Published in IET Software
Received on 30th April 2011
Revised on 10th October 2011
doi: 10.1049/iet-sen.2011.0077

Special Section: Management of Global Software Development: Opportunities, Challenges and Lessons Learned

ISSN 1751-8806

Selective availability: coordinating interaction initiation in distributed software development

R.R. Palacio[1] *A.L. Morán*[2] *V.M. González*[3] *A. Vizcaíno*[4]

[1]*DES Navojoa, Instituto Tecnológico de Sonora, Navojoa, Mexico*
[2]*Facultad, de Ciencias, Universidad Autonoma de Baja California, Ensenada, Mexico*
[3]*División Académica de Ingeniería, ITAM, Mexico*
[4]*Instituto de Tecnologías Sistemas de Informacíon, University of Castilla-La Mancha, Ciudad Real, Spain*
E-mail: alberto.moran@uabc.edu.mx

Abstract: The software industry is facing a recent trend called distributed software development (DSD), in which distributed teams require continuous support in their communication and coordination. However, there is a lack of communication tools that actually support the coordination of DSD activities. Current communication mechanisms appear to favour the issuer of an interaction, because the context of the receiver is not always considered. In this study, the authors introduce selective availability (SA), a mechanism with which to provide information about the current activities of the members in a distributed team, in order to motivate a more suitable means to initiate interactions, thus facilitating the communication and coordination of DSD activities. Moreover, the authors describe the CWS-IM tool, an extended instant messaging application that supports SA, by notifying collaborators about each of their colleague's activities. Therefore, issuers can decide whether the time is right to start the interaction. The results of an evaluation of the actual use of the tool in a DSD software development company are also presented. These results indicate that developers perceive CWS-IM to be more useful and easier to use than other traditional instant messaging applications when initiating collaboration in DSD environments.

1 Introduction

The software industry has recently been confronted with a work paradigm known as distributed software development (DSD) [1]. This paradigm involves separating the different development processes and collaborators into multiple geographic locations. The typical scenarios for DSD are generally defined by a number of characteristics. One of these is distance, which can vary from a few to tens, hundreds or thousands of metres.

The software industry is motivated to follow the DSD work trend as it provides the following benefits: (a) advantage can be taken to work continuously on a project (up to 24 h/day) through collaboration among colleagues who are in different time zones [2, 3], which in turn allows (b) lower labour costs [4], (c) the rapid creation of virtual development groups [5] and (d) being geographically closer to the customer [6]. In spite of these benefits, the distance between the developers involved in joint projects creates some challenges for software development. Previous studies have shown that communication may be inadequate as distance makes face-to-face interaction more difficult [6, 7]. Other challenges include finding optimal mechanisms for knowledge management [4], time differences [8] and lack of awareness and trust relationships [8, 9], all of which may lead to problems caused by misunderstandings.

Previous studies provide some evidence of how activities are coordinated in DSD (e.g. [10, 11]). In general, these studies argue that coordination among developers, processes, information and technology are key challenges for this type of work environments. These research efforts also provide concepts and models with which to understand how coordination helps to cope with these challenges. For instance, Wiredu [10] proposes a concept of coordination which is focused on global software development (GSD) and is defined as 'managing interdependencies, uncertainties and equivocalities, conflicts, technology representations and their interrelations'. This concept is used to create a framework for the analysis of coordination in GSD, which offers information about how coordination occurs in the workplace. This framework is a starting point for further research into coordination issues in DSD. Moreover, Herbsleb and Mockus [12] state that teams working in DSD confront disadvantages when communicating and coordinating their work given the negative impact caused by distance, thus confirming that a coordination problem exists in DSD.

Within the DSD context it is also common to find that activities are planned and coordinated through the use of project management schemes, which help to reduce uncertainty in the workplace [13]. However, despite being a common practice in these organisations, project management in the DSD context is limited by the lack of informal communication [14], which generates low levels of trust between colleagues, and poor knowledge about their work and progress on remote sites [15, 16].

Other works (e.g. [17]) attempt to deal with these limitations by proposing the use of instant messaging (IM) to support informal communication, thus complementing the management of individual work in collaborative projects. They propose providing awareness information about multiple collaborators in multiple activities through an IM interface. However, these proposals do not consider using awareness information to promote the initiation of collaboration, and focus mainly on providing information about the user's identity, project team membership and general availability (offline, online, available, busy etc.).

Communication via IM presents a natural challenge: how can people communicate in such a way that interruptions are minimised and communication is optimal? In other empirical studies (e.g. [18]), researchers have investigated how workers are interrupted while carrying out their activities, showing how the complexity of the task, its duration, the number of interruptions and the type of task have an impact on the difficulty of returning to the interrupted task (task switching) [19]. The results of these studies thus characterise how workers behave when confronted with interruptions (interruption management) [20]. However, these studies do not provide sufficient elements concerning how to select the right time to start an interaction.

A possible starting point to discover the optimal means to begin an interaction could be based on the way in which people organise and manage their work. Among other models, a more personal approach to project management can be defined from personal activity management (PAM) [21], which offers a documented and informed perspective of the work that individual workers have to do. PAM is based on the analysis of the processes and strategies that are involved in the way in which workers confront the planning and management of their activities. PAM is related to the concept of working sphere, which explains how people as individuals organise their work, and is sufficiently flexible to represent the activities with the required degree of granularity [22, 23]. The use of the working sphere concept may be useful as it provides elements of the activities or tasks performed by people (e.g. resources, repositories or related applications used by the individual). These can potentially be used to identify an opportunity to initiate collaboration through the monitoring of context (e.g. through potential collaboration awareness [24]). The concept of potential collaboration awareness is important because it enables the most appropriate time at which to establish an interaction to be discovered.

Our understanding of some of the coordination problems in software development is derived from the way in which the development process is carried out. That is, in the case of co-located development, project members are on sight or are easily accessible, so that it is possible to see or estimate what they are doing, without any significant effort. It is even possible to judge, usually with a high degree of certitude, whether that precise moment is appropriate to interrupt what others are doing in order to establish communication and maintain a coordination effort, while attempting to minimise the impact of the interruption. In contrast, in the case of DSD, participants are located on distant sites, signifying that the contextual information existing in a co-located situation no longer exists. This hinders the communication, coordination and production processes.

It is therefore essential to know the activity status of the person one wishes or needs to contact, and thus attempt to find an appropriate moment for both participants, in order to initiate an interaction while minimising the negative effects of an interruption [18]. That is, good coordination requires good communication [25], and good communication requires appropriate initial interactions. Information is thus required to discover the development context of the target participant in order to potentiate collaboration [24, 26].

We are interested in understanding how to initiate interaction in a more informed manner, implying fewer interruptions. Modern communication technologies (e.g. IM) appear to favour the person who is interested in initiating the interaction (issuer), as they help to identify when it is possible to contact a remote colleague (e.g. identity and presence). In contrast, the features and ease of access of such technologies increase the sources of interruption for the person being contacted (receiver) [20], potentially resulting in negative interruption effects (e.g. prospective memory failure). These interruptions may result in a possible miscommunication, and poor communication can significantly contribute to the poor coordination of development activities.

The aforementioned analysis provided us with a guideline that could be used to establish the research question that guided this work:

- How can DSD workers initiate collaboration in an informed way, which is appropriate for both the issuer and the receiver, while they perform their development activities?

We propose to answer this question by extracting information from the individual activities of each worker to infer what they are doing with regard to the assigned project activities. This is done in order to provide the distributed team with more information about the activities of all workers at any given time, and to thus obtain the information elements needed to decide whether or not the time is appropriate to initiate collaboration, both from the perspective of the interaction issuer and that of the receiver.

In a previous work [27] we proposed the concept of collaborative working spheres (CWS), and used it as a basis to inform the design of the graphical user interface (GUI) of the CWS-IM tool. The proposed GUI of the tool was evaluated through projected scenarios with staff from three sites of a Mexican software development company. In this paper, we present the system design and actual implementation of CWS-IM, along with the results of an empirical evaluation of its actual use comprising a 3-week study in Novutek, a Mexican DSD software development company specialised in the development of custom software applications for their clients.

The remainder of the paper is organised as follows. Section 2 briefly presents the initial findings regarding the coordination of a DSD team in a software development company. Section 3 presents the selective availability (SA) criterion, along with current and projected use scenarios. Section 4 presents CWS-IM, an extended Instant Messaging (IM) application implementing SA support. Section 5 shows the main results of an actual use evaluation of CWS-IM. Section 6 presents related work, and finally, our concluding remarks and some directions of future work are shown in Section 7.

