Computer Supported Cooperative Work (2013) 22:113–158
DOI 10.1007/s10606-012-9164-4
© Springer 2012

Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature

Igor Steinmacher[1,2], Ana Paula Chaves[1] & Marco Aurélio Gerosa[2]
[1]Coordination of Computer Sciences, Federal University of Technology—Paraná, BR 369–km 0,5 CEP: 87301-006, Caixa Postal: 271Campo Mourão, PR, Brazil (E-mail: igorfs@utfpr.edu.br; E-mail: anachaves@utfpr.edu.br); [2]Computer Science Department, University of São Paulo (USP), Rua do Matão, 1010 CEP 05508-090 São Paulo, SP, Brazil (E-mail: igorfs@utfpr.edu.br; E-mail: gerosa@ime.usp.br)

Abstract. The developers' physical dispersion in Distributed Software Development (DSD) imposes challenges related to awareness support during collaboration in such scenario. In this paper, we present a systematic literature review and mapping that gathered, analyzed, and classified studies that improve awareness support in DSD, providing an overview of the area. Our initial search returned 1967 papers, of which 91 were identified as reporting some awareness support to DSD. These papers were then analyzed, and classified according to the 3 C collaboration model and to the Gutwin et al. Awareness Framework. Our findings suggest that awareness in DSD is gaining increasingly attention, 71 out of 91 papers were published from 2006 to 2010. Most part of the papers presented tools with some awareness support. The classification showed that the coordination is by far the most supported dimension of the 3C model, while communication is the less explored. It also showed that workspace awareness elements play a central role on DSD collaboration.

Key words: Awareness, Communication, Coordination, Cooperation, Distributed software development, Systematic mapping, Systematic review

1. Introduction

The software development industry has been gaining competitive advantage in terms of cost and quality by using qualified professionals from all over the world (Robinson and Kalakota 2004). This approach, called Distributed Software Development (DSD), is based on geographically dispersed teams working collaboratively in a software project. The software industry started obtaining—in some cases—higher quality products at lower costs than that of collocated teams (Jimenez et al. 2009).

In DSD, developers interact in a distributed way along the software lifecycle, creating a network of distributed sub-teams. In some cases, these teams comprise members of the same organization; in others, they involve members from different organizations. Therefore, DSD brings new challenges such as contextual, cultural, organizational, geographical, temporal, language, and political differences among developers (Ivcek and Galinac 2008). Traditional face-to-face meetings become less common, and interaction among members requires the use of technology to support communication, coordination, and cooperation (Jimenez et al. 2009). Temporal distance hinders and limits opportunities for direct contact, cooperation, and development of trust (Carmel 1999). Linguistic distance limits the ability for coherent communication (Jensen et al. 2007). Communication and cooperation in DSD projects are normally electronic with limited opportunities for synchronous contact, due to geographical location and time zone differences (Casey 2010). Initiating casual interaction in this scenario is difficult, since people do not see who is around, do not know if others are available for conversation, and lack the awareness cues that naturally lead to serendipitous interaction (Tee et al. 2009). The increasing adoption of DSD and its characteristics raise several challenges and new opportunities for research of new awareness support techniques, as well as the investigation of the applicability of the existing ones. Due to the collaborative nature of DSD, team members need to be aware of activities of others, progress of tasks, status of artifacts, owners of artifacts, etc. The lack of awareness about what is happening in other parts of the project leads to misunderstandings and errors, affecting the work of the whole team (Gutwin et al. 2005).

A great number of studies and recent advances deal with awareness support in distributed development environments (Steinmacher et al. 2010; Treude and Storey 2010; Omoronyia et al. 2010). However, knowledge is scattered among CSCW and Software Engineering literature. This article surveys the current state of the art and highlights research directions, by means of a systematic review and mapping. As a result, we offer an overview of the research area, which can be used as a guide for researchers and practitioners looking for solutions or challenges in DSD domain.

Systematic reviews gather and evaluate the available evidence concerning specific topics (Mian et al. 2005). In contrast to the usual process of literature review, a systematic review aims to reduce bias and follows a precise and strict sequence of methodological steps that relies on a well-defined protocol. The protocol presents the topic being investigated in a very specific, predefined, focused, and structured question, as well as instructions for the selection, analysis, and summarization of relevant papers. The systematic review aims to present a fair evaluation of a research topic by using a trustworthy, rigorous, and auditable methodology, increasing the likelihood of detecting effects that individual smaller studies are unable to detect (Kitchenham 2004).

In this paper, our goal was to identify awareness studies and elements that aim to improve the DSD scenario and classify them according to commonly used groupware perspectives. For the purpose of analysis and classification, we used the 3C Collaboration Model (Fuks et al. 2007) and the Awareness Framework proposed by Gutwin et al. (1996). According to the former, the collaboration is analyzed from communication, coordination, and cooperation points of view. The latter defines a classification for awareness based on a list of elements that people may keep track of when working with others in a shared space.

In addition to the systematic review, we also conducted a systematic mapping, which provides an overview of a research area, identifies the quantity, type of research, and results available within it (Petersen et al. 2008). The goal of our systematic mapping was to check out the evolution of the research area, as well as to find out in which forums the studies of the area are published, and who the main researchers on the area are.

Thus, the main goals of this paper are: to organize and classify the high number of papers found on this review; to provide a quick reference for DSD environment developers and researchers regarding which awareness features have already been investigated and implemented; and to identify missing coverage and opportunities for possible future research. Our contributions also include: characterization and synthesis of a research area; selection of relevant papers; and comprehensive formulation and integration of findings by analysis of selected papers.

This paper is organized as follows. Section 2 presents a discussion regarding DSD, awareness, and the classification models used. Section 3 presents the systematic mapping and review planning. Section 4 and Section 5 present the results of our systematic mapping and review, respectively. Section 6 concludes the paper, with a discussion about our findings and the limitations of our study.

2. Awareness in DSD

Awareness enables participants to pick up what their colleagues are doing and to adjust their own individual activities accordingly. A deep discussion about the concept of awareness is presented by Schmidt (2002). According to him, awareness denotes the practices through which actors tacitly and seamlessly align and integrate their distributed and yet interdependent activities. De Souza and Redmiles (2011) state that although awareness has been usually associated with individuals' achievements, checking a tool log, receiving an email, or starting a conversation are also valid examples of work practices deftly used by group members to become aware of their colleagues' actions.

DSD offers specific awareness requirements. During collaboration, developers need to be aware of how particular tasks or project artifacts are progressing, what fellow developers are (or have been) doing, and the current state of resources associated to the project (Omoronyia et al. 2009a). According to Gutwin et al. (2004), awareness is vital for distributed developers to coordinate their efforts, smoothly add code, make changes that affect other parts of the code, and avoid rework.

Espinosa et al. (2007) state that geographic distances influence the effectiveness of coordination because of: lack of familiarity with cross-site colleagues and working environments, lean communication media, fewer opportunities for interaction, and other geographically distributed collaboration problems (e.g. poor telephone lines, time zones, language and cultural differences). The last reasons listed are specific to distributed development on a global scale (Global Software Development), when the teams are allocated in different countries. In these cases, not only the physical distance, but also temporal and cultural distances increase the difficulties of supporting awareness in computer systems. The participants of a collaborative work often do not know other participants in person, work in different time zones, do not speak the same language, and do not share the same culture. These, among other factors, hinder the information sharing (Chang and Ehrlich 2007), increase the possibility of conflicts (Sarma et al. 2008), and inhibit informal interaction among team members (Calefato et al. 2009).

2.1. Awareness for communication, coordination, and cooperation (3C Model)

The 3C collaboration model has often been used in the literature to classify collaborative tools or to support CSCW analysis (Muhammad et al. 2005; Neale et al. 2004; Kaariainen and Valimaki 2011; Wirth and Albayrak 2011; Fuks et al. 2008). Organizing collaborative tools according to this model facilitates the analysis of issues related to each dimension separately while composing a complete solution (Fuks et al. 2007). Moreover, as they are interrelated concepts, dealing with them separately may reveal factors that otherwise would be forgotten. During communication, people negotiate and make decisions, generating commitments. Coordination organizes and manages tasks and people. Cooperation is the execution of tasks by group members by means of a joint operation in a shared space manipulating shared artifacts. The analysis of collaboration based on these dimensions was originally proposed by Ellis et al. (1991), with some terminological differences, and later extended by Fuks et al. (2007). Figure 1 presents some applications positioned in the triangle formed by the three dimensions.

We used the 3C model to categorize the papers that improve awareness support in DSD according to the dimension they mainly support. We adopted the following criteria to classify them:

Communication: when the paper reports improvements to the way messages and information are exchanged among people, reducing gaps, ambiguity, or the effort needed to understand, establish, or continue a conversation (for example, presenting the context in which the communication takes place is a communication awareness feature);

Coordination: when the paper reports improvements to the support offered for people managing themselves or being aware of the activities flow

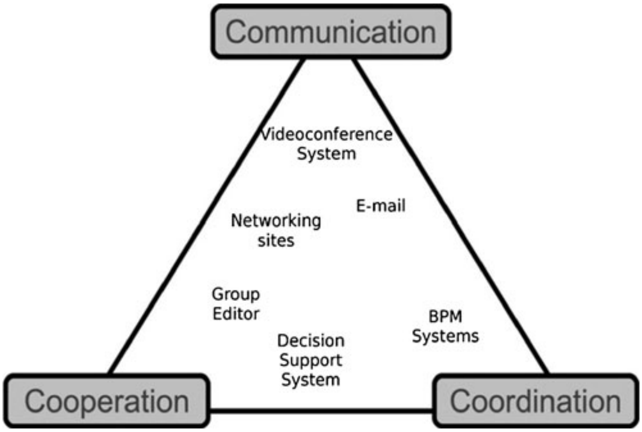

Figure 1. Applications spread in the 3 C triangle based on (Borghoff and Schlichter 2000; Wirth and Albayrak 2011).

(availability status and simultaneous editing indication as well as the level of activity of each participant are examples of coordination awareness information);

Cooperation: when the paper reports improvements to the shared space or to the way users interact with shared artifacts synchronous or asynchronously (source code annotation, screen sharing, and code coloring are example of cooperation awareness features). In this context, shared space is a virtual workspace in which people are able to collaborate, sharing artifacts and information related to their collaborative work.

2.2. Gutwin et al. awareness framework

Gutwin et al. (1996) proposed a framework to describe the elements and mechanisms of awareness on groupware systems. They present four types of awareness (workspace, informal, group-structural, and social awareness), as shown in Fig. 2. The four types of awareness proposed are defined as follows:

- **Workspace Awareness**: the collection of up-to-the-minute knowledge that a person holds about the state of another's interaction with the workspace. It helps people move between individual and shared activities, provides a context in which it is possible to interpret other's utterances, allows anticipation of others' actions, and reduces the effort needed to coordinate tasks and resources.
- **Informal Awareness**: the general sense of whom is around and what they are up to (the kinds of things that people know when they work together at the same office). Informal awareness is the glue that facilitates casual interaction.

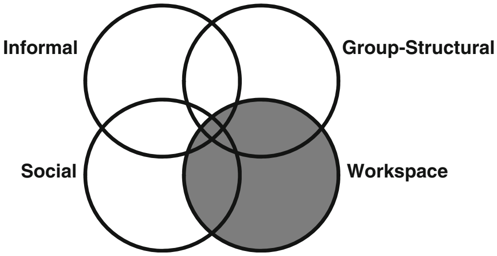

Figure 2. Types of awareness according to (Gutwin et al. 1996).

- **Group-Structural Awareness**: involves knowledge about such things as people's roles and responsibilities, their positions on an issue, their status, and group processes.
- **Social Awareness**: the information that a person maintains about others in a social or conversational context, e.g. whether another person is paying attention, their emotional state, or their level of interest. Social awareness is maintained by means of conversational cues such as back-channel feedback, and by non-verbal cues like eye contact, facial expression, and body language.

The conceptual framework of awareness was built to present a set of basic ideas that are critical for the design of awareness support. It is composed of a set of key elements that collaborators keep track of during collaboration in a shared space. The elements are seen in many kinds of workspace collaboration, and can be related to questions that people ask themselves during group work. This set of elements and the related questions are presented in Table 1.

Table 1. Elements of Gutwin et al. Awareness Framework (Gutwin et al. 1996).

Element	Questions
Identity	Who is participating in the activity?
Location	Where are they?
Activity Level	How active are they in the workspace? How fast are they working?
Actions	What are they doing? What are their current activities and tasks?
Intentions	What are they going to do? Where are they going to be?
Changes	What changes are they making? Where are the changes being made?
Objects	What objects are they using?
Extents	What can they see?
Abilities	What can they do?
Sphere of Influence	Where can they have effects?
Expectations	What do they need me to do next?

The set of elements and questions presented by Gutwin et al. (1996) covers a great part of awareness elements present on collaborative software development, and plays a central role on any collaborative work. We decided to use the framework as a way to classify the elements of awareness presented by the current literature in DSD because of its comprehensive view of workspace awareness and the fact that it is widely used in the general awareness literature (Omoronyia et al. 2010).

3. Systematic review and mapping

A systematic review is a way of identifying, evaluating, and interpreting the available research related to a research question, topic area, or phenomenon. The main purpose for conducting a systematic review is to gather evidences to base conclusions in order to support or contradict claims made by researchers, identify gaps in existing research, provide motivation for new research, and supply a context for the new research (Williams and Carver 2010). On the other hand, Systematic Mapping Studies (also known as Scoping Studies) are designed to provide an overview of a research area, and to provide an indication about the quantity and type of research available within it . Hence, the mapping-related questions are concerned with the structure of the area and the review-related questions with the evidence (Petersen 2011).

Kitchenham (2004) summarizes the stages of a systematic review into three main activities: (i) Planning the Review, (ii) Conducting the Review, and (iii) Reporting the Review. We organized our systematic review and mapping according to this approach.

3.1. Review and mapping planning

The planning activity includes the identification of the objectives and the definition of a protocol. The protocol specifies the method to be used to undertake the systematic review and mapping in a way to reduce the researcher bias (Kitchenham 2004). Systematic reviews are thus auditable and replicable, in the sense that other professionals may reproduce the same protocol and judge the adequacy of the chosen standards. This section summarizes our protocol.

3.1.1. *Research questions*

Formulating the research questions is the most important activity during the protocol definition (Kitchenham 2004), as they guide the systematic mapping and review. The research questions are divided into mapping questions and review questions. To perform our systematic mapping, we defined the following questions:

MQ1: How are the publications on DSD distributed across the years?

- We expect that the answer to this question points out the growth or shortening of the research area.

MQ2: Who are the most active authors in the area?

- With this question we expect to have an indication of the main researchers in the area, offering a reference for related publications in future researches.

MQ3: Which publication forums are the main targets for DSD research production?

- By answering this question we expect to point to where papers can be found, as well as identify good targets for publication of future studies.

The review questions (RQ) drive the systematic review research (Kitchenham and Charters 2007). In our systematic review, the questions were:

RQ1: What awareness studies aimed to improve the DSD scenario?

- The answer to this question includes all papers presenting tools, surveys, case studies, experiments, or theoretical studies that contribute to the way awareness is supported in a Distributed Software Development scenario.

RQ2: Which elements of the 3C model and of the Gutwin et al. Awareness framework do the DSD studies support?

- The answer to this question involves the identification of which kind of contribution is offered by the papers, classifying them according to the 3C collaboration model and to the Gutwin et al. Awareness Framework.

3.1.2. *Search strategy*

The research questions drive the definition of a query string to be executed in scientific repositories. To compose this query, key terms (along with their synonymous) are defined. The query string presents a set of logical expressions that combine keywords and their synonymous, arranged in a way that increases the number of relevant papers obtained from search engines (Mian et al. 2005).

