Gwendolyn Kolfschoten
Thomas Herrmann
Stephan Lukosch (Eds.)

LNCS 6257

Collaboration and Technology

16th International Conference, CRIWG 2010
Maastricht, The Netherlands, September 2010
Proceedings

Springer

Lecture Notes in Computer Science 6257

Gwendolyn Kolfschoten
Thomas Herrmann
Stephan Lukosch (Eds.)

Collaboration and Technology

16th International Conference, CRIWG 2010
Maastricht, The Netherlands, September 20-23, 2010
Proceedings

Springer

Volume Editors

Gwendolyn Kolfschoten
Stephan Lukosch
Delft University of Technology
Faculty of Technology, Policy and Management
Department of Systems Engineering
Jaffalaan 5, 2628BX, Delft, The Netherlands
E-mail: {g.l.kolfschoten,s.g.lukosch}@tudelft.nl

Thomas Herrmann
Ruhr-Universität Bochum
Informations- und Technikmanagement
Institut für Arbeitswissenschaft (IAW)
Universitätsstr. 150, 44780 Bochum, Germany
E-mail: thomas.herrmann@rub.de

Library of Congress Control Number: 2010934001

CR Subject Classification (1998): D.2, H.3, H.4, H.5, J.1, H.2.8

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-642-15713-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15713-4 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

This volume constitutes the proceedings of the 16^{th} Collaboration Researchers' International Working Group (CRIWG 2010) Conference on Collaboration and Technology. The conference was held in Maastricht, The Netherlands. The previous ten CRIWG conferences were organized in Madeira, Portugal (2000), Darmstadt, Germany (2001), La Serena, Chile (2002), Autrans, France (2003), San Carlos, Costa Rica (2004), Porto de Galinhas, Brazil (2005), Medina del Campo, Spain (2006) Bariloche, Argentina (2007), Omaha NE, USA (2008), and Peso da Régua, Douro, Portugal (2009). CRIWG conferences follow a simple recipe for success: good papers, a relatively small number of attendees, extensive time for lively and constructive discussions, and a high level of cooperation both within and between paper sessions. CRIWG 2010 continued this tradition.

This 16^{th} CRIWG exemplified the continuing interest in the groupware research area. Papers were reviewed by at least three members of an internationally renowned Program Committee, using a double-blind reviewing process. Based on the reviewers' recommendations 27 papers were finally accepted: 18 long papers presenting mature work, and 9 short papers describing work in progress. The accepted papers were grouped into seven themes that represent current areas of interest in groupware research: Knowledge Elicitation, Construction and Structuring, Collaboration and Decision Making, Collaborative Development, Awareness, Support for Groupware Design, Social Networking and Mobile Collaboration. In addition, we featured a paper describing the history of CRIWG research. We were further very pleased to have Jay Nunamaker, Director of the Center for Management of Information at the University of Arizona, USA, a renowned specialist in group support systems as keynote speaker. Finally we had a demo and distinguished lecture on a new generation of thinkLet-based group support systems by Bob Briggs.

CRIWG 2010 would not have been possible without the work and support of a great number of people. First of all we thank all members of the Program Committee for their valuable reviews of the papers. We were grateful for the advice and support provided by the CRIWG Steering Committee, and we thank the Doctoral Consortium Chairs Rafael Gonzalez, from Delft Univeristy of Technology, The Netherlands, Michael Koch, from Bundeswehr University Munich, Germany, and Jan Marco Leimeister, from Kassel University, Germany.

Last, but certainly not least, we thank you for your interest in CRIWG 2010, and hope find the proceedings valuable.

July 2010

Gwendolyn Kolfschoten
Thomas Herrmann
Stephan Lukosch

Conference Organization

Steering Committee

Pedro Antunes	Universidade de Lisboa, Portugal
Marcos Borges	Federal University of Rio de Janeiro, Brazil
Gert-Jan de Vreede	University of Nebraska at Omaha, USA
Jesus Favela	CICESE, Mexico
Jörg M. Haake	FernUniversität in Hagen, Germany
Stephan Lukosch	Delft University of Technology, The Netherlands
José A. Pino	Universidad de Chile, Chile
Ana Carolina Salgado	Universidade Federal de Pernambuco, Brazil

Program Committee Chairs

Gwendolyn Kolfschoten	Delft University of Technology, The Netherlands
Thomas Herrmann	Ruhr-Universität Bochum, Germany

Program Committee

Adriana Vivacqua	Universidade Federal do Rio de Janeiro, Brazil
Aurora Vizcaíno Barceló	Universidad de Castilla-La Mancha, Spain
Benjamim Fonseca	Universidade de Trás-os-Montes e Alto Douro, Portugal
Bertrand David	Ecole Centrale de Lyon, France
Carla Simone	University of Milan-Bicocca Italy
Carlos Duarte	Universidade de Lisboa, Portugal
César Collazos	Universidad del Cauca, Colombia
Christoph Rensing	TU Darmstadt, Germany
Claus-Peter Klas	FernUniversität in Hagen, Germany
Dominik Heutelbeck	FTK - Forschungsinstitut für Telekommunikation, Germany
Dominique Decouchant	LSR-IMAG, Grenoble, France
Doug Druckenmiller	Western Illinois University, USA
Erik Proper	Radboud University Nijmegen, The Netherlands
Etienne Rouwette	Radboud University Nijmegen, The Netherlands

Doctoral Colloquium Chairs

Rafael Gonzalez	Delft Univeristy of Technology, The Netherlands
Michael Koch	Bundeswehr University Munich, Germany
Jan Marco Leimeister	Universität Kassel, Germany

Organizing Committee Chair

Stephan Lukosch	Delft University of Technology, The Netherlands

Organizing Committee

Tanja Buttler	Delft University of Technology, The Netherlands
Rick van Krevelen	Delft University of Technology, The Netherlands

Table of Contents

Collaboration Support

Awareness

Groupware Design

Social Networking

Mobile Collaboration

A Review of CRIWG Research

Pedro Antunes[1] and José A. Pino[2]

[1] University of Lisbon, Department of Informatics of the Faculty of Sciences, Campo Grande, Lisbon, Portugal
paa@di.fc.ul.pt
[2] Universidad de Chile, Department of Computer Science, Avenida Blanco Encalada 2120, Tercer Piso, Santiago, CP 837- 0459, Chile
jpino@dcc.uchile.cl

Abstract. This paper presents a meta-analysis of the CRIWG conference. The study is organized in three main sections: bibliometric analysis, analysis of references and subject analysis. The bibliometric analysis indicates that CRIWG is significantly above the average citation index of similar papers published in LNCS. The analysis of references shows a significant dependence on ACM papers and very low cross-referencing between CRIWG papers. The subject analysis reveals that CRIWG slightly favors positivist evaluations, although almost half of the papers do not present any type of evaluation. We conclude this study with a discussion over strengths, weaknesses, opportunities and threats.

Keywords: Meta-analysis, CRIWG review.

1 Introduction

This paper is intended for the 16th edition of the CRIWG[1] conference. As a round number (2^4), it may trigger an appropriate time to reflect on the past contributions of this conference and the outlook for the future. The authors would then like to present some data gathered from the previous proceedings and elaborate some analysis and discussion. We expect this to be the starting point of a rich and controversial interchange of viewpoints during the conference itself.

CRIWG started in Lisbon, Portugal, in September 1995. It was initially thought to be a meeting to exchange research approaches in the field of Groupware for a few groups. Instead of establishing an informal gathering, a decision was made to make it scientifically valid through the commonly accepted peer-reviewing process, using anonymous submissions and reviews, and circulating the papers in proceedings distributed by international scientific publishers. The conference has kept this tradition up to now.

CRIWG has some unique features when compared with other conferences in the field. It has just one track, with full papers and work-in-progress ones. The presentation of the papers provides ample time for discussion, giving thus authors very valuable

[1] Which now is an acronym for the Collaboration Researchers International Working Group on Groupware.

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 1–15, 2010.

feedback. The conference also encourages social interaction among participants. Finally, the conference has been organized in interesting places around the world, but this has not been obstacle to sustained presence by participants to all sessions.

We will present some data extracted from the proceedings, we will analyze it and try to make suggestions for the future of the conference. The analysis, of course, is the authors' responsibility and it does not represent an official statement from the conference committees.

2 Methodology

Most of this review is based on information provided by Thomson Reuters ISI Conference Proceedings Citation Index (ISI, for short). We analyzed all CRIWG papers published between 2000 and 2008. Our intention was to cover the set of proceedings published by LNCS and IEEE between 1999 and 2009. However, for some unknown reason the joint SPIRE/CRIWG 1999 conference was not found in ISI; and the 2009 papers were not yet available in December 2009, when the data was gathered.

Gathering the list of papers from ISI was not completely straightforward, requiring combined searches using the CRIWG acronym and the Groupware keyword, plus manual inspection to remove spurious references to other journal and conference papers. The consolidated data set used in most of our analysis consists of 246 papers.

The review was separated in three main goals: bibliometric analysis, analysis of references and subject analysis. The bibliometric information, such as the Hirsch index (h) [1], was automatically produced by ISI and gives a summative assessment of CRIWG.

Our main purpose to analyze the CRIWG references to other papers was to understand how CRIWG views and positions itself relatively to other research fields. The references were automatically obtained from ISI, exported to Endnote and exported again to Excel, which was then used to discover the main referenced papers, authors and sources. It should be emphasized the references exported by ISI present some shortcomings. For instance, they do not identify all authors. Furthermore, conferences and journals are formatted with multiple short names. A considerable effort was done to normalize the data and obtain the consolidated information reported in section 4.

The subject analysis follows a method that has been adopted by similar reviews (e.g., [2-4]). The method uses qualitative data analysis techniques to code the data set using multiple tags [5]. Unlike some reviews that use a predefined set of tags, we adopted a grounded approach [6] where the keywords emerge as the analysis progresses [5]. The main categories considered during the grounded coding were: research objective, research topic and type of evaluation. Two coding rounds were performed to ensure consistency.

The coding process was applied to the 246 paper abstracts obtained through ISI. This might be viewed as a controversial decision, since coding could instead be applied to the full paper bodies. The basis for our decision to only code abstracts was: (1) the abstracts should be considered accurate and concise summaries of the authors' research, done by the persons most fit to accomplish them, i.e. the authors themselves; (2) abstracts have an implicit structure that is totally aligned with the coding categories mentioned above; and (3) this structure is typically checked and enforced by peer reviewers.

During the coding process we confirmed the type of information we were reviewing was available in the abstracts, which made the access to the remaining information unnecessary. We nevertheless point out the search for more fine-grained information, including for instance reviewing which specific techniques, tools and algorithms were researched, would necessarily mandate a full body analysis.

We finally note the trend lines shown in the paper are all polynomial functions. The R^2 appearing near the trend lines was automatically calculated using Apple's Numbers.

3 Bibliometric Analysis

The total number of CRIWG papers considered in the bibliometric analysis is 246. The total number of citations to these papers is 336, which gives an average citation per paper of 1.37. The obtained h-index is 6.

The distribution of citations per year is shown in Figure 1. After an expected ramp up of citation activity from 2002 to 2005, the number of citations has stabilized since 2006 at an average of 48 cites/year.

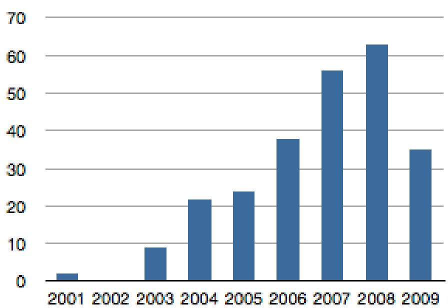


Fig. 1. Distribution of citations per year

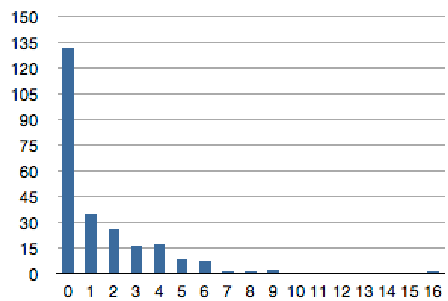


Fig. 2. Distribution of citations per paper (number of citations in horizontal axis)

As shown in Figure 2, 132 papers (53%) have not received any citation. This indicates that CRIWG, although having a selection of good papers, has to improve the selection process.