2 Understanding coordination in a DSD team

In order to identify some of the coordination problems that developers confront during DSD, we conducted semi-structured interviews with 16 workers from a Mexican DSD

doi: 10.1049/iet-sen.2011.0077

software development company based in Mexico and the US. This software development company bases its production processes on the rational unified process (RUP) and uses the unified modelling language (UML) for software modelling and design. It has also achieved an SEI CMMI maturity level 3. The project management methodology used is based on the standards of the Project Management Institute (PMI), and has a group of project leaders in the process of certification as Project Management Professionals.

The participants (four project leaders, two testers, four programmers and six software architects) had an average amount of work experience in DSD of 2.9 years, and were involved in the definition and execution stages of their respective projects, developing ASP and C# .NET applications for desktop and mobile devices, with database backends on SQL Server 2000/2005.

These participants were located in three cities (Obregón, Guaymas and Navojoa) in the state of Sonora, México and in one city (Tucson) in the state of Arizona, USA. The, participants were thus distributed throughout different buildings on the same site (separated by 40–80 m), sites in the same city (separated by approximately 4–20 km) and different cities (separated by approximately 60–700 km). In this case the time zone was the same (Sonora, Mexico and Arizona, USA). Cultural differences were not considered as the workers were bilingual and moved among sites for training and on a project basis.

The instrument for data acquisition during the interviews was a questionnaire which included questions regarding four main topics concerning DSD coordination: project planning, PAM, work articulation and communication. A sample of these questions is shown in Table 1.

The data obtained from the interviews were interpreted by conducting a qualitative analysis focused on identifying the software development company's specific coordination needs. The data were extracted from the interviews through the use of a comparative method based on grounded theory techniques [28]. This process began with open coding, in which the data were divided into categories (recurrent concepts) and subcategories. We later continued with axial coding, where we examined the relationships between categories and the possible causes of their behaviour. Finally, in the selective coding phase, the categories were integrated and conclusions were obtained.

A summary of the results of this process is shown in Table 2. Coordination problems were identified during work stages in the software development company (definition and execution), and were categorised either as scale-, uncertainty-, interdependence- or communication-related problems, inspired by the proposal in [29]. A brief description of these coordination problems is presented below.

In the definition stage, the problems identified include:

SD1 – Scale estimation during project proposal planning: During proposal planning, the project marketing and management people should, among other things, determine the size and scope of the project to determine the type and number of human and material resources to schedule for it. Incomplete or incorrect information, or misunderstandings among the parties, will lead to an incorrect estimation of the requirements and will generate problems regarding the scale of the project.

UD1 – Uncertainty during effort estimation: There is a high uncertainty in the estimation (for budgeting purposes) of the work hours required for an activity. This principally depends on the project manager's experience and on the ease of access to other reference sources, including other remote actors.

UD2 – Uncertainty about developers' specialisation levels: A high amount of uncertainty is caused by not knowing the skill level of remote developers in certain technical areas. This has implications for the allocation of activities.

ID1 – Dependencies on resource availability: There is heavy dependence on the allocation of activities to each worker which, in turn, leads to uncertainty as a result of the possible indisposition of human and material resources among dependent activities.

CD1 – Critical communication among key roles to deliver a proposal on time: The pressure to deliver a project proposal on time makes communication among key actors more critical. As deadlines approach, people get in a rush and tend to avoid communication and coordination activities. Managers should estimate the proposal based on their personal experience in order to be able to submit it on time.

Table 1 Sample questions from the questionnaire for the interviews

Project planning	Personal activity management
how do you plan a project?	do you engage in activities from DPs at the same time?
what criteria do you use to assign tasks to developers?	how do you identify the activities from DPs in your workspace?
how do you communicate assignments to developers?	how do you organise work from DPs?
how do you assign resources?	how do you go about managing things and getting things done?
what people are involved in project management?	what tools do you use to support personal organisation?
what are the main challenges that you usually experience during project planning?	what problems have you encountered when attempting to get things done?
Articulation work	**Communication**
do you use any tool to integrate the work undertaken as a group?	how do you communicate with your co-workers?
how do you organise the work groups?	what means or tools are used?
how do members notify others when they conclude their work assignments on the project?	do you follow any standard process to communicate with colleagues?
what are the main problems found when organising group work?	how do you report progress to your teammates?
	what kind of problems have you encountered when communicating with other members of your team?

Table 2 Identification of coordination problems in the software factory

	Scale	Uncertainty	Inter-dependence	Communication
definition stage	SD1: Scale estimation during project proposal planning	UD1: Uncertainty during effort estimation UD2: Uncertainty about specialisation levels of developers	ID1: Dependencies on resource availability	CD1: Dependency on project manager to deliver the proposal on time
execution stage	SE1: Assigned resources and activities according to scale of the project	UE1: Uncertainty about individual activities during project execution UE2: Uncertainty about projects at risk UE3: Uncertainty about parallel roles of workers	IE1: Interdependence among workers on collaborative activities	CE1: Uncertainty about most appropriated way to make contact for urgent communication

Some of the problems identified in the execution stage include:

SE1 – Assigned resources and activities according to the scale of the project: Among others, the size of the project and the number of resources assigned to it determine the amount of activities to be assigned to members of the development team. However, during the execution of the project, managers may assign or re-assign project activities to team members. Managers may also add developers who were not initially included, which will certainly add to the complexity of the project. There are thus dependences on the project manager's experience when allocating activities to developers according to the scale of the project.

UE1 – Uncertainty about individual activities during project execution: There is uncertainty regarding what activities developers carry out individually in their workspace. It is not easy to know which activity a developer is working on at any given time.

UE2 – Uncertainty about projects at risk: This refers to the high dependency between project managers to decide when a project is at risk, and to the uncertainty of whether and when to add other workers to a project that could become a failure.

UE3 – Uncertainty about parallel roles of workers: This refers to the case in which developers work in different roles [and even on different projects (DPs)] during their work day. This creates confusion and misunderstandings between collaborators as regards the activity that these developers are performing at a given time.

IE1 – Interdependence among workers on collaborative activities: During the execution of a project there is high dependence between collaborators in the same role and in different roles. Uncertainty exists regarding the way in which the efforts of each team member will be integrated to accomplish group activities.

CE1 – Uncertainty about the most appropriate way in which to make contact when urgent communication is necessary: When it is urgent to contact collaborators there is no certainty about what the most appropriate way to contact them is. The mechanisms that are used to attempt to establish contact are currently defined by considering the level of urgency of the interaction.

It is important to note that some of the problems presented here could be classified into more than one of the proposed categories. For instance, SD1 could be also classified under Uncertainty. It is also important to highlight that the list of problems shown is not intended to be comprehensive or complete, but rather an illustrative sample of the problems identified. These problems allowed us to identify an ensemble of needs regarding coordination in DSD, which were in turn used to establish an initial set of requirements for a tool whose aim is to provide awareness support in the initiation of timely appropriate interactions in a DSD environment. The requirements identified include the provision of:

R1. Easy access to knowledge regarding the status of activities and resources by means of the collaborators' current activities (SD1, SE1, UE1, UE2, UE3, ID1 and IE1).
R2. Fast and easy access to information about the collaborators' profiles (UD1 and UD2).
R3. Lightweight communication mechanisms that automatically group the contacts (potential receivers of the communication) based on the activity that is performed at a given time (SD1 and CE1).
R4. Lightweight mechanisms to integrate information from the project management system into the communication tools to access a history of the executed projects, and to communicate with a set of expert consultants, with whom communication is easy and transparent (CD1).
R5. Simple and constant access to information concerning project collaborators' current activities and roles (SD1, ID1, UE1, UE2, UE3, IE1 and CE1).

Some of the problems identified clearly emerge as a result of a lack of information and communication between the participants in the development process.

3 Selective availability: establishing a more suitable moment to start collaboration

As evidenced by the findings described in the previous section, there is a set of problems in the software development company that affects certain coordination aspects of DSD activities. Part of the challenge is therefore to identify how to establish optimal communication among developers. This signifies that communication must be initiated at an opportune time, not only for the person who initiates the interaction (issuer) but also for the recipient of that communication (receiver). From this perspective we suggest that it is possible to facilitate the initiation of an interaction if the issuer's reason for or interest in contacting the receiver is aligned to the activity being performed by the latter. We based this assumption on the receiver's 'Willingness to Interact'; Ye [30] considers that an expert will be more willing to interact (or provide help) if the topic of interaction is closely related to her area of expertise.

In co-located environments, physical proximity usually allows the issuer to obtain sufficient information about the

doi: 10.1049/iet-sen.2011.0077

receiver's situation with a simple glance, and to then use this information to decide whether the time is right to contact her or to attempt an interaction. The distributed case, however, lacks physical proximity, and one means of achieving the above could be to provide the issuer with sufficient information about the possible receivers' current activities.