The keywords used in our review are presented in Table 2 and were defined based on terms related to DSD and to awareness. The words of category C1 were defined based on previous systematic reviews conducted on the DSD domain ((Prikladnicki and Audy 2010; Jimenez et al. 2009). Category C1 has more keywords and reflects the fact that the DSD area is maturing, and there are many variations of the same term. The keywords related to the classification of papers were not included in the query string because there are papers related to awareness that might not explicitly present any of the related terms, but can be classified according to them. Category C2 has just the word "Awareness," because it is the key term in this research and it is largely used on CSCW and Software Engineering literature.

Table 2. Keywords defined based on the research questions.

Reference	Category	Keywords
C1	Distributed Software Development	Distributed software development, Global software development, Collaborative software development, Global software engineering, Globally distributed work, Collaborative software engineering, Distributed development, Distributed teams, Global software teams, Globally distributed development, Geographically distributed software development, Offshore software development, Dispersed teams, Virtual teams
C2	Awareness	"Awareness"

The query string was defined as a combination of C1 and C2 presented in Table 2 using the logical connectors "AND" and "OR." The query string was used to retrieve the candidate papers in the following search sources, selected based on previous studies (Dybå and Dingsøyr 2008; Prikladnicki and Audy 2010):

- Science@Direct (http://www.sciencedirect.com);
- El Compendex (http://www.engineeringvillage.com);
- IEEE Digital Library (http://ieeexplore.ieee.org);
- ACM Digital Library (http://portal.acm.org);
- Springer Link (http://springerlink.com);
- ISI Web of Science (www.isiknowledge.com);
- Scopus (http://www.scopus.com).

3.1.3. *Inclusion and exclusion criteria*

All the papers obtained by running the query string on each selected source were analyzed to check their relevance to this research. The process used to include or exclude a paper was based on (Kitchenham and Charters 2007; Prikladnicki and Audy 2010) and comprised the following steps:

1. The papers titles were read, and those papers that were considered clearly irrelevant to the research questions, were excluded.
2. The remaining papers were gathered and repetitions were excluded. For the exclusion, sources that actually publish the papers were prioritized over those that just index articles from other publishers.
3. The included papers were then analyzed based on their abstracts and keywords, considering the research questions.
4. Papers included in the previous step were further analyzed based on the reading of introduction, conclusion, and specific parts related to the contributions.

5. All papers selected so far were fully read by the researchers and documented on a proper form. Those papers which, despite addressing awareness issues, did not focus on DSD domain were dismissed. Papers included after this step were checked using quality criteria (presented in Section 3.1.4) and the resulting papers were considered our primary studies.

To reduce bias, the first three steps were performed by two researchers independently, and the paper passed to the next step if at least one of them decided so. In the other steps, the paper was included after some discussion and consensus. All the steps were reviewed by a third—more experienced—researcher, responsible for checking the information generated. It is worth noticing that only papers written in English were considered.

3.1.4. *Quality assessment*

According to (Kitchenham and Charters 2007), quality assessment is considered critical to assess the primary studies to guide the interpretation of findings and determine the strength of inferences, among other reasons. Each of the 91 papers that remained after step 4 was assessed by one researcher and discussed with the other, according to a set of criteria. This set, presented in Table 3, was defined based on (Dybå and Dingsøyr 2008) and (Kitchenham et al. 2010). Each of the 10 criteria was graded on a dichotomous scale (Yes, No). Only criterion QA1 was used for excluding a paper. The other criteria were used to assist individual data analysis. Taken together, these 10 criteria provided a measure of the extent to which we could be confident that a particular study's findings could make a valuable contribution to the review.

3.1.5. *Data collection and analysis*

All selected papers were categorized based on the classification used in (Jimenez et al. 2009): (i) case studies; (ii) theoretical studies (also including conceptual/

Table 3. Quality assurance criteria used in this review.

No.	Question
QA1	Does the study answer primarily or secondarily the research questions?
QA2	Are the objectives and research aims clearly defined?
QA3	Is there an adequate description of the context in which the research was carried out?
QA4	Was there a control group/comparison with other studies?
QA5	Have the authors explained how the study sample (participants or cases) was identified and selected, and what was the justification for such selection?
QA6	Is it clear how the data were collected (e.g. through interviews, forms, observation, tools, etc.)?
QA7	Do the authors explain how the analysis/interpretation of the collected data was done?
QA8	Does the study clearly report the results?
QA9	Are the results consistent with the objectives presented?
QA10	Are limitations and validity of the study discussed explicitly?

theoretical frameworks); (iii) experiments; (iv) literature reviews; and (v) tools (also including frameworks and architectures).

The categorization according to the 3C collaboration model was based on the evaluation of the dimensions (communication, coordination, or cooperation) supported by the paper under analysis. Each dimension was graded from 0 to 3, according to the level of support the paper presented:

3: Mainly supports (main focus of awareness study is on that dimension);
2: Also supports (the dimension is not the main focus, but it is also supported);
1: Indirectly supports (it has no focus, but brings indirect improvement);
0: Does not support (when no support or improvement is presented).

For the categorization based on the Gutwin et al. Awareness Framework , we used the set of elements and related questions presented in Table 1. We have also considered the possibility of past events (asynchronous interactions) appearing as workspace awareness elements, as the framework "does not mention elements for maintaining awareness of asynchronous changes, required for people to catch up with events that have already taken place in the workspace" (Tam and Greenberg 2004). Action and artifact history concern the details of events that have already occurred, and event history concerns the timing of when events happened (Gutwin and Greenberg 2002). Table 4 presents the full list of elements and questions considered in our analysis. The researchers could choose as many elements as they wanted for each paper analyzed.

3.2. Conducting the review

After executing the whole process defined in Section 3.1.3, 91 primary studies were selected. More details about the process, the papers included, and their classification can be found at http://www.igor.pro.br/JCSCW/awarenessDSD.

3.2.1. *Inclusion and exclusion process results*

As observed in Fig. 3, confirming the observations made by Prikladnicki and Audy (2010), the lack of a standard terminology in DSD resulted in a large number of papers to start with (total of 1967 papers found), but only a few were selected. A high number of papers of unrelated areas (like computer networks, ubiquitous computing, e-analysis, learning, and psychology) also contributed to the large number of papers discarded at the beginning. Table 5 presents the number of papers per phase, distributed according to the source of the paper.

For the first step, the titles were analyzed by two researchers independently of each other. From the 1967 papers returned by the query, 728 (37.0 %) were selected. All selected papers were checked to dismiss duplications. Then, 167 papers were excluded (22.9 %) and 561 were kept.

The next step involved abstract reading. All abstracts were read by two researchers independently. After the analysis, 277 papers were selected. We also computed the

Table 4. Elements and questions considered for papers classification (based on (Gutwin et al. 1996)).

Awareness Dimension	Element	Questions
Workspace	Identity	Who is/was participating in the activity?
Workspace	Location	Where are/were they?
Workspace	Activity Level	How active are/were they in the workspace? How fast are/were they working?
Workspace	Actions	What are/were they doing? What are/were their current activities and tasks?
Workspace	Intentions	What are they going to do? Where are they going to be?
Workspace	Changes	What changes are they making (did they make)? Where are the changes being made?
Workspace	Objects	What objects are/were they using?
Workspace	Extents	What can they see?
Workspace	Abilities	What can they do?
Workspace	Sphere of Influence	Where can they have effects?
Workspace	Expectations	What do they need me to do next?
Informal	Presence	Who is around? What are they up to?
Informal	Opinion	What is their opinion?
Social	Interest Level	How interested are they?
Social	Emotional Feelings	How are they feeling?
Social	Availability	Are they available?
Group Structural	Roles and Responsibilities	What are their roles/positions?

Kappa coefficient to check the degree of agreement among researchers. The Kappa for step 3 assessments was 0.723, which is characterized as “substantial agreement” according to the classification by Landis and Koch (1977).

On step 4, the researchers read the introductions and conclusions and specific parts related to the contributions of the paper. The papers that did not present any clear contribution to awareness on DSD were dismissed. As a result, 134 papers were included by at least one researcher. The agreement was very good, with the Kappa coefficient of 0.816 (classified as “almost perfect” (Landis and Koch 1977)). These 134 papers were selected as relevant papers and were submitted to an in-depth analysis.

In the deeper analysis, performed by reading the full papers, 39 papers were considered not relevant to the review. The main reason for dismissing papers at this step was that they did not present contributions to the DSD area.

3.2.2. *Quality assessment*

All 91 remaining papers were submitted to the quality assessment presented in Section 3.1.4. Table 6 presents the percentage of papers that received ‘yes’ as the answer for each criterion. The results enable us to be confident in the inclusion of

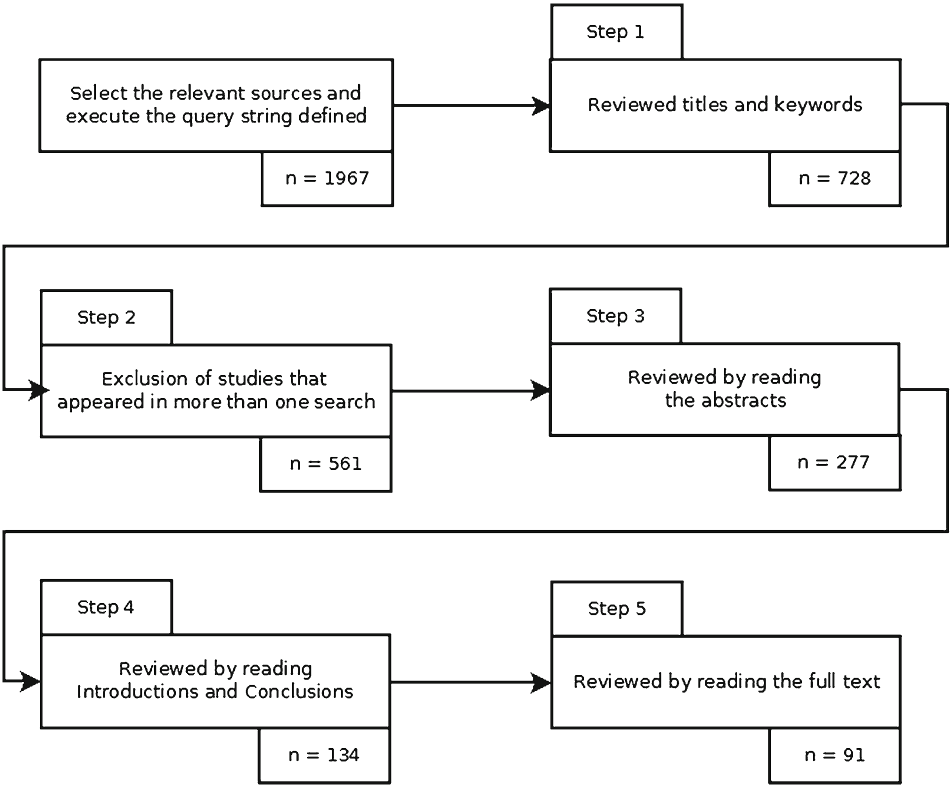
Select the relevant sources and execute the query string defined
n = 1967
Step 1
Reviewed titles and keywords
n = 728
Step 2
Exclusion of studies that appeared in more than one search
n = 561
Step 3
Reviewed by reading the abstracts
n = 277
Step 4
Reviewed by reading Introductions and Conclusions
n = 134
Step 5
Reviewed by reading the full text
n = 91

Figure 3. Paper selection process.

Table 5. Distribution of papers selection according to their source.

Source/	Query date	Query result	Step 1	Step 2	Step 3	Step 4	Step 5
			Title/ keywords Analysis	Repetitions	Abstract Analysis	Introduction/ Conclusion Analysis	Full Text Reading
ACM	21 October 2010	470	148	130	73	52	33
IEEE	21 October 2010	52	34	32	23	11	10
Scopus	22 October 2010	641	352	256	123	48	32
Science Direct	26 October 2010	292	47	46	17	6	3
El Compendex	27 October 2010	122	37	4	3	0	0
Springer	27 October 2010	352	83	83	32	15	11
ISI	29 October 2010	38	27	10	6	2	2
TOTAL		1967	728	561	277	134	91

Table 6. Quality Assessment overall results.

Question	% of papers that received 'yes'
1. Does the study answer primarily or secondarily the research questions?	100.00 %
2. Are the objectives and research aims clearly defined?	98.90 %
3. Is there an adequate description of the context in which the research was carried out?	80.20 %
4. Was there a control group/comparison with other studies?	76.92 %
5. Has the researcher explained how the study sample (participants or cases) was identified and selected, and what was the justification for such selection?	53.85 %
6. Is it clear how the data was collected (e.g. through interviews, forms, observation, tools, etc.)?	59.34 %
7. Do the authors explain how the analysis/interpretation of the collected data was done?	54.95 %
8. Does the study clearly report the results?	79.12 %
9. Are the results consistent with the objectives presented?	95.60 %
10. Are limitations and validity of the study discussed explicitly?	81.32 %

the primary studies in this review. Most papers define their objectives clearly and present results in a consistent way. It is noteworthy the high number of papers that presented their limitations and threats to validity, enabling other researchers to help filling the gaps left opened. However, we found out that many of the publications lack details about the research methods used, not explaining how they have chosen the samples, how the data were collected, or how the data were analyzed. The high number of papers presenting new tools can be one explanation for this lack of research methods details.

4. Systematic mapping report

A systematic mapping aims to provide an overview of a research area and to identify the quantity and type of research available within it. In this section, we present the answers to our mapping questions (MQ).

To answer MQ1—How are the publications of the area distributed across the years?—Figure 4 is presented. The first conclusion we can draw from the figure is that awareness in DSD is an area that was not widely studied until 2005, and recently gained attention. We can observe that 71 out of 91 papers were published from 2006 to 2010. Among them, 21 appeared in 2010. It is important to highlight that the search was performed in late 2010, so it is possible that more papers could have been published in 2010 and early 2011, and had not been yet indexed.

Regarding MQ2 (who are the most active researchers in the area), Table 7 depicts the authors that most appeared on the selected primary studies. If we check the number of different papers published by the top five authors, we come to a number of 24 (26.4 % of the papers). Some of them work together. Anita Sarma, for example, worked with David Redmiles, André van der Hoek, and James Herbsleb.

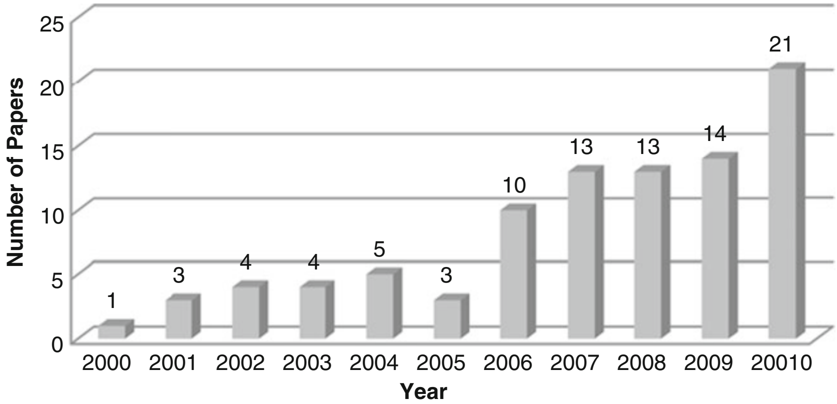

Figure 4. Number of relevant papers per year (queries from October 2010).

Question MQ3 is related to the main forums of publication for awareness studies in DSD. To answer this question, we first analyzed the type of publication: Journal *versus* Conference. The result shows that 79 papers (86.8 %) were published in conference proceedings, while just 12 (13.2 %) appeared in journals. An interesting fact is that each of the 12 journal papers appeared in a different vehicle.

The 79 papers published in conference proceedings appeared in 43 different forums. In Fig. 5 we show the 13 conferences that present more than one paper published. We must highlight ICSE (International Conference on Software Engineering), whose main track accommodated 13 papers (14.3 % of all primary studies / 16.5 % of conference papers). If we consider ICSE workshops (MSR, CHASE, SDG, and Web2SE) together, we come to a total of 21 papers. Therefore, ICSE is definitely the main forum regarding awareness studies on DSD domain. The conference on Global Software Engineering (ICGSE) published six of the primary studies (6.6 %). Regarding CSCW related forums, five primary studies were published on CSCW (ACM Conference on Computer

Table 7. Most active researchers and number of primary studies.