3.1 Comparison with Other Conferences

The information conveyed above might be difficult to analyze without a frame of reference. In order to create such reference we tried to compare CRIWG with other conferences related with CRIWG and also reported by ISI. The following conferences were selected: COOPIS (Conference on Cooperative Information Systems, now part of OTM - On The Move Confederated Conferences), WET-ICE (Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises), DSV (International Workshop on Design, Specification and Verification of Interactive Systems) and CONTEXT (International and Interdisciplinary Conference on Modeling and Using Context). Furthermore, we also contrasted the CRIWG bibliometrics with a broad set of papers collected from ISI using a search for “LNCS” and “Conference”.

Table 1. Comparison with other conferences

Name	**COOPIS**	**DSV**	**CRIWG**	**CONTEXT**	**WET-ICE**	**LNCS**
Years	01-07	98-07	**01-08**	00-07	96-06	90-09
Nr. papers	379	163	**246**	194	524	8746
Times cited	703	237	**336**	235	274	9103
Av. citation	1.85	1.45	**1.37**	1.21	0.52	1.04

The results shown in Table 1 indicate that CRIWG, in terms of quality measured by citation indexes, is above average when compared with the other selected conferences (1.37 against an average of 1.28) and significantly above the average citation index of a large collection of papers published in LNCS.

3.2 Top 5 Cited Papers

The top 5 cited papers are shown in Table 2.

Table 2. Top 5 cited papers (number of citations on the left)

16	Rosa MGP, Borges MRS, Santoro FM (2003) A conceptual framework for analyzing the use of context in groupware, 9th International Workshop on Groupware, 2003, Autrans, France, LNCS, vol. 2806, p. 300-313
9	Haake JM, Schummer T, Haake A, et al. (2003) Two-level tailoring support for CSCL, 9th International Workshop on Groupware, Autrans, France, LNCS, vol. 2806, p. 74-81
9	Collazos CA, Guerrero LA, Pino JA, et al. (2002) Evaluating collaborative learning processes, 8th International Workshop on Groupware, La Serena, Chile, LNCS, vol. 2440, p. 203-221
8	Neyem A, Ochoa SF, Pino JA (2006) Supporting mobile collaboration with service-oriented mobile units, 12th International Workshop on Groupware, Medina del Campo, Spain, LNCS, vol. 4154, p. 228-245
7	Moran AL, Favela J, Martinez-Enriquez AM, et al. (2002) Before getting there: Potential and actual collaboration, 8th International Workshop on Groupware, La Serena, Chile, LNCS, vol. 2440, p. 147-167

3.3 Top 10 Authors

Table 3 presents the list of authors according to the number of papers published in the proceedings.

Table 3. Top 10 authors (number of papers on the left)

19 Pino JA	12 Ochoa S
17 Antunes P	12 Guerrero L
15 Borges MRS	10 Baloian N
14 Favela J	9 Lukosch S
14 Collazos C	7 Vreede GJ
	7 Fuks H

3.4 Country Distribution

A total of 29 countries have been represented in the conference series. Figure 3 indicates the internationalization ratio has slightly increased between 2000 and 2004 and slightly decreased in 2007 and 2008.

Table 4. Country distribution

44	Brazil	3	Costa Rica
34	Chile	3	Finland
30	Germany	3	Japan
27	Portugal	2	England
24	Spain	2	Greece
22	France	2	Norway
22	USA	2	Taiwan
21	Mexico	1	Australia
13	Netherlands	1	Austria
8	Colombia	1	Belgium
6	Argentina	1	Luxembourg
6	Canada	1	Romania
4	China	1	Scotland
4	Korea	1	Sweden
		1	Switzerland

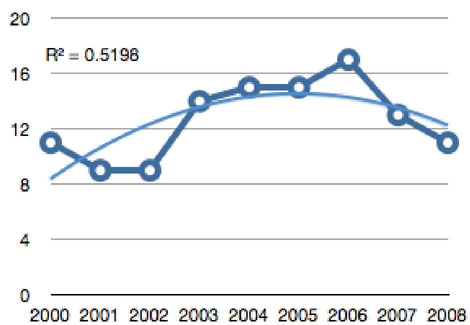


Fig. 3. Internationalization (countries/year)

3.5 Collaborative Research

As mentioned in the introduction, CRIWG aimed to promote the participants' socialization. This should allow researchers to meet potential partners for future projects. These researchers will probably return to the CRIWG conference to present the new joint results. A reasonable hypothesis then is to assume the proportion of papers presented by researchers from two or more institutions would increase in time.

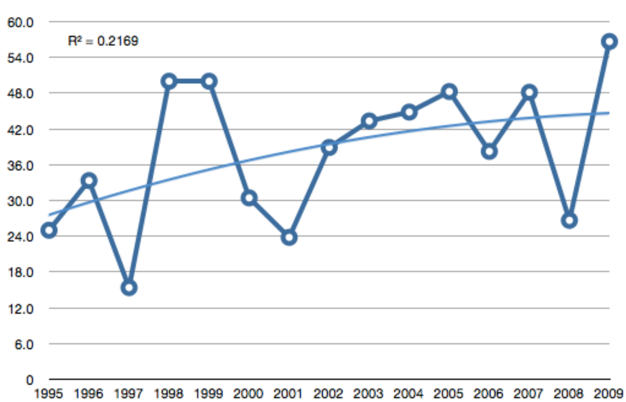


Fig. 4. Percentage of collaborations by year

Figure 4 shows the evolution of the number of collaborations for the whole CRIWG period (1995-2009), represented as a percentage of the number of papers published in each year. This data was manually gathered from the proceedings. It shows the proportion of papers presented by two or more research groups slightly growing in time (groups from different departments of the same university were considered as just one group). Of course, the contribution from the conference may not be the only reason for this increase. Other factors may influence this result, like appearance of grants supporting research from more than one country, etc. However, in our own personal experience, CRIWG helped us finding research partners.

3.6 Special Issues in ISI Journals

Table 5 summarizes the special issues in ISI Journals published with extended versions of papers presented at CRIWG conferences. The total number of papers is 46 (18.7% of all papers published by the Proceedings in the 2000-2008 period). The number of citations is 99. The average citation index is 2.15, which is significantly higher than the one obtained by the proceedings. There is also in press one special issue of the Group Decision and Negotiation journal containing extended versions of papers presented at the 2008 conference.

Table 5. Special issues in ISI journals

Journal name	No. of issues	No. of papers published
Int. J. of Cooperative Information Systems	2	10
Int. J. of Human-Computer Studies	1	5
Journal of Universal Computer Science	2	10
Computing and Informatics	1	5
Group Decision and Negotiation (*)	2	12
Multimedia Tools and Applications	1	4

(*) The 2008 issue is in press.

Besides the special issues in ISI-indexed journals, there have been special issues in other journals, as reported in Table 6.

Table 6. Special issues in other journals

Journal name	No. of issues	No. of papers published
Int. J. of Computer App. and Technology	1	9
International Journal of e-Collaboration	1	4
e-Service Journal	1	5
Journal of CLEI	1	1

4 Analysis of References

The analysis of references gives a good indication of how CRIWG perceives its research community. The references were automatically obtained from ISI using the set of 246 papers published between 2000 and 2008. References to technical documentation and web sites were manually removed from the data set.

Table 7. Analysis of references

Total number of cited papers: 4524 Average number of references per paper: 19.15 (stdev: 8.6)
References to ACM papers: 653 (14%) ACM Transactions: 98 Communications of ACM: 134 Proceedings of ACM: 338 Proceedings of ACM CSCW: 187 Proceedings of ACM CHI: 61
References to LNCS papers: 229 (5%) References to IEEE papers: 196 (4.3%) References to Thesis: 112 (2.5%) References to HICSS papers: 90 (2%) References to CRIWG papers: 84 (1.9%) References to ECSCW papers: 35 (0.8%) References to LNAI papers: 14 (0.3%)

The results summarized in Table 7 indicate a significant dependence on ACM papers. The data also indicate very low cross-referencing between CRIWG papers. In the one hand, this shows there is very low inbreeding in the CRIWG community, but in the other hand it also points out a lack of community building. The number of references to thesis might be interpreted as indicating a focus on exploratory rather than summative research.

4.1 Most Cited References

The references most cited by CRIWG are shown in Table 8.

Table 8. Most cited references (number of citations on the left)

15	ELLIS CA, 1991, COMMUN ACM, V34, P38
12	BRIGGS RO, 2003, J MANAGE INFORM SYST, V19, P31
12	GAMMA E, 1995, DESIGN PATTERNS ELEM
10	FJERMESTAD J, 1999, J MANAGEMENT INFORMA, V15, P7
10	NUNAMAKER JF, 1991, COMMUN ACM, V34, P40
8	ROSEMAN M, 1996, ACM T COMPUTER HUMAN, V3, P66
8	SCHUCKMANN C, 1996, P ACM 1996 C COMP SU, P30
7	DOURISH P, 1992, P ACM C COMP SUPP CO, P107
7	GUERRERO LA, 2001, INFORM SOFTWARE TECH, V43, P457
7	MALONE TW, 1994, ACM COMPUT SURV, V26, P87
6	DEVREEDE GJ, 2006, INT J COMPUTER APPL, V25, P140
6	GRUDIN J, 1994, COMMUN ACM, V37, P92
6	CHABERT A, 1998, COMMUN ACM, V41, P69

4.2 Most Referenced First Authors

The researchers most cited by CRIWG are shown in Table 9. Only the first authors are taken into account, since ISI does not produce the full reference records.

Table 9. Most referenced authors (number of citations on the left)

50	C. Gutwin	23	L. Guerrero
46	R. Briggs	22	M. Borges
42	S. Greenberg	22	G. Stahl
35	C. Ellis	21	J. Nunamaker
34	P. Dourish	21	J. Haake
34	G. Kolfschoten	21	T. Malone
26	P. Dillenbourg	20	M. Roseman
23	D. Pinelle	20	B. Myers

4.3 Main Referenced Research Areas

The seven categories shown in Table 10 emerged after systematic data analysis of the data set using manual search. These categories may again be categorized by order of importance, where CSCW (Computer Supported Cooperative Work) emerges as the most important research area, followed by Decision Support Systems (DSS) and Computer Supported Collaborative Learning (CSCL). It should however be noted that these categories cover a small percentage of the data set (more precisely, 25.6%). The remaining 3/4 of data could not be associated to a category.

Table 10. Main research areas referenced by CRIWG

1 – About 10%	413 (9.1%)	CSCW
2 – About 5%	238 (5.3%) 230 (5%)	DSS CSCL
3 – About 1-2%	75 (1.7%) 58 (1.2%) 105 (2.3%) 45 (1%)	Software engineering Human Computer Interaction Business Process Management Artificial Intelligence

5 Subject Analysis

5.1 Research Objectives

As shown in Figure 5, nine different research objectives were codified during the data analysis. The most prevailing research objective is prototype development, followed by design and theory/model building. Interestingly, although a major focus is on prototyping and designing collaborative systems and tools, CRIWG has not focused on building guidelines for developers.

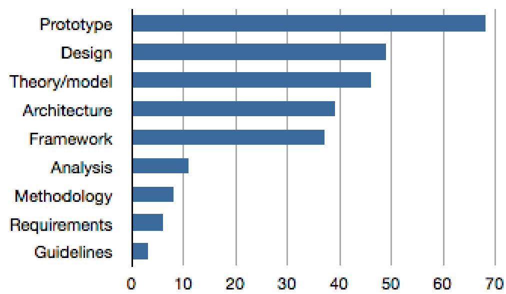


Fig. 5. Distribution of papers by research objective

5.2 Research Topics

The data set was manually coded with the purpose to uncover the main research topics. The coding process was done in multiple rounds for consistency. Thirty-two research topics emerged after consolidation. Figures 5 and 6 show the most and least significant topics, respectively. Learning stands out as the most prevailing research topic, considered by 51 papers (20%).