In this respect, our proposal is to introduce the criterion of SA which considers that a user is selectively available to collaborators whose activity is related to the work unit she is currently dealing with, and is less (or not) available to other collaborators.

The definition of CWS [31] is fundamental for the SA criterion. A CWS is a combination of working spheres [32] and potential collaboration awareness [24] that allows workers to detect, identify or create opportunities for collaboration (potential collaboration) with each other based on information managed in their individual work units. It also allows them to identify an appropriate moment at which to initiate collaboration in a more informed way by means of the information obtained from the actions that collaborators are performing to execute their individual activities. Moreover, CWS allow collaborators to have a meeting point with their potential collaborators, where they are actually offered a manner in which to begin an appropriate interaction and from which they may begin a formal interaction with the collaborators concerned. CWS also provides them with easy access to the work activities involved, and the consistent triggering of actual collaboration [24].

This approach can be more clearly illustrated by considering two scenarios which: (i) provide an example of how the situation currently occurs; and (ii) illustrate a similar situation, but include the proposed support to deal with the problem. These scenarios are presented in Sections 3.1 and 3.3, respectively.

3.1 Typical scenario

At the software development company, John, a programmer from the DSD group accesses a UML file through a diagramming application. This file was sent to him by one of his teammates at the client's site as part of a GUI design activity. Whenever John has a doubt about the contents of the UML file, he usually tries to contact the person responsible for the artefact by any means of communication (e.g. telephone or IM application). Since current communication technologies do not inform the issuer about the recipient's current activity, the receiver is usually interrupted. This means that the receiver may be involved in activities related to another project (parallel roles), which are not aligned with what the programmer (issuer) wishes to consult him about at that time. One approach with which to provide information about this user status is that used in traditional IM applications. However, these applications are limited with regard to the kind of information (e.g. available, busy, away) and the level of detail they can automatically gather and provide about the user's status (e.g. 'Busy' rather than 'Writing the project proposal'). Furthermore, these applications have no means of indicating different statuses for different users or groups (e.g. 'Available' for some, and 'Busy' for others).

3.2 Selective availability through CWS

Based on the scenario shown above, we propose the use of CWS as a means to provide SA. Fig. 1 represents an instance of a shared collaboration space populated by a set of workers and resources, during concurrent use of the tool. Fig. 1 is divided into three main sections: a virtual work group (1A), the CWS (1B) and the project repository (1C).

The 'virtual work group' is composed of four developers who work in different geographical locations. Group members focus their efforts on performing a particular activity with a set of associated resources and collaborators within a particular project. Each ensemble represents a working sphere [32] (e.g. Jimmy is working on activity *A*1 of the 'ERPPharmacy' project modifying document *D*1).

CWS are composed of a set of individual working spheres (WS) (Fig. 1*b*). The information in each WS can be used to identify who is working on: the same activity and the same project (SACT/SP); different activity and same project (DA/SP); and different activity and DP (DA/DP).

The 'project repository' contains information about project activities. These activities are associated with documents (e.g. project 'ERPPharmacy' contains activity *A*1 which is associated with documents *D*1 and *D*2).

In the case shown (Fig. 1), Jimmy (project manager) is working on document *D*1 (issues.ppt), a document that is associated with activity *A*1 (integrated report) from the ERPPharmacy project. Similarly, Austin (tester) is working on activity *A*1, which was detected when Austin opened document *D*2 (Effort.xls) and started working on it. The CWS therefore categorises Jimmy's and Austin's WS as SACT/SP. In contrast, John (programmer) is working on the SP as Jimmy and Austin but on a DA, *A*2 (coding of inventories), and on document *D*3 (InvMod.sln). The CWS distinguishes that John is working on a DA, but on the same project (DA/SP) as Jimmy and Austin. In addition Frank (analyst) is working on activity *A*3 (Use Cases) of the 'Digital Library' project. This was identified from his manipulation of document *D*3 (cases.vsd). The CWS thus categorises Frank's WS as a DA, and DP (DA/DP). Based on these elements, let us consider the following projected scenario.

3.3 Projected scenario

In this scenario, Jimmy (project manager) is assigned to various projects within the software development company. During a review of some of the documents relating to project P1 ('ERPPharmacy'), he realises that there is an error in a design for which Frank (Analyst) is responsible. When Jimmy opens the document, the CWS-IM system (an extended IM application) detects that the document is associated with activity *A*1 ('Integrated report') of project *P*1 and creates a contact group related to it. In this case, Jimmy reviews the contact list and realises that Frank is working on activity *A*3 ('Use Cases') of project *P*2 ('Digital Library'), John (programmer) is coding a module of project *P*1 and Austin (tester) is working on activity *A*1 of project *P*1. This information was easily and quickly determined by simply opening the project document. Jimmy was able to use the CWS-IM system to determine the different situations of the working group by reading the following user status:

Available: Available for those colleagues working in the SACT and on the SP (e.g. Austin).

Reachable, but busy: Reachable to be contacted by workers from the SP, but letting them know that the user is busy doing a DA (e.g. John).

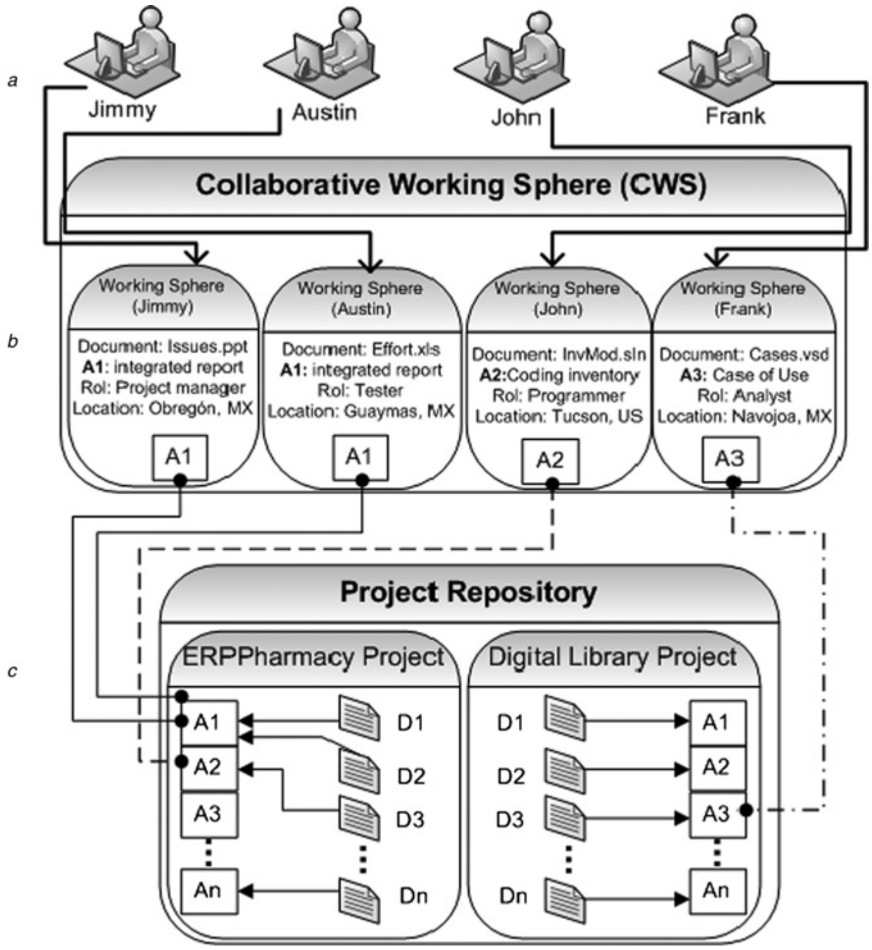

Fig. 1 *Representation of a shared collaboration space*

a Virtual working group
b The CWS
c The project repository

Busy: Currently performing activities for another project, but online if urgent contact needed to be made (e.g. Frank).

In this case, Jimmy knows that Frank is directly responsible for the artefact; however, given Frank's current activity (*A*3 of *P*2), CWS-IM shows him as Busy. Furthermore, Jimmy knows that Austin is working on the SACT (*A*1 of *P*1) since the system shows that he is available. This information enables Jimmy to decide not to contact Frank at that moment. Instead, Jimmy decides to initiate an interaction with Austin to discuss a possible solution.

As previously noted, CWS-IM promotes SA by providing availability and activity information about the other participants in the group (group view). As illustrated in the projected scenario, the project manager (Jimmy) decides to contact an available colleague (Austin) when he notices that that precise moment is not suitable to contact his first contact option (Frank). Moreover, based on information concerning the current activity of collaborators, it might be possible to promote informed interaction attempts that would establish communication in a more appropriate manner, which would in turn allow the proper coordination of activities to take place.