Author	# of papers
André van der Hoek	11
Anita Sarma	9
James D. Herbsleb	6
Lile Hattori	5
David Redmiles	4
Cleidson R. B. de Souza; Clinton Jeffery; Daniela Damian; Hani Bani-Salameh; Inah Omoronyia; John Ferguson; Kurt Schneider; Marc Roper; Marcelo Cataldo; Murray Wood; Prasun Dewan; Ziad Al-Sharif	3

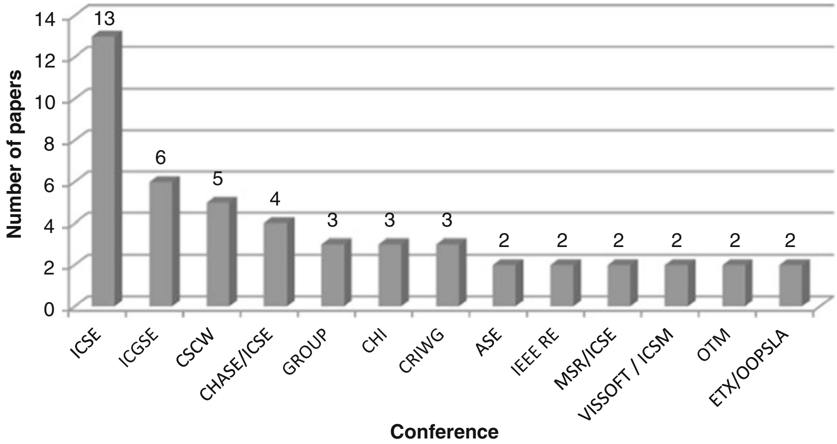

Figure 5. Publications per forum. Keys: ICSE = International Conference on Software Engineering; ICGSE = International Conference on Global Software Engineering; CSCW = ACM Conference on Computer Supported Cooperative Work; CHASE = Workshop on Cooperative and Human Aspects of Software Engineering; GROUP = International Conference on Supporting Group Work; CHI = Conference on Human Factors in Computing Systems; CRIWG = Conference on Collaboration and Technology; ASE = International Conference on Automated Software Engineering; IEEE RE = International IEEE Requirements Engineering Conference; MSR = Working Conference on Mining Software Repositories; VISSOFT = IEEE International Workshop on Visualizing Software for Understanding and Analysis; ICSM = International Conference on Software Maintenance; OTM = On The Move Conferences; ETX = Eclipse Technology Exchange workshop; OOPSLA = Object-Oriented Programming, Systems, Languages & Applications.

Supported Cooperative Work), three were published on CRIWG, and other three on the GROUP conference.

5. Systematic review report

In this section, we present an overall analysis of the primary studies regarding the type of studies found (Section 3.1.5) and a discussion of the works involving the improvements, opportunities, and research gaps identified in this review. The discussion is based on the 3 C collaboration model (Section 5.2) and on Gutwin et al. Awareness Framework analysis (Section 5.3). Hereafter, we use the notation [PSXX] to cite the primary studies analyzed, where XX is a unique identification of the primary study (see Appendix A for the list of primary studies and their source, type, and reference).

5.1. Studies type

Figure 6 presents the type of studies encountered (according to the classification presented in Section 3.1.5). All papers were classified in, at least, one category.

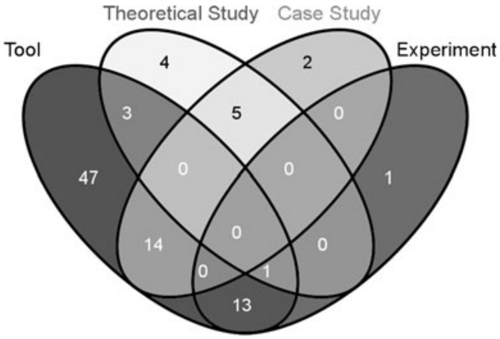

Figure 6. Venn diagram for types of primary studies found.

The large number of tools-related papers may be noted. Among the 91 primary studies, 68 (74.7 %) presented a new tool, most of them (47) without any experimental analysis. Thus, we can observe that the focus of the research so far was to present new tools to support DSD activities. It is worth pointing out that some of these papers were related to the same tool, or presented different features of the same tool. For example, the tool Palantìr appears in seven papers [PS74] [PS76] [PS82] [PS81] [PS80] [PS78] [PS79], Syde appears in five papers [PS52] [PS53] [PS54] [PS51] [PS55], and COLLECE [PS29] [PS30] and SCI [PS04] [PS05] appear in two different papers. The other 23 papers identified in this review were categorized as case studies, theoretical studies, and/or experiments. Just one of them was classified only as experiment [PS70]; it reports a controlled experiment to check the potential benefits of information visualization in supporting newcomers.

Of the four theoretical studies, three present conceptual frameworks for awareness support [PS31] [PS13] [PS90]; and one introduces the concept of interface translucence as a mechanism for enabling scalable collaboration in software ecosystems [PS14]. Two papers of the review present case studies conducted within the industry [PS21] [PS20]. One of them focuses on investigating whether current awareness, general awareness, and availability are correlated with communication patterns in Requirements Centered Social Networks [PS21]. The other case study goal was to check the consequences of awareness gaps, verifying possible relationships among broken builds and these gaps [PS20]. In addition to the papers presented in Fig. 6, one paper [PS88] was classified as a Survey, presenting and classifying a set of visualization tools which support awareness of human activities in distributed software development.

5.2. Papers classification according to the 3C model

Figure 7 shows the classification of the primary studies according to the 3C collaboration model. This Venn diagram presents the number of awareness

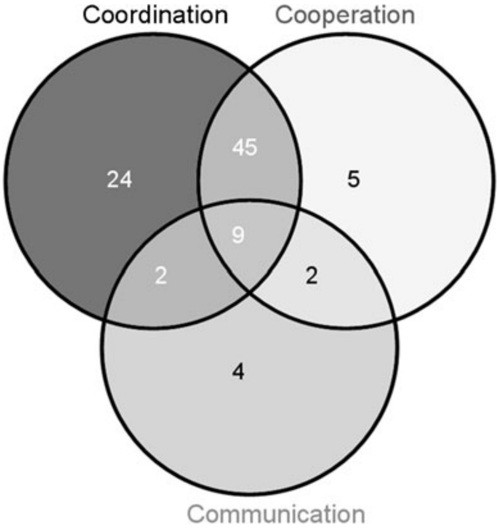

Figure 7. Venn diagram for the 3C-model classification.

studies that bring support (at any level) for each 3C-dimension. Therefore, the first conclusion is that communication is scarcely studied, appearing in just 17 papers (18.7 %), with just four focusing only on communication. We can also observe that a great focus is given to coordination and cooperation, as 87 out of 91 papers (95.6 %) present some support to one of these dimensions; and 45 papers (49.5 %) present support to coordination and cooperation concurrently. Nine other papers presented some level of support in all dimensions.

Figure 8 presents the distribution of papers according to the level of awareness support provided to each of the 3C model dimensions. It is clear that coordination is by far the main focus of awareness studies on DSD domain due to the number of papers that mainly support it (evaluated with 3 in the scale presented in Section 2.1): a total of 58 papers out of 91 (63.7 %). When we verified papers that support coordination and that were evaluated with 2 or 3, this number grows to 74 (81.3 %). On the other hand, communication is mainly supported by only eight papers (8.8 %). In the following subsections, we briefly present the 3C awareness elements encountered in this literature review.

5.2.1. *Communication*

According to Jimenez et al. (2009), the software life cycle requires a great deal of communication using different tools and formats to avoid misunderstandings and delays. Awareness initiatives are needed to avoid ambiguity and misunderstandings, as cultural differences imply on different vocabulary that may lead to mistakes.

As discussed in the previous subsection, awareness supporting communication is poorly explored within the DSD domain. Although many researchers cite communication issues as motivation, only eight awareness initiatives mainly

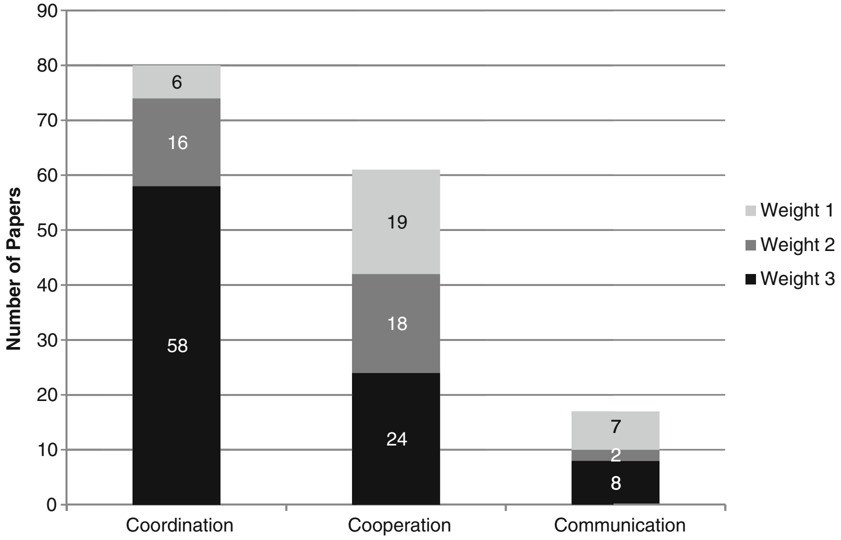

Figure 8. Number of papers classified according to the dimensions of the 3C model.

support communication. Most of these papers are focused on offering some kind of **context for the conversation**. A tickertape tool was introduced by Fitzpatrick [PS39] to present CVS commit messages to members of a project, enabling them to start a private or a group chat within the context of the CVS message. Communico [PS28] is a tool that presents awareness mechanisms that enable participants to check ongoing and past discussions, to know who is talking, and the their level of knowledge. Jazz [PS17] includes a chat tool that enables developers to include links to transcripts of older chats and team event notifications. A Jazz extension, called COFFEE [PS09], provides a discussion tool embedded within eclipse, enabling developers to conduct threaded and categorized discussions within the development environment. EGRET [PS86] also provides contextualized chat and email tools that enable the automatic or manual insertion of navigable links to relevant artifacts into messages.

Another concern of the studies is how to **improve informal communication**. The social network application FriendFeed[1] was integrated to Jazz [PS11] to bring personal interests to workspace and to support informal and social communication by means of microblogs within their development environment. To enable informal, verbal, or ad-hoc communication during project analysis, Stapel et al. [PS87] proposed a specific notation to model and depict information flows, stimulating point-to-point communication.

Ehrlich and Chang [PS32] report a case study conducted within the industry that analyzes the effects of awareness in communication in a DSD environment. The authors found out that frequency of communication was associated with awareness of the person's current work and familiarity to the other person. Some papers [PS30]

[PS88] [PS06] [PS71] [PS83] present limited means to support communication in DSD environments (e.g. chats, forums) without any specific contribution to provide an easier or more effective communication in a DSD scenario.

To conclude, we can argue that some topics about communication are scarcely studied, but present challenges and research opportunities. For example, privacy issues regarding access to sensitive user data during communication is a topic that is starting to be studied on DSD domain (Godefroid et al. 2000; Patil et al. 2009) and can be explored on awareness studies. Ways to support cultural, political, and language differences are frequently cited as issues in DSD studies, but papers addressing them were not found. Semantics extraction and communication formalization could be used to address these issues, for example.

5.2.2. *Coordination*

To coordinate themselves, team members need to be aware of their tasks and interdependencies (Damian et al. 2007a). 80 out of 91 (87.9 %) papers present some level of support for coordination and 58 (63.7 %) mainly support it.

To provide awareness for coordination, some papers focused on presenting **information regarding the tasks** of a project. A list of tasks is presented in TeamSpace [PS43], enabling team members to be aware of related tasks and to track them. ProjectWatcher [PS48] observes and records fine-grained information about user modifications and provides visualizations of who works on the project, what artifacts they have been working on, and where in the project they have been working.

Another approach used to provide awareness to support coordination is gathering **information on source code version management repository**. CAISE [PS18] is a tool based on commits to notify developers regarding dependent code, user dependencies, and impacts of code changes. Celine [PS34] offers information about artifact conflicts in chaotic, star, and extended star models, by means of an awareness mechanism that supports conflict management in Software Configuration Management (SCM) systems. Syde [PS52] [PS53] [PS54] [PS51] [PS55] is a collaborative tool that brings awareness of team members actions by propagating changes as they happen, and by warning about potential conflicting changes. Many other papers revealed that code repositories are a rich source of information for awareness generation [PS82] [PS17] [PS22] [PS19] [PS68].

Also using information from source code version management repository, but applying **social network analysis** (SNA) techniques, de Souza et al. [PS25] introduce Ariadne, a tool that enables visual analysis of sociotechnical dependencies, thus helping to find coordination problems. TraVis [PS24] provides a real-time visualization of the project traceability network of artifacts and users, containing valuable information about dependencies between software developers and about what other people work on. Other papers also present sociotechnical network analysis to improve awareness, such as [PS77], [PS66], [PS33] and [PS01].

Some papers propose the usage of **visualization techniques** to support coordination. Ye et al. [PS91] present SecondWATCH, a tool that enables visualization of local workspaces, version control repository, and bug tracking system in real-time as a common view shared by the whole team using a 3D city metaphor. Augur [PS41] is a visualization tool that creates visual representations of both software artifacts and software development activities, besides enabling developers to explore the relationship between such activities and artifacts. SoftChange [PS42] provides an interactive application that enables the exploration of the project history as a collection of graphs: file authorship (sociotechnical), coupling (co-changes), change overview, authorship overview, and file evolution graphs. StarGate [PS66] uses information visualization techniques to assist the user in understanding the complex interactions between developers and software repositories, helping team members to understand the evolution of the software project over time. Palantír [PS74] provides a three-dimensional activity viewer to explore workspace activities and their evolution on a project-wide basis.

The usage of version management repository information and SNA to **search for experts** (or identify artifact ownership [PS53]) is another mechanism largely studied to provide awareness to support coordination. Expertise Browser [PS64] and SmallBlue [PS33] are tools to assist users in identifying experts for a specific artifact or task, making them aware of how experienced the candidates are. Minto and Murphy [PS63] present Emergent Expertise Locator, which uses Instant Messenger clients embedded in developers IDE to make them aware of experts. Minto and Murphy use a method proposed by Cataldo et al. [PS15] to measure task dependencies among people, by checking the "congruence" between these task dependencies and the coordination activities performed by individuals. Begel et al. [PS07] present Hoozizat, a tool developed to find people who own and are responsible for a feature, API, product, or service, built as a web search portal over Codebook. One of the features proposed by Omoronyia [PS67] [PS68] is the possibility of ranking developers according to their relevance to a task or artifact, as a result of a real-time analysis of interactions generated from every developers' workstation.

Tracing artifacts and relationships among them is also used to support coordination on DSD projects. Helming et al. [PS57] present UNICASE, a tool that implements a unified model that explicitly combines system specification and project management models into one fully traceable model, used to notify interested parties. Sysiphus [PS10] enables users to trace artifacts and related users by modeling the links among artifacts as directed graphs and allowing subscription for changes notification. TraVis [PS24] provides a traceability of artifacts and users using social network graphs. Share [PS02] aims to help developers keeping track of their code, presenting a practice of loosely bound cooperation in which individuals are able to pursue independent goals and assist each other.

