used to support interaction in each quadrant of the matrix (Fig. 3) are briefly described in Table 4 (adapted from [Herskovic 2011]). Table 5 shows which services (from those indicated in Table 4) can be used to support the interaction in each quadrant. The row and column labeled as "transition" indicate the services that can be used to generate a transition between two interaction scenarios; typically from one where the people are unreachable to one where they are reachable. This configuration of services is used to provide suggestions to developers based on the analysis of each particular RI graph.

Considering these requirements and the interaction needs of each role, the identification of the potentially useful collaboration services for each pair of roles becomes simple. In order to do so, it is only necessary to analyze the link between a role and other roles in the RI graph. Although the set of groupware requirements are automatically suggested after analyzing the RI graph, the developer is in charge of deciding whether or not a certain requirement should be kept in the requirement list. Next section describes the supporting tool and also the decision-making process conducted by the developers to obtain the final set of groupware requirements.

4.4 Supporting Tools for Developers

Fig. 7 shows the main interface of the Graph Modeling Tool (GMT), which is a software application that allows developers to create and edit RI graphs (Fig. 7-iii) by instantiating roles (Fig. 7-i) and relationships between them (Fig. 7-ii). The tool allows automatic validation of the consistency of the graphs as explained in Section 4.2. As previously mentioned, it is not possible to diagnose the exact problem with the graph, but only its inconsistent arcs. Then, it is up to the developer to review and correct the inconsistency.

Once the graph is made consistent, this tool uses the defined RI graph to display, for each role, the list of requirements for supporting its interactions (Fig. 8). These requirements are obtained automatically from those in Table 5. In case of Fig. 8, we can see the requirements suggested by the system to support the interactions between users playing the roles of "team leaders" and "operations" officers. The last column of the figure shows the requirements that developers decide to keep, thus indicating services that should be embedded in the collaborative system supporting interactions between these two roles. Using the upper menu, the developers can change the pair of roles to be analyzed. Once all pairs of roles indicated in the RI graph have been analyzed, the developers have an initial set of groupware requirements to be considered in the conception of the system. The same list of requirements can be used to determine if the services provided by an already implemented tool are suitable to assist the collaborative process represented by the graph.

#	Requirement	Description
1	Automatic Peer Detection	Automatically keep information about reachable users and availability.
2	Explicit Data Replication	Share data with other users during connection periods.
3	Caching	Replicate data in collaborators' workspace to keep it up-to-date.
4	Conflict Resolution	Help solve inconsistencies in shared data when two collaborators update files autonomously.
5	Automatic Connection	The network should be automatically formed (either through wireless internet services or mobile ad-hoc networks).
6	Synchronous Messaging	Exchange messages between simultaneously available users.
7	File Transfer	Provide access to a user to some of the shared files in a workspace.
8	Online Awareness	Online awareness mechanisms, e.g., connected users, locations.
9	Ad hoc work sessions	Provide private work sessions between users.
10	Connection/disconnection	Switch between offline and online use on-demand.
11	Message Routing	Use intermediary mobile workers to provide reachability between actors with more than one-hop distance between them.
12	User Gossip	Messaging unreachable partners through intermediary neighbor nodes, whose movement eventually may allow message delivery.
13	Transition Awareness	Awareness about transitions between connection and disconnection.
14	Offline awareness	Offline awareness mechanisms, e.g., last modification to a document, text authorship.
15	Asynchronous Messaging	Send messages that the other user will receive when available.
16	Push Notifications	Deliver messages with push notifications when users become available.

Table 4: Groupware requirements description (adapted from [Herskovic 2011])

The list of requirements for each role in the USAR case study was then checked with people who actually play such a role. Particularly, two focus groups were done at a firefighting company in Ñuñoa, Chile, and they lasted between 90 and 120 minutes. Two authors and 6-8 firefighters participated in each session. All of them regularly play (or know how to play) the roles considered in the RI graph.

	Reachable	**Transition**	**Unreachable**
Simultaneous	1, 2, 3, 4, 5, 6, 7, 8, 9	1, 3, 10, 11, 12, 13	11, 12, 14
Transition	1, 3, 4, 10, 12, 13, 15		12, 14
Non Simultaneous	2, 4, 12, 14, 15, 16	12, 14	14

Table 5: Groupware requirements and their scenarios

Session dynamics were as follows: the authors used slides to present three situations (all of them considered in the RI graph) that require interactions among several roles, and then asked the participants to indicate how to address these interactions. After a short exchange of ideas (5-10 minutes) the authors presented, using visual examples, the services that a system could provide to each role according to the GMT tool. Besides the analysis of each suggested service, the participants indicated if such a suggestion was appropriate, inappropriate or neutral.

This checking showed that all suggested requirements were appropriate; however, some of them were more often needed than others. In this case, the developers should prioritize (and discard) services by checking with end-users. Following a similar procedure, next section describes a hospital work collaboration scenario.