As shown in this scenario, and as suggested by the results of our scenario-based evaluation of the tool's GUI [27], CWS-IM could be useful for remote workers as it provides information about the individual activities of each group member. Furthermore, we believe that the tool could also be useful for co-located workers, as in addition to the information provided about their physical working environment, the tool may also provide them with fine-grain information about their work in the logical environment (e.g. Integrated Development Environment, Project Management System etc.). The following section provides more details about how CWS-IM looks and works.

4 CWS-IM: instant messaging with selective availability support

We decided to provide DSD developers with SA support by developing CWS-IM, an extended IM system that has been adapted with several mechanisms which gather and provide specific awareness information elements that permit its implementation.

Based on the requirements identified in Section 2, we identified specific features with which to implement the SA concept by means of CWS. The following features were identified for the development of CWS-IM [33]:

- *Knowledge about the progress status per activity:* Provide mechanisms that allow the status of project members and the tasks they are performing to be discovered.
- *Awareness of the status of members per activity:* Provide mechanisms to share and filter project information among colleagues working on a related activity.

- *Coordination in common or dependent work units among members of the activity:* Provide mechanisms that allow the progress level of the tasks that each member of the project is executing to be discovered, along with mechanisms that permit the location of, and interaction with, members of common or dependent work units.
- *Initiate an interaction with the right person at the right moment:* Provide mechanisms to identify when one user may interact with another, based on the needs profile, status and activity under execution, along with mechanisms for asynchronous and synchronous communication.

Further details concerning the design of the GUI of the CWS-IM system, and a scenario-based evaluation on the perception of usefulness and ease of use, are reported elsewhere [27, 31].

4.1 Support for selective availability

The CWS-IM tool allows collaboration to be initiated with target collaborators in a more informed manner by means of the implementation of the SA criterion. The issuer can use this to estimate whether the time is right to address a potential receiver. If he interprets that the receiver is working on the SACT and on the SP, that is, performing work closely related to the reason for initiating the interaction, then it might be expected that, considering the receiver's context, she will be more predisposed to deal with the request made by the issuer. It is thus assumed that she will have better conditions in which to respond to the request ('willingness to interact' [30]).

On the other hand, if the issuer knows that the potential receiver is working on the SP but on a DA, he may draw information from the receiver's role (icon illustration), signifying that if the issuer's interest is related to the role that the receiver is currently playing, the issuer may decide to establish communication at that time, or may otherwise decide to wait for a more opportune moment. Finally, when the potential receiver is working on a DP, the issuer can interpret that it is more likely that the receiver's context at that time is not right to address his request, and decide to contact her only in urgent situations.

SA will thus allow users to take into consideration that a user is available to collaborators whose activity is related to the activity she is currently dealing with and not (or less) available to other collaborators. This can certainly not be stated as a rule, as there will be times when the interaction attempt will interrupt or disturb. However, we consider that it increases the possibility of incrementing the receiver's willingness to interact.

4.2 Colour coding and role icons for selective availability

In this implementation SA is achieved by representing a user's status with a colour code (see Fig. 2, notice that labels to indicate the icon's colour in a black and white image were not added, e.g. Green icon. These labels are not shown in the actual implemenation.): if the user's status is presented in green it means that the users are working on the SP and on the SACT (Available); if the status is represented in yellow it means that the users are working on the SP, but on different activities (Reachable, but busy); and if the status is presented in red it means that they are working on DPs (Busy). With this, we aim to provide information elements during a situation of distribution to enable an awareness level to be achieved based on

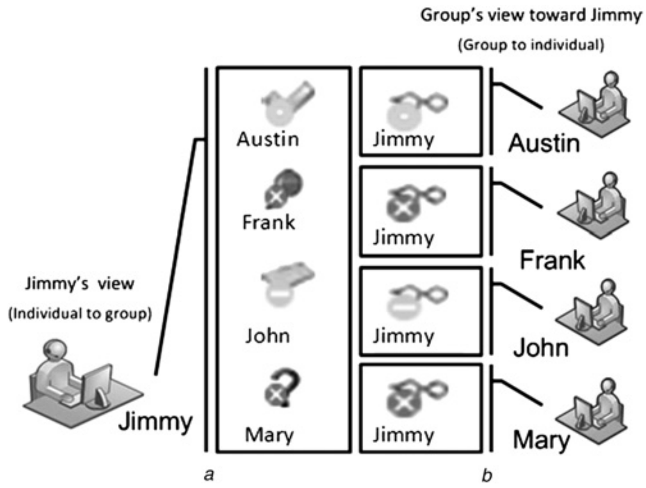

Fig. 2 *Perspectives from the contact list*
a Jimmy's view: individual to group
b Group's view: group to individual

information that is similar to that which is available to an issuer in a situation of co-location, for example, by simply observing the potential recipient.

In order to interpret the work situation in a shared collaboration space (e.g. Fig. 1), it is necessary to understand the colour coding and role icons proposed for CWS-IM. This interpretation has two perspectives: 'from the individual to the group' and 'from the group to the individual'. An illustrative application scenario follows:

'In this case, Jimmy (project manager) is accessing the document 'issues.ppt', and receives a list of potential contacts that are related to the project he is working on at that time. The system simultaneously provides him with information regarding the roles that his colleagues are currently playing. This is achieved by means of role icons (Check mark = Tester, Magnifying glass = Analyst, Keyboard = Programmer, eyeglasses = Project manager and Question mark = No specific role) and the colour codes explained above. Jimmy may thus use the colour associated with the icon to establish whether his co-worker is working on the SP and on the SACT (Available, green), on the SP but on another activity (Reachable but busy, yellow) or performing a DA in a DP (Busy, red). Jimmy may also consider the role information in order to decide whether or not to initiate an interaction with his colleague (e.g. if the reason for the interaction is related to '*A*1' (integrated report), the best time to contact a co-worker is probably when her activity is related to the SP or when her role is that of a 'project manager').

It is important to stress that the role shown to each member of the group represents the potential receiver's role based on her current activity and project, and this from the perspective of the potential issuer. Thus, if a user is playing several roles (not concurrently – e.g. because she is working on multiple projects or activities at different times) the tool will only show the role information regarding the project and activity she is working on at that particular time (a special case is when a developer is performing activities that do not correspond to any of the joint projects – represented by means of a question mark icon). It should also be stressed that in the current implementation of CWS-IM, the information regarding the role, activity, project and status is automatically gathered, inferred and delivered by the system, and the user has no control over the status change. The rationale behind this was based on our aim to preserve the actual reciprocity of subscriptions in CWS-IM ('You can see actual changes in my status if and only if I can see actual changes in your status').

By considering the perspective from the individual to the group ('how I see the group'), Jimmy becomes aware of the current role and activities of his co-workers, as illustrated in Fig. 2*a*. On the other hand, by considering the perspective from the group to the individual ('how the group sees me'), as is shown in Fig. 2*b*, Jimmy's role (project manager) is shown to his co-workers by decorating the eyeglasses icon with different colours in order to depict the difference in the status. In the example, Austin sees Jimmy as working on the SACT and the SP (Green), whereas Frank sees him working on a DA in a DP (Red). Furthermore, John sees Jimmy as working on the SP but a DA (Yellow), whereas Mary sees him as being busy (Red).

With this model, the system automatically customises the information provided and presented to each co-worker considering the current 'role–activity–project' relation between the issuer and the potential receivers of the interaction (i.e. SACT/SP, DA/SP, DA/DP). The implications for this are two-fold: (i) to provide information – once the user has explicitly accepted that she will participate in the bi-directional information sharing among project co-workers (one subscription per project), the system obtains the required information (e.g. role, activity, project, status) in an implicit manner while the user performs her assigned work (e.g. monitoring artefact use); and (ii) to obtain information – the user only has to 'glance' at the visual representation of the tool to gather information on presence, status or role, and based on this information, she is able to determine whether the moment is suitable for both the issuer and the receiver to start an interaction. Moreover, we decided not to provide information on 'glances' whenever a potential issuer checks the status of a potential receiver in order not to create an additional source of interruption for the latter, and this is without the certainty existing that the glance will result in an interaction attempt.

4.3 Current implementation of CWS-IM

The current version of CWS-IM is implemented as an extended IM system in C# .NET 2008. It uses the agsxmpp library [34], which provides the functionality required to connect to an XMPP server (Openfire 3.5.1), retrieve the list of registered users and artefacts, and send and deliver instant messages (see Fig. 3).

It is worth mentioning that this implementation uses data from the DIGITÉ's project management system [35] deployed in the software development company.