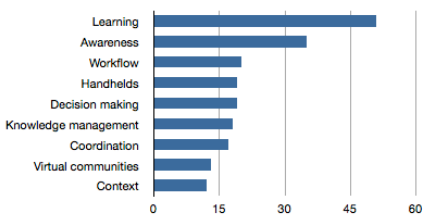


Fig. 6. Research topics (most significant)

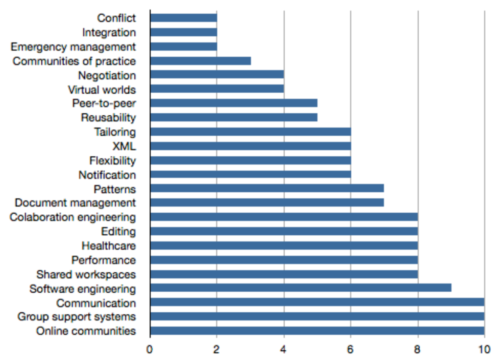


Fig. 7. Research topics (least significant)

We emphasize the coding process was performed at various conceptual levels before the main research topics were completely settled. Learning, for instance, is a code that actually encompasses the following sub-codes: classroom activities/ composition, online materials/courses, teaching, reflection, infrastructure/platform, scenarios, processes, discussion forums, modeling knowledge, interdependencies/ mediation, performance, attitudes, dialectic reasoning, participation and self-organization. Awareness, in turn, includes the following sub-codes: semantic awareness, situation awareness, document awareness, awareness model, group awareness, presence awareness, workspace awareness, and change awareness.

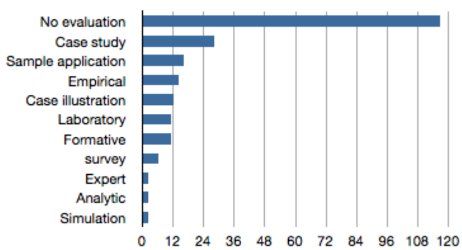
After the coding process, the research topics were aggregated in main research areas. Five main areas emerged this way, covering the foundations of collaboration support, application areas, group decision-making, system development issues, and communities. Table 11 shows that collaboration support is the main research concern expressed by CRIWG (35% of papers). It is also interesting to note that infrastructural issues related with collaboration support have not received sizeable attention from CRIWG researchers (10% of papers).

Table 11. Main research areas

128 (35%)	**Collaboration support** (awareness, coordination, context, tailoring, flexibility, notification, performance, shared workspaces, communication, editing, document management)
98 (27%)	**Applications areas** (learning, workflow, handhelds, healthcare)
70 (19%)	**Decision making** (conflict, decision making, emergency management, group support systems, negotiation, knowledge management, collaboration engineering)
36 (10%)	**Systems development** (integration, peer-to-peer, reusability, XML, patterns, software engineering)
31 (8.5%)	**Communities** (virtual communities, communities of practice, virtual worlds, online communities)

5.3 Evaluation Methods

Almost half of the papers (118 papers, 47.5%) do not present any type of evaluation. Of the adopted evaluation methods, case studies are the most prevalent one. CRIWG balances positivist and interpretivist evaluations, with a slight advance given to positivist evaluations. Interpretivism addresses questions of meaning while positivism addresses questions of cause and effect [7].



47.5% - No evaluation

25.2% - Positivist (laboratory, survey, empirical, sample application, formative, simulation, analytic)

16.3% - Interpretivist (case study, case illustration)

Fig. 8. Distribution of papers per evaluation method

Trying to study the 118 papers without any evaluation, we found they are distributed according to Table 12.

The large number of papers with no evaluation perhaps is related to one category of papers the conference has: the Work-in-Progress class. These papers, short in length, are supposed to present initial ideas and thus may tend to be speculative, reporting initial stages of research projects. We could not confirm this hypothesis with the available data, since the proceedings do not distinguish the CRIWG type of paper.

Table 12. Subject addressed by papers without any evaluation (number of papers on the right)

Propose a framework/architecture	39
Describe a prototype	29
Concern design issues	18
Address implementation issues (e.g., flexibility, synchronization, heterogeneity, interoperability)	16
Propose a model	12
Concern workflow	3
Concern decision making	3
Concern knowledge management	2
Concern a methodology	2
Concern software engineering	3

5.4 Trends in Research Objectives

The trend lines shown in Figures 9-13 account the different research objectives according to year of publication. The framework and architecture topics exhibit trend lines showing that, after an increase of importance, the pursuit of these research lines is in decline (note also that R^2 is high). The other categories do not exhibit any definite trend.

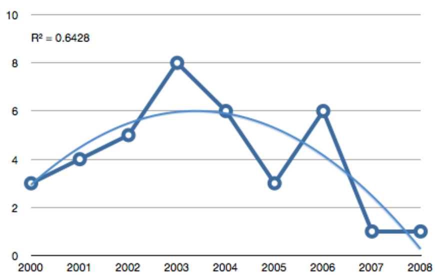


Fig. 9. Framework

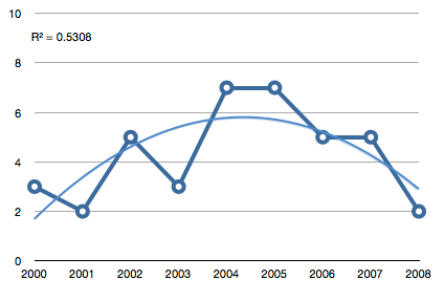


Fig. 10. Architecture

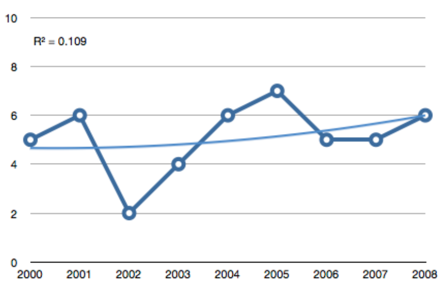


Fig. 11. Theory/models

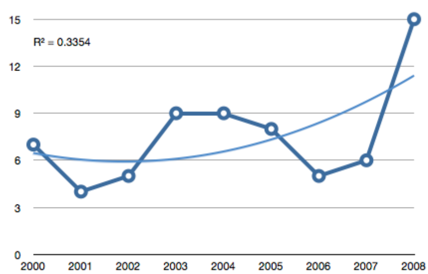


Fig. 12. Prototype

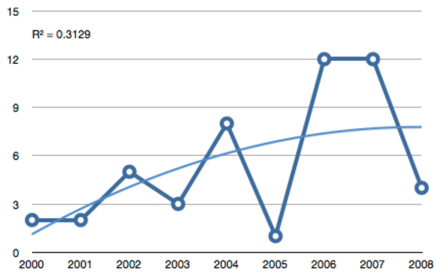


Fig. 13. Design

5.5 Trends in Research Topics

The trends shown in Figures 14-18 indicate a clearly increasing interest over the communities theme (comprising issues such as virtual communities, communities of practice, virtual worlds and online communities). The CRIWG interest over application development (including various issues such as integration, peer-to-peer, reusability, XML and patterns) seems to persist as rather low when compared with the other categories.

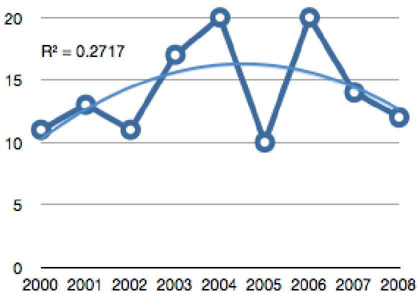


Fig. 14. Support

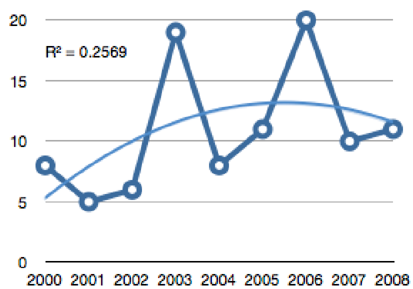


Fig. 15. Application areas

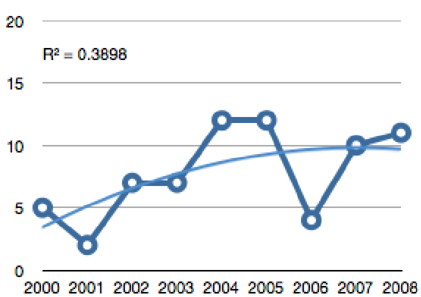


Fig. 16. Decision making

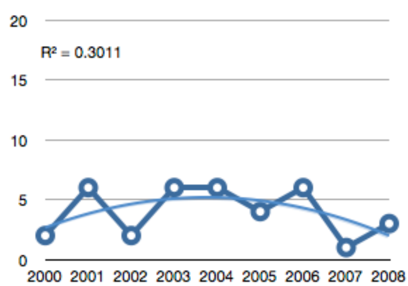


Fig. 17. Application development

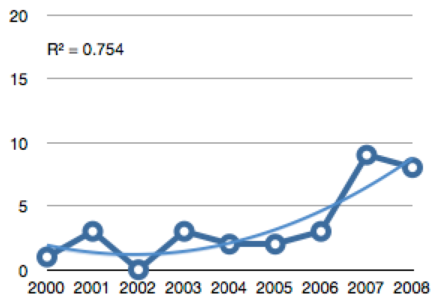


Fig. 18. Communities

5.6 Trend in the Use of the "Groupware" Term

As mentioned in the Introduction, the initial goal of the Conference was to ease collaboration of researchers working in the Groupware field. We have now the feeling that groupware is being phased out in favor of other terms, including "collaborative technology", "group support", or more specific keywords. Figure 19 shows that the percentage of occurrences of the keyword in the proceedings of each year is declining, thus supporting our hypothesis.

The percentages were calculated using Word Counter 2.10 to scan PDF files, count words and convert PDF to text; and then using TextEdit to count how many times the groupware keyword appears in the text. To avoid counting occurrences in the papers' references sections, we manually removed them from the PDF files.

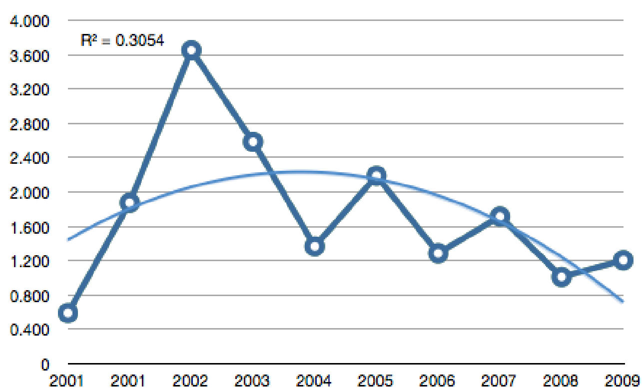


Fig. 19. Number of "groupware" words in the conference proceedings per mil

6 SWAT Analysis

A traditional Strengths/Weaknesses/Opportunities/Threats analysis is provided in Table 1. The entries are by no means definitive truths, but rather the authors' interpretations of the data elements presented above; they are subject to contrast with other viewpoints.

7 Conclusions

Several conclusions are already embedded in the SWAT analysis. However, the authors would like to emphasize a few of them.

The CRIWG conference is a well-established conference with a positive SWAT chart. Of course, weaknesses should be faced and maybe they can be corrected. Although the average number of cites received is higher than the average LNCS conference, it certainly would be desirable to raise that figure.

Table 13. SWAT analysis of CRIWG

Strengths	**Weaknesses**
- Average citation index within range of other reputed conferences and above LNCS average. - More than 18% papers were extended and republished in ISI journals. - Variety of evaluation methods. - Significant focus on prototyping development and design issues. - Balance between positivistic and interpretivist evaluation methods. - High number of collaborations (currently at 57%). - Long-term agreement to publish proceedings in LNCS series. - ISI visibility.	- Lack of cohesive research topics and trends. - Small focus in communities of practice (although it is increasing). - Small number of references to CRIWG papers. - Dependence on ACM conferences. - About half of the papers do not have any impact. - Small overall h-index. - About half of the papers do not address evaluation.
Opportunities	**Threats**
- The largest number of papers comes from Brazil. This country also has a strong local conference on CSCW. A possible synergy? - Emergent application areas outside the core of the conference could be used to attract interesting papers. - Some journals already know the conference and accept special issues; it could be possible to associate the conference to a specific prestigious journal.	- Dependence on a small set of authors. - Great challenges in the field already worked out (e.g., awareness). - A relatively small number of accepted papers may not make economically viable to organize the conference in the future.

Perhaps the opportunities can be taken. The authors particularly suggest the CRIWG Steering Committee should discuss with the Brazilian members the advantages and disadvantages of organizing the conference more often in Brazil. For example, the CRIWG conference could eventually merge with the Brazilian CSCW local event; the conference could be held in Brazil and abroad in alternating years. This is especially relevant since the 2011 conference will be held in Brazil.