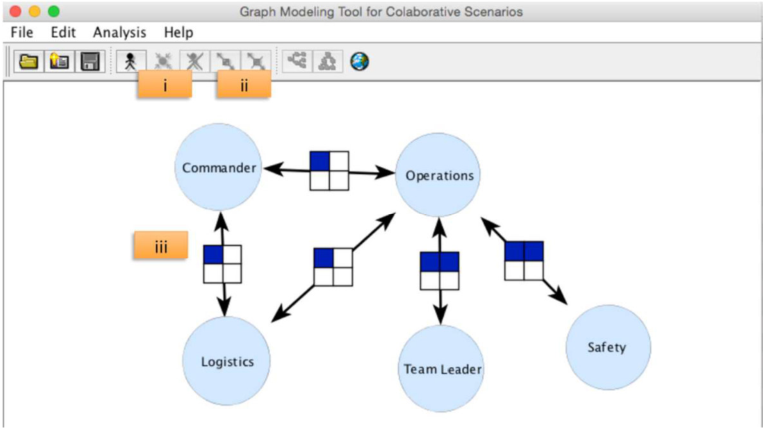


Figure 7: GMT main user interface

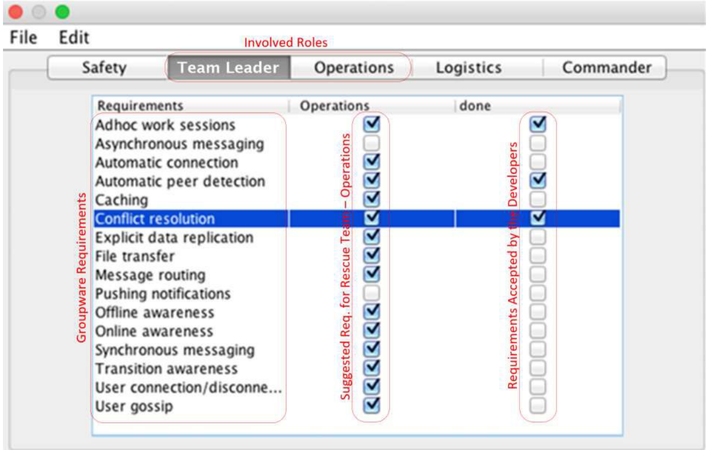
5 Case Study: Hospital Work

Hospitals are settings in which teams of healthcare professionals must collaborate to address medical problems and provide care to patients. The CSCW field has studied collaboration in healthcare for over 30 years, including coordination and mobility as key issues [Fitzpatrick 2012]. This section describes the setting and characteristics of hospital collaborative work and shows the interactions model for this scenario.

5.1 Observing Hospital Work Activities

We first reviewed research on qualitative studies done in hospital settings. We summarize some of the findings below. Then, we conducted a semi-structured interview and two focus groups to learn about mobile work in hospitals. In these focus groups, several health professionals remarked that the problems of unavailability and locating workers are important and frequent limitations to perform collaboration.

Hospital work shows need for coordination, task switching, information exchange, data integration and a high rate of mobility in terms of staff, patients, and resources [Bossen 2002]. A shadow study was done in which ten physicians were observed for two work shifts (capturing all interactions). This study indicates that these hospital workers spend 53% of their time outside their base location [Moran 2007]. These workers do not only use face-to-face interactions, but also co-located and distributed work [Bardram 2005]. Health professionals working in sequential shifts leave instructions in artifacts, such as medical records, to share information and coordinate activities with co-workers. These records are not just artifacts, rather, they are a central part of collaborative work and patients care [Fitzpatrick 2012]. The observation of these activities is the first step considered in the process described in Section 3.



Requirements	Operations	done
Adhoc work sessions	☑	☑
Asynchronous messaging	☐	☐
Automatic connection	☑	☐
Automatic peer detection	☑	☑
Caching	☑	☐
Conflict resolution	☑	☑
Explicit data replication	☑	☐
File transfer	☑	☐
Message routing	☑	☐
Pushing notifications	☐	☐
Offline awareness	☑	☐
Online awareness	☑	☐
Synchronous messaging	☑	☐
Transition awareness	☑	☐
User connection/disconne...	☑	☐
User gossip	☑	☐

Figure 8: Functional requirements obtained from the RI graph shown in Fig. 7

5.2 Characterizing the Interaction Scenarios

Once the dynamics of hospital work activities were understood, we identified the participating roles as a way to characterize the interaction scenarios. The main roles to be supported were physicians, specialists, medical interns and nurses. We distinguish between physicians, specialists and interns since hospital work is partly hierarchical: specialists are reluctant to be located by interns, but interns accept being located as part of their work. Workers perform their activities in shifts; Table 6 describes information flows among roles. We note nurses and interns are only able to exchange information among them and provide information to physicians and specialists to acknowledge their privacy requirements.

From/To	Physician	Specialist	Intern	Nurse
Physician	■	■	■	■
Specialist	■	■	■	■
Intern			■	■
Nurse			■	■

Table 6: Information flows description

These RI graphs provide a view of the interactions that exist in a collaboration scenario by identifying the roles and the scenarios in which they interact. All users work non-simultaneously at some point in this scenario. This is because they work in shifts, therefore e.g. a nurse leaving their shift may require asynchronous communication with the nurse beginning the next shift. This is currently done through medical records, but a system could include support for this activity. Users become unreachable in a hospital because they may be unavailable (e.g., busy with an emergency surgery). In contrast, firefighters doing USAR operations usually become unreachable because they lack an available communication channel.