The CWS client automatically determines which files have recently changed on the client side by listening to the OnChanged (Creation, Change and Delete) and OnRenamed (Rename) file events of the FileSystemWatcher class of the .NET Framework Class library, while it determines which application and files are currently in use (receiving focus at the user screen) by using the GetForegroundWindow method from the Windows User32.dll (dynamic link library). The classpath of the file that is currently in use or has recently been modified is then sent to the Activity Presence Server by means of an XMPP-based message. Once there, it is compared with the information on the dataview (SQLServer) to determine whether the file is associated with an activity or project. Once the activity and/or project have been determined, information about the participants involved is obtained from the dataview (SQLServer). This information is used to add an entry to the activity log to register that user *U* is currently working on document *D* associated with activity *A* from project *P*. This information is also automatically forwarded to the User and Artifact Presence Server by means of an XMPP-based message, which in turn updates the user and artefact statuses and sends a notification to the CWS clients of all involved users by means of another XMPP-based message. Finally, the actual data used to populate the GUI components (e.g. project names, assigned activities, involved users and documents, start and ending dates etc.) are retrieved by the CWS client from the dataview (SQLServer) through an HTTP connection (IIS WebService).

Having established how the SA concept was implemented, in the following section we present the results of an evaluation of the actual use of the CWS-IM tool in a DSD setting.

5 Evaluation of actual use

The CWS-IM tool was evaluated through its use in the software development company at which the initial

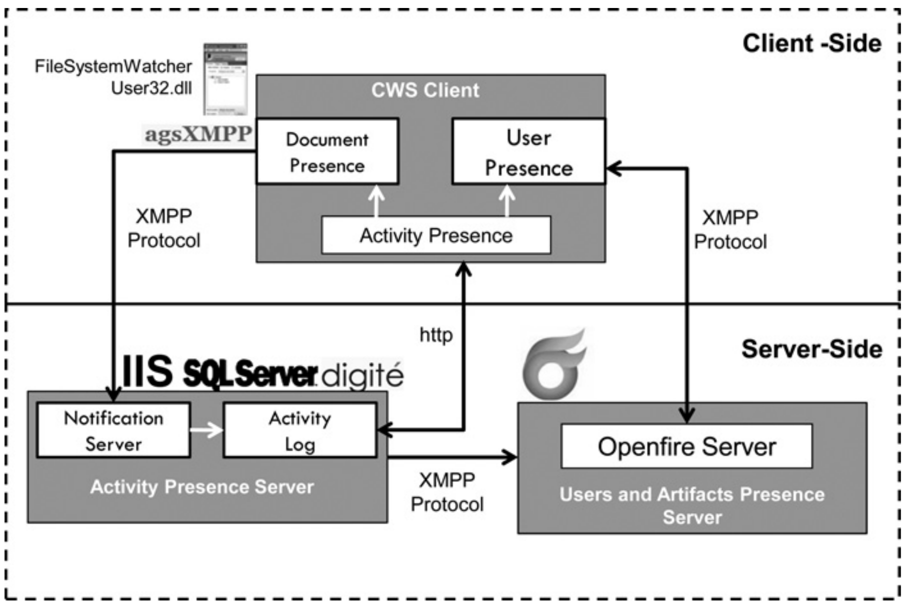

Fig. 3 *Architecture of CWS-IM*

IET Softw., 2012, Vol. 6, Iss. 3, pp. 185–198
doi: 10.1049/iet-sen.2011.0077

understanding study was conducted (see Section 2). Our aim was to discover how developers perceived the SA criterion during its actual use in their work environment. We also intended to determine how a more informed interaction initiation might help DSD workers to interpret when the initiator's efforts are aligned with those of the receiver.

5.1 Experimental design and procedure

The following hypotheses were established in order to explore whether the CWS-IM system provides adequate support for the initiation of collaboration at appropriate times, and whether this supports the coordination of activities:

With regard to determining the impact of SA on the DSD environment:

H1a: The participants will perceive that the CWS-IM system is more useful than the traditional IM applications during the execution of their DSD activities.
H1b: The participants will perceive that the CWS-IM system is easier to use than the traditional IM applications for the execution of their DSD activities.
H1c: The participants will perceive that the CWS-IM system is more suitable for adoption than the traditional IM application for DSD work.

Regarding initiating collaboration:

H2a: Participants acting as issuers will tend to gather more information regarding the status of the receiver when the status of the potential receiver is 'DA/SP' or 'DP' than when the status is 'SACT/SP'.
H2b: The participants will perceive that interruptions were more detrimental when using the traditional IM than when using the CWS-IM system.

5.2 Participants

The evaluation was conducted with a group of distributed software developers from the aforementioned Mexican software development company at which the initial understanding study was conducted (see Section 2).

The evaluation involved 16 workers, including 4 project leaders, 2 testers, 4 programmers and 6 software architects, distributed among the four cities (six in Obregon, three in Guaymas, four in Navojoa and three in Tucson). This distribution signified that the participants from the same city knew each other personally, but no participant knew all the other participants from the other cities personally, although some of them had worked together on previous DSD projects. They were divided into four teams according to their assigned activities. The average age of the sample was 24.5. There were ten women and six men. Their average work experience in DSD was 2.9 years (minimum 1 year, maximum 10 years). Regarding IM usage experience, most participants stated that they had been IM users since they were at high school, and this reflected an average experience of 7.4 years.

During the 3-week study, the participants worked on 33 different activities of the company's 11 projects. The average number of assigned projects per collaborator was 2.1 projects. During this period, the participants were working on the development of custom mobile and desktop applications for one of the software development company's commercial clients (ASP .NET and C# .NET billing and payments applications, with database backends on SQL Server 2000/2005).

5.3 Setting

The design paradigm for this study was within subjects (i.e. all the subjects participated in all the conditions). To avoid the carryover effect, two teams (50% of the participants) started using CWS-IM for a workweek and then switched to the traditional IM system for an additional workweek, whereas the other two teams (the other 50%) started using the traditional IM system and then switched to CWS-IM. Regarding IM communication, during the study the participants only used the IM tools provided: CWS-IM or a traditional IM client. The latter provided typical IM functionality, including identity, presence and availability awareness, private and group online and offline text messaging and file transfer. As regards other means, they had access to other communication tools (e.g. email, telephone etc). The participants performed their development work in their usual spaces at the software development company.

5.4 Procedure

A total of six activities were conducted for the evaluation (see Fig. 4). These are described below:

1. *Initial meeting:* We conducted a general meeting with the participants in order to inform them of the intention of the study, obtain their agreement to participate and explain what their participation would be and how it would take place.
2. *Preliminary interview:* We conducted a preliminary interview with each of the subjects to identify what their roles in the software development company were.
3. *Systems installation:* We installed the CWS-IM and the traditional IM systems, one system per work team.
4. *Training:* The subjects were individually trained in the use of each system. During the training session an individual ID

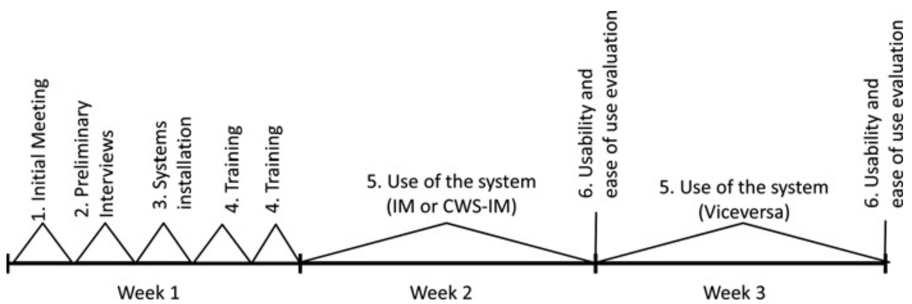

Fig. 4 *Evaluation procedure*

was assigned to them for use with the system. The participants were also asked for the following personal information: age, gender, years of experience (software development, DSD and IM) and number of projects in which they were involved.
5. *Use of the system:* They were asked to use the corresponding system (either CWS-IM or IM) during the whole workweek. They subsequently changed their system, and worked with the other one for an additional workweek.
6. *Usability and ease of use evaluation:* Upon completing the use of each system, the participants were asked to answer the SUS (System Usability Scale) [36] and TAM (Technology Acceptance Model) [37] questionnaires.

5.5 Results and discussion

This subsection shows and discusses the results of the study carried out to determine the effect that the SA criterion had on developers: whether or not the CWS-IM tool helped them to initiate interactions in a more appropriate manner than traditional IM systems.

The information gathered from the SUS and TAM questionnaires was processed through the use of descriptive statistics and means comparison.

We worked with the following question and related hypotheses:

What is the impact in terms of the perception of usefulness and ease of use in starting an interaction in an informed manner in DSD?