Another suggestion to the CRIWG conference would be to explore possibilities of long-term association with prestigious journals in the area. Finally, the incorporation of new promising research fields in areas of interest of the conference could also be considered.

Acknowledgements. This paper was supported by the Portuguese Foundation for Science and Technology (PTDC/EIA/102875/2006) and Fondecyt (Proj. 1080352).

References

1. Hirsch, J.: An Index to Quantify an Individual's Scientific Research Output. Proceedings of the National Academy of Sciences of the United States of America 102, 16569–16572 (2005)
2. Pervan, G.P.: A Review of Research in Group Support Systems: Leaders, Approaches and Directions. Decision Support Systems 23, 149–159 (1998)
3. Pervan, G.P., Atkinson, D.J.: GDSS Research: An Overview and Historical Analysis. Group Decision and Negotiation 4, 475–483 (1995)
4. Serenko, A., Bontis, N.: Meta-Review of Knowledge Management and Intellectual Capital Literature: Citation Impact and Research Productivity Rankings. Knowledge and Process Management 11, 185–198 (2004)
5. Miles, M., Huberman, A.: Qualitative Data Analysis. SAGE, Thousand Oaks (1994)
6. Glaser, B., Strauss, A.: The Discovery of Grounded Theory Strategies for Qualitative Research. Aldine, Chicago (1967)
7. Davison, R.: An Action Research Perspective of Group Support Systems: How to Improve Meetings in Hong Kong. PhD Dissertation. Department of Information Systems, City University of Hong Kong (1998)

Supporting Collaborative Knowledge Creation in Mobile Working Scenarios

Gustavo Zurita[1] and Nelson Baloian[2]

[1] Information systems and Management Department, University de Chile, Diagonal Paraguay 257, RM, Santiago de Chile, Chile
gzurita@fen.uchile.cl
[2] Department of Computer Science, Universidad de Chile, Blanco Encalada 2010, RM, Santiago, Chile
nbaloian@dcc.uchile.cl

Abstract. Knowledge Creation (KC) is a critical activity inside organizations. It has been said to be a differentiating factor and an important source of competitiveness. Tacit knowledge is an important asset of any organization. Because it is not formalized is difficult to share. KC supporting systems help people inside an organization to share this tacit knowledge. This paper presents the design, and implementation of a KC system called MCKC, for Mobile Collaborative Knowledge Creation, supporting face-to-face knowledge creation and sharing in mobile scenarios, allowing people to create new knowledge and share their tacit knowledge with their co-workers, using visual metaphors, gestures and sketches to implement the human-computer interface.

Keywords: Knowledge Creation, Mobile, Face-to-face Collaboration.

1 Introduction

Explicit knowledge is systematized and standardized knowledge, which can be expressed by a formal language, using records, reports, or files. People acquire tacit knowledge through verbal face-to-face conversations, exchange of personal experiences or by processing previous knowledge they have, but they are not able to communicate it in a systematical way. This might be due to reasons, for example, because it is not structured enough, [1]; it has to be transferred in a certain social context in order to be interpreted by the receiver, [2]; or it is difficult to represent. Nonaka and Takeuchi [3] proposed the Socialization, Externalization, Combination, Internalization (SECI) model in order to convert tacit knowledge into explicit knowledge trough a process they called Knowledge Creation (KC). It might be difficult to take advantage of tacit knowledge in order to achieve KC, since it is difficult retrieve and share it unless it is converted to explicit knowledge. This process has been conceptualized as a never-ending spiral [3]. Sometimes explicit knowledge fails to be shared because of the presence of free-riding practices, evaluation apprehension, production blocking, [1] or lack of trust among people who create knowledge in the organization.

Technologies used in KC include from successful databases storing best practices to artificial intelligence systems supporting human decision making processes. These solutions

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 16–24, 2010.

have been so far designed for static workplaces. This implies that mobile workers cannot be supported by the knowledge pool available in their organizations, nor can they contribute to enlarge the knowledge pool while performing their duties outside the office [1, 4].

This work aims to explore the role of mobile technology as support for this kind of meetings, providing simple but helpful visual mechanisms to support knowledge creation, especially targeted to manage tacit knowledge [1]. The prototype's design takes into consideration previous works including: 1) The SECI knowledge transformation model to developed a platform which supports the knowledge creation [5]; 2) Results of researchers about the loss of productivity in the generation of ideas (free-riding, production blocking, and evaluation apprehension), and the relevance of face-to-face social interactions, [6].

2 Knowledge Creation in a Face-to-Face Scenarios

2.1 Visual Mechanisms, Sketching and Brain-Sketching

According to [7], visualization enables knowledge "mapping" facilitating its creation and sharing. In KC, visualization is used to support the creation of tacit knowledge individually or collaboratively by means of sketches, concept maps, graphical representations, etc. It facilitates the clarification of the tacit knowledge for an individual herself or when trying to share her knowledge with others.

Previous works on the field [8] highlight the following advantages of sketching in idea face-to-face generation meetings: a) in relation to thinking, sketching stimulates a re-interpretative cycle in the individual participant's idea generation process, b) in relation to the talking, sketching stimulates the participants to re-interpret each other's ideas; and c) in relation to the storing, sketching stimulates the use of earlier ideas in the idea generation process by enhancing their accessibility. The visualization technique called 'brainsketching' [8] was used to describe idea generation techniques that use sketching. Brainsketching is a graphic variation of the more widely known brainwriting technique. Commonly during brainsketching, participants first sketch ideas individually and share them by switch papers. Then they use the ideas already present on the worksheet as a source of inspiration.

2.2 Brainwriting-Based Knowledge Creation in Mobile Groups

According to [9], most people believe that knowledge creation is best performed in a face-to-face groups because interaction with other people stimulates creativity. However, controlled research has consistently shown that people produce fewer and lower quality ideas working in a group as compared with when working alone or in nominal groups. Nowadays, much has become clear about the causes hindering productivity in face-to-face brainstorming groups: a) **free riding** is the tendency to let to others group members do the work; b) **evaluation apprehension** is when groups start out with a low rate of production which are maintained in the rest of the session; c) **production blocking** refers to the fact that group members have to wait to express their ideas because they have to wait for their turns to talk.

In accordance to [9], electronic brainwriting and brainsketching can be used to reduce or even eliminate production blocking, evaluation apprehension, and also free riding. Moreover, the findings of [9, 10] indicate that sharing written ideas in groups may enhance creativity. In [9] authors propose an initial phase of individual writing of ideas. These ideas are then shared with the group in a round-robin turn-taking session where they are summarized and selected using a blackboard. Then the group discusses the ideas for clarification and evaluation. Finally, they rank the ideas in an individual and/or collaborative process in order to take a final decision.

Nonaka et. al. [3] developed a model of KC that involves a continuous interaction between tacit and explicit knowledge in order to produce new knowledge within groups or communities. It has therefore been argued that tacit or implicit knowledge can be converted to explicit knowledge by “reflection in action”, by the use of metaphors and analogies or by using mentoring and storytelling. Although it is possible to “externalize” some parts of implicit knowledge, some aspects of tacit knowledge, particularly those related to creativity, intuition, emotions, and skills, are unlikely to be ever made completely explicit.

Becerra-Fernandez and Cousins [11], say knowledge is increasingly being created and applied on the move workers working face-to-face. The potential of KC is usually limited to static workplaces because most KC support systems are designed for being used in desktop PCs connected to a central server. This might exclude valuable mobile workers in charge of knowledge intensive activities. An organization’s capabilities to support KC may be extended through the introduction of mobile technology usage.

The authors in [11] argue that mobile KC supporting situated work has not attracted as much attention as it should, considering its potential. There are mobile systems developed with the aim of extending the information access everywhere and anytime by using PDAs .

2.3 Design Principles Applied to a Face-to-Face Mobile KC Support

In [12] authors conclude that rather than focusing on systems to codify knowledge, we should instead concentrate on systems facilitating collaboration between knowledge holders, creators and those needing the knowledge. Indeed, recent research has already begun to recognize the need to incorporate support for face-to-face KC and sharing when designing KM systems in order to facilitate the transfer of complex, context-specific knowledge [6].

In accordance to [13], there are three key functional requirements for KC systems: **1) facilitate information contextualization, 2) facilitate social interactions and networking, and 3) present a ease to use human-computer interface - providing visual representation and organization of information.**

Considering the arguments and ideas mentioned before, we develop a prototype of a system supporting KC in mobile scenarios with people collaboratively working face-to-face, which we will refer to as MCKC, for Mobile Collaborative Knowledge Creation. Its design principles are derived from the results of various previous empirical and experimental KC research works. They also consider the characteristics of mobile devices and human-computer interface design principles.

MCKC runs on Tablet-PCs and PDAs wirelessly interconnected by an ad-hoc network, with the possibility of synchronizing the data of the mobile devices with a central repository when the required networking infrastructure is available. In this way, people have access to the existing explicit knowledge anytime and anywhere. The touch-screen mechanism of their displays is used as the main human-computer interaction mean to input information to the system. As we mentioned before, sketching is used as a visual tool for information management and for supporting brainwriting and brainsketching processes. It is also used to implement command inputs by gestures, in order to implement a simple and easy-to-use application interface. By using sketching, the system implements the electronic paper and pencil paradigm, which facilitate the communication of tacit knowledge.

The system uses various visualization mechanisms for information management. MCKC implements the free-hand-based input paradigm which means user are able to draw sketches, edit graphic information and free-hand text writing, as well as using visual metaphors for information management like conceptual maps, the usage of gestures to trigger options, object dragging, visual presentation of the concept maps' nodes, etc.

The system's design is oriented to support collaborative KC based on the SECI model. The system has three modes: **1) brainwriting/brainsketching,** or knowledge externalization support mode, **2) selection of relevant information,** or socialization and combination support mode, **3) visual presentation and knowledge semantics of the created knowledge** mode, which supports the socialization process. The three modes of the application are aimed to provide the environment supporting tacit to tacit knowledge sharing and creation, without the need to convert tacit knowledge to explicit knowledge before sharing it.

MCKC considers social interaction as a key factor for collaborative KC, although it can be also used to support individual KC, in this way we expect to reduce the productivity problem described before. Writing ideas instead of speaking them inside a group minimizes the problem of **production blocking** since individuals do not have to wait their turn to generate ideas. It may also reduce **evaluation apprehension** since the written format eliminates the need for public speaking and is typically more anonymous than oral brainstorming. Also the **free-riding** problem might be reduced because it will be easier to identify not contributing people. Employees should not be forced to use a knowledge creation or sharing system. The system must be created to support their work and their social behavior, [1]. The system has to adapt to employees' social interactions in order to let them be a member of a successful KC community and gain recognition. The relevant information selection mode of MCKC allows a person to explain the semantic associated to the knowledge that has been made explicit in order to build a common language.

3 MCKC Description and Scenario of Application

As already establish, sketches help to externalize the tacit knowledge and hence express ideas and concepts from people's minds. They can also help people to order and clarify their own ideas before communicating them to others [8]. This is why MCKC allows specifying the explicit and tacit knowledge by means of its interface. This interface allows the manipulation of information in a simple way using the device's stick to activate options using gestures. It uses visual mechanisms for presenting and manipulating information. It allows the edition of sketches and freehand writing and facilitates the interaction among members of a group working face-to-face collaboratively in KC. MCKC can be used anytime, anywhere which means it can be brought to any physical place of the organization, and can be even used while people are on the move. In this section we describe briefly the functionalities of MCKC, its three working modes and the characteristics that make it a suitable tool for supporting KC. Each mode is oriented to support one stage of the SECI model. The system does not impose a certain order of sequence for using each mode which allows a spiral kind of development. It is always possible to go back to a previous mode in order to make corrections or even start from scratches again. The following description of the tool is based on a scenario where three persons of the marketing department of an organization are trying to figure out how does the new poster advertising an all-terrain car should look like.