It is also important to note that the physician becomes a sort of hub in this collaboration, as they are the organizer of patient care. The collaborators (e.g., nurses and interns) provide input for the physician to make decisions (including calling a specialist). An RI graph may be validated to check for inconsistencies and can be used to decide which software capabilities should be considered for a system. For instance, in this case, providing support for hospital workers to become up-to-date with new information when they transit from an unreachable to a reachable state.

5.3 Deriving the Groupware Requirements

Fig. 10 shows the groupware requirements that were obtained from the hospital setting graph (Fig. 9-right). The requirements between the on-call specialist and the roles they interact with (physician and specialist) are shown. The user may also choose to view the requirements for other roles by selecting the corresponding tabs (nurse, physician, intern or specialist). The software developer may then review the suggestions, choosing to implement some or all of the required groupware services.

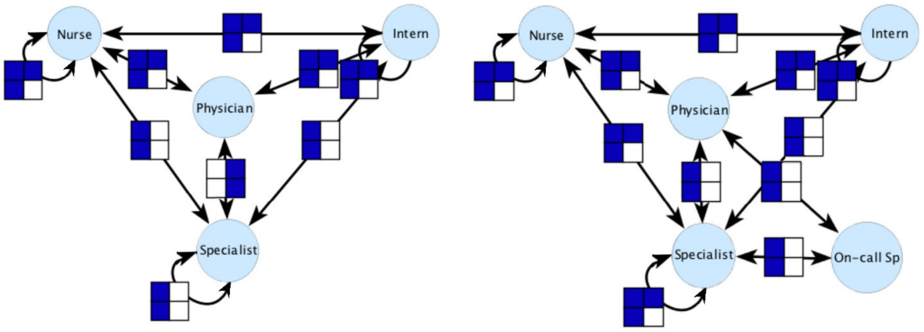
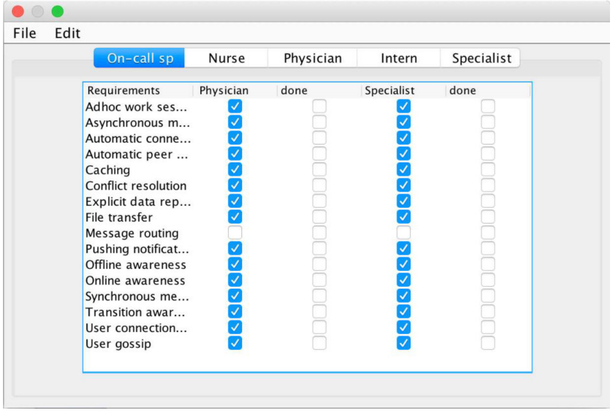


Figure 9: RI graphs of hospital work case. (left) Initial graph, (right) Final graph



Requirements	Physician	done	Specialist	done
Adhoc work ses...	☑	☐	☑	☐
Asynchronous m...	☑	☐	☑	☐
Automatic conne...	☑	☐	☑	☐
Automatic peer ...	☑	☐	☑	☐
Caching	☑	☐	☑	☐
Conflict resolution	☑	☐	☑	☐
Explicit data rep...	☑	☐	☑	☐
File transfer	☑	☐	☑	☐
Message routing	☐	☐	☐	☐
Pushing notificat...	☑	☐	☑	☐
Offline awareness	☑	☐	☑	☐
Online awareness	☑	☐	☑	☐
Synchronous me...	☑	☐	☑	☐
Transition awar...	☑	☐	☑	☐
User connection...	☑	☐	☑	☐
User gossip	☑	☐	☑	☐

Figure 10: Requirements for hospital work setting, derived from Fig. 9-right

6 Evaluation of the MCM notation

The MCM notation has been used by the authors and several graduate students over the last six years, to discover user interaction requirements in mobile collaborative applications. It has also been used to evaluate the suitability of the interaction services provided by prototypes of mobile collaborative applications. This section reports the results of experiments done in various projects that contribute to assess usability and usefulness of the MCM notation for its use by developers.

The evaluation of the usability was focused on determining whether the developers can learn the notation and use it to model particular interaction scenarios belonging to a people-driven collaborative process. The usefulness of the notation was evaluated in terms of its capability to represent the interaction scenarios and derive suitable groupware requirements from them. Next, we describe both evaluation processes and present and discuss the obtained results.

6.1 Usability of the Notation

We conducted two user studies to determine the usability of MCM; both involved Computer Science students at the University of Chile, who have no work or study links with the authors. The recruitment process was done using a snowball mechanism, and participation was voluntary. All participants were informed that they could quit the evaluation whenever they wanted without consequences. Graduate students and final year undergraduate students (who were working on their engineering thesis) were allowed to participate in these evaluations. This restriction ensures that these participants were knowledgeable in system and process modeling.

The first study involved the modeling of the USAR interaction scenario, and it allowed us to have a preliminary feedback about the usability of the notation. It also served as a pilot study that helped design the second study in which a mixed method (qualitative and quantitative) was used to get a more in depth understanding of the usability of MCM. The second study involved modeling the thesis proposal elaboration process used in the Computer Science Department of the University of Chile. Next we describe each experience and discuss the evaluation results.