Three papers present flexible ways to make members aware of **quality metrics results**, helping developers and leaders/managers to coordinate their work. EvolTrack presents software evolution and enables customizations to show specific metrics in user-defined color scales. Proença et al. [PS72] present a tool that shows quality metrics (e.g. number of developers, number of recent bugs, code volatility), and an overall quality measure represented by a colored border in UML class diagrams. CoderChrome [PS50] is a code coloring plugin for Eclipse that provides means to derive coloring or decorations on source code for information such as code author, code age, cyclomatic complexity, and line length.

Cataldo and Herbsleb [PS13] propose a framework based on product features to support coordination within distributed environments, providing information about members' activities and their relations with product features.

Most part of the papers that support coordination presented features based on historical information extracted from source code repositories. However, key information used to obtain awareness is based on items that change on a daily, hourly, or minute-by-minute basis (Biehl et al. 2007). Therefore, there is still room for research on awareness based on recent information [PS68]. The use of Social Networks is another research topic appearing on DSD domain. A possible fruitful area is the use of social network to maintain awareness on emerging and unplanned interactions that can appear during the development cycle [PS20].

5.2.3. *Cooperation*

To cooperatively edit a shared artifact, developers need to be aware of current and previous actions affecting that artifact to avoid loss of efforts. Some papers propose awareness elements embedded in the shared artifacts to provide notifications. In this sense, two papers [PS58] [PS60] use **code annotation** to present changes being concurrently made by other developers in a shared artifact. Dekel and Herbsleb [PS26] also use code annotation to provide awareness on a tool called eMoose, which enables developers to write informal comments in the code, stores them in a central database, and spreads them to other developers using the annotated module.

Warning conflicts on shared artifacts is a common way to provide awareness to support cooperation. Lighthouse [PS19] is a tool that captures code change events directly on developers' workspace to avoid conflicts by keeping a shared and up-to-date UML design representation of the actual code, displaying information about who is updating the code and which actions are being done. CASI [PS85] is a Lighthouse plugin that aims to avoid indirect conflicts by dynamically showing the source code entities that are being modified by any developer along with their Spheres of Influence. Guimarães and Rito-Silva [PS46] propose the concept of real-time integration and implement it on WeCode (Eclipse plugin) by automating the detection of actual conflicts while programming, enabling developers to share fine-grained changes before they check in. Estublier and Garcia [PS34] present a study based on cooperative policies to control concurrent engineering in order to avoid conflicts considering different concurrent models. Other papers also present elements to avoid conflicts during cooperative handling of artifacts [PS55] [PS60] [PS72].

Regarding **support for synchronous artifact handling**, CE4WEB [PS71] is a cooperative CASE tool that supports the cooperative edition of UML diagram, offering a shared blackboard and telepointers. OdisseyShare SDE [PS62] provides a solution based on a telepointer and a radar view as awareness elements to support synchronous handling of UML diagrams. There are some other papers focused on the coding phase, in particular supporting the XP pair programming practice for distributed members, enabling developers to cooperate over a shared piece of code [PS38] [PS49] [PS84] [PS75] [PS56]. Usually, these tools also provide means for the developers to communicate and coordinate their actions.

Despite the number of papers supporting cooperation, this area requires further development. An example is that none of the papers presents ways to make developers aware of pieces of code to complete a given function based on similar codes extracted from other developers' code. So far, coding is the most supported phase. The other development activities may be further explored, for example, offering support to clients and analysts cooperate during requirement specification or acceptance tests writing.

5.2.4. *Summary of the classification according to the 3C Model*

In this subsection, we summarize the classification of awareness features according to the 3C collaboration model. This classification was made to (i) organize the high number of papers found on this review, and (ii) to provide a quick reference for DSD environment developers and researchers regarding which awareness features have already been investigated.

Table 8 presents the references for the papers classified according to the 3C model dimensions they support. The table does not include frameworks or architectures. It is possible to notice that some elements provide support to more than one C, while others seem to be more specific. The level of support and the axis supported depends on the way the element is used on each study.

5.3. Papers classification according to the Gutwin et al. Awareness Framework

In addition to the classification based on the 3C model, the primary studies were also classified according to the Gutwin et al. Awareness Framework (Gutwin et al. 1996), based on the questions and elements presented in Table 4 (Section 3.1.5). The data extracted are depicted in Fig. 9 and show the number of papers related to each element. Six papers were not classified according to this framework as they present experimental studies not mentioning any specific element [PS13][PS20][PS21][PS31][PS70][PS88].

Figure 10 depicts the Venn diagram representing the distribution of awareness elements on DSD domain according to their types. We can see that workspace awareness is by far the most explored type of awareness, playing a central role on DSD domain. All papers present at least one workspace awareness element. Fifty four (59.3 %) papers present only workspace awareness elements. The other 31 papers present some other type of element.

Table 8. Identified awareness features supporting each 3C dimension.

	Communication	Coordination	Cooperation
Activity control (workflow, agenda, worklist)		[PS10] [PS39] [PS43]	
Artifact change indication		[PS05] [PS10] [PS17] [PS19] [PS22] [PS41] [PS46] [PS54] [PS55] [PS56] [PS59] [PS68] [PS72] [PS76] [PS81] [PS85] [PS86] [PS91]	[PS05] [PS22] [PS34] [PS54] [PS55] [PS56] [PS60] [PS72] [PS85]
Code coloring		[PS50]	[PS02] [PS75]
Collaborative artifact synchronous handling			[PS19] [PS30] [PS35] [PS38] [PS43] [PS56] [PS71] [PS72] [PS75] [PS84] [PS85]
Conflict indication		[PS03] [PS18] [PS19] [PS34] [PS46] [PS54] [PS55] [PS56] [PS60] [PS61] [PS72] [PS76] [PS81] [PS83] [PS85]	[PS19] [PS34] [PS46] [PS55] [PS60] [PS61] [PS72] [PS85]
Context/subject-aware message exchange	[PS09] [PS17] [PS28] [PS30] [PS39] [PS43] [PS86]		
Developers activity		[PS05] [PS45] [PS48]	[PS02]
Historic based expert search/ recommendation		[PS07] [PS15] [PS33] [PS48] [PS55] [PS56] [PS63] [PS64] [PS68] [PS69] [PS72]	
Informal/Social Communication	[PS11]		
Metrics visualization		[PS16] [PS50] [PS72]	
Presence/status indication		[PS05] [PS12] [PS17] [PS28] [PS30] [PS35] [PS38] [PS43] [PS46] [PS49] [PS56] [PS62] [PS63] [PS65] [PS75] [PS83] [PS84] [PS86] [PS89] [PS91]	
Screen Sharing		[PS17] [PS27] [PS49] [PS62] [PS71] [PS83] [PS84]	[PS05] [PS17] [PS30] [PS38] [PS56] [PS62] [PS71] [PS75] [PS83] [PS84]
Social/socio-technical network		[PS02] [PS07] [PS24] [PS25] [PS40] [PS42] [PS66] [PS68] [PS77]	[PS25] [PS68]
Software Visualization / Evolution		[PS16] [PS41] [PS42] [PS55] [PS66] [PS68] [PS70] [PS74] [PS91]	
Source code annotation			[PS26] [PS59] [PS60]
Tracing artifacts and relationships		[PS02] [PS10] [PS24] [PS57]	

As it can be observed, seven (7.7 %) papers present elements of all types [PS84] [PS75] [PS86] [PS04] [PS05] [PS17] [PS11]. Some other papers presented overlap of three types of awareness elements: six present social, informal, and workspace elements [PS28] [PS89] [PS43] [PS56] [PS63] [PS91]; and one presents Group-structural, informal, and workspace elements [PS65].

In the next subsections, we detail some elements found in our systematic review.

5.3.1. *Workspace awareness*

All classified papers present some element of workspace awareness, most part of them presenting only elements of this type.

Identity is the most present element, appearing in 83 papers. In a collaborative environment, it is essential to know who is responsible for the action that generated an event. In some papers, this information is presented on communication tools (e.g. forums and chats), enabling users to be aware of who posted a message, or who is discussing it [PS28] [PS86] [PS30] [PS09] [PS11]. Other papers provide information related to the person who is using or has been using an artifact, enabling users to know who are the other members collaborating with them, like in [PS02] [PS55] [PS60] [PS85] [PS91] [PS39] [PS40] [PS50] [PS19] [PS76].

Change awareness elements appear on 63 papers (68.1 %). Some of these papers present awareness related to **past changes**, enabling members to visualize the history of artifacts [PS07] [PS08] [PS16] [PS41] [PS66] [PS48]. Others focus

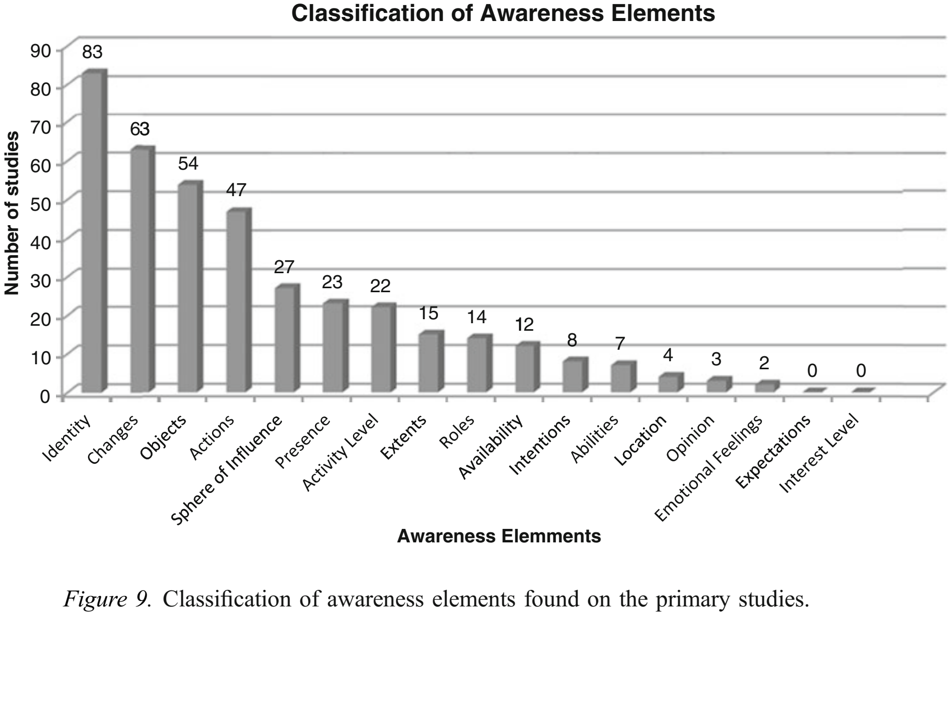

Figure 9. Classification of awareness elements found on the primary studies.

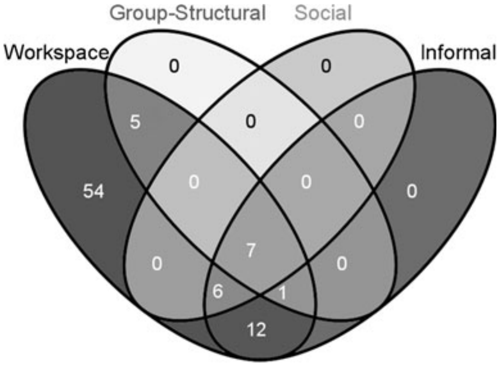

Figure 10. Awareness elements found, according to the dimensions proposed by (Gutwin et al. 1996).

on awareness of **real time changes** that occur on distributed workspaces, visualized directly on the shared artifacts [PS59] [PS78] [PS76] [PS02] [PS75] [PS71] [PS61] [PS85] [PS55].

Sphere of Influence information makes people aware of the effects an action can cause. On the selected DSD papers, this element is implemented by showing up the conflicts that a change on an artifact can cause or is causing to other artifacts or team members [PS19] [PS68] [PS36] [PS37] [PS24].

Activity level elements appear on 22 papers (24.2 %). Some of these papers present the activity level using software visualization approaches like, for example, in Stargate [PS66] and SoftChange [PS42]. Other papers focus on showing the activity of the developers within a given task or contributions to a given artifact. For instance, [PS44] [PS45] present a timeline with the amount of changes made by the developers and his/her communication activity.

Location elements were found in four papers, making team members aware of where in the world [PS82] [PS73] and within a building [PS43] are other members located. Ehrlich and Chang [PS32] study the effects of being aware of team members' location while searching for experts and communicating.

Eight papers presented **intention** elements, focusing on revealing what the members' next activities are [PS08] [PS29] [PS30] [PS43] [PS56] [PS65] [PS68] [PS79]. Seven other papers offer some information regarding members' **abilities** [PS04] [PS05] [PS08] [PS33] [PS47] [PS63] [PS25].

We also found 54 papers concerned on providing some awareness on what **objects** are/were being used and 47 on making people aware of the **actions** performed on the workspace.

5.3.2. *Informal awareness*

Informal awareness refers to who is around and what they are up to. This kind of information is typically related to presence awareness, since it is associated with technologies, such as Instant Messengers, which enable colleagues at distributed locations be aware of who is online, offline, active, inactive, etc (Omoronyia et al. 2010). Moreover, informal awareness is related to team members' ability to express their opinion and be aware of others' opinions about different aspects of collaborative work.

Informal awareness appears in 25 papers. 23 of them provide information related to **presence indication** within the workspace. The use of Buddy lists is the most explored way to show this kind of information. When dealing with shared sessions (like using screen sharing), some papers bring presence indication by showing a list of users on the session, telepointers, colored text selection [PS75] [PS30] [PS84] [PS83], and radar view [PS38] [PS62]. A presence indication element based on physical cameras and panels is presented by Everitt et al. [PS35], indicating the presence of an individual by a blue shadow in the remote workspaces.

Only two papers in this category present team members' **opinion**. In the COFFEE tool [PS09], users can post comments, arguments, questions, and other contributions during a discussion. By adding FriendFeed to the developer workspace, Calefato et al. [PS11] enabled users to share their opinion via messages posts to a social network application available for a private corporative group.

5.3.3. *Social awareness*

The social dimension creates opportunities to publish personal information helping to establish a shared context among remote developers, with little or no chances to physically meet [PS11].

Some **availability** feature was found on 11 out of 12 papers that presented social awareness. Availability differs from presence element because it offers specific information about the possibility of being interrupted or not, or if it is an opportune moment to discuss a specific subject, for example. ICI/SCI [PS06] [PS05], and SecondWATCH [PS91] improve social awareness by providing collaborative virtual worlds/environments as part of the developers workspace, enabling them to interact even when dispersed. Some papers use the status of a member to indicate what he/she is doing (participating of a conversation [PS28], or visualizing/editing an artifact [PS75] [PS56]). Some others present the availability using cameras in real time [PS89] [PS43]. When working in shared spaces, such as distributed pair programming environments, availability is indicated by presenting the sessions in which the members are participating [PS84].

Just two papers improve the social area by means of **emotional feelings** or social/personal information. FriendFeed [PS11] enables team members to express their feelings by posting microblog messages, offering personal context and culture sharing. Status message feature of Jazz [PS17] can also be used to express emotions, since team members can change and share their status according to their feeling at the moment.

We think that this dimension has a lot of possibilities and can grow in order to help current workspaces to become closer to collocated teams space. Since social awareness helps to develop an organizational culture and to consolidate connections and trust-based relationships between distant collaborators, it contributes to project success by improving team's well-being and social health (Robinson and Sharp 2005). Researches involving interests, day-to-day opinions, and emotions of group members can improve the distributed scenario developing team spirit and help building personal relationships among distant members.

5.3.4. *Group structural awareness*

Group-structural awareness is related to the expertise of the collaborators based on the roles and responsibilities they assume. This knowledge is important when choosing whom to initiate an interaction with, for mentoring purposes on project activities [PS68].

Most part of the 14 papers presenting **roles and responsibilities** awareness elements focus on finding out experts on a given subject, extracting information from historical data and contextual information. Minto and Murphy [PS63], for example, propose an Eclipse plugin to show a ranked list of experts about the piece of code a developer is currently working. Carter and Dewan [PS12] propose an approach to create awareness information automatically based on developers' interactions with the IDE, indicating who is having difficulties. On their work, Hattori et al. [PS55] present an approach based on IDE history logs to measure code ownership. In this sense, other papers also present ways to find experts or owners of specific artifacts (or parts of them) [PS33] [PS68] [PS72] [PS64].