H1a: The participants will perceive that the CWS-IM system is more useful than the traditional IM applications during the execution of their DSD activities.
H1b: The participants will perceive that the CWS-IM system is easier to use than traditional IM applications for the execution of their DSD activities.
H1c: The participants will perceive that the CWS-IM system is more suitable for adoption than the traditional IM application for DSD work.

We first used the SUS questionnaire and the first six questions of the TAM questionnaire focusing on perceived usefulness to address H1a.

The SUS questionnaire uses a Likert scale of 1–5 (with a range from 0 to 100). The results from the SUS questionnaire indicate that CWS-IM (mean = 82.5, sd = 2.02892) was perceived as being more useful than the traditional IM system (mean = 70.94, sd = 3.05232). This difference was validated by conducting a Student's *t*-test with a significance level $p < 0.01$. Given that $t_{0.01} = -2.147$, we can establish that this difference is significant.

The TAM questionnaire uses a Likert scale of 1 (strongly disagree) to 7 (strongly agree), considering an acceptable usability level to be in a range of 25–42. The average (standard deviation) of the responses to the TAM questions is shown in Table 3.

The averages for the TAM questionnaire as regards its usefulness factor (first six items) indicate a trend favouring CWS-IM (mean = 35.31, sd = 4.51248) in comparison to the traditional IM system (mean = 24.69, sd = 8.81452). This difference was validated by conducting a Student's *t*-test with a significance level of $p < 0.01$. Given that $t_{0.01} = -4292$, we can establish that this difference is significant. These results thus signify that H1a can be accepted.

In order to validate H1b and H1c, we worked with the last six questions of the TAM questionnaire which focused on perceived ease of use (see Table 3). We found similar average values regarding ease of use for both systems, with a trend favouring the CWS-IM (mean = 35.69, sd = 4.55659) in comparison with the traditional IM system (mean = 32.75, sd = 5.97216). However, a Student's *t*-test confirmed that these differences were not significant ($t_{0.01} = -1.564$). The results indicate that both systems were considered easy to use and were perceived as likely to be adopted. These results thus signify that H1b can be accepted. Finally, the participants considered that CWS-IM was more useful (H1a) than the traditional IM and as easy to use (H1b), and H1c is thus accepted.

With regard to the question How can DSD workers initiate collaboration in an informed manner which is appropriate both for the issuer and the receiver, while they perform their development activities?, we firstly aimed to understand the behaviour of the sender in the CWS-IM condition when he realised that the receiver was doing a DA (status 'DA/SP' or 'DP') or doing the SACT (status 'SACT/SP'). We therefore evaluated the following hypothesis:

H2a: Participants acting as issuers will tend to gather more information regarding the status of the receiver (through the 'Contact Information ToolTip' component of the CWS-IM tool) when the status of the potential receiver is 'DA/SP' or 'DP' than when the status is 'SACT/SP'.

In order to obtain data and test this hypothesis, we kept logs of the following information: date, time and the specific action performed on the contact list component of the CWS-IM tool. This information was collected whenever the user was in the issuer position and acted on the contact list. The actions that we collected were: double clicks on the potential receiver (start interaction) and single right button clicks on the receiver (Contact Information ToolTip). It is important to mention that all the information collected came from the CWS-IM system. We collected 455 data points, and found that there were 354 of them for the 'SACT/SP' status, 62 for the 'DA/SP' status and 39 for the 'DP' status. The data were codified in the following way: If there was an interaction with the 'Contact Information ToolTip' component the data point was coded with a one (1) value, and if the action was a double click on the contact list the data point was coded with a two (2) value.

The results obtained after analysing the interaction with the Contact Information Tooltip (with a range from 1 to 2) indicate that the users were more likely to request additional information about the receiver when they had a status of 'DA/SP' or 'DP' (mean = 1.71, sd = 0.442) than when they had a status of 'SACT/SP' (mean = 1.51, sd = 0.502). A Student's *t*-test with a significance level of $p < 0.01$ was conducted to confirm this information. Given that $t_{0.01} = 4.268$, we can establish that this difference is significant, and H2a is thus accepted. In this case we can argue that the users understood the information provided by CWS-IM.

These results indicate that the tool provides the issuer with sufficient information to allow him to determine a receiver's current activity, and contact her only when he considers it to be appropriate (e.g. they are working on similar activities). This in turn may help the receiver to reduce problems related to hidden costs caused by the interruption, such as prospective memory failure [19] and mental workload [38].

Table 3 TAM results

TAM		
Questions	CWS-IM	IM
Usefulness		
Q1: Using the system would enable me to accomplish tasks more quickly	6.06 (0.680)	4.06 (1.982)
Q2: Using the system would improve my performance	5.81 (0.911)	3.81 (1.759)
Q3: The system would make it easier to do my work activities	5.50 (1.155)	3.81 (1.559)
Q4: Using the system would enhance my effectiveness at work	5.63 (1.147)	3.81 (1.682)
Q5: Using the system would increase my productivity	5.44 (1.031)	3.88 (1.586)
Q6: I would find the system useful in my work	6.19 (0.834)	4.44 (1.632)
Ease of use		
Q7: I find it easy to get the system to do what I want it to do	5.44 (1.094)	3.88 (2.094)
Q8: My interaction with the system is clear and understandable	6.13 (1.025)	6.31 (1.250)
Q9: Learning to operate the system would be easy for me	6.50 (0.632)	6.31 (1.537)
Q10: It would be easy for me to become skillful at using the system	6.50 (0.816)	6.69 (0.479)
Q11: I would find the system easy to use in my work	6.19 (0.750)	4.69 (1.448)
Q12: I would find the system flexible to interact with	5.81 (0.750)	5.75 (1.915)

In order to address the receiver's perspective, we considered the degree of benefit or detriment that she perceived from the interruption, based on the status of the issuer. We thus evaluated the following hypothesis:

H2b: The participants will perceive that interruptions were more detrimental when using the traditional IM than when using the CWS-IM system.

This was evaluated by applying a random mini-survey that appeared at the beginning of the interaction (receiver) with a specific question asking 'To what extent has this interaction been detrimental to you?' and 'To what extent has this interaction been beneficial to you'. We used a 5-point Likert scale for these questions and kept a log with these data plus the information of the status, date and time. The data were collected when the user was acting as the receiver of the interaction. We collected 105 data points. The data were grouped on the basis of the users' status (issuer and receiver) at the moment of the interaction.

The results from the detrimental level perceived (from the mini-survey) were used for comparison, using a Student's *t*-test. The differences of means between the traditional IM system (mean = 2.04, sd = 1.115) and the CWS-IM system (mean = 1.32, sd = 0.47), resulted in a statistically significant difference ($t_{0.01} = 5888$), and H2b is thus accepted. We can therefore state that the interactions performed using the traditional IM system were perceived to be more detrimental than those performed using CWS-IM. A possible explanation for this may be that CWS-IM provides the sender with information concerning the current activity of the receivers, which allows him to interpret their work situations, and thus provides him with more elements to decide whether the time is right to initiate collaboration. This is supported by the results of hypotheses H1a (usefulness), H1b (ease of use), H1c (adoption) and H2a (understanding of the current activity).

5.6 *Limitations: threats to validity*

In this subsection we present some of the limitations of this evaluation, regarding internal and external validity.

5.6.1 Internal validity: With regard to data collection, this study was conducted by one researcher, and the data were collected through standardised SUS [36] and TAM [37] tests, which the participants answered anonymously. Furthermore, the participants were randomly divided into two groups, as the selected design paradigm was the within subject paradigm. We ensured that the participants had to carry out activities in at least one common project during the evaluation period. To avoid the carryover effect, one group started with the CWS-IM system and the other with the traditional IM system. The participants were subsequently asked to switch systems.

With regard to user perception, there was no possibility of the researcher influencing the participants' perception, because the researcher was not present during the use of the system (a workweek in each participant's work setting). None of the participants in this study had a previous relation or conflict of interest with the researcher.

5.6.2 External validity: This study was conducted at a software development company with CMMI level 3 and in a DSD group with participants located on three sites in Sonora, Mexico and one site in Arizona, USA. This software development company was chosen because its production processes are based on internationally recognised standards such as: PMI, CMMI, RUP and UML. However, the results concerning the preference of CWS-IM for DSD environments depend on multiple contextual variables, including organisational factors (e.g. structure, dependency relationships, remunerations etc.) and on factors relating to the work unit (e.g. familiarity, knowledge etc.). These results might also be valid in other similar software development companies. Nonetheless, we cannot guarantee that these results could be generalised to other work settings. Our intention is to establish an initial set of results that may serve as a foundation from which others could start to explore this topic at greater length.