6.1.1 Modeling a USAR Interaction Scenario

Eight people participated in this study; one of them had previous formal training in USAR processes and two were knowledgeable in this study domain. The experiment presented a one-page summary of the notation, and a one-page summary of a modified USAR scenario, described as an observation of the firefighters' work. We asked the students to individually model the interaction scenario using the notation. They had one week to perform this activity, so they had ample time to do it.

The average time the participants reported for learning the notation was 36 minutes. After such a period, all participants were able to complete the activity and generate a correct RI graph that included over 80% of the interactions indicated in the process description. Only two participants were not able to identify all interactions; they were particularly people without any knowledge about the USAR process before this exercise. The interactions that were not identified by the participants were those not explicitly indicated in the process description, but which can be inferred from the explanation. We initially assumed this situation was generated by lack of both, previous participants' knowledge, and details in the interaction scenario description; however, we addressed this aspect more in depth in the second evaluation experience.

Once the participants finished the modeling activity, we asked them to answer a 10-item survey (a modified SUS [Brooke 1996]), in which each question was scored in a 5-point Likert scale (1: strongly disagree to 5: strongly agree). Table 7 presents the items (translated from Spanish), their average scores and the standard deviation.

#	Item	Score	SD
1	If I had to model mobile collaborative work again, I would use this notation.	3.57	0.90
2	I found the notation unnecessarily complex.	1.86	0.83
3	I thought the notation was easy to use.	3.86	0.99
4	I would need the support of an expert person to be able to use this notation.	1.57	0.73
5	I found the elements of the notation were well integrated.	3.86	0.64
6	I thought the notation was inconsistent.	2.00	1.07
7	I would imagine that most people would learn to use this notation quickly.	3.71	0.88
8	I found the notation cumbersome to use.	1.86	0.64
9	I felt confident using the notation.	3.57	0.90
10	I needed to learn a lot of things before I could start using the notation.	1.14	0.35

Table 7: Usability results of the MCM notation

The results obtained in every item of the survey were consistent, and most people indicated to feel comfortable using the notation and willing to using it to model mobile collaborative scenarios. These results show the usability of the MCM notation is good enough as to use it in practice, at least in research scenarios, although there is also space to improve it.

The scores were normalized in a 0 (worst) – 100 (best) range, where the average usability score was 75.3, with scores ranging from 42.5 to 92.5. This is aligned with our initial understanding about the usability of the notation. The next section presents the second evaluation scenario, with in depth analysis about the usability of MCM.

6.1.2 Modeling the Thesis Proposal Elaboration Process

The thesis proposal elaboration is a particular collaborative process used at the Computer Science Department of the University of Chile; several roles formally participate in it (e.g., students, advisors, a process coordinator, reviewers and supporting staff). The process has a high-level workflow that involves four macro-activities and considers much variability. There is not a workflow coordinating the tasks that compose the macro-activities, and the collaboration among the involved people is done on-demand. This aspect makes the process a people-driven collaborative one.

Sixteen people participated in this experiment; twelve of them were PhD students, two were instructors and two were undergraduate students enrolled in the thesis process. All of them knew quite well the process to be modeled, and none of them participated in the previous evaluation.

The evaluation involved two sessions in which participated six and ten people respectively. These sessions had a coordinator who guided the process, and one of the authors acted as observer during these experiences. Each session considered five steps: (1) a brief explanation of the notation delivered by the coordinator (and one page with the notation was also given to the participants to support their modeling activities), (2) a brief overview of the thesis proposal elaboration process, (3) the modeling of such a process that was conducted individually by the participants, (4) the filling of an evaluation form, and (5) an open discussion to try to understand evaluation results.

The explanation of the MCM notation took 6 minutes on the average and it was presented using slides; a sheet with a summary of the notation was provided to the participants as support to the modeling activity. The second step, i.e., the overview of the process to model, was explained describing nine interaction scenarios shown using slides; each of them included an explanatory caption. These scenarios were specified with the nomenclature proposed by [Antunes 2013] and their explanation took 7-8 minutes given all participants knew the process. A printed copy of those scenarios was also provided to the participants as support of the modeling activity.

After a brief review of the nomenclature (2-3 minutes), the participants were able to specify the RI graphs in 8 minutes on the average. All graphs were correct and complete according to the specification given to the participants, which indicates that the notation was not a limitation for them to specify the interactions.

After representing the RI graph the participants filled a questionnaire relating the attributes of the notation and key measures derived from the goal-question-metric model [Buchanan 2009]. The questionnaire considered six items to evaluate the usability of the MCM notation (Table 8): expressiveness, aesthetic, understandability, intuitiveness, and learnability.