Other papers aim to make teams aware of **roles** assigned to teams or developers within an activity or project. On EGRET [PS86], a user is able to list all stakeholders along with their status and roles in different modules. In World View [PS82], it is possible to display maps presenting team's location, time zones and responsibilities, enabling one to check the dynamics of the whole project.

Papers regarding group-structural awareness are expected to address developers' awareness regarding the full distributed process, the roles involved in each activity, and who is playing the roles on the project. Although there are a great number of papers focusing on expertise and ownership search, the area can be extended to support other development phases or to find experts on previous development phases. For example, an architect may need to be aware of experts on the domain or infrastructure of a solution he/she needs to design; a developer may need to find the analysts able to help explaining the feature affected by the code he/she is changing, etc.

5.3.5. *Summary of the classification according to the Gutwin et al. Awareness Framework*

In this subsection, we summarize the classification of the awareness features found in the papers according to the Gutwin et al. Awareness Framework. Table 9 presents the references for the primary studies classified according to the framework. The table does not include Interest Level and Expectations elements, because they were not found in the papers. Additionally, the table includes just 85 papers, because six primary studies were not classified according to this framework as they present experimental studies not mentioning any specific element.

6. Threats to validity

Systematic review is a powerful method to search for primary studies within a given domain (Kitchenham 2004). As any other method, it also presents some threats to validity and limitations.

This review may have missed some papers that address the use of awareness to support DSD, since we did not perform our search into every possible source and some relevant papers may not contain the chosen terms. To reduce bias, we defined the sources and the keywords based on other studies (Jimenez et al. 2009; Prikladnicki and Audy 2010; Dybå and Dingsøyr 2008; Williams and Carver 2010; Petersen 2011; Petersen et al. 2008), and on the subject under review. We adjusted the criteria to cover all relevant papers that were of our knowledge and conducted pilot studies. A prior version of this review was published at (Steinmacher et al. 2010). We considered the feedbacks obtained during the submission and presentation of that paper.

The findings of this review may have also been affected as the classification is a human process and it is based on some subjective criteria. In particular, the terms of the area do not have a common definition among all studies, thus we need to interpret and sometimes reword some concepts. This was particularly relevant for the classification of the elements according to the 3C Model and Gutwin et al. framework. For example, some authors called collaboration what we consider cooperation (and *vice versa*), or called coordination what we consider cooperation, etc. Finally, we only considered awareness elements that were specific to the DSD domain. Thus, general awareness elements that support communication, coordination, and cooperation, but did not have any particularity for the DSD domain were discarded. This interpretation of what is specific to the DSD domain may also suffer from interpretation bias.

To reduce interpretation bias, this review involved two researchers cross checking each paper for inclusion, and a third researcher responsible for reviewing and discussing the information generated after each step. One of the researchers, besides the academic background, has also actual experience on coordinating DSD teams, being responsible for industry global software development (GSD) projects. Another one has academic experience in coordinating a DSD project.

Table 9. Distribution of papers according to Gutwin et al. Awareness Framework elements.

	Elements	References
Workspace	Identity	[PS01] [PS02] [PS04] [PS05] [PS06] [PS07] [PS08] [PS09] [PS10] [PS11] [PS12] [PS14] [PS15] [PS16] [PS17] [PS18] [PS19] [PS22] [PS23] [PS24] [PS25] [PS26] [PS27] [PS28] [PS29] [PS30] [PS32] [PS33] [PS34] [PS36] [PS37] [PS38] [PS39] [PS40] [PS41] [PS42] [PS43] [PS44] [PS45] [PS46] [PS47] [PS48] [PS49] [PS50] [PS51] [PS52] [PS53] [PS54] [PS55] [PS56] [PS57] [PS58] [PS59] [PS60] [PS61] [PS62] [PS63] [PS64] [PS65] [PS66] [PS67] [PS68] [PS69] [PS71] [PS72] [PS73] [PS74] [PS75] [PS76] [PS77] [PS78] [PS79] [PS80] [PS81] [PS82] [PS83] [PS84] [PS85] [PS86] [PS87] [PS89] [PS90] [PS91]
	Location	[PS32] [PS43] [PS73] [PS82]
	Activity level	[PS01] [PS02] [PS07] [PS14] [PS18] [PS28] [PS42] [PS44] [PS45] [PS48] [PS50] [PS53] [PS55] [PS64] [PS66] [PS67] [PS68] [PS69] [PS71] [PS72] [PS75] [PS91]
	Actions	[PS04] [PS05] [PS06] [PS08] [PS14] [PS17] [PS18] [PS19] [PS22] [PS27] [PS29] [PS30] [PS35] [PS36] [PS37] [PS38] [PS40] [PS42] [PS46] [PS48] [PS49] [PS51] [PS52] [PS54] [PS55] [PS56] [PS57] [PS58] [PS59] [PS60] [PS62] [PS65] [PS68] [PS71] [PS73] [PS74] [PS75] [PS76] [PS77] [PS78] [PS79] [PS80] [PS84] [PS85] [PS89] [PS90] [PS91]
	Intentions	[PS08] [PS29] [PS30] [PS43] [PS56] [PS65] [PS68] [PS79]
	Changes	[PS02] [PS03] [PS04] [PS05] [PS07] [PS08] [PS10] [PS14] [PS16] [PS17] [PS18] [PS19] [PS22] [PS23] [PS27] [PS29] [PS30] [PS34] [PS35] [PS38] [PS39] [PS40] [PS41] [PS42] [PS46] [PS47] [PS48] [PS49] [PS51] [PS52] [PS53] [PS54] [PS55] [PS56] [PS57] [PS58] [PS59] [PS60] [PS61] [PS62] [PS64] [PS65] [PS66] [PS67] [PS68] [PS69] [PS71] [PS72] [PS73] [PS74] [PS75] [PS76] [PS77] [PS78] [PS79] [PS80] [PS81] [PS82] [PS84] [PS85] [PS86] [PS90] [PS91]
	Objects	[PS04] [PS05] [PS10] [PS16] [PS17] [PS18] [PS19] [PS22] [PS23] [PS24] [PS25] [PS27] [PS29] [PS30] [PS36] [PS37] [PS38] [PS39] [PS40] [PS41] [PS42] [PS46] [PS48] [PS49] [PS51] [PS52] [PS53] [PS54] [PS56] [PS57] [PS58] [PS59] [PS60] [PS61] [PS62] [PS65] [PS66] [PS67] [PS68] [PS69] [PS71] [PS72] [PS74] [PS75] [PS77] [PS79] [PS80] [PS81] [PS83] [PS84] [PS85] [PS87] [PS90] [PS91]
	Extents	[PS04] [PS05] [PS17] [PS27] [PS35] [PS38] [PS41] [PS49] [PS52] [PS56] [PS62] [PS71] [PS83] [PS84] [PS91]
	Abilities	[PS04] [PS05] [PS07] [PS25] [PS33] [PS48] [PS63]
	Sphere of Influence	[PS03] [PS07] [PS10] [PS14] [PS15] [PS16] [PS19] [PS24] [PS25] [PS36] [PS37] [PS40] [PS41] [PS42] [PS46] [PS55] [PS56] [PS58] [PS59] [PS61] [PS67] [PS68] [PS69] [PS76] [PS77] [PS79] [PS85]
Informal	Presence	[PS04] [PS05] [PS06] [PS12] [PS17] [PS28] [PS29] [PS30] [PS35] [PS38] [PS43] [PS46] [PS49] [PS56] [PS62] [PS63] [PS65] [PS75] [PS83] [PS84] [PS86] [PS89] [PS91]
	Opinion	[PS09] [PS11] [PS43]
Social	Availability	[PS04] [PS05] [PS17] [PS28] [PS43] [PS56] [PS63] [PS75] [PS84] [PS86] [PS89] [PS91]
	Emotional Feelings	[PS11] [PS17]
Group-Structural	Roles and Responsibilities	[PS04] [PS05] [PS11] [PS17] [PS33] [PS48] [PS49] [PS55] [PS65] [PS75] [PS82] [PS84] [PS86] [PS90]

7. Conclusion

In this paper, we presented the results of a systematic review and mapping on the support for awareness in DSD environments. By using a systematic method, we obtained with some degree of confidence a comprehensive picture of the state of art, which may guide other researchers to position their current researches or to find new research opportunities. We estimate that we spent around 1500 men/hour on the overall process.

During the selection of the papers, we could note the lack of standard terminology in the area, which resulted in 1967 papers to start with, which filtered, resulted in 91 relevant papers in the area. Mapping these papers (Section 4), we showed that awareness in DSD had not been widely studied until 2006 and we could note a high number of papers appearing in 2010, the year in which we ran our query.

During the mapping, we also identified the main publication forums for awareness studies in DSD and the most active researchers of the area. We could observe a small propensity for Software Engineering forums, which comprised 53

(58 %) of the relevant studies. ICSE and its workshops was the main forum, being responsible for 21 (23 %) of all the papers. Regarding the researchers of the area, we could note some groups and some concentration. The five most active researchers of the area published 26 % of the papers.

Analyzing the content of the selected papers, we could note that most part of them (70 %) focused on introducing a new tool with some awareness support to DSD. The predominant approach was to gather information from source code version management repositories to provide some kind of awareness. As depicted in Table 8, the most common awareness features identified in the papers were **change indication**, **conflict notification**, and **presence/status indication**.

In terms of classification according to the 3C collaboration model, we may conclude that most of the literature focused on awareness to support coordination, while awareness support for communication was very poorly explored. However, it is worth to note that we indeed found several tools in the area with support for communication. Many studies we have analyzed, like (Bani-Salameh et al. 2008; Duque et al. 2010; Penichet et al. 2008; Schümmer 2001), present means to support communication, such as text chat, teleconference, and forum. However, they do not present any awareness element to improve the communication in DSD scenario; they presented just general communication awareness elements. As communication issues are usually considered among the main difficulties when moving to a DSD scenario, we consider this gap a good opportunity for future research. Awareness to support communication can be fostered, for example, by taking advantage of semantics extraction, and language formalization to reduce ambiguity and misunderstandings.

In our opinion, all three dimensions have open opportunities and gaps that can become fruitful research areas (some of them were already stated in Section 5). Regarding cooperation, an open research topic is offering awareness to support software engineering phases that require cooperation among stakeholders with different roles, like clients and business analysts during requirement specification and functional tests. Coordination, even appearing as the main focus of awareness within the DSD area, still presents some open opportunities like those described in (Omoronyia et al. 2009a) and (Damian et al. 2007a), and summarized in Section 5.2.2.

Regarding Gutwin et al. Awareness Framework, we have observed that workspace elements play a central role on DSD domain. However, some elements are still scarcely studied. Embedding social environments within collaborative software engineering environments, like Jazz [PS17] and SCI [PS05] can help overcoming isolation, lack of team spirit, and other geographically related issues found on DSD. To do so, more initiatives on Social and Informal awareness elements are necessary. In addition, it is possible to observe that several concepts and techniques already investigated by the CSCW community in other contexts may be adapted or extended to the DSD domain.

In addition to the analysis made in sections 5.2 and 5.3, it is possible to highlight some other potential research topics that may be explored. We have not found papers dealing with awareness to overcome issues related to cultural, political, and geographical differences, although these issues are frequently presented as motivation on DSD studies. An opportunity is the definition of policies to provide awareness within a DSD environment and to maintain privacy and information security for team members and organizations. Another opportunity that may be explored is the use of awareness elements not only to support the coding phase, but also to support other software engineering phases, including project management, quality, and measurement. We also did not find papers linking awareness in DSD and ubiquitous and mobile computing. Merging these areas may be a promising research topic, since awareness is already being discussed in ubiquitous computing for some time (Baldauf et al. 2007).

The results presented here can be applied in distinct ways, such as for aiding the design and implementation of new DSD applications. It can also be considered a starting point to establish issues upon which subsequent researches may focus on or upon which developers may consider while designing DSD environments. This review also presented an overview of how awareness is used on a collaborative domain. The methodology applied here may also be used or adapted by other groupware researchers while conducting systematic reviews for other CSCW domains.

Acknowledgments

This work was partially funded by Fundação Araucária. The authors would also like to thank Gustavo Ansaldi Oliva and Giovanna Avalone for proofreading preliminary versions of this document. Marco Aurélio Gerosa receives individual grant from Brazilian National Research Council (CNPq).

Appendix A

Number	Source	Type of Study	Reference
PS01	Scopus	Case Study Theoretical Study	(Amrit and Hillegersberg 2008)
PS02	ACM	Case Study Tool	(Assogba and Donath 2010)
PS03	ACM	Tool	(Bang et al. 2010)
PS04	Scopus	Tool	(Bani-Salameh et al. 2009)
PS05	Scopus	Tool	(Bani-Salameh et al. 2010)
PS06	Springer	Tool	(Bani-Salameh et al. 2008)
PS07	ACM	Tool	(Begel et al. 2010)
PS08	ACM	Tool	(Begel and Zimmermann 2010)
PS09	Springer	Tool	(Belgiorno et al. 2010)
PS10	Scopus	Case Study Tool	(Bruegge et al. 2006)
PS11	ACM	Tool	(Calefato et al. 2009)
PS12	ACM	Tool	(Carter and Dewan 2010)
PS13	ACM	Theoretical Study	(Cataldo and Herbsleb 2009)
PS14	ACM	Theoretical Study	(Cataldo and Herbsleb 2010)
PS15	ACM	Case Study Theoretical Study	(Cataldo et al. 2006)

PS16	Springer	Experiment Tool	(Cepêda et al. 2010)
PS17	ACM	Tool	(Cheng et al. 2003)
PS18	IEEE	Tool	(Cook et al. 2004)
PS19	ACM	Tool	(da Silva et al. 2006)
PS20	IEEE	Case Study	(Damian et al. 2007a)
PS21	IEEE	Case Study	(Damian et al. 2007b)
PS22	ACM	Tool	(de Lucia et al. 2007a)
PS23	Science Direct	Tool	(de Lucia et al. 2007b)
PS24	ACM	Tool	(de Souza et al. 2007a)
PS25	Scopus	Case Study Tool	(de Souza et al. 2007b)
PS26	ACM	Tool	(Dekel and Herbsleb 2008)
PS27	IEEE	Experiment Tool	(Dewan et al. 2009)
PS28	IEEE	Tool	(Dullemond et al. 2010)
PS29	Springer	Case Study Tool	(Duque et al. 2008)
PS30	Science Direct	Tool	(Duque et al. 2010)
PS31	IEEE	Theoretical Study	(Dustdar and Gall 2002)
PS32	Scopus	Case Study Theoretical Study	(Ehrlich and Chang 2006)
PS33	ACM	Case Study Tool	(Ehrlich et al. 2007)
PS34	ACM	Theoretical Study Tool	(Estublier and Garcia 2005)
PS35	ACM	Tool	(Everitt et al. 2003)
PS36	ISI	Tool	(Farshchian 2000)
PS37	Science Direct	Tool	(Farshchian 2001)
PS38	Scopus	Experiment Tool	(Favela et al. 2004)
PS39	ACM	Case Study Tool	(Fitzpatrick et al. 2006)
PS40	Scopus	Tool	(Fokaefs et al. 2009)
PS41	Scopus	Experiment Tool	(Froehlich and Dourish 2004)
PS42	Scopus	Case Study Tool	(German and Hindle 2006)
PS43	ACM	Tool	(Geyer et al. 2001)
PS44	Scopus	Case Study Tool	(Gilbert and Karahalios 2007)
PS45	Scopus	Case Study Tool	(Gilbert and Karahalios 2009)
PS46	Scopus	Tool	(Guimarães and Rito-Silva 2010)
PS47	ACM	Case Study Theoretical Study	(Gutwin et al. 2004)
PS48	Scopus	Tool	(Gutwin et al. 2005)
PS49	Scopus	Experiment Tool	(Hanks 2008)
PS50	Scopus	Tool	(Harward et al. 2010)
PS51	ACM	Tool	(Hattori 2010)