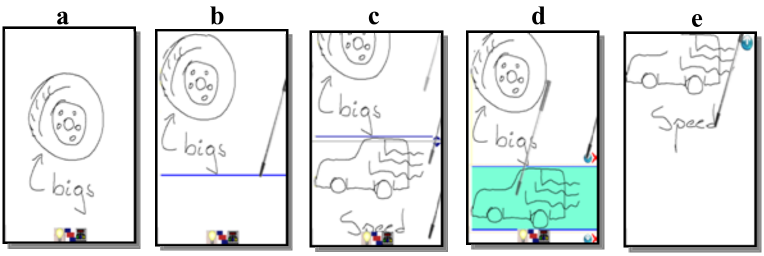


Fig. 1. Specification of an idea through sketching and freehand writing. a) An idea of a wheel and the text "bags" is generated. b) A gesture of writing a horizontal line through the whole screen will mark the separation between two ideas. c) The second idea is produced (a car sketch the "speed" text). d) An idea is being selected and then the "enter" icon is clicked. e) The idea is being edited, which will be seen updated when the user leaves the editing mode.

3.1 SECI Model - Externalization: Mode Brain-Writing/Sketching

The brainwriting/brainsketching mode supports the knowledge externalization allowing users to explain their tacit or explicit knowledge by jeans of freehand writing or sketching. This mode works in a non-collaborative way by default, allowing users to freely prepare their ideas before sharing them, reducing the free-riding, production blocking, and evaluation apprehension problems. Users generate their ideas in parallel despite they are in a face-to-face situation (see Fig. 1).

If a previous idea has to be edited, the user selects the area where it is by a single click and "enters" the edition mode clicking the "arrow down" icon (see Fig. 1d.).

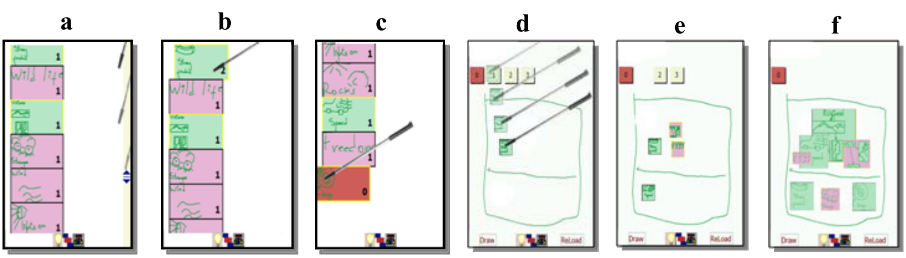


Fig. 2. List of ideas to be selected. a). Scrolling the ideas by moving the pen on the right of the screen; b). An idea is ranked from the level 1 to the level 2; c. An idea is ranked to the level 0 (irrelevant level). d). An idea select, drag and drop on the visual representation; e). All ideas of level 1 were use on the KC representation, and this set disappears; f). Final result.

In order to work collaboratively, participants have to activate the collaborative work option of the application. After this, they can share their ideas and start editing them collaboratively in the same way they did individually. This supports the knowledge socialization process of the SECI model. Since ideas are shown one below the other a scrolling function is necessary to go through them, which is done by a gesture of sliding the stick up and down parallel to the right vertical border of the screen (Fig. 1c.).

3.2 SECI Model-Socialization/Combination: Relevant Information Selection Mode

After each user has externalized her ideas, it is necessary to refine them involving all group members. In order to select the ideas it is necessary to define which are relevant and which can be discarded. In order to support this process, MCKC generates a list of all created ideas, which will be visually shown as rectangular boxes of similar proportions. In this stage, the list of ideas is visible to all participants, as shown in Figure 1. In order to rank them, participants have to vote for them positively or negatively. They can issue a positive vote for a certain idea by making a tick gesture on the left area of the rectangle representing it (see Fig. 1.e.). A negative vote is issued by making a tick on the right area of the rectangle (see Fig. 1.f.). Numbers from 1 to 5 represent the ranking of each idea according to the votes received, being 5 the most relevant. A 0 means the idea is not relevant at all. Because there might be many ideas, a scroll mechanism is also available in this mode (Fig. 1.d.).

At the beginning, before receiving any vote, the ranking number for an idea will be 1. The ranking number for an idea appears in the bottom-right corner of the rectangle. As ideas get ranked, they will be rearranged and grouped according to the ranking level. In this way, relevant ideas are easily differentiated from the irrelevant ones, supporting their selection. An idea can be collaboratively edited while working in this mode by clicking in the middle area of the rectangle. Collaborative editing allows the socialization of the tacit and explicit knowledge, allowing participants to combine their knowledge and perspectives about the ideas.

3.3 SECI Model - Internalization: Visual Presentation and Semantic of the KC Mode

This mode allows users to concretize the KC process using a final visual representation of the ideas. This process is done collaboratively with the agreement of all participants. For the example case, participants have to agree which ideas will be expressed visually and which in a written form on the poster.

At the beginning of this mode, an empty page will appear with a list of small squares at the top representing the ideas generated ordered according to their ranking. In this stage, participants have to make a visual arrangement of the ideas. This is an important stage during the KC because it is expected that the tacit and explicit knowledge to be expressed here with sketches and other visual representations. It is expected that participants will first draw a sketch where ideas will be placed in a particular order according to the meaning of the sketch. In the example, users will determine the position where each idea will be placed in the poster. Ideas can be dragged from the list and dropped in the desired place (Fig. 2.). The placement of the ideas inside the sketch should represent a meaning collaboratively defined by all participants. The square representing an idea can be reshaped as desired (see Fig. 2.d.). After placing the ideas on the schema, participants may finalize their proposal by skating which one would be fundamental to the project or they can go back to a previous mode in order to edit, the existing ideas or include new ones. Not used ideas might be deleted.

4 Usability and Utility Evaluation

For the usability and utility evaluation processes we used two external consulters, two car sellers of a certain brand and two drivers (users). We proposed them the task of creating a commercial spot for a car. They had six working sessions during two weeks, during which they used MCKC in order to collaboratively work in a face-to-face modality to generate the ideas for the commercial spot. This work was monitored with inspection techniques to bring more context to the inspection task. Then we conducted a workshop where all participants had to analyze the MCKC tool in the context of the predefined scenario. From the workshop we finally obtained a set of comments and observations regarding the usability and utility of MCKC tool. The workshop revealed that the free-riding, production blocking, and evaluation apprehension problems were partially mitigated by the use of technological support, which contrasted with the results of previous experiences where technology was not used. The explicit knowledge could be easily specified and communicated with the help of MCKC. It was also noted that that sketches helped to exteriorize and share tacit knowledge. The visualization of the artifacts on the system interface associated to data, information and functionalities triggered by gestures was well accepted and easy to use. However, more experimented user missed the menus, choice boxes and fast access keys. The “visual presentation and semantic of the created knowledge” mode, was by far perceived as the most helpful one because it’s flexible and enriched way to represent knowledge as a final result of a goal. Second to this, the “brain-writing/sketching” mode, was also perceived as a very helpful way to easily specify ideas through sketches and the possibility to organize them as concept maps. In the whole MCKC was perceived as a relevant tool to support collaborative work because it enables people to contribute, explain, exteriorize and share their ideas. Regarding the usability of the MCKC, in general the participants suggested some additional improvements. The participants regarded a major challenge to keep the awareness information and collaboration constantly up-to-date. The learning curve of MCKC was satisfactory completed during the second working session. Some difficulties were perceived on users who declared not having too much experience with technology.

5 Conclusions

People posses a big amount of tacit “hidden” knowledge which has to be converted into “new knowledge", in order to promote its delivery, sharing and innovation. In this way, this knowledge can be effectively used in the organizations where people work. MCKC is a tool that helps externalize this knowledge. Our work is based on the empirical and experimental findings of KC related Works, which have been incorporated into the system presented in this paper. The visualization technology of knowledge and the use of mobile devices as support for KM is a new field, which has already generated applications for different scenarios such as engineering, education and economy.

Our application supports the visualization of information in a free and extensible way. It also promotes the collaboration in mobile scenarios by making use of ad-hoc wireless networks, which helps to transform tacit into explicit knowledge, promoting the elicitation, transmission and sharing of information based on sketches. The KM success model developed by emphasizes the need for KM systems to include both types of knowledge (tacit and explicit) and linkages or pointers to people with knowledge expertise. A better understanding of the various characteristics of the tacit knowledge dimension, as detailed in the present study, will assist researchers and practitioners in the development of more sophisticated knowledge management systems that can adequately address knowledge users' needs for both codified knowledge and interaction with human sources of knowledge.

Acknowledgements

The work of this paper was funded by Fondecyt project 1085010 and Latin American Collaborative Research Federation (LACCIR), project number R1209LAC002.

References

1. Matuszewski, M., Balandin, S.: Peer-to-Peer Knowledge Sharing in the Mobile Environment. In: The Fifth International Conference on Creating, Connecting and Collaborating through Computing, C5 2007 (2007)
2. Wiberg, M.: Knowledge Management in Mobile CSCW: Evaluation Results of a Mobile Physical/Virtual Meeting Support System. In: Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS'34). IEEE, Maui (2001)
3. Nonaka, I., Takeuchi, H.: The Knowledge Creating Company. Oxford University Press, New York (1995)
4. Becerra-Fernandez, I., Cousins, K.: Nomadic context-aware knowledge management systems: applications, challenges and research problems. International Journal of Mobile Learning and Organisation 1(2), 103–121 (2007)
5. Nonaka, I., von Krogh, G., Voelpel, S.: Organizational Knowledge Creation Theory: Evolutionary Paths and Future. Advances, in Organization Studies (01708406), pp. 1179–1208 (2006)
6. Brown, S.A., Dennis, A.R., Gant, D.B.: Understanding the Factors Influencing the Value of Person-to-Person Knowledge Sharing. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, HICSS 2006 (2006)
7. Yongjin, Z., et al.: Study on the Knowledge Visualization and Creation Supported Kmap Platform. In: Proceedings of the First International Workshop on Knowledge Discovery and Data Mining. IEEE Computer Society, Los Alamitos (2008)
8. van der Lugt, R.: How sketching can affect the idea generation process in design group meetings. Design Studies 26, 101–122 (2005)
9. Paulus, P.B., Huei-Chuan, Y.: Idea Generation in Groups: A Basis for Creativity in Organizations. Organizational Behavior & Human Decision Processes 82, 76–87 (2000)

10. Herring, S.R., Jones, B.R., Bailey, B.P.: Idea Generation Techniques among Creative Professionals. In: Hawaii International Conference on System Sciences, pp. 1–10 (2009)
11. Becerra-Fernandez, I., Cousins, K.: Nomadic context-aware knowledge management systems: applications, challenges and research problems. International Journal of Mobile Learning and Organisation 1(2), 103–121 (2007)
12. Malhotra, A., et al.: Radical Innovation without Collocation: A Case Study at Boeing. MIS Quarterly 25, 229–249 (2001)
13. Balmisse, G., Meingan, D., Passerini, K.: Technology Trends in Knowledge Management Tools. International Journal of Knowledge Management 3(2), 118–131 (2007)

Collaborative Conceptual Modeling Using an Ontology

Arkalgud Ramaprasad, Steven Diduca, Abhinaya Javier, Tarun Misra,
Ganesh Parasuraman, Pratibha Prabhakaran, Himanshu Sharma, Devi Subramanian,
Anuroopa Thangaraja, Ankita Vij, David Yi, and Jolita Zukaite

University of Illinois at Chicago, 601 S Morgan Street (MC 294)
Chicago, IL 60607, USA
{prasad,ajavi2,tmisra4,gparas2,pprabh3,hsharm3,dsubra3,
athang3,avij3,dyi2,jzukai2}@uic.edu,
steven.diduca@gmail.com

Abstract. This paper describes the method of collaborative conceptual modeling using an ontology. The method was used for the design of a student lifecycle management system at a University. The problem was presented to the students of a Knowledge Management class by the Decision Support staff of the University System. Starting with a core knowledge management framework used for the class the students, in collaboration with the instructor, developed the student lifecycle management ontology to synthesize the requirements. The class presented the ontology to the Assistant VP of Decision Support and her staff for validation, together with prototype dashboards based on the ontology. They found the ontology to be comprehensive, insightful, and useful.

Keywords: Ontology, Collaborative Modeling, Conceptual Modeling.

1 Introduction

This paper presents the method of collaborative conceptual modeling using an ontology. It is based on the design of a Student Lifecycle Management (SLM) system using the method. The project was done by a class of graduate and undergraduate students in conjunction with their instructor. The Assistant Vice President for Decision Support for the University System and two of her staff members were the clients.

A large group working on the design needs a systematic method to synthesize the disparate knowledge and apply it to the design. This paper describes using an ontology as a conceptual model for such a synthesis, to collaboratively model the problem to design a solution.