Goal	Question (asks if...)	Metric
Expressiveness	1. The interaction model carried out is complete.	- % of specified interactions
Aesthetic / appearance	2. Visual appearance is nice. 3. The notation symbols are appropriate.	- Attractiveness - Appropriateness
Understandability of the terminology	4. The used symbols are understandable.	- Comprehension
Intuitiveness	5. The used symbols are intuitive.	- Intuitiveness
Learnability	6. The notation is easy and quick to learn.	- Easiness to learn

Table 8: Usability questions and metrics

The participants rated each item using a 4-point Likert scale (from 1: strongly disagree, 2: disagree, 3: agree, 4: strongly agree) and thus assume a positive or negative position about the MCM usability. Nine participants (56%) rated the notation as good or very good, and the rest considered MCM as acceptable. During the open discussion conducted at the end, participants indicated that the main aspects that limited learnability were the need of using the concept of reachability and quadrants to represent interaction scenarios between two roles. These concepts were easier to understand for people who had already modeled mobile collaborative systems, but this shows MCM still lacks intuitiveness. These results show how the usability of MCM may be improved; they also show the nomenclature is already suitable to use it in practice (Table 9).

Item / Score	1	2	3	4	Avg.	S.D.
Expressiveness	0%	25%	50%	25%	3.0	0.7
Aesthetic / appearance	0%	12.5%	37.5%	50%	3.4	0.7
Understandability	6.25%	25%	25%	43.75%	3.1	1.0
Intuitiveness	6.25%	18.75%	43.75%	31.25%	3.0	0.9
Learnability	6.25%	43.75%	31.25%	18.75%	2.6	0.9

Table 9: Evaluation results (1=Strongly Disagree, 4=Strongly Agree)

6.2 Usefulness of the Notation

The usefulness of the notation was evaluated in terms of how much the RI graph representations help developers determine the groupware services required to support the collaborative work. In order to do that, we analyzed a mobile collaborative application developed to assist USAR activities, and also an equivalent application designed to facilitate mobile work in hospitals. Each system was conceived to support the people-driven processes explained in the previous sections.

The first version of these mobile collaborative systems was developed by PhD students, initially considering a set of groupware requirements obtained in an ad hoc manner. In the first case, the main developer belonged to the University of Chile, and in the second case it belonged to CICESE, Mexico.

This evaluation process started once the first version of these systems was completed, and it considered these versions as a baseline to try to identify the usefulness of the notation to help improve the functionality of these systems. Thus, the first step was to assess the functionality of these applications. Then, we created an RI graph for each process to obtain the groupware requirements for the improved versions of these systems. None of the authors participated in the development of the systems; only the PhD students and some supporting developers participated in it.

The results were positive; they show the capability of the proposed method to identify interaction requirements; particularly, the potential usefulness of the MCM notation to help developers elicit interaction requirements of mobile collaborative systems. Next, we briefly explain both evaluation processes and the obtained results.

6.2.1 Improving an USAR Mobile Ad hoc Collaborative Application

In this evaluation, the collaborative functionality of the MobileMap application [Monares 2008] was used as baseline to determine the capability of MCM (and its RI graph validation algorithms) for improving such functionality, based on the representation and analysis of the RI graph of the USAR process.

6.2.1.1. Establishing the baseline

The functionality of the first version was reviewed during a working session lasting about 1 hour; the participants were the system developer and the authors. The goals were to determine the groupware services that should be considered by the next versions of MobileMap, and also identify the already implemented interaction services. In this first review, none of the previously presented processes or tools was used. The evaluation results show that MobileMap 1.0 considered only one general role (firefighter) and addressed two out of 14 groupware requirements identified as mandatory, which was insufficient to support the USAR process.

Then, in order to conceive the second and third version of the system, the main developer and his team followed the process shown in Fig. 1, and built an RI graph. It allowed them to specialize the user roles and also the type and number of groupware services required by these users. Next we explain these processes.

6.2.1.2. Identifying the groupware requirements

In order to identify the groupware requirements of the USAR process, we created the RI graph of the process to be supported, and also validated and refined it with people who play the relevant roles; particularly, firefighters of the 2nd Fire company of Ñuñoa (Chile). The goal of this activity was to make sure the RI graph was correct and considers all key interactions among roles participating in the process.

Then, we used the GMT tool to derive groupware requirements from the RI graph. The resulting requirements for MobileMap 2.0 and 3.0 were analyzed in two focus groups where the participants were the authors, a couple of system developers and at least one end-user per role included in the graph. The focus groups lasted between 40 and 65 minutes. In each session, the interaction supporting services provided by a particular version of the system were reviewed and compared to those suggested by the GMT tool. The discrepancies (typically, additional services suggested by GMT) were analyzed by developers and end-users to assess their suitability and criticality. When the tool suggested more than one option to provide a particular service, the developers and end-users agreed which service to embed into the system. The evaluation results (Table 10) show that using the RI graphs and GMT tool allowed developers to specialize the user roles and identify almost 50 new requirements.

The first row of the Table shows the number of roles included in each MobileMap version. The second row gives the number and percentage of groupware requirements that were addressed in each version of the system, out of the total number of mandatory requirements identified by the developers after using the GMT tool.

This example shows the use of the MCM notation and its processing strategy are useful to derive the groupware requirements from a collaborative process observation, and also to improve the functionality of already implemented systems. This latter aspect is also clear when comparing versions 2.0 and 3.0: the percentage of mandatory groupware requirements addressed by the product grew from 50.0% to 58.7%. Given the granularity used to analyze the interaction support between roles, the number of requirements suggested by the GMT application is usually high, but they can be easily grouped by involved roles or functionality concerns.