PS52	ACM	Tool	(Hattori and Lanza 2010)
PS53	Scopus	Tool	(Hattori and Lanza 2009a)
PS54	Scopus	Experiment Tool	(Hattori and Lanza 2009b)
PS55	Springer	Case Study Tool	(Hattori et al. 2010)
PS56	Scopus	Experiment Tool	(Hegde and Dewan 2008)
PS57	Springer	Experiment Tool	(Helming et al. 2009)
PS58	ACM	Tool	(Holmes and Walker 2008)
PS59	ACM	Tool	(Holmes and Walker 2010)
PS60	IEEE	Tool	(Ignat 2008)
PS61	Springer	Tool	(Ignat and Oster 2008)
PS62	Scopus	Tool	(Mangan et al. 2002)
PS63	ACM	Experiment Tool	(Minto and Murphy 2007)
PS64	ACM	Tool	(Mockus and Herbsleb 2002)
PS65	IEEE	Tool	(Nutter and Boldyreff 2003)
PS66	Scopus	Case Study Tool	(Ogawa and Ma 2008)
PS67	Scopus	Tool	(Omoronyia et al. 2007)
PS68	Scopus	Theoretical Study Experiment Tool	(Omoronyia et al. 2009a)
PS69	Scopus	Tool	(Omoronyia et al. 2009b)
PS70	Scopus	Experiment	(Park and Jensen 2009)
PS71	Springer	Experiment Tool	(Penichet et al. 2008)
PS72	Springer	Tool	(Proença et al. 2010)
PS73	ISI	Theoretical Study Tool	(Redmiles et al. 2007)
PS74	Scopus	Case Study Tool	(Ripley et al. 2007)
PS75	ACM	Tool	(Salinger et al. 2010)
PS76	ACM	Experiment Tool	(Sarma et al. 2007)
PS77	ACM	Experiment Tool	(Sarma et al. 2009)
PS78	Scopus	Tool	(Sarma et al. 2003)
PS79	ACM	Experiment Tool	(Sarma et al. 2008)
PS80	IEEE	Tool	(Sarma and van der Hoek 2002)
PS81	Scopus	Tool	(Sarma and van der Hoek 2003)
PS82	IEEE	Theoretical Study Tool	(Sarma and van der Hoek 2006)
PS83	Scopus	Tool	(Schümmer 2001)
PS84	Springer	Tool	(Schümmer and Lukosch 2008)
PS85	ACM	Tool	(Servant et al. 2010)

PS86	Scopus	Case Study Tool	(Sinha et al. 2006)
PS87	Scopus	Case Study Theoretical Study	(Stapel et al. 2009)
PS88	ACM	Survey or Systematic Review	(Storey et al. 2005)
PS89	ACM	Case Study Tool	(Venolia et al. 2010)
PS90	Springer	Theoretical Study	(Wahyudin et al. 2008)
PS91	Scopus	Tool	(Ye et al. 2009)

Note

1. http://www.friendfeed.com

References

Amrit, Chintan and Jos van Hillegersberg (2008). Detecting Coordination Problems in Collaborative Software Development Environments. *Information Systems Management*, vol. 25, no. 1, pp. 57–70.

Assogba, Yannick and Judith Donath (2010). Share: a programming environment for loosely bound cooperation. In E. D. Mynatt, D. Schoner, G. Fitzpatrick, S. E. Hudson, W. K. Edwards and T. Rodden (eds): *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI'10), Atlanta, USA, April 10 to 15, 2010*. New York: ACM Press, pp. 961–970.

Baldauf, Matthias, Schahram Dustdar and Florian Rosenberg (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277.

Bang, Jae Young, Daniel Popescu, George Edwards, Nenad Medvidovic, Naveen Kulkarni, Girish M. Rama and Srinivas Padmanabhuni (2010). CoDesign: a highly extensible collaborative software modeling framework. In J. Kramer, J. Bishop, P. T. Devanbu and S. Uchitel (eds): *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10)—Volume 2, Cape Town, South Africa, May 1 to 8, 2010*. New York: ACM Press, pp. 243–246.

Bani-Salameh, Hani, Clinton Jeffery, Ziad Al-Sharif and Iyad A. Doush (2008). Integrating Collaborative Program Development and Debugging within a Virtual Environment. In R. Briggs, P. Antunes, G. de Vreede and A. Read (eds): *Proceedings of the 14th International Workshop on Groupware: Design, Implementation, and Use (CRIWG 2008), Omaha, USA, September 14 to 18, 2008*. LNCS 5411. Berlin / Heidelberg: Springer-Verlag, pp. 107–120

Bani-Salameh, Hani, Clinton Jeffery and Jafar Al-Gharaibeh (2009). SCI: Towards a Social Collaborative Integrated Development Environment. In: *Proceedings of the 12th IEEE International Conference on Computational Science and Engineering (CSE'09)—Volume 4, Vancouver, Canada, August 29 to 31, 2009*. Los Alamitos: IEEE Computer Society, pp. 915–920.

Bani-Salameh, Hani, Clinton Jeffery and Jafar Al-Gharaibeh (2010). A social collaborative virtual environment for software development. In W. K. McQuay and W. W. Smari (eds): *Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS), Chicago, USA, May 17 to 21, 2010*. Los Alamitos: IEEE Computer Society, pp. 46–55.

Begel, Andrew and Thomas Zimmermann (2010). Keeping up with your friends: function Foo, library Bar.DLL, and work item 24. In C. Treude, M. Storey, K. Ehrlich and A. van Deursen (eds): *Proceedings of the 1st Workshop on Web 2.0 for Software Engineering (Web2SE 2010), Cape Town, South Africa, May 4, 2010*. New York: ACM Press, pp. 20–23

Begel, Andrew, Yit P. Khoo and Thomas Zimmermann (2010). Codebook: discovering and exploiting relationships in software repositories. In J. Kramer, J. Bishop, P. T. Devanbu and S. Uchitel (eds): *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (*ICSE'10)*—Volume 1*, *Cape Town, South Africa, May 1 to 8, 2010*. New York: ACM Press, pp. 125–134.

Belgiorno, Furio, Ilaria Manno, Giuseppina Palmieri and Vittorio Scarano (2010). Argumentation Tools in a Collaborative Development Environment. In Y. Luo (ed): *Proceedings of the 7th International Conference on Cooperative Design, Visualization, and Engineering (CDVE 2010), Calvia, Spain, September 19 to 22, 2010*. LNCS 6240. Berlin / Heidelberg: Springer-Verlag, pp. 39–46.

Biehl, Jacob T., Mary Czerwinski, Greg Smith and George G. Robertson (2007). FASTDash: a visual dashboard for fostering awareness in software teams. In M. B. Rosson and D. J. Gilmore (eds): *Proceedings of the 2007 Conference on Human Factors in Computing Systems (CHI 2007)—Volume 2, San Jose, USA, April 28 to May 3, 2007*. New York: ACM Press, pp. 1313–1322.

Borghoff, Uwe M. and Johann H. Schlichter (2000). *Computer-Supported Cooperative Work: Introduction to Distributed Applications*. Secaucus, USA: Springer-Verlag New York.

Bruegge, Bernd, Allen H. Dutoit and Timo Wolf (2006). Sysiphus: Enabling informal collaboration in global software development. In P. Fernandes, D. Damian, C. Ebert, A. Avritzer and D. Paulish (eds): *Proceedings of the 2006 IEEE International Conference on Global Software Engineering (ICGSE 2006), Florianopolis, Brazil, October 16 to 19, 2006*. Los Alamitos: IEEE Computer Society, pp. 139–148.

Calefato, Fabio, Domenico Gendarmi and Filippo Lanubile (2009). Embedding social networking information into jazz to foster group awareness within distributed teams. In I. Hammouda, F. Lanubile, J. Bosch and M. Jazayeri (eds): *Proceedings of the 2nd International Workshop on Social Software Engineering and Applications (SoSEA'09), Amsterdam, The Netherlands, August 24, 2009*. New York: ACM Press, pp. 23–28.

Carmel, Erran (1999). *Global software teams: collaborating across borders and time zones*. Upper Saddle River, USA: Prentice Hall PTR.

Carter, Jason and Prasun Dewan (2010). Are you having difficulty? In K. Inkpen, C. Gutwin and J. C. Tang (eds): *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, *(CSCW 2010), Savannah, Georgia, USA, February 6 to 10*. New York: ACM Press, pp. 211–214.

Casey, Valentine (2010). Developing Trust In Virtual Software Development Teams, *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 5, no. 2, pp. 41–58.

Cataldo, Marcelo and James D. Herbsleb (2009). End-to-end features as meta-entities for enabling coordination in geographically distributed software development. In Y. Dubinsky and P. Kruchten (eds): *Proceedings of the 2009 ICSE Workshop on Software Development Governance (SDG'09), Vancouver, Canada, May 17, 2009*. Los Alamitos: IEEE Computer Society, pp. 21–26.

Cataldo, Marcelo and James D. Herbsleb (2010). Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration. In C. E. Cuesta (ed): *Fourth European Conference on Software Architecture (ECSA 2010): Doctoral Symposium, Industrial Track and Workshops—Companion Volume, Copenhagen, Denmark, August 23 to 26, 2010*. New York: ACM Press, pp. 65–72.

Cataldo, Marcelo, Patrick A. Wagstrom, James D. Herbsleb and Kathleen M. Carley (2006). Identification of coordination requirements: implications for the Design of collaboration and awareness tools. In P. J. Hinds and D. Martin (eds): *Proceedings of the 20th Anniversary Conference on Computer Supported Cooperative Work (CSCW 2006), Banff, Canada, November 4 to 8, 2006*. New York: ACM Press, pp. 353–362.

Cepêda, Rafael D., Andréa Magdaleno, Leonardo Murta and Cláudia Werner (2010). EvolTrack: improving design evolution awareness in software development. *Journal of the Brazilian Computer Society*, vol. 16, no. 2, pp. 117–131.

Chang, Klarissa T. and Kate Ehrlich (2007). Out of sight but not out of mind?: Informal networks, communication and media use in global software teams. In B. Spencer, M. Storey and D. Stewart (eds): *Proceedings of the 2007 Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2007), Richmond Hill, Canada, October 22 to 25, 2007*. New York: ACM Press, pp. 86–97.

Cheng, Li-Te, Susanne Hupfer, Steven Ross and John Patterson (2003). Jazzing up Eclipse with collaborative tools. In M. G. Burke (ed): *Proceedings of the 2003 OOPSLA Workshop on Eclipse Technology eXchange, Anaheim, USA, October 27, 2003*. New York: ACM Press, pp. 45–49.

Cook, Carl, Neville Churcher and Warwick Irwin (2004). Towards synchronous collaborative software engineering. In K. H. Kwon (ed): *Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC 2004), Busan, Korea, November 30 to December 3, 2004*. Los Alamitos: IEEE Computer Society, pp. 230–239.

Damian, Daniela, Luis Izquierdo, Janice Singer and Irwin Kwan (2007a). Awareness in the Wild: Why Communication Breakdowns Occur. In F. Paulisch, P. Kruchten and A. Mockus (eds): *Proceedings of Second IEEE International Conference on Global Software Engineering (ICGSE 2007), Munich, Germany, August 27 to 30, 2007*. Los Alamitos: IEEE Computer Society, pp. 81–90.

Damian, Daniela, Sabrina Marczak and Irwin Kwan (2007b). Collaboration Patterns and the Impact of Distance on Awareness in Requirements-Centred Social Networks. In A. Sutcliffe and P. Jalote (eds): *Proceedings of the 15th IEEE International Requirements Engineering Conference (RE'07), New Delhi, India, October 15 to 19, 2007*. Los Alamitos: IEEE Computer Society, pp. 59–68.

da Silva, Isabella A., Ping H. Chen, Christopher van der Westhuizen, Roger M. Ripley and André van der Hoek (2006). Lighthouse: coordination through emerging design. In M. G. Burke, A. Orso and M. P. Robillard (eds): *Proceedings of the 2006 OOPSLA workshop on Eclipse Technology eXchange, Portland, USA, October 22 to 23, 2006*. New York: ACM Press, pp. 11–15.

de Lucia, Andrea, Fausto Fasano, Rocco Oliveto and Domenico Santonicola (2007a). Improving Context Awareness in Subversion Through Fine-grained Versioning of Java Code. In M. di Penta and M. Lanza (eds): *9th International Workshop on Principles of Software Evolution: in conjunction with the 6th ESEC/FSE Joint Meeting (IWPSE'07), Dubrovnik, Croatia, September 3 to 7, 2007*. New York: ACM Press, pp. 110–113.

de Lucia, Andrea, Fausto Fasano, Giuseppe Scanniello and Genny Tortora (2007b). Enhancing collaborative synchronous UML modelling with fine-grained versioning of software artefacts. *Journal of Visual Languages and Computing*, vol. 18, no. 5, pp. 492–503.

de Souza, Cleidson R.B. and David F. Redmiles (2011). The Awareness Network, To Whom Should I Display My Actions? And, Whose Actions Should I Monitor? *IEEE Transactions on Software Engineering*, vol. 37, no. 3, pp. 325–340

de Souza, Cleidson R.B., Tobias Hildenbrand and David Redmiles (2007a). Toward visualization and analysis of traceability relationships in distributed and offshore software development projects. In B. Meyer and M. Joseph (eds): *Proceedings of the First International Conference on Software Engineering Approaches for Offshore and Outsourced Development (SEAFOOD), Zurich, Switzerland, February 5 to 6, 2007*. LNCS 4716. Berlin / Heidelberg: Springer-Verlag, pp. 182–199.

de Souza, Cleidson R., Stephen Quirk, Erik Trainer and David F. Redmiles (2007). Supporting collaborative software development through the visualization of socio-technical dependencies. In T. Gross and K. Inkpen (eds): *Proceedings of the 2007 International ACM Conference on Supporting Group Work (GROUP'07), Sanibel Island, USA, November 4 to 7, 2007*. New York: ACM Press, pp. 147–156.

Dekel, Uri and James D. Herbsleb (2008). Pushing relevant artifact annotations in collaborative software development. In B. Begole and D. W. McDonald (eds): *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW'08), San Diego, USA, November 8 to 12, 2008*. New York: ACM Press, pp. 1–4.

Dewan, Prasun, Puneet Agarwal, Gautam Shroff and Rajesh Hegde (2009). Distributed side-by-side programming. In J. Singer, H. Sharp, Y. Dittrich and C. de Souza (eds): *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2009), Vancouver, Canada, May 17, 2009*. Los Alamitos: IEEE Computer Society, pp. 48–55.

Dullemond, Kevin, Ben van Gameren and Rini v. Solingen (2010). Virtual Open Conversation Spaces: Towards Improved Awareness in a GSE Setting. In A. Avritzer, Y. Dubinsky and A. Milewski (eds): *Proceedings of the 2010 IEEE International Conference on Global Software Engineering (ICGSE 2010), Princeton, USA, August 23 to 26, 2010*. Los Alamitos: IEEE Computer Society, pp. 247–256.

Duque, Rafael, María L. Rodríguez, María V. Hurtado, Manuel Noguera and Crescencio Bravo (2008). An Architecture to Integrate Automatic Observation Mechanisms for Collaboration Analysis in Groupware. In R. Meersman, Z. Tari and P. Herrero (eds): *Proceedings of the OTM Confederated International Workshops and Posters on On the Move to Meaningful Internet Systems, Monterrey, Mexico, November 9 to 14, 2008*. LNCS 5333. Berlin / Heidelberg: Springer-Verlag, pp. 354–363

Duque, Rafael, María Luisa Rodríguez, María Visitación Hurtado, Crescencio Bravo and Carlos Rodríguez-Domínguez (2010). Integration of collaboration and interaction analysis mechanisms in a concern-based architecture for groupware systems. *Science of Computer Programming*, vol. 77, no. 1, pp. 29–45.