6 Related work

Although the software industry has acquired different tools [11] to enable developers to coordinate their activities during the different stages of development, some studies have recently been conducted on how these tools are used to coordinate activities in the work environment (e.g. [10]). This has prompted researchers to conduct studies to understand coordination problems during the execution of their activities. This has been addressed through contributions regarding awareness that intend to keep the development group informed, in a quick and easy manner, as regards their activities and their use and creation of artefacts. There are also proposals for the redesign of traditional collaboration tools (e.g. email and IM) so that, in addition to allowing communication to be initiated and maintained, these tools could also provide information and follow up the pending activities.

With regard to the work on coordinating DSD activities, there are several works whose intention is to explain how these coordination problems emerge. This has been addressed by understanding the concept of coordination in DSD within an organisational approach. For example, Cataldo *et al.* [25] present four case studies in which coordination problems are identified in GSD (a special case of DSD), under the approach of modularity of tasks. A further contribution is a framework for the analysis of coordination in GSD [11]. The framework leads to an understanding that in GSD the management of interdependencies of the process, inter-unit and inter-personal conflicts, information uncertainty and ambiguity, and representations of technology will be significantly constrained by distance, technology and other socio-cultural aspects. An analysis in this direction was performed in [11], in which six types of development coordination tools were conceptualised. The results of this work identify some insights related to the use of several tools for the coordination of software development. Qualitative studies that pay special attention to distance and the unwillingness to cooperate have also been conducted (e.g. [39]). In general, these works seek to detect activity coordination problems in GSD from a theoretical approach. To the best of our knowledge, there are no other works that evaluate aspects of communication from the perspective of appropriate interaction initiation in DSD environments, and we thus consider our proposal to be complementary to current efforts towards DSD activity coordination.

With regard to the work on awareness in DSD, several works focus on reporting the activity taking place within a development group. For example Palantir [40] is a system that complements existing configuration systems by providing distributed awareness of the project's progress. This is done through a graphic display which shows measurements of severity and the impact of changes on the artefacts, thus permitting developers to anticipate the problems that may result from those changes. ProjectWatcher [41] is a system designed to provide awareness support regarding two questions ('who is who in general?' and 'who works in this area of the code?'). There are also mechanisms (e.g. Eclipse Plugging [42]) to inform users about the location of a concurrent change at the class, method and line level. Tesseract offers an interactive exploratory environment in which developers can investigate relationships of source code, errors or failures, e-mail and discussions around the error [43]. It is also possible to discover each developer's contributions. In this respect, our proposal provides a new manner in which to automatically reflect user status changes by providing (near) real-time information about potential collaborators' current activities. It also provides a more detailed perspective by means of different availability statuses based on the activities of both the interaction issuer and the receiver (SA). That is, the user is shown as being available to a collaborator working on the SACT and SP, but busy to collaborators working on other projects. Furthermore, none of the cited works focuses on informing a group with this granularity (dynamicity and level of detail) about the activity that individuals perform in their own workspace.

With regard to work on initiating collaboration, various communication tools have been proposed for software development environments. These include ProjectView-IM [17], which suggests changes to the interface design of an IM application considering three characteristics: presence awareness (i.e. online, away and offline), user information and reminders. This design facilitates the division of attention between multiple projects without penalising a primary task. ProjectView-IM is focused on monitoring personal trajectories by acting as a reminder about the activities to be performed. It is also possible to know how many people are working on a project at any given time. However, during interaction initiation it does not inform the issuer about the type of activity and the role played by co-workers, nor does it keep different perspectives of the users' statuses with regard to their current activities as CWS-IM does. Rear view mirror (RVM) [44] is a presence-based IM with features that help to support work teams. RVM offers the creation of groups. These groups are similar to buddy lists with the exception that buddy lists are defined independently by each user. Any user can create any number of groups to interact. The group owner becomes the manager. Groups can be open or restricted as decided by the manager. RVM focuses on managing working groups, but these groups still have to be created manually. In contrast, in CWS-IM the groups are created dynamically and automatically based on the assigned activities in the project management system, to which our tool is linked. CWS-IM also provides insights into the activities in which the user is currently involved, thus allowing a user to discover whether someone is working on a certain activity at a specific moment. It is also possible to identify when a co-worker starts an activity in which other workers have a special interest, so as to trigger or initiate interaction. Finally, the aforementioned tools for the initiation of collaboration were not designed to attempt to establish an interaction that is aligned to the interests of both the issuer and the receiver of the interaction. This is possible with CWS-IM, as it considers information regardless of the stage of the project (e.g. planning or execution) in order to keep other collaborators informed about it. Furthermore, the design of CWS-IM is based on the features of DSD activities and on the CWS model presented in Sections 2 and 3.

7 Conclusions and future work

DSD provides many benefits that make it an attractive paradigm for the software industry. These benefits include continuous work on projects, lower labour costs and the rapid creation of development teams that are geographically close to the customer, among others. However, DSD also creates an ensemble of challenges that are inherent to its characteristics, including lack of opportunities for face-to-face interaction, distributed knowledge management and lack of awareness and trust relations among the distributed team, which may lead to misunderstandings. These challenges can only be met with new coordination methods, tools and processes.

In this work, an initial understanding of the DSD features and an analysis of the practices of a DSD team were used as a starting point to identify an ensemble of coordination problems that DSD developers confront in the execution of their activities. In order to deal with these problems, we

doi: 10.1049/iet-sen.2011.0077

identified a set of requirements and design insights which provide evidence for the need for informal communication support in DSD environments as a means to improve collaboration and trust among distributed individuals. These requirements and insights have been used as a basis for the introduction of the SA criterion as a means to provide the issuer of an interaction with detailed information regarding the status of the receiver, so that she can decide whether the moment is appropriate for both of them to start an interaction. We have also presented current and projected scenarios to illustrate the concept. These insights and the proposed concept were later used as a starting point for the design and development of CWS-IM, an extended IM application that implements support for SA. The actual use of CWS-IM, was evaluated through its use by a DSD team in a software development company, where a 3-week study was carried out. Perception of usefulness and ease of use were evaluated with 16 developer participants by means of SUS and TAM scales.

Evidence from the evaluation results indicates that the SA concept was well received by the participating DSD developers. They perceived that CWS-IM was more useful than and as easy to use as traditional IM applications in the execution of their DSD activities.

It is important to note that the functionality provided by CWS-IM, although well perceived by the potential users, is by no means comprehensive. For instance, our tool requires both the issuer and the receiver to be working at the same time in order to benefit from the proposed functionality (as when there is a time zone overlap between the users).

CWS-IM could deal with this if it were enhanced with mechanisms that allow the issuer to be informed of any of the receiver's important events (e.g. the potential receiver is about to leave work after finishing her workday or is about to enter a work meeting). A special case of the former would be to include information on workday overlaps among team members from distributed sites with important local time differences. This could be achieved by means of a special animation icon which indicates this. We may additionally include a mechanism that enables the receiver to manually configure her SA status regardless of her current activity (e.g. only available to the project manager, and unavailable to all others, including those performing the SACT in the SP). In this case it would be interesting to study how the participants would use this feature, and how they would perceive that SA information could be shared either in a 'configured' or an 'automatic' manner.

Finally, with regard to our future work, among other things we plan to include the functionalities discussed above to investigate whether there are differences in the perceived usefulness and ease of use of the tool among co-located and distributed participants in the same team, and to refine our approach to the evaluation of the CWS-IM system in order to be able to determine whether the information provided by SA could be taken in account by the receiver to establish the value of an interaction. Our hope is that with concepts like SA and CWS, and with tools like CWS-IM, we may contribute to the provision of better support for communication and coordination in DSD efforts.

8 Acknowledgments

The authors would like to thank Novutek, S.C. and the 16 participants for their valuable support and participation that made the evaluation possible. This work was supported by the PEGASO/MAGO project (Ministerio de Ciencia e Innovación MICINN and Fondos FEDER, TIN2009-13718-C02-01), by ENGLOBAS (PII2I09-0147-8235) funded by Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha), and by Fondos FEDER and ORIGIN [IDI-2010043 (1-5)] funded by CDTI and FEDER. This work was also supported by UABC under grant 207 of the Convocatoria Interna de Proyectos de Investigación, by PROMEP under grant 103.5/11/6698 of the Convocatoria de Apoyo a la Reincorporación de Exbecarios 2011 and by Asociación Mexicana de Cultura A.C.