Evaluated version	**1.0**	**2.0**	**3.0**
# of supported roles	1	5	5
% of addressed requirements	2/14 (14.3%)	31/62 (50.0%)	37/63 (58.7%)
False positives	--	34/96 (35.4%)	33/96 (34.3%)
False negatives	0/16	0/62	0/63

Table 10: Usefulness of MCM to improve the collaborative features of MobileMap

The third row of Table 10 includes the number of *false positives*, i.e., the number of groupware requirements suggested by the tool as potentially relevant (in this case 96 requirements), versus the number of requirements that were determined by the evaluators (i.e., the developers and end-users) as actually not required. When using MCM, a 35.4 and 34.3% of the requirements proposed by the GMT system were false positives. Examining the evaluation results in detail, we found that many of them were suggestions to include groupware services with low relevance for the end-users. This happened because the system intends to maximize the number of requirement suggestions given to the developers. This issue represents an opportunity to improve the requirements recommendations from the GMT system, more than a weakness of the notation or the processing strategy for the RI graphs.

The fourth row indicates the number of *false negatives*, i.e., the number of mandatory requirements that were not identified by the GMT system. This number is 0 in all cases, indicating the MCM notation can properly represent the interactions in a certain work scenario, and the mechanisms to derive the requirements is effective. These results support the usefulness of this proposal for using it in real settings.

6.2.2 Improving a Mobile Collaborative Application for Hospital Work

Similar to the previous case, we first analyzed the collaborative functionality of the application named cTracker [Mejia 2007], and we used it as baseline to improve its functionality using MCM models and its processing mechanisms. This first review lasted 45 minutes and involved the authors and the main developer of this application.

Versions 2.0 and 3.0 of cTracker were not actually implemented, but discussed with the developers and planned by them. The RI graphs of these systems (shown in Fig. 9-left and 9-right) were analyzed using the GMT system and their groupware requirements were obtained accordingly. In order to analyze these requirements a focus group (for each version) was conducted involving the authors (as recorders and coordinators), a couple of the system developers and four end-users mimicking the dynamics explained for the evaluation of MobileMap.

Table 11 shows a summary of the roles and groupware requirements identified by the developers in ad hoc manner for the version 1.0, and then with the support of the GMT tool for versions 2.0 and 3.0 of the system. As in the previous case, the representation of the interaction scenarios using MCM allowed developers to identify new roles (specializations of the previous ones) to be supported. Due to the specification of interaction scenarios and the identification of new roles, the number of mandatory requirements increased from 16 (when they were identified in an ad hoc manner) to 107 and 128 (when the proposal was used) for cTracker 2.0 and 3.0 respectively. The percentage of addressed requirements is indicated as null since those versions were not actually implemented, but their functionality was analyzed in detail by a subset of the cTracker developers and users.

The number of false positives was between 20.5 and 23.6%, for the same reasons as in the previous case; i.e., the number of options suggested by the GMT system to support a certain interaction requirement. In case of the false negatives, they were always 0%, showing that the proposal is useful to identify the mandatory groupware requirements of collaborative systems such as these.

Evaluated version	**1.0** (implemented)	**2.0** (envisioned)	**3.0** (envisioned)
# of supported roles	2	4	5
% of addressed requirements	5/16	--/107	--/128
False positives	--	33/140 (23.6%)	33/161 (20.5%)
False negatives	--	0/140	0/161

Table 11: Usefulness of MCM to improve the collaborative features of cTracker

In both examples we can see the evolution of the software functionality due to the use of this proposal; therefore, these results support that MCM models and their validation are useful to identify interaction requirements of this type of system.

7 Discussion and Limitations

Based on the observation of the collaboration process, software developers can identify the participating roles and the interactions among them. Using the MCM notation it is possible to specify the Roles Interaction (RI) graph for a people-driven collaborative process. This graph may be used to communicate how people interact in a standard way when compared to the results of social science observation techniques.

The coherence of the graph can be validated automatically. RI graphs play a role similar to workflows during the development of information systems; i.e., these artifacts inform the groupware requirements identification. Using GMT system and coherent RI graphs as input, developers can get groupware requirements (particularly, interaction supporting services) that could be potentially required by mobile workers while they participate in a people-driven collaborative process. Thus, it is possible to identify, at an early stage, whether the services considered to be embedded into the supporting application are enough to allow users perform the collaborative work.

Given the GMT system suggests several options to support the interactions between every pair of roles, the developer has to choose which requirements to keep as part of the product backlog and which to discard. This requirement identification helps increase the suitability and effectivity of the obtained collaborative systems. Therefore, from a technology adoption perspective, we believe the trade-off between the benefits obtained due to the specification of interaction scenarios and the cost and burden to do it, is biased towards the use of this new collaboration analysis approach.

There are some limitations that we would like to acknowledge; first, the USAR process is not necessarily the only example that could have been used to motivate this paper. One specific advantage of it is that the collaboration is observable. However, this is only one specific example, which led us to generalize our observations to other scenarios, and to test our proposal under other conditions. Our graphs do not include all possible information about the collaboration, trying instead to maintain a level of simplicity. As a consequence, individual tasks are not taken into account, nor are other aspects such as frequency, length, or periodicity of the interactions.