2 Collaborative Conceptual Modeling Using an Ontology

An ontology is a "specification of a conceptualization" [1, p. 907]. "Ontologies are used in information systems design to standardize terminologies, map requirements, organize them systematically, facilitate integration of systems, promote knowledge exchange, etc. [1, 2] Ontologies are related to but different from taxonomies, typologies, concept hierarchies, thesauri, and dictionaries. [3] They are tools for systematizing the description of

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 25–32, 2010.

complex systems [5]; a way of deconstructing the architecture of complexity [4]. Such systematization, in turn, facilitates analysis and design of these systems." [5]

Ontological analysis [6] is "the method of parsing the problem into its component dimensions and taxonomies to capture its complexity with natural language descriptions using a structured terminology." [5] The dimensions of the problem and the taxonomies of the dimensions are logically derived from the statement of the problem – in this case student lifecycle management. The sentences derived from the dimensions and taxonomies provide a complete closed description of the problem. This is similar to the ontology maturing process [7] in which ontology building is a learning process, formality and complexity of use is seen as a barrier, and there is a continuous evolution in work processes.

"The concept of ontology and ontological analysis has been proposed for and used in information systems design [8-13]. They have primarily sought to use ontologies to represent concepts. We use it at a higher level of abstraction and coarser granularity, but guided by the same need for a parsimonious and economical representation as a 'strategic ontology'[11]." [5] In the following we discuss the concept of using an ontology for collaborative conceptual modeling, the problem of SLM, the ontology developed for SLM, validation of the ontology, and the conclusion.

The concept of using an ontology for design is described by Ramaprasad [5]. It is a shared graphical/textual representation which allows all stakeholders to be "actively engaged in the construction and modification of [conceptual] models" [14, p. 61] "With increasing complexity of systems and organizations, creating shared understanding and joint representations of those systems [such a SLM] becomes increasingly important." [14, p. 62] Thus the proposed method can be described as "[t]he joint creation of a shared graphical[/textual] representation of a system." [14, p. 62]

An ontology has the advantage of visually representing the complexity parsimoniously at different levels of detail – thus allowing the collaborators to describe the system at different levels of granularity. Structurally it is scalable and extensible to fit the collaborators' perspectives. Additional dimensions can be added or redundant dimensions eliminated; additional categories and subcategories can be added or existing ones conflated. It can be used to simultaneously represent the macro with the micro – the big picture with the detail, the whole and the parts, metaphorically and literally.

The above characteristics make an ontology an effective collaborative conceptual modeling tool for design. Each system requirement can be seen as a partial description of the system. All the requirements of all the collaborators taken together represent an incomplete description of the system. By collectively inducting the underlying dimensions of the requirements and describing the taxonomy of categories in each dimension suggested by the requirements, the collaborating group can encapsulate the requirements in an ontology.

In the following we present collaboratively developed and validated ontology to conceptually model the requirements of a SLM system in a University.

3 The Problem of Student Lifecycle Management

The university has a very comprehensive Electronic Data Warehouse (EDW) which will be the foundation for SLM. The objectives of the SLM are to: (a) Identify success factors, (b) Monitor matriculation, (c) Monitor academic progress, and (d) Identify retention issues.

Two groups of students chose two of the four goals in the top layer of the schematic, namely: (a) Identify success factors, and (b) Monitor academic progress. Details of the Electronic Data Warehouse (EDW) and the BANNER data model were made available to the students. The staff members were available for consultation.

In addition to the presentation by the Decision Support staff and the documents provided by them the students used the following sources of data to formulate the ontology: (a) presentation by and discussion with an Assistant Dean of Undergraduates and the Vice Provost of Admissions and Records of the University, (b) exhaustive review of academic and practitioner literature on SLM, (c) exhaustive review of practices, policies, and procedures across the three University campuses, (d) thorough review of practices, policies, and procedures in other US universities, (e) informal discussions with undergraduate students outside the class, and (f) Personal experience of students.

4 An Ontology of Student Lifecycle Management

Starting with a core knowledge management ontology used for the class the students, in collaboration with the instructor, developed the ontology shown in Figure 1 to synthesize the requirements. In the following we describe the dimensions and taxonomies constituting the ontology, and the requirements encapsulated in it.

4.1 Outcome

The outcome dimension is shown in the rightmost column of the ontology in Fig. 1. The four desired outcomes of an SLM system are improvements in recruitment, admission, retention, and graduation of students. These outcomes are sequentially dependent and ordered top to bottom in the ontology. Thus, recruitment affects admission, admission affects retention, and retention affects graduation. Together they determine the key parameters of a university's performance such as the time to graduation and retention rate.

The information requirements of the four outcomes overlap sequentially. Each outcome has its own unique requirements. At the same time it draws upon the information for the earlier outcome. Thus recruitment information feeds into admission, admission information into retention, and retention information into graduation. The stages of the information lifecycle parallel the stages of the student lifecycle.

4.2 Users

The users dimension is the second column from the right in Fig. 1. It is shown as a two-level taxonomy of all users who have a role in SLM – those that 'touch' and are 'touched' by the students. The taxonomy covers the university laterally across its different functions from teaching to lodging the students, and vertically along the hierarchy from the highest administrator down. The users affect the outcomes in different ways and at different stages. Consequently their information requirements will be different. The taxonomy of users can be further refined based on their information requirements – new categories with specific requirements can be added, existing categories with similar requirements can be aggregated.

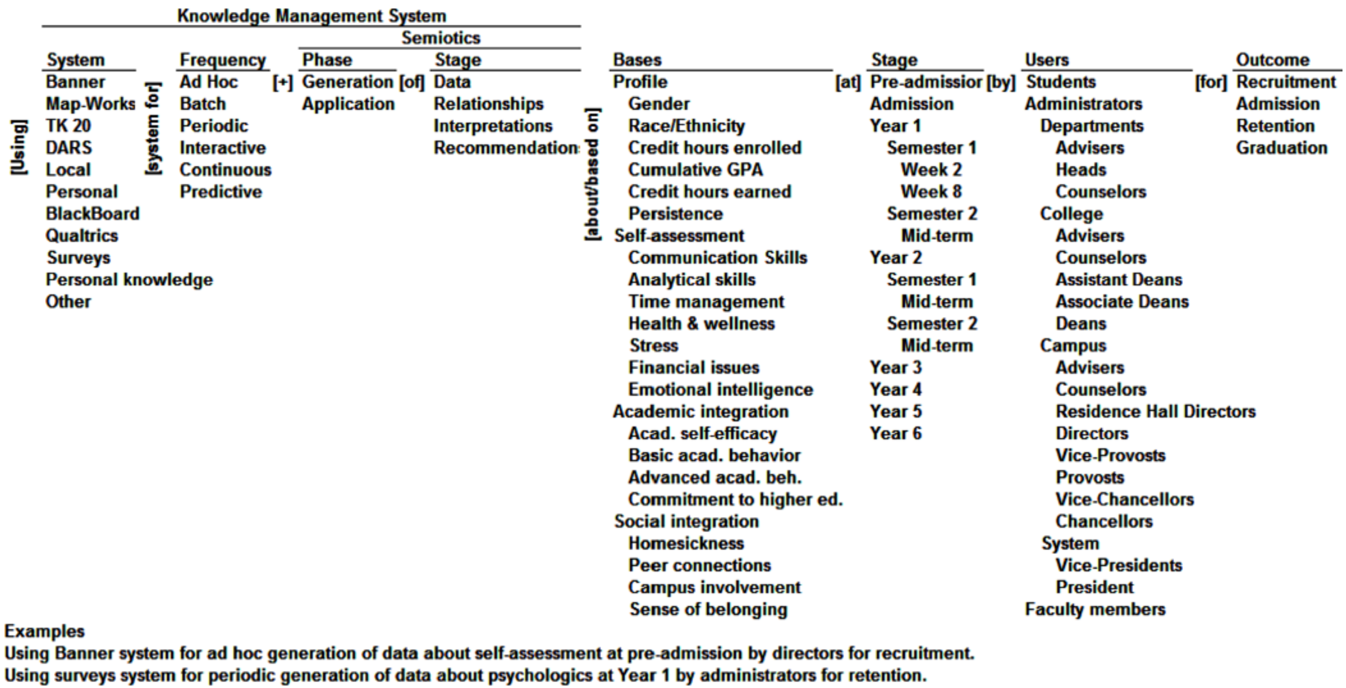


Fig. 1. Ontology of Student Lifecycle Management (abbreviated)

4.3 Stages

There are a number of stages in the student life cycle which are critical to obtain the four desired outcomes. It is a temporal dimension. However, the taxonomy is very fine-grained in the early stages of SLM and coarse-grained later. The difference in granularity reflects the need for early, frequent, and timely intervention for students at risk of not continuing with their studies. Thus, in the first semester of the freshman year it is important to monitor the student's performance at weeks 2 and 8. However, in the junior and senior years yearly monitoring may be adequate.

4.4 Bases

There is a large variety of information necessary to obtain the four outcomes. The 'Bases' dimension is a two-level taxonomy of the types of information. It is shown in the fourth column from the right of Fig. 1. Academic information alone is not sufficient for effective SLM; the users need psychological, social, financial, and other information too to ensure the continuity of the student in the University through graduation. All the bases may not be needed at all stages by all users for all outcomes. Each user acts on the bases of different information at different stages to obtain the desired outcomes. Use of more bases of information for SLM can bring about a significant increase in retention and graduation rates resulting in increased savings of money for the University.

4.5 System

The system dimension is the leftmost column in the ontology. It is a list of systems in the University which are the potential sources of information for SLM. The initial problem was formulated to use BANNER for SLM. It soon became clear that BANNER could not cover all the bases of information required. Other systems in the University had or could obtain the information. These systems would have to be linked, if not integrated, for effective SLM. More systems may have to be added as they are discovered or developed.

4.6 Frequency

This is the second temporal dimension in the ontology. It represents the common categories of the frequency of information management. It ranges from ad hoc through continuous to predictive. For effective SLM it is insufficient to depend upon past data; continuous monitoring of some variables is important; and in many cases predictive modeling too is important in avoiding potential pitfalls for students.

4.7 Semiotics

The Semiotics dimension forms the core of the knowledge management model used for the class. Knowledge management was defined as the generation/application of data/relationships/interpretations/recommendations. Generation/application is called the Semiotic Phase dimension in the ontology; the data/relationships/interpretations/

recommendations is called the Semiotic Stage. They are shown in the third and fourth columns from the left respectively.

4.8 Requirements

The requirements encapsulated in the taxonomy can be enumerated by concatenating natural language sentences using a word/phrase from each column together with the conjunctive word/phrase between the columns. Two examples are given at the bottom of Fig. 1. A very large number of such sentences can be concatenated. Logically, each of them could be a requirement of the SLM system. However, practically, they will not all be relevant or feasible. The selection of the necessary requirements from the universe of all possible requirements encapsulated in the ontology will determine the efficacy of the system.

5 Validation of the Ontology

The process of validation was two-fold. First, there was an ongoing internal debate and discussion among the students, and between the students and the instructor. This debate was spread over many weeks during the latter part of the semester. During this debate the ontology was modified iteratively to incorporate new information. A conscious attempt was made not to exclude any information about SLM. When a new requirement for SLM was obtained, the ontology was checked to verify whether it could accommodate the requirement or needed to be modified. When necessary, the ontology was modified by adding a dimension or a category. An example of such addition is the 'Stage' dimension (primarily initiated by the presentations by the Assistant Dean and the Vice Provost). At the conclusion of these iterations the students and the instructor were comfortable with the validity of the dimensions and their taxonomies.

Second, the ontology was presented twice to the clients (Assistant VP of Decision Support and her two staff members) for their critique and comments. They were surprised and pleased by the comprehensiveness of the dimensions and taxonomies. They very quickly found it to be easy to understand and insightful. For example, they had not thought of using 'Financial factors' as part of SLM although the data is in BANNER. Its importance and utility became apparent to them immediately.

The students created dashboards to show how some of the key combinations derived from the ontology could be implemented in a SLM system. These dashboards used real and simulated data. The clients found the dashboards to be very useful in conceptualizing the SLM system and understanding the implications of the ontology.

The clients also remarked during the discussions that the ontology could be overwhelming in two ways: (a) cognitively overloading the designer with the large number of possibilities, and (b) generating requirements which could overwhelm any design.