Dustdar, Schahram and Harald Gall (2002). Process awareness for distributed software development in virtual teams. In M. Fernandez, I. Crnkovic, G. Fohler, C. Griwodz, T. Plagemann and P. Gruenbacher (eds): *Proceedings of the 28th Euromicro Conference (Euromicro 2002), Dortmund, Germany, September 4 to 6, 2002*. Los Alamitos: IEEE Computer Society, pp. 244–250.

Dybå, Tore and Torgeir Dingsøyr (2008). Strength of evidence in systematic reviews in software engineering. In H. D. Rombach, S. G. Elbaum and J. Münch (eds): *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM'08), Kaiserslautern, Germany, October 9 to 10, 2008*. New York: ACM Press, pp. 178–187.

Ehrlich, Kate and Klarissa Chang (2006). Leveraging expertise in global software teams: Going outside boundaries In P. Fernandes, D. Damian, C. Ebert, A. Avritzer and D. Paulish (eds): *Proceedings of the 2006 IEEE International Conference on Global Software Engineering (ICGSE 2006) Florianopolis, Brazil, October 16 to 19, 2006*. Los Alamitos: IEEE Computer Society, pp. 149–158.

Ehrlich, Kate, Ching-Yung Lin and Vicky Griffiths-Fisher (2007). Searching for experts in the enterprise: combining text and social network analysis. In T. Gross and K. Inkpen (eds): *Proceedings of the 2007 International ACM Conference on Supporting Group Work (GROUP'07) Sanibel Island, USA, November 4 to 7, 2007*. New York: ACM Press, pp. 117–126.

Ellis, Clarence A., Simon J. Gibbs and Gail Rein (1991). Groupware: some issues and experiences. *Communications of ACM*, vol. 34, no. 1, pp. 39–58.

Espinosa, J. A., Sandra A. Slaughter, Robert E. Kraut and James D. Herbsleb (2007). Team Knowledge and Coordination in Geographically Distributed Software Development. *Journal of Management Information Systems*, vol. 24, no. 1, pp. 135–169.

Estublier, Jacky and Sergio Garcia (2005). Process model and awareness in SCM. In E. J. Whitehead, Jr,. A. P. Dahlqvist (eds): *Proceedings of the 12th International Workshop on*

Software Configuration Management (SCM'05), Lisbon, Portugal, September 5 to 6, 2005. New York: ACM Press, pp. 59–74.

Everitt, Katherine M., Scott R. Klemmer, Robert Lee and James A. Landay (2003). Two worlds apart: bridging the gap between physical and virtual media for distributed design collaboration. In G. Cockton and P. Korhonen (eds): *Proceedings of the 2003 Conference on Human Factors in Computing Systems (CHI'03), Fort Lauderdale, USA, April 5 to 10, 2003*. New York: ACM Press, pp. 553–560.

Farshchian, Babak A. (2000). Gossip: An awareness engine for increasing product awareness in distributed development projects. In B. Wangler and L. Bergman (eds): *Proceedings of the 12th International Conference on Advanced Information Systems Engineering (CAiSE 2000), Stockholm, Sweden, June 5 to 9, 2000. LNCS 1789*. Berlin / Heidelberg: Springer-Verlag, pp. 264–278.

Farshchian, Babak A. (2001). Integrating geographically distributed development teams through increased product awareness. *Information Systems*, vol. 26, no. 3, pp. 123–141.

Favela, Jesús, Hiroshi Natsu, Cynthia B. Pérez, Omar Robles, Alberto L. Morán, Raul Romero, Ana M. Martínez-Enríquez and Dominique Decouchant (2004). Empirical evaluation of collaborative support for distributed pair programming. In G. de Vreede, L. A. Guerrero and G. M. Raventós (eds): *Proceedings of the 10th International Workshop on Groupware: Design, Implementation and Use (CRIWG 2004), San Carlos, Costa Rica, September 5 to 9, 2004*. LNCS 3198. Berlin / Heidelberg: Springer-Verlag, pp. 215–222.

Fitzpatrick, Geraldine, Paul Marshall and Anthony Phillips (2006). CVS integration with notification and chat: lightweight software team collaboration. In P. J. Hinds and D. Martin (eds): *Proceedings of the 20th Anniversary Conference on Computer Supported Cooperative Work (CSCW 2006), Banff, Canada, November 4 to 8, 2006*. New York: ACM Press, pp. 49–58.

Fokaefs, Marios, Ken Bauer and Eleni Stroulia (2009). WikiDev 2.0: Web-based software team collaboration. In A. Aguiar, U. Dekel and P. Merson (eds): *Proceedings of the ICSE Workshop on Wikis for Software Engineering* (*Wikis4SE'2009), Vancouver, Canada, May 19, 2009*. Los Alamitos: IEEE Computer Society, pp. 67–77.

Froehlich, Jon and Paul Dourish (2004). Unifying artifacts and activities in a visual tool for distributed software development teams. In : A. Finkelstein, J. Estublier and D. S. Rosenblum (eds)*: Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), Edinburgh, Scotland, May 23 to 28, 2004*. Los Alamitos: IEEE Computer Society, pp. 387–396.

Fuks, Hugo, Alberto B. Raposo, Marco A. Gerosa, Mariano Pimentel and Carlos J.P. Lucena (2007) The 3 C Collaboration Model, In N. Kock (ed) *The encyclopedia of e-Collaboration*, Hershey, USA: Information Science Reference, pp. 637–644.

Fuks, Hugo , Alberto Raposo, Marco A. Gerosa, Mariano Pimentel, Denise Filippo and Carlos Lucena (2008). Inter- and intra-relationships between communication coordination and cooperation in the scope of the 3 C Collaboration Model. In W. Shen, Q. Zheng, J. Barthès, J. Luo, J. Yong, Z. Duan and F. Tian (eds): *Proceedings of the 2008 12th International Conference on Computer Supported Cooperative Work in Design, 2008 (CSCWD 2008), Xi'an, China, April 16 to 18, 2008*. Los Alamitos: IEEE Computer Society, pp. 148–153.

German, Daniel and Abram Hindle (2006). Visualizing the Evolution of Software Using SoftChange. *International Journal of Software Engineering and Knowledge Engineering*, vol. 16, no.1, pp. 5–21.

Geyer, Werner, Heather Richter, Ludwin Fuchs, Tom Frauenhofer, Shahrokh Daijavad and Steven Poltrock (2001). A team collaboration space supporting capture and access of virtual meetings. In C. Ellis and I. Zigurs (eds): *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work (GROUP'01), Boulder, USA, September 30 to October 3, 2001*. New York: ACM Press, pp. 188–196.

Gilbert, Eric and Karrie Karahalios (2007). CodeSaw: A social visualization of distributed software development. In M. C. C. Baranauskas, P. A. Palanque, J. Abascal and S. D.J. Barbosa (eds): *Proceedings of the 11th IFIP TC 13 International Conference on Human-Computer Interaction, (INTERACT 2007), Part II, Rio de Janeiro, Brazil, September 10 to 14, 2007*. LNCS 4663. Berlin / Heidelberg: Springer-Verlag, pp. 303–316.

Gilbert, Eric and Karrie Karahalios (2009). Using social visualization to motivate social production. *IEEE Transactions on Multimedia*, vol. 11, no. 3, pp. 413–421.

Godefroid, Patrice, James D. Herbsleb, Lalita J. Jagadeesany and Du Li (2000). Ensuring privacy in presence awareness: an automated verification approach. In W. A. Kellogg and S. Whittaker (eds): *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (CSCW 2000), Philadelphia, USA, December 2 to 6, 2000*. New York: ACM Press, pp. 59–68.

Guimarães, Mário L. and António Rito-Silva (2010). Towards real-time integration. In Y. Dittrich, C. De Souza, M. Korpela, H. C. Sharp, J. Singer and H. Winshchiers-Theophilus (eds): *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2010), Cape Town, South Africa, May 2, 2010*. New York: ACM Press, pp. 56–63.

Gutwin, Carl and Saul Greenberg (2002). A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work,* vol. 11, no. 2, pp. 411–446.

Gutwin, Carl, Saul Greenberg and Mark Roseman (1996). Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. In M. A. Sasse, J. Cunningham, and R. L. Winder (eds): *Proceedings of the Eleventh Conference of the British Computer Society Human Computer Interaction Specialist Group—People and Computers XI (HCI'96), London, United Kingdom, August 20 to 23, 1996*. London: Springer-Verlag, pp. 281–298.

Gutwin, Carl, Reagan Penner and Kevin Schneider (2004). Group awareness in distributed software development. In J. D. Herbsleb and G. M. Olson (eds): *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW 2004), Chicago, USA, November 6 to 10, 2004*. New York: ACM Press, pp. 72–81.

Gutwin, Carl, Kevin Schneider, David Paquette and Reagan Penner (2005). Supporting group awareness in distributed software development. In R. Bastide, P. A. Palanque and J. Roth (eds): *Engineering Human Computer Interaction and Interactive Systems: Joint Working Conferences EHCI-DSVIS 2004, Hamburg, Germany, July 11 to 13, 2004*. LNCS 3425. Berlin / Heidelberg: Springer-Verlag, pp. 383–397.

Hanks, Brian (2008). Empirical evaluation of distributed pair programming. *International Journal of Human Computer Studies*, vol. 66, no. 7, pp. 530–544.

Harward, Matthew, Warwick Irwin and Neville Churcher (2010). In situ software visualisation. In J. Noble and C. Fidge (eds): *Proceedings of the 21st Australian Software Engineering Conference (ASWEC 2010), Auckland, New Zealand, April 6 to 9, 2010*. Los Alamitos: IEEE Computer Society, pp. 171–180.

Hattori, Lile (2010). Enhancing collaboration of multi-developer projects with synchronous changes. In J. Kramer, J. Bishop, P. T. Devanbu and S. Uchitel (eds): *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering* (*ICSE '10)—Volume 2, Cape Town, South Africa, May 1 to 8, 2010*. New York: ACM Press, pp. 377–380.

Hattori, Lile and Michele Lanza (2009a). An environment for synchronous software development. In S. Fickas, J. M. Atlee and P. Inverard (eds): *Proceedings of the 31st International Conference on Software Engineering (ICSE 2009)—Companion Volume, Vancouver, Canada, May 16 to 24, 2009*. Los Alamitos: IEEE Computer Society, pp. 223–226.

Hattori, Lile and Michele Lanza (2009b). Mining the history of synchronous changes to refine code ownership. In M. W. Godfrey and J. Whitehead (eds): *Proceedings of the 6th International Working Conference on Mining Software Repositories (MSR 2009), Vancouver, Canada, May 16 to 17, 2009*. Los Alamitos: IEEE Computer Society, pp. 141–150.

Hattori, Lile and Michele Lanza (2010). Syde: a tool for collaborative software development. In J. Kramer, J. Bishop, P. T. Devanbu and S. Uchitel (eds): *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10)—Volume 2, Cape Town, South Africa, May 1 to 8, 2010*. New York: ACM Press, pp. 235–238.

Hattori, Lile, Michele Lanza and Romain Robbes (2010). Refining code ownership with synchronous changes. *Empirical Software Engineering*, pp. 1–33.

Hegde, Rajesh and Prasun Dewan (2008). Connecting programming environments to support ad-hoc collaboration. In A. Ireland and W. Visser (eds): *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering (ASE 2008), L'Aquila, Italy, September 15 to 19, 2008*. Los Alamitos: IEEE Computer Society, pp. 178–187.

Helming, Jonas, Maximilian Koegel, Helmut Naughton, Joern David and Aleksandar Shterev (2009). Traceability-Based Change Awareness. In A. Schürr and B. Selic (eds): *Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2009), Denver, USA, October 4 to 9, 2009*. LNCS 5795. Berlin / Heidelberg: Springer-Verlag, pp. 372–376.

Holmes, Reid and Robert J. Walker (2008). Promoting developer-specific awareness. In L. Cheng, J. Sillito, M. D. Storey, B. Tessem, G. Venolia, C. R. B. de Souza, Y. Dittrich, M. John, O. Hazzan, F. Maurer, H. Sharp, J. Singer and S. E. Sim (eds): *Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2008)*, Leipzig, Germany, May 13, 2008. New York: ACM Press, pp. 61–64.

Holmes, Reid and Robert J. Walker (2010). Customized awareness: recommending relevant external change events. In J. Kramer, J. Bishop, P. T. Devanbu and S. Uchitel (eds): *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10)—Volume 1, Cape Town, South Africa, May 1 to 8, 2010*. New York: ACM Press, pp. 465–474.

Ignat, Claudia-Lavinia (2008). Annotation of concurrent changes in collaborative software development. In I. A. Letia (ed): *Proceedings of the 2008 IEEE 4th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, August 28 to 30, 2008*. Los Alamitos: IEEE Computer Society, pp. 137–144.

Ignat, Claudia-Lavinia and Gérald Oster (2008). Awareness of Concurrent Changes in Distributed Software Development. In R. Meersman and Z. Tari (eds): *Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Part I, Monterrey, Mexico, November 9 to 14, 2008*. LNCS 5331. Berlin / Heidelberg: Springer-Verlag, pp. 456–464.

Ivcek, Mario and Tihana Galinac (2008). Aspects of Quality Assurance in Global Software Development Organization. In P. Biljanovic and K. Skala (eds): *Proceedings of the International Conference on Telecommunications and Information of the 31st International Convention MIPRO 2008, Opatija, Croatia, June 26 to 30, 2008*. Rijeka: MIPRO, pp. 150–155.

Jensen, Morten, Shashi Menon, Lars E. Mangset and Vibeke Dalberg (2007). Managing Offshore Outsourcing of Knowledge-intensive. In F. Paulisch, P. Kruchten and A. Mockus (eds): *Proceedings of Second IEEE International Conference on Global Software Engineering (ICGSE 2007), Munich, Germany, August 27 to 30, 2007*. Los Alamitos: IEEE Computer Society, pp. 186–196.

Jimenez, Miguel, Mario Piattini and Aurora Vizcaíno (2009). Challenges and Improvements in Distributed Software Development: A Systematic Review. *Advances in Software Engineering*, vol. 2009, pp. 1–14.

Kaariainen, Jukka and Antti Valimaki (2011). Get a grip on your distributed software development with application lifecycle management. *International Journal of Computer Applications in Technology*, vol. 40, no. 3, pp. 181–190.

Kitchenham, Barbara (2004). *Procedures for performing systematic reviews*. Technical Report, TR/SE-0401, Department of Computer Science, Keele University, UK.

Kitchenham, Barbara and Stuart Charters (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Technical Report, EBSE 2007–01, Keele University and Durham University Joint Report, UK.

Kitchenham, Barbara, Dag I.K. Sjøberg, O. P. Brereton, David Budgen, Tore Dybå, Martin Höst, Dietmar Pfahl and Per Runeson (2010). Can we evaluate the quality of software engineering experiments? In G. Succi, M. Morisio and N. Nagappan (eds): *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2010), Bolzano-Bozen, Italy, September 16 to 17, 2010*. New York: ACM Press, pp. 2:1–2:8.

Landis, J. R. and Gary G. Koch (1977). The measurement of observer agreement for categorical data. *Biometrics*, vol. 1, no. 33, pp. 159–174.

Mangan, Marco A.S., Renata M. Araujo, Marcos Kalinowski, Marcos R.S. Borges and Cláudia M.L. Werner (2002). Towards the Evaluation of Awareness Information Support Applied to Peer Reviews of Software Engineering Diagrams. In W. Shen, J. M. de Souza, J. A. Barthès and Z. Lin (eds): *Proceedings of the Seventh International Conference on CSCW in Design (CSCWD 2002), Rio de Janeiro, Brazil, September 25 to 27, 2002*. Los Alamitos: IEEE Computer Society, pp. 49–54.