Regarding our method, we do not indicate or prescribe a method to extract the required information from users to be able to build a RI graph, and we acknowledge that some methods may be more appropriate than others in specific situations.

Concerning the usability evaluation of the MCM notation, some participants were from the authors' university, and most were graduate students. However, none of them were related to the authors and their participation was voluntary. A validated questionnaire for assessing the usability of information systems, SUS, was adapted to assess usability. This adaptation means that the results of our evaluation are not directly comparable to SUS results in other domains. The evaluation on the usefulness of the notation is based on the framework in the earlier examples/case studies presented in the article. The results could be biased due to the authors' participation in the definition of the RI graph for the case studies. However, the authors did not work in establishing the suitability of the services proposed by the GMT tool.

8 Conclusions and Future Work

This article presented MCM, a role interactions notation, which was conceived to represent and analyze interactions among the participants in a people-driven mobile collaborative process. The goal of this notation is to help identify the groupware requirements to be embedded in a certain mobile ad hoc collaborative system.

The MCM notation was used to specify interactions in two particular scenarios: urban search-and rescue and hospital work. Both interaction scenarios were specified, validated, and the list of groupware services for each role were obtained. The list was validated with developers and potential users of these mobile collaborative applications, and the obtained results were highly positive. Moreover, the usability and usefulness of the proposal was rated with high scores, showing that it can be used to support the development of systems in research and industrial scenarios.

In this sense, the MCM notation, the RI graphs and the mechanism to derive the groupware requirements represent a tangible contribution to the conception and analysis of mobile ad hoc collaborative systems. The proposed notation provides a new way to study and understand mobile collaboration scenarios, with a goal of building mobile systems that truly support the collaboration needs of their end-users.

Next research steps include using the MCM notation to specify interaction scenarios in other collaborative application domains, e.g. home healthcare and tourism. We plan to extend the notation to incorporate additional information about the interactions and collaborations that take place, e.g., which roles and interactions are the most important, which are the most usual collaboration scenarios, and which services can be implemented to support interactions.

Acknowledgements

We thank Maximiliano Canché for the valuable help provided in the evaluation. This research has been partially supported by the grant Fondecyt Enlace, VID, University of Chile, and by CONICYT Fondecyt (Chile), grants 1191516 and 1181162.

References

[Antunes 2013] Antunes, P., Simões, D., Carriço L., Pino, J.A.: "An end-user approach to business process modeling". J. of Network and Computer Applications 36, 6 (2013), 1466–1479.

[Antunes 2018] Antunes, P., Baloian, N., Zurita, G., Pino, J.A.: "Supporting people-driven, dynamic and geo-located work processes". Proc. 10th International Conference on Subject-Oriented Business Process Management (S-BPM One '18). Linz, Austria (2018).

[Buchanan 2009] Buchanan, S., Salako, A.: "Evaluating the usability and usefulness of a digital library". Library Review 58, 9 (2009), 638–651.

[Arciniegas-Mendez 2017] Arciniegas-Mendez, M., Zagalsky, A., Storey, M., Hadwin, A.F.: "Using the model of regulation to understand software development collaboration practices and tool support". ACM Conf. on CSCW and Soc. Computing. Portland (2017), 1049-1065.

[Bardram 2005] Bardram, J.E., Bossen, C.: "A web of coordinative artifacts: Collaborative work at a hospital ward". ACM SIGGROUP Conf. on Supporting Group Work, (2005), 168–176.

[Bjorn 2016] Bjørn, P.: "New fundamentals for CSCW research: From distance to politics". Interactions 23, 3 (2016), 50-53.

[Blomberg 2013] Blomberg, J., Karasti, H.: "Reflections on 25 years of ethnography in CSCW". Computer Supported Cooperative Work 22, 4-6 (2013), 373-423.

[Bossen 2002] Bossen, C.: "The parameters of common information spaces: The heterogeneity of cooperative work at a hospital ward". Proc. ACM Conf. on CSCW, (2002), 176–185.

[Brooke 1996] Brooke, J.: "SUS-A quick and dirty usability scale". Usability Evaluation Industry 189, 194 (1996), 4–7.

[Canche 2018] Canche, M., Ochoa, S.F.: "Modeling Computer-mediated User Interactions in Ubiquitous Collaboration Systems". Proc. 12th Int. Conf. on Ubiquitous Comp. and Ambient Intelligence (UCAmI'18), MDPI Proceedings 2, 19, article 1250, Punta Cana (2018).

[Conca 2018] Conca, T., Saint-Pierre, C., Herskovic, V., Sepúlveda, M., Capurro, D., Prieto, F., Fernández-Llatas, C.: "Analysis of multidisciplinary collaboration in the treatment of patients with type 2 diabetes in primary care using process mining". J. of Med. Internet Res. 20, 4 (2018).

[Christensen 2017] Christensen, L.R., Hildebrandt, T.: "Modelling cooperative work at a medical department". Proc. 8th Int. Conference on Communities and Technologies, (2017), 46-55.