6 Conclusion

We present a functional ontology for the design of a SLM system. The sentences derived from the ontology make it easy to judge the semantic quality of the model [14].

Thus collaborative conceptual modeling using an ontology can be used as a tool in Collaboration Engineering [15], for "collaborative learning of design" [16, p. 170], and "architectural decision capturing" [17, p. 15]. The entire ontology, the dimensions, or parts of the taxonomies of each dimension can be used in "recurring collaboration tasks". [15, p. 611] Similarly, the dimensions of the ontology, the taxonomy of each dimension, and the ontology itself are design patterns which can be reused [18]. They can also provide focus for generating inquiry and synthesizing knowledge. The method will allow the designers to "integrate both ontological (top-down) and discursive (bottom-up) approaches to knowledge elicitation and structure" [19, p. 672] in designing a system. The discursive knowledge is encapsulated in the concatenations derived from the ontology.

References

1. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies 43, 907–928 (1995)
2. Gruber, T.R.: Ontology. In: Liu, L., Ozsu, M.T. (eds.) Encyclopedia of Database Systems. Springer, Düsseldorf (2008)
3. Gilchrist, A.: Thesauri, taxonomies and ontologies – an etymological note. Journal of Documentation 59, 7–18 (2003)
4. Simon, H.A.: The architecture of complexity. Proceedings of the American Philosophical Society 106, 467–482 (1962)
5. Ramaprasad, A., Papagari, S.S.: Ontological Design. In: Proceedings of DESRIST 2009 (2009)
6. Ladkin, P.B.: Ontological Analysis. Safety Systems 14, 1–7 (2005)
7. Braun, S., Schmidt, A., Walter, A., Nagypal, G., Zacharias, V.: Ontology maturing: a collaborative web 2.0 approach to ontology engineering. In: Proceedings of the Workshop on Collaborative Construction of Structured Knowledge (CKC), Banff, Canada (2007)
8. Burton-Jones, A., Gallivan, M.J.: Toward a Deeper Understanding of System Usage In Organizations: A Multilevel Perspective. MIS Quarterly 31, 657–679 (2007)
9. Gregor, S.: The Nature of Theory in Information Systems. MIS Quarterly 30, 611–642 (2006)
10. Hevner, A.R., March, S.T., Jinsoo, P., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75–105 (2004)
11. Nelson, K., Nelson, H.: The Need for a Strategic Ontology. In: Proceedings of the MIS (Management Information Systems) Quarterly Special Issue Workshop on Standard Making: A Critical Research Frontier for Information Systems (2003)
12. Parsons, J., Wand, Y.: Using Cognitive Principles to Guide Classification in Information Systems Modeling. MIS Quarterly 32, 839–868 (2008)
13. Wand, Y., Monarchi, D.E., Parsons, J., Woo, C.C.: Theoretical foundations for conceptual modelling in information systems development. Decision Support Systems 15, 285–304 (1995)
14. Renger, M., Kolfschoten, G.L., Vreede, G.-J.: Challenges in Collaborative Modeling: A Literature Review. Advances in Enterprise Engineering I, 61–77 (2008)
15. Kolfschoten, G.L., Briggs, R.O., de Vreede, G.-J., Jacobs, P.H.M., Appelman, J.H.: A conceptual foundation of the thinkLet concept for Collaboration Engineering. International Journal of Human-Computer Studies 64, 611–621 (2006)

16. Bravo, C., Redondo, M.Á., Ortega, M., Verdejo, M.F.: Collaborative environments for the learning of design: a model and a case study in Domotics. Computers & Education 46, 152–173 (2006)
17. Zimmermann, O., Gschwind, T., Küster, J., Leymann, F., Schuster, N.: Reusable Architectural Decision Models for Enterprise Application Development. In: Overhage, S., Szyperski, C., Reussner, R., Stafford, J.A. (eds.) QoSA 2007. LNCS, vol. 4880, pp. 15–32. Springer, Heidelberg (2008)
18. Kolfschoten, G., Lukosch, S., Verbraeck, A., Valentin, E., Vreede, G.-J.: Cognitive learning efficiency through the use of design patterns in teaching. Computers & Education 54, 652–660 (2010)
19. Pike, W., Gahegan, M.: Beyond ontologies: Toward situated representations of scientific knowledge. International Journal of Human-Computer Studies 65, 674–688 (2007)

A Mobile Computer System to Support Collaborative Ethnography: An Approach to the Elicitation of Knowledge of Work Teams in Complex Environments

Luiz Carlos L. Silva Junior, Marcos R.S. Borges, and Paulo Victor R. de Carvalho

Graduate Program in Informatics, Universidade Federal do Rio de Janeiro,
Rio de Janeiro, Brasil
luiz_junior@petrobras.com.br, mborges@nce.ufrj.br,
paulov@ien.gov.br

Abstract. The paper presents a collaborative ethnography approach for knowledge elicitation of work teams in complex environments. It discusses the concepts of cognitive systems, distributed cognition, and presents a review of methods commonly used in the elicitation of knowledge both in the case of traditional and complex environments. Then, it points to some advantages of a collaborative approach in comparison to other non-collaborative approaches. An evaluation plan of the collaborative ethnography approach based on experimentation, and the development of a mobile system to support the proposed methodology is also presented. This system aims to stimulate collaboration and an organization in the ethnographic knowledge elicitation process.

Keywords: Ethnography, mobile.

1 Introduction

The human interactions in group work activities are important objects in the study of human problem solving and decision-making processes, and give important clues to investigate the tacit knowledge that teams uses during their work activity. Behind these interactions we can find important mechanisms of reasoning that will drive people, according to their experience, to select and manage their actions in context-dependent work situations. The analysis of these interactions can lead to the discovery of basic requirements for the construction of artifacts that can efficiently support the process of team members decision-making.

The aim of this paper is to show how collaborative ethnography (i.e. an ethnography performed by many agents who can interact with each other) can be used to elicitate the knowledge that shapes team members interactions. We also show in which way the knowledge obtained using the collaborative ethnography can be very important to elicitate useful knowledge for the implementation of artifacts, training and groupware systems that will give adequate support to people tasks.

We begin the paper with cognitive systems definitions, including the concept of distributed cognition, and some examples of these systems in real life. Then we present some cognitive task analysis (CTA) methods used in knowledge elicitation

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 33–48, 2010.

process describing their strong and weak points. Later, the collaborative ethnography approach for knowledge elicitation will be presented, with a collaborative ethnography mobile support system and evaluated by an experiment that will point their efficiency in the elicitation of knowledge in a real complex environment of a nuclear reactor control room.

2 Understanding Cognition behind Human Work

A cognitive system is a self regulated and adaptable system that functions using knowledge about itself and the environment to planning and modify their actions [12]. An adequate design of a cognitive system depends basically on: The existence of a common vocabulary between their parts; the study of the agents' cognition during their actual activity, analyzing the man-technology interactions in real environments [13] rather than studies that analyze each system component apart (i.e. man, machine and interface) in more controlled settings.

Our research focus on cognitive systems design because their ever growing socio-technical complexity due to new technologies (automation, communication, information) and new system structures (more regulation levels, consumer feedback and control) claims to a change of the traditional (linear) models and old paradigms of human cognition in system design to they take in consideration the actual human-system interactions.

Nowadays we can observe the technology facilitating the execution of tasks and operations with high degree of automation. Cognitive systems are present in energy power plants, air traffic management systems, vehicles, and in work environments that demand high degree of coordination. To produce effective cognitive systems, i.e. cognitive systems that can be adequately handled by the human agents, there is a need to consider actual cognitive requirements in the system design phase.

However, the increasing use of technology results in an increase of the complexity of cognitive systems. The digital control rooms with large video display units, computers with ever growing processing capacity, and other technological devices to support cognitive tasks generate other problems, such as the search for efficiency, making systems operate closer to their maximum capacity and safety boundaries, an increasing performance dependence among the various tasks, and a significant growth of the amount of data that have to be processed by human beings [13].

Such situations can generate a considerable degradation in these systems. To cope with complexity, human agents adapt their activities generating variability in the tasks execution, which may lead to systems malfunction (the drift to failure mechanism). It is important to note that in the last few decades serious accidents have happened in socio-technical complex systems, indicating the need to change the paradigm on how human cognition is treated in complex systems design [23].

Many of these system failures and accidents emerge in situations, in which operators did not understand the actual situation, i.e., the systems do not provide the right information in the right moment to the operators who were not able to adapt their behaviors according to the actual situation demands and make their decisions in a safe way. To deal with this problem is necessary to emphasize design processes that facilitate the adaptation of the human cognition to the system functionality.

Hutchins [16] states that cognition is best understood as a distributed phenomenon. The theory of distributed cognition, [11] seeks to understand the organization of cognitive systems by extending the scope of what is considered cognitive beyond the individual and cover the interactions between people and with the resources and materials into the environment. This concept is very important in this work, since the aim of the study is that the interrelationships between people and artifacts, as well as groups of people, in complex environments.

Distributed cognition looks for cognitive processes, wherever they may occur, depending on the functional relationships of the elements that participate together in the process. It is important to note in this context that a process is not cognitive simply because it happens in a brain, or ceases to be cognitive, simply because it happens in the interactions among many brains. In distributed cognition, is expected to find a system that can dynamically configure itself to put in coordination to perform many functions.

A cognitive process is delimited by the functional relationships between its elements and by placing the same space. In the field, applying these principles, there is mainly three types of distribution of cognitive processes [11]:

- Cognitive processes may be distributed over the members of a group.
- Cognitive processes may involve coordination between internal and external structure (material or environmental).
- These processes can be distributed over time, so that the products of earlier events can transform the nature of later events.

The cognitive requirements, in the scope of this work, can be defined as the functional requirements of the system that will give critical support to the cognitive activities of the operator throughout the execution of his/her work. This means that these requirements will be paramount for the construction of information systems capable to enable people to achieve adequate situation awareness (enhance perception possibilities, decision-making and action planning support).

These cognitive requirements must be incorporated in the design of displays, man-machine interfaces, and are present in the collaborative characteristics of many systems. We can observe some examples in: control rooms [5], [24]; military decision-making systems [3]; air traffic control [2], [9].

To elicitate knowledge for complex systems design, we argue that is necessary to carry out cognitive task analysis in actual work situations, understanding how and why operators make their actions, and the intrinsic human-system relations (human-organization, human-technology, and human-human actions). Therefore, we claim that collaborative ethnography is one of the most adequate approach to elicitate knowledge. In the next section, we will describe some methods traditionally used to elicitate knowledge for information systems design, in order to compare them to the collaborative ethnography approach we propose in this paper.

3 Knowledge Elicitation Approaches for Cognitive Task Analysis

In this section, some CTA methods already used for in information system design in several environments will be described. Some of these methods, like interviews and direct observation are often used in the elicitation of requirements for information system design and business-oriented processes. Ethnographic approaches are not so used in information systems design, but they are widely used in the elicitation of experts' knowledge and social factors.

3.1 Interviews

Interview is one of the most common methods for requirements elicitation, however nor always does this technique capture all the information needed to a complete requirements definition. To a large extent, the data generated from interviews are field notes, meeting reports, which are difficult to analyze later on, and can be complemented with the existing documentation in the organization.

In many situations, the information elicitated from interviews was incomplete, due the existence of difficult questions to be answered, i.e., questions related to activities where tacit knowledge is used. Bell [1] noted that is not rare the situation in which people tell the idealized information, i.e., people say how they should do their tasks rather than how they actually do them.

There are research to refine the interview technique [4], helping the organization and integration of the data using argument structures or domain models, as the Joint Application Development (JAD) approach that provide a way to requirement elicitation allowing the identification of different points of views, and means to solve conflicts and achieve consensuses.

3.2 Observation

The observation of work activities in the organizations is a useful way to understand the interactions, practical skills, and tacit knowledge developed by work teams, as well as the organization culture. According to Bell [1], we do not have to ask people what they do. Rather, we must observe how people carry out their activities, because from observation findings it is possible to discover the reasons and motivations behind systems use, and the problems or difficulties related with system usability.

The use of observation appears in some domains for knowledge elicitation and system improvement. In ergonomics, direct observations are the most used method to understand the actual work conditions, the operators' activities and their influence on the people health and in the system production [7]. However, to describe the distributed regulation mechanisms, in the sense of how operators work as a crew, both in nominal and degraded situations, to understand how agents regulate (adapt) collectively their work when confronted different situations, we must perform deeper and more systematic observations in actual work settings. Therefore we need an ethnographic approach.

3.3 Ethnography

The ethnography is a methodology that comes from the Social Anthropology, which consists of studying an object living the same reality of this object. This approach aims to understand and to describe, a nation, its people or culture, using natural or participative observation for long periods.

Using ethnography, the observed group or culture issues are understood by living in the same environment, being present, and making the things that the people make and as they make [1]. It is possible to understand how and, mainly, why the activities are done in one determined way, because the phenomenon is studied inside the social, cultural and organizational context [21].

The ethnography contains traces of the two previous techniques, because the information collected are complemented with data from observations and from informal interviews. However, the difference in the ethnography is that all individuals (observers and observed) are situated in the same context. Therefore, the method has many possible advantages [18]: Bigger familiarization with the domain of the organization; discovery of the informal activities, issues related to cognitive and collaborative mechanisms; better understanding of the actions, decisions, strategies, behaviors and interactions of the agents in the context where they occur; perception on how the interaction with the resources or devices occurs; and more visibility about usability problems of the existing technologies.

Moreover, the ethnography shows how the physical disposal of the work environment and the layout of the devices and equipment have influence, positive or negative, on the efficiency of the activities as well as in the difficulties, and the respective adaptations to face work problems.

Using the ethnography for the elicitation of requirements, we are able to get more details and a more complete description of these requirements, because the activities are observed instead of described by the users. These factors are especially important for the analysis and design of complex systems.

One of the ethnographic techniques we will use ahead in the Collaborative Ethnography we propose in this research is the Fast Ethnography [14] that recognizes the time and resource limitations to carry out detailed and intensive ethnographic studies, and uses short term observations together with the debrief meetings feedback.

The ethnographic approach presents some problems. Normally, its execution is complex, due the impossibility of one person alone to capture and to analyze all the information of the social environment. The method demands extensive planning and coordination if the ethnography would be carried out by teams, because people may have different perceptions and viewpoints about the observed activities. Another limitation is the need of a long time to capture and analyze what occurs in the field, to get rich and detailed information [20]. In some cases, it is necessary a familiarization with the domain to be studied. Moreover, it is difficult to use the technique in complex and distributed settings due the huge number of variables to be observed.

The technique also can present risks for the researchers, or to be impracticable if the presence of the researches/observers jeopardize the work activities observed. In some cases, there is the need of getting the permission for entrance in the work settings and for registering the information, and also the acceptance and the assent from those who will be observed [7].

4 The Collaborative Ethnography Approach for Knowledge Elicitation

In this section we will discuss how the collaborative approach can address the knowledge elicitation of work teams. We first stress the importance of the combination of collaboration and ethnography, particularly how this combination can help the study of complex activities. Then, we present a succinct review of previous work, which have used this combined approach to elucidate interactions and the social dimension of teamwork.

4.1 Why Collaborative Ethnography?

First, it is pertinent to clarify why to add the collaboration to the ethnography method. A straightforward combination between observers and observed subjects is presented in Table 1. We can have a single person observing another person or a group, and a team observing a single person or a group.

Table 1. Combination of ethnography studies

Observer/ Observed	Individual	Teams
Individual	A	B
Teams	C	D

Scenario A is the most common case: An individual observing another individual carrying out some task. This situation also refers to a number of observer/observed pairs being done in parallel. If the observation refers to the same task some variation may occur between different observers/observed. The results are quite dependent on the background and on the previous experience of the observer. There may also be some information loss during the observation. In this scenario, the aggregation of information from different sources is not an easy task and may show some inconsistencies.

Scenario B is possible but not common. In this scenario the loss of information is potentially high. It is very difficult to a single observer to capture and to understand all tasks and interactions, particularly if the activities are complex. A way to overcome these difficulties is to do the observation in several sessions.

The scenario C is normally used when the observation requires multiple perspectives over the same observer or task. In this scenario it is expected that the information gathered and processed would be richer that that collect by a single observer. This can be considered a particular case of scenario D, assuming that the observers would meet and discuss about their findings before, during and after the ethnography. This is not easy to realize without the support of an appropriate process and a supporting tool.

Finally, the scenario D is the circumstance that will be dealt with in this work. The observation of groups and their interaction being performed by a team of ethnographers is a very challenging goal. The huge amount of information collected that needs to be organized and processed requires not only an organized process but also an appropriate supporting tool. The complexity of this alternative is a consequence of combined circumstances originated by the multiple, and perhaps conflicting, perspectives from the observers´ part and the potentially high number of interactions among the members of the observed group. We will try to avoid the loss of relevant information intrinsic to scenario B.

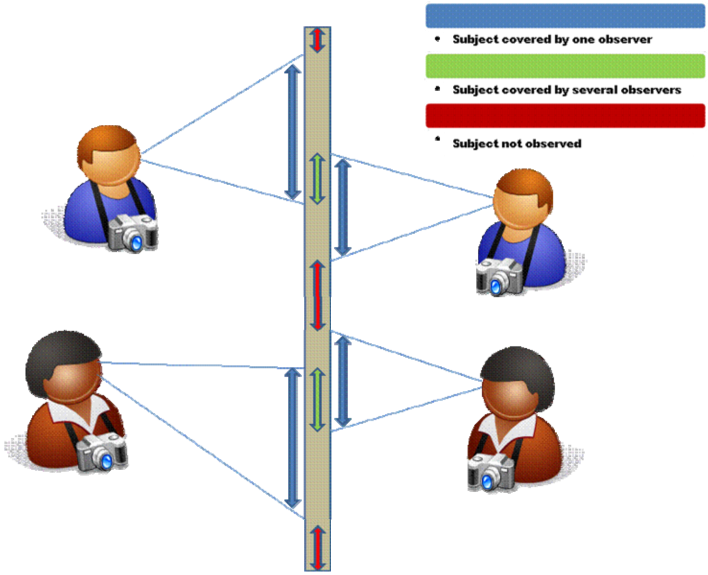


Fig. 1. Collaborative ethnography in the field

Even a collaborative ethnography can generate losses and in some cases conflicts. As shown in Figure 1, when people are observing a complex event, losses may occur due to several possibilities: the complexity of cognitive actions required, the geographically dispersed event (the same observer cannot be in two places at the same time), or even lack of attention from the observer. Redundancy in these cases is essential.

4.2 Applications of Collaborative Ethnography

The collaborative ethnography is a technique that has been used extensively in other domains for the social analysis of work activities. May and Pattillo-McCoy [19] reported a work reproducing the way the ethnographers discuss their findings in a collaborative way generating a single document with the information collected and their different viewpoints. They reported how different the ethnographers´ perceptions about the environment are. They claim that variety is due to several cognitive factors that also help the reach of consensus in the discussion.

It is also important to note the value of the collaborative ethnography to complement the details of the object of study. The inconsistencies and ambiguities of observations are more easily identified and solved. In other words, the ethnographers play a positive influence on the results of the process.

On the other hand, it becomes clear that the collaborative ethnography has some drawbacks. First, it shares, or perhaps increases, the time spent to collect, analyze and conclude about the collected information. The need to join, to compare and to discuss the information collected requires time. The collaboration required needs extra effort and is time consuming. The field notes needs to be combined in order to reflect the different perspectives of observers. According to May and Patillo-McCoy [19] the observation itself influences the results of the ethnography. Thus, the use of several observers opens the possibility of analyzing this impact in the process and, at the same time, enriching the information collected.

We expect that the approach contributes to the expose of tacit information, specialized knowledge and the decision rationale, mostly due to the interaction between the ethnographer with the task, the environment and especially with other ethnographers.

Common ethnography is also collaborative [17]. Lassiter makes it clear that the communities in which they conducted their work would be very difficult to perform the studies without an effective involvement of researchers in real life and day after day. In this case, ethnography becomes a collaborative technique from the time when there is field interaction between ethnographers and also between the interlocutors. The aim here is to show that the term "collaborative ethnography" puts collaboration at the heart of ethnographic technique, rather than simply assume it sporadically.

The goal of this collaborative approach is in fact deliberately and explicitly emphasize collaboration across the ethnographic process. Since the conceptualization of the project, through fieldwork and during the writing process. Collaboration in this case is important to allow the researchers comment, make it part of the ethnographic and may be reinstated within the process field, generating inputs for new discoveries.

Machado [18] also carried out a collaborative ethnography study aimed to elicitate software requirements. The study reports an experience with field observations and with the use of a tool that facilitates the consolidation of data from multiple sources into a shared repository. In comparison with traditional techniques based on interviews, the observation approach have shown more efficient and generated more complete information.

The experiments carried out in that study has shown that a groupware plays an important role in the coordinating of activities and on the organization of the data (videos, photos, comments, notes, etc.) reducing the burden generated by the collaborative ethnography.

5 Mobile System to Support Collaborative Ethnography Approach

According to Guerlain [8], the evaluation of activities from a coordinated team involves the independent observation of multiples individuals. The analysis of these independent observations requires they should be coordinated, codified and correlated before a subjective evaluation can be performed.

Thus, similar to other works [8] [15] [18], the support of a tool aimed to assist the information exchange and the discussion among ethnographers is desired. The tool that supports the collaborative ethnography should facilitate collaboration and interaction during and after the gathering of field data in order to identify the relevant knowledge with less effort.

In this work, the tool that will support the ethnographic method aims at structuring of the before (preparation phase), during the (capture phase), encouraging collaboration and interaction among specialists in the gathering of information in the field, and after ethnography, producing a consolidated report of the field. The process is depicted on Figure 2.

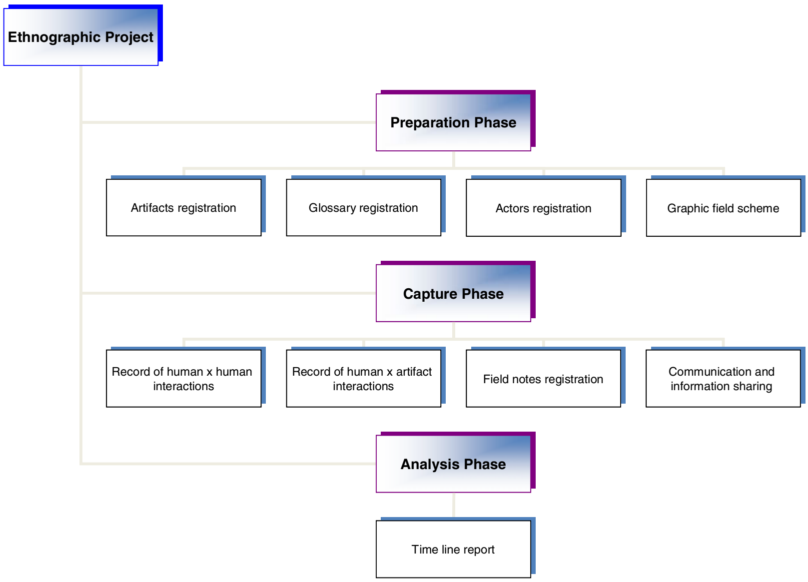


Fig. 2. Structuring of ethnography and mobile system features

Breaking apart this proposal, the mobile support system for ethnography aim to provide technological support to the researcher running the ethnographic study in the preparation, capture and analysis of collaborative ethnography, specifically the following points:

- Preparation phase: Definition of tasks to the team, the choice of perspectives and key people, choice of observable variables, choice of registration methods.
- Phase Capture: Collecting records and elements of observation, storage and cataloging, transcript of records
- Analysis Phase: Organization in the timeline and categorized records

The basic requirements for the tool to support collaborative ethnography approach are:

- Allow the registration of those involved in ethnographic study (actors, operators, researchers and others involved)
- Allow the registration of artifacts used by those involved in ethnographic study (tools, systems and other objects that are the subject of interactions)
- Allow the record of field characteristics and related graphics.
- Allow the record of terminology and expressions used by those involved in field study.
- Allow the record of different types of interaction between the involved and / or artifacts previously registered (human x human, human x artifact, artifact x artifact).
- Allow the record of field notes, activities, events and disturbances in the execution of tasks.
- Support the information sharing between ethnographers during the implementation of the ethnographic study

- Support the need for communication between ethnographers at certain times of ethnography.
- Support the division of labor during the ethnographic study.
- Generate reports to the chronological order of events recorded (timeline), different categories, and all information gathered in preparation.