Mian, Paula, Tayana Conte, Ana Natali, Jorge Biolchini and Guilherme Travassos (2005). A Systematic Review Process for Software Engineering. In: *Proceedings of 2nd Experimental Software Engineering Latin American Workshop (ESELAW 2005), Uberlandia, Brazil*, *October 3, 2005.* Porto Alegre : SBC, pp. 1–6.

Minto, Shawn and Gail C. Murphy (2007). Recommending Emergent Teams. In H. Gall and M. Lanza (eds): *Proceedings of Fourth International Workshop on Mining Software Repositories (MSR'07), Minneapolis, USA, May 19 to 20, 2007*. Los Alamitos: IEEE Computer Society, pp. 5.

Mockus, Audris and James D. Herbsleb (2002). Expertise browser: a quantitative approach to identifying expertise. In M. Young and J. Magee (eds): *Proceedings of the 24th International Conference on Software Engineering (ICSE'02), Orlando, USA, May 19 to 25, 2002*. New York: ACM Press, pp. 503–512.

Muhammad, Aslam, Ana María M. Enríquez and Dominique Decouchant (2005). Awareness and Coordination for Web Cooperative Authoring. In P. S. Szczepaniak, J. Kacprzyk and A. Niewiadomski (eds): *Proceedings of Third International Atlantic Web Intelligence Conference—Advances in Web Intelligence (AWIC 2005), Lodz, Poland, June 6 to 9, 2005.* LNCS 3528. Berlin / Heidelberg: Springer-Verlag, pp. 327–333.

Neale, Dennis C., John M. Carroll and Mary B. Rosson (2004). Evaluating computer-supported cooperative work: models and frameworks. In J. D. Herbsleb and G. M. Olson (eds): *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW 2004), Chicago, USA, November 6 to 10, 2004*. New York: ACM Press, pp. 112–121.

Nutter, David and Cornelia Boldyreff (2003). Historical awareness support and its evaluation in collaborative software engineering. In G. Kotsis and S. Reddy (eds): *Proceedings of Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2003), Linz, Austria, June 9 to 11, 2003*. Los Alamitos: IEEE Computer Society, pp. 171–176.

Ogawa, Michael. and Kwan-Liu Ma (2008). StarGate: A unified, interactive visualization of software projects. In I. Fujishiro, H. Li and K. Ma (eds): *Proceedings of IEEE Pacific Visualization Symposium 2008 (PacificVIS'08), Kyoto, Japan, March 4 to 7, 2008*. Los Alamitos: IEEE Computer Society, pp. 191–198.

Omoronyia, Inah, John Ferguson, Marc Roper and Murray Wood (2007). A 3-dimensional relevance model for collaborative software engineering spaces. In F. Paulisch, P. Kruchten and A. Mockus (eds): *Proceedings of Second IEEE International Conference on Global Software Engineering (ICGSE 2007), Munich, Germany, August 27 to 30, 2007*. Los Alamitos: IEEE Computer Society, pp. 204–216.

Omoronyia, Inah, John Ferguson, Marc Roper and Murray Wood (2009a). Using developer activity data to enhance awareness during collaborative software development. *Computer Supported Cooperative Work*, vol. 18, no. 5–6, pp. 509–558.

Omoronyia, Inah, Guttorm, Sindre, Marc Roper, John Ferguson and Murray Wood (2009b). Use case to source code traceability: The developer navigation view point. In K. Ryan and B. Robinson (eds): *Proceedings of the 2009 17th IEEE International Requirements Engineering Conference (RE 2009), Atlanta, USA, August 31 to September 4, 2009*. Los Alamitos: IEEE Computer Society, pp. 237–242

Omoronyia, Inah, John Ferguson, Marc Roper and Murray Wood (2010). A review of awareness in distributed collaborative software engineering. *Software: Practice and Experience*, vol. 40, no. 12, pp. 1107–1133.

Park, Y. and C. Jensen (2009). Beyond pretty pictures: Examining the benefits of code visualization for open source newcomers. In M. Lanza, H. A. Müller and M. Storey (eds): *Proceedings of the 5th IEEE International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 2009), Edmonton, Canada, September 25, 2009*. Los Alamitos: IEEE Computer Society, pp. 3–10.

Patil, Sameer, Alfred Kobsa, Ajita John, Lynne Brotman and Doree Seligmann (2009). Interpersonal Privacy Management in Distributed Collaboration: Situational Characteristics and Interpretive Influences. In T. Gross, J. Gulliksen, P. Kotzé, L. Oestreicher, P. Palanque, R. Prates and M. Winckler (eds): *Proceedings of 12th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT 2009)—Part II, Uppsala, Sweden, August 24 to 28, 2009*. LNCS 5727. Berlin / Heidelberg: Springer-Verlag, pp. 143–156.

Penichet, Victor, Jose Gallud, Ricardo Tesoriero and Maria Lozano (2008). Design and Evaluation of a Service Oriented Architecture-Based Application to Support the Collaborative Edition of UML Class Diagrams. In M. Bubak, G. van Albada, J. Dongarra and P. Sloot (eds): *Proceedings of the 8th International Conference on Computational Science (ICCS 2008)—Part III, Kraków, Poland, June 23 to 25, 2008*. LNCS 5103. Berlin / Heidelberg: Springer-Verlag, pp. 389–398.

Petersen, Kai (2011). Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology*, vol. 53, no. 4, pp. 317–343.

Petersen, Kai, Robert Feldt, Shahid Mujtaba and Michael Mattsson (2008). Systematic Mapping Studies in Software Engineering. In G. Visaggio, M. T. Baldassarre, S. Linkman and M. Turner (eds): *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE 2008), Bari, Italy, June 26 to 27, 2008*. London: British Computer Society, pp. 71–80.

Prikladnicki, Rafael and Jorge L.N. Audy (2010). Process models in the practice of distributed software development: A systematic review of the literature. *Information and Software Technology*, vol. 52, no. 8, pp. 779–791.

Proença, Tiago, Nilmax Moura and André van der Hoel (2010). On the Use of Emerging Design as a Basis for Knowledge Collaboration. In K. Nakakoji, Y. Murakami and E. McCready (eds): *New Frontiers in Artificial Intelligence: JSAI-isAI 2009 Workshops LENLS, JURISIN, KCSD, LLLL, Tokyo, Japan, November 19 to 20, 2009*. LNCS 6284. Berlin / Heidelberg: Springer-Verlag, pp. 124–134.

Redmiles, David, Andre van der Hoek, Ban Al-Ani, Tobias Hildenbrand, Stephen Quirk, Anita Sarma, Roberto S. Silva, Cleidson R.B.M. de Souza and Erik Trainer (2007). Continuous Coordination: A new paradigm to support globally distributed software development projects. *Wirtschaftsinformatik,* vol. 49, pp. S28–S38.

Ripley, Roger M., Anita Sarma and André van der Hoek (2007). A visualization for software project awareness and evolution. In J. I. Maletic, A. Telea and A. Marcus (eds): *Proceedings of the 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis*

(VISSOFT 2007), Banff, Canada, June 25 to 26, 2007. Los Alamitos: IEEE Computer Society, pp. 137–144.

Robinson, Marcia and Ravi Kalakota (2004). *Offshore Outsourcing: Business Models, ROI and Best Practices*. Alpharetta, USA: Mivar Press.

Robinson, Hugh and Helen Sharp (2005). Organisational culture and XP: three case studies. In: *Proceedings of the Agile Development Conference 2005, Denver, USA, July 24 to 29, 2005*. Los Alamitos: IEEE Computer Society, pp. 49–58.

Salinger, Stephan, Christopher Oezbek, Karl Beecher and Julia Schenk (2010). Saros: an eclipse plug-in for distributed party programming. In Y. Dittrich, C. R. de Souza, M. Korpela, H. C. Sharp, J. Singer and H. Winshchiers-Theophilus (eds): *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2010), Cape Town, South Africa, May 2, 2010*. New York: ACM Press, pp. 48–55.

Sarma, Anita and André van der Hoek (2002). Palantír: Coordinating Distributed Workspaces. In: *Proceedings of 26th International Computer Software and Applications Conference (COMPSAC 2002), Oxford, England, August 26 to 29, 2002*. Los Alamitos: IEEE Computer Society, pp. 1093–1097

Sarma, Anita and André van der Hoek (2003). Visualizing parallel workspace activities. In M. H. Hamza (ed): *Proceedings of the IASTED International Conference on Software Engineering and Applications, Marina del Rey, USA, November 3 to 5, 2003*. Calgary, Canada: Acta Press, pp. 435–440.

Sarma, Anita and André van der Hoek (2006). Towards Awareness in the Large. In P. Fernandes, D. Damian, C. Ebert, A. Avritzer and D. Paulish (eds): *Proceedings of the 2006 IEEE International Conference on Global Software Engineering (ICGSE 2006), Florianopolis, Brazil, October 16 to 19, 2006*. Los Alamitos: IEEE Computer Society, pp. 127–131.

Sarma, Anita, Zahra Noroozi and André van der Hoek (2003). Palantìr: Raising awareness among configuration management workspaces. In L. Clarke, L. Dillon and W. Tichy (eds): *Proceedings of the 25th International Conference on Software Engineering (ICSE 2003), Portland, USA, May 3 to 10, 2003*. Los Alamitos: IEEE Computer Society, pp. 444–454.

Sarma, Anita, Gerald Bortis and André van der Hoek (2007). Towards supporting awareness of indirect conflicts across software configuration management workspaces. In R. E. K. Stirewalt, A. Egyed and B. Fischer (eds): *Proceedings of the 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE'07), Atlanta, USA, November 5 to 9, 2007*. New York: ACM Press, pp. 94–103.

Sarma, Anita, David Redmiles and André van der Hoek (2008). Empirical evidence of the benefits of workspace awareness in software configuration management. In M. J. Harrold and G. C. Murphy (eds): *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2008, Atlanta, USA, November 9 to 14, 2008*. New York: ACM Press, pp. 113–123.

Sarma, Anita, Larry Maccherone, Patrick Wagstrom and James Herbsleb (2009). Tesseract: Interactive visual exploration of socio-technical relationships in software development. In S. Fickas, J. M. Atlee and P. Inverard (eds): *Proceedings of the 31st International Conference on Software Engineering (ICSE 2009), Vancouver, Canada, May 16 to 24, 2009*. Los Alamitos: IEEE Computer Society, pp. 23–33.

Schmidt, Kjeld (2002). The Problem with 'Awareness': Introductory Remarks on 'Awareness in CSCW'. *Computer Supported Cooperative Work*, vol. 11, no. 3, pp. 285–298.

Schümmer, Till (2001). Lost and found in software space. In R. H. Sprague (ed): *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)—Volume 9, Maui, Hawaii, January 3 to 6, 2001*. Los Alamitos: IEEE Computer Society, pp. 1–10.

Schümmer, Till and Stephan Lukosch (2008). Supporting the Social Practices of Distributed Pair Programming. In R. Briggs, P. Antunes, G. de Vreede and A. Read (eds): *Proceedings of the*

14th International Workshop on Groupware: Design, Implementation, and Use (CRIWG 2008), Omaha, USA, September 14 to 18, 2008. LNCS 5411. Berlin / Heidelberg: Springer-Verlag, pp. 83–98.

Servant, Francisco, James A. Jones and André van der Hoek (2010). CASI: preventing indirect conflicts through a live visualization. In Y. Dittrich, C. De Souza, M. Korpela, H. C. Sharp, J. Singer and H. Winshchiers-Theophilus (eds): *Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2010), Cape Town, South Africa, May 2, 2010*. New York: ACM Press, pp. 39–46.

Sinha, Vibha, Bikram Sengupta and Satish Chandra (2006). Enabling collaboration in distributed requirements management. *IEEE Software*, vol. 23, no. 5, pp. 52–61.

Stapel, Kai, Eric Knauss and Kurt Schneider (2009). Using FLOW to improve communication of requirements in globally distributed software projects. In: *Proceedings of the 2009 Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills (CIRCUS'09), Atlanta, USA, August 31, 2009*. Los Alamitos: IEEE Computer Sciences, pp. 15–22.

Steinmacher, Igor, Ana Paula Chaves and Marco Aurélio Gerosa (2010). Awareness support in global software development: a systematic review based on the 3 C collaboration model. In G. L. Kolfschoten, T. Herrmann and S. Lukosch (eds): *16th International Conference Collaboration and Technology (CRIWG 2010), Maastricht, The Netherlands, September 20 to 23, 2010*. LNCS 6257. Berlin / Heidelberg: Springer-Verlag, pp. 185–201.

Storey, Margaret-Anne D., Davor Cubranic and Daniel M. German (2005). On the use of visualization to support awareness of human activities in software development: a survey and a framework. In T. L. Naps and W. de Pauw (eds): *Proceedings of the ACM 2005 Symposium on Software Visualization (SoftVis'05), Saint Louis, USA, May 14 to 15, 2005*. New York: ACM Press, pp. 193–202.

Tam, James and Saul Greenberg (2004). A Framework for Asynchronous Change Awareness in Collaboratively-Constructed Documents. In G. de Vreede, L. A. Guerrero and G. M. Raventós (eds): *Proceedings of the 10th International Workshop on Groupware: Design, Implementation and Use (CRIWG 2004), San Carlos, Costa Rica, September 5 to 9, 2004*. LNCS 3198. Berlin / Heidelberg: Springer-Verlag, pp. 67–83.

Tee, Kimberly, Saul Greenberg and Carl Gutwin (2009). Artifact awareness through screen sharing for distributed groups. *International Journal of Human-Computer Studies*, vol. 67, no. 9, pp. 677–702.

Treude, Christoph and Margaret-Anne Storey (2010). Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds. In J. Kramer, J. Bishop, P. T. Devanbu and S. Uchitel (eds): *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering (ICSE'10)—Volume 1, Cape Town, South Africa, May 1 to 8, 2010*. New York: ACM Press, pp. 365–374.

Venolia, Gina, John Tang, Ruy Cervantes, Sara Bly, George Robertson, Bongshin Lee and Kori Inkpen (2010). Embodied social proxy: mediating interpersonal connection in hub-and-satellite teams. In E. D. Mynatt, D. Schoner, G. Fitzpatrick, S. E. Hudson, W. K. Edwards and T. Rodden (eds): *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI'10), Atlanta, USA, April 10 to 15, 2010*. New York: ACM Press, pp. 1049–1058.

Wahyudin, Dindin, Matthias Heindl, Benedikt Eckhard, Alexander Schatten and Stefan Biffl (2008). In-Time Role-Specific Notification as Formal Means to Balance Agile Practices in Global Software Development Settings. In B. Meyer, J. Nawrocki and B. Walter (eds): *Balancing Agility and Formalism in Software Engineering. Second IFIP TC 2 Central and East European Conference on Software Engineering Techniques (CEE-SET 2007), Poznan, Poland*. LNCS 5082. Berlin / Heidelberg: Springer-Verlag, pp. 208–222.

Williams, Byron J. and Jeffrey C. Carver (2010). Characterizing software architecture changes: A systematic review. *Information and Software Technology*, vol. 52, no. 1, pp. 31–51.

Wirth, Carsten and Sahin Albayrak (2011). A classification for Supportive User Interfaces derived from Collaborative User Interfaces. In G. Lehmann, A. Demeure, M. Petit and G. Calvary (eds): *1st International Workshop on Supportive User Interfaces (SUI 2011) at the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, Pisa, Italy, June 13, 2011*. New York: ACM Press, pp. 37–39.

Ye, En, Lev A. Neiman, Hiep Q. Dinh and Chang Liu (2009). SecondWATCH: A workspace awareness tool based on a 3-d virtual world. In S. Fickas, J. M. Atlee and P. Inverard (eds): *Proceedings of the 31st International Conference on Software Engineering (ICSE 2009), Companion Volume, Vancouver, Canada, May 16 to 24, 2009*. Los Alamitos: IEEE Computer Society, pp. 291–294.