[Dong 2007] Dong, W., Lepri, B., Cappelletti, A., Pentland, A.S., Pianesi, F., Zancanaro, M.: "Using the influence model to recognize functional roles in meetings". Proc. 9th International Conference on Multimodal Interfaces, 271–278, 2007.

[Erickson 2016] Erickson, I., Jarrahi, M.H.: "Infrastructuring and the challenge of dynamic seams in mobile knowledge work". Proc. ACM Conference on CSCW, (2016), 1323–1336.

[Feldman 2003] Feldman, M.S., Pentland, B.T.: "Reconceptualizing organizational routines as a source of flexibility and change". Adm. Sci. Q., 48, 1 (2003), 94–118.

[Fitzpatrick 2012] Fitzpatrick, G., Ellingsen, G.: "A review of 25 years of CSCW research in healthcare: contributions, challenges and future agendas". J of CSCW 22, 4 (2012), 609–665.

[Fowler 2000] Fowler, M., Scott, K.: "UML Distilled (2nd Ed.): A Brief Guide to the Standard Object Modeling Language". Addison-Wesley Longman Publishing, Boston, MA (2000).

[Gottanka 2012] Gottanka, R., Meyer, N.: "ModelAsYouGo: (Re-) Design of S-BPM Process Models during Execution Time". S-BPM ONE '12, Springer, Vienna (2012), 91–105.

[Gideon 2012] Gideon, L.: "Handbook of survey methodology for the social sciences". New York, NY: Springer (2012).

[Grudin 1994] Grudin, J.: "Groupware and social dynamics: eight challenges for developers". Communications of the ACM 37, 1 (1994), 92–105.

[Gutierrez 2017] Gutierrez, F.J., Ochoa, S.F.: "It takes at least two to tango: Understanding the cooperative nature of elderly caregiving in Latin America". Proc. ACM Conf. on CSCW, Portland (2017), 1618-1630.

[Harr 2002] Harr, R.: "Exploring the concept of group interaction through action in a mobile context". Proc. DEXA'02, LNCS 2453, (2002), 567-576.

[Herskovic 2009] Herskovic, V., Ochoa, S.F., Pino, J.A.: "Modeling groupware for mobile collaborative work". 13th IEEE Int. Conf. on CSCWD, Santiago (2009), 384–389.

[Herskovic 2011] Herskovic, V., Ochoa, S.F., Pino, J.A., Neyem, A.: "The iceberg effect: behind the user interface of mobile collaborative systems". J.UCS 17, 2 (2011), 183–201.

[Huth 2001] Huth, C., Erdmann, I., Nastansky, L.: "GroupProcess: using process knowledge from the participative design and practical operation of ad hoc processes for the design of structured workflows". 34th Annual Hawaii Int. Conf. on System Sciences, Maui, (2001).

[Kuhn 2018] Kühn, R., Schlegel, T.: "Mixed-focus collaboration activities for designing mobile interactions". Proceedings of MobileHCI (2018), 71-78.

[Mateti 1976] Mateti, P., Deo, N.: "On algorithms for enumerating all circuits of a graph". SIAM J. Comput. 5, 1 (1976), 90–99.

[Mejia 2007] Mejia, D.A., Morán, A.L., Favela, J.: "Supporting informal co-located collaboration in hospital work". 13th Int. Workshop CRIWG, LNCS 4715 (2007), 255–270.

[Monares 2008] Monares, A.: "MobileMap: A mobile collaborative environment for firefighters (in Spanish)". Diploma Thesis for Eng. Degree, Computer Science Dept., University of Chile, 2008.

[Moran 2007] Morán, E.B., Tentori, M., González, V.M., Favela, J., Martínez-García, A.I.: "Mobility in hospital work: towards a pervasive computing hospital environment". Int. J. Electron. Healthcare 3, 1 (2007), 72–89.

[Pan 2012] Pan, W., Dong, W., Cebrian, M., Kim, T., Fowler, J.H., Pentland, A.S.: "Modeling dynamical influence in human interaction: Using data to make better inferences about influence within social systems". IEEE Signal Processing Magazine 29, 2 (2012), 77–86.

[Pinelle 2005] Pinelle, D., Gutwin, C.: "A groupware design framework for loosely-coupled workgroups". 9th European Conf. on Comp. Supported Coop. Work (2005), 65–82.

[Poltrock 2009] Poltrock, S., Handel, M.: "Modeling collaborative behavior: Foundations for collaboration technologies". Proc. 42nd Hawaii Int. Conf. on System Sc. (2009), 1–10.

[Reilly 2009] Reilly, D.F., Inkpen, K.M., Watters, C.R.: "Controlling, integrating, and engaging context in urban computing research". Proc. 42nd HICSS Conference, (2009), 1–10.

[Rodriguez-Covili 2011] Rodríguez-Covili, J., Ochoa, S.F., Pino, J.A., Herskovic, V., Favela, J., Mejía, D., Moran, A.L.: "Towards a reference architecture for the design of mobile shared workspaces". Future Generation Computer Systems 27, 1 (2011), 109–118.

[Suthers 2011] Suthers, D.: "Interaction, mediation, and ties: An analytic hierarchy for socio-technical systems". 44th Hawaii Int. Conference on System Sciences (2011), 1–10.