9 References

1 Prikladnicki, R., Audy, J.L.N., Evaristo, J.R.: 'Distributed software development: toward an understanding of the relationship between project team, users and customers'. Proc. Fifth Int. Conf. on Enterprise Information Systems (ICEIS' 03), 2003, pp. 417–423

2 Carmel, E., Agarwal, R.: 'Tactical approaches for alleviating distance in global software development', *IEEE Softw.*, 2001, **18**, (2), pp. 22–29

3 Herbsleb, J., Moitra, D.: 'Guest eds.' Introduction: global software development', *IEEE Softw.*, 2001, **18**, (2), pp. 16–20

4 Conchúir, E., Holmstrom, H., Agerfalk, P., Fitzgerald, B.: 'Exploring the assumed benefits of global software development'. Proc. IEEE Int. Conf. on Global Software Engineering (ICGSE'06), 2006, pp. 159–168

5 Lloyd, W., Rosson, M., Arthur, J.: 'Effectiveness of elicitation techniques in distributed requirements engineering'. Proc. Tenth Anniversary IEEE Joint Int. Conf. on Requirements Engineering, RE'02, 2002, pp. 311–318

6 Damian, D., Moitra, D.: 'Guest editors' introduction: global software development: how far have we come?', *IEEE Softw.*, 2006, **23**, (5), pp. 17–19

7 Bellur, U.: 'An academic perspective on globalization in the software industry'. Proc. 30th Annual Int. Computer Software and Applications Conf., (COMPSAC'06), 2006, pp. 52–54

8 Damian, D., Zowghi, D.: 'The impact of stakeholders'? Geographical distribution on managing requirements in a multi-site organization'. Proc. Tenth Anniversary IEEE Joint Int. Conf. on Requirements Engineering, 2002, pp. 319–330

9 Babar, M., Verner, J., Nguyen, P.: 'Establishing and maintaining trust in software outsourcing relationships: an empirical investigation', *Syst. Softw.*, 2007, **80**, (9), pp. 1438–1449

10 Wiredu, G.O.: 'A framework for the analysis of coordination in global software development'. Proc. 2006 Int. Workshop on Global Software Development for the Practitioner, 2006, pp. 38–44

11 Tiwana, A.: 'Impact of classes of development coordination tools on software development performance: a multinational empirical study', *ACM Trans. Softw. Eng. Methodol.*, 2008, **17**, (2), pp. 1–47

12 Herbsleb, J.D., Mockus, A.: 'An empirical study of speed and communication in globally distributed software development', *Softw. Eng.*, 2003, **29**, (6), pp. 481–494

13 Stinchcombe, A., Heimer, C.: 'Organization theory and project management: administering uncertainty in Norwegian Offshore Oil' (A Scandinavian University Press Publication, 1986)

14 McChesney, I.: 'Effective coordination in the software process – historical perspectives and future directions', *Softw. Qual. Control*, 1997, **6**, (3), pp. 235–246

15 Kraut, R., Fish, R., Root, B., Chalfonte, B.: 'Informal communication in organizations: form, function and technology'. Proc. People's Reactions to Technology in Factories, Offices and Aerospace, The Claremont Symp. on Applied Social Psychology, 1990, pp. 145–199

16 Layman, L., Williams, L., Damian, D., Bures, H.: 'Essential communication practices for extreme programming in a global software development team', *Inf. Softw. Technol.*, 2006, **48**, (9), pp. 781–794

17 Fussell, S., Kiesler, S., Setlock, L.D., Scupelli, P.: 'Effects of instant messaging on the management of multiple project trajectories'. Proc. SIGCHI Conf. on Human Factors in Computing Systems, 2004, pp. 191–198

18 Czerwinski, M., Horvitz, E., Wilhite, S.: 'A diary study of task switching and interruptions'. Proc. SIGCHI Conf. on Human Factors in Computing Systems, 2004, pp. 175–182

19 Ellis, J., Kvavilashvili, L.: 'Prospective memory in 2000: past, present and future directions', *Appl. Cogn. Psychol.*, 2000, **14**, pp. 1–9

20 Dabbish, L., Kraut, R.E.: 'Controlling interruptions: awareness displays and social motivation for coordination'. Proc. 2004 ACM Conf. on Computer Supported Cooperative Work, 2004, pp. 182–191

21 González, V., Galicia, L., Favela, J.: 'Supporting the planning and organization of multiple activities in the workplace'. Proc. INTERACT 2007, 2007, pp. 235–238

22 Morteo, R., González, V., Favela, J., Mark, G.: 'Sphere Juggler: fast context retrieval in support of working spheres'. Proc. ENC 2004, 2004, pp. 361–367

23 Camacho, J., Favela, J., González, V.: 'Supporting the management of multiple activities in mobile collaborative working environments'. Proc. CRIWG 2006, 2006, pp. 381–388

24 Morán, A.L., Favela, J., Romero, R., *et al.*: 'Potential and actual collaboration support for distributed pair-programming', *Computación y Sistemas*, 2008, **11**, (3), pp. 211–229

25 Cataldo, M., Bass, M., Herbsleb, J.D., Bass, L.: 'On coordination mechanisms in global software development'. Proc. Int. Conf. on Global Software Engineering IEEE Computer Society, 2007, pp. 71–80

26 Morán, A.L., Favela, J., Martinez, A., Decouchant, D.: 'On the design of potential collaboration spaces', *Int. J. Comput. Appl. Technol.*, 2004, **19**, (3/4), pp. 261–271

27 Palacio, R.R., Vizcaíno, A., Morán, A.L., González, V.M.: 'Tool to facilitate appropriate communication in global software development', *IET Softw.*, 2011, **5**, (2), pp. 157–171

28 Strauss, A., Corbin, J. (Eds.): 'Basics of qualitative research: grounded theory procedures and techniques' (Sage, Newbury Park, EUA, 1990)

29 Kraut, R.E., Streeter, L.A.: 'Coordination in software development', *Commun. ACM*, 1995, **38**, (3), pp. 69–81

30 Ye, Y.: 'Supporting software development as knowledge-intensive and collaborative activity'. Proc. 2006 Int. Workshop on Interdisciplinary Software Engineering Research, 2006

31 Palacio, R.R., Morán, A.L., González, V.M.: 'CWS: an awareness tool to support starting collaboration in global software development', *Open Softw. Eng. J. (special issue on Global Softw. Dev. Chall)*, 2009, **4**, pp. 38–51

32 Gonzalez, V., Mark, G.: 'Constant, constant, multi-tasking craziness: managing multiple working spheres'. Proc. SIGCHI Conf. on Human Factors in Computing Systems, 2004, p. 2004

33 Palacio, R.R., Morán, A.L., Vizcaíno, A., González, V.M.: 'Knowledge flow as facilitator for getting into collaboration in distributed software development'. Proc. 44th Hawaii Int. Conf. on System Sciences (HICSS-44), 2011

34 Software, A.: 'agsXMPP SDK', http://www.ag-software.de/agsxmpp-sdk.html

35 Digité: 'Globally distributed Agile ALM from Digité'. 2003, http://www.digite.com/products/digite-products.htm

36 Brooke, J.: 'SUS: a quick and dirty usability scale'. Proc. Usability Evaluation in Industry, 1996, pp. 189–194

37 Davis, F.: 'Perceived usefulness, perceived ease of use, and user acceptance of information technology', *MIS Q.*, 1989, **13**, pp. 319–340

38 Bailey, B.P., Iqbal, S.T.: 'Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management', *ACM Trans. Comput.–Hum. Interact.*, 2008, **14**, (4), pp. 1–28; doi http://doi.acm.org/10.1145/1314683.1314689

39 Panjer, L.D., Damian, D., Storey, M.: 'Cooperation and coordination concerns in a distributed software development project'. Proc. 2008 Int. Workshop on Cooperative and Human Aspects of Software Engineering, 2008, pp. 77–80

40 Sarma, A., Hoek, v.A.d.: 'Palantír: increasing awareness in distributed software development'. Proc. Workshop on Global Software Development (ICSE 2002), 2002, pp. 28–32

41 Gutwin, C., Schneider, K.A., Paquette, D., Penner, R.: 'Supporting group awareness in distributed software development'. Proc. EHCI/DS-VIS, 2004, pp. 383–397

42 Ignat, C., Oster, G.: 'Awareness of concurrent changes in distributed software development'. Proc. OTM 2008 Confederated Int. Conf., Coopis, Doa, Gada, Is, and ODBASE 2008. Part I On the Move to Meaningful internet Systems, 2008, (*LNCS*), **5331**, pp. 456–464

43 Sarma, A., Maccherone, L., Wagstrom, P., Herbsleb, J.: 'Tesseract: interactive visual exploration of socio-technical relationships in software development'. Proc. 2009 IEEE 31st Int. Conf. on Software Engineering, 2009, pp. 23–33

44 Handel, M., Herbsleb, J.D.: 'What is chat doing in the workplace?'. Proc. 2002 ACM Conf. on Computer Supported Cooperative Work, 2002, pp. 1–10

doi: 10.1049/iet-sen.2011.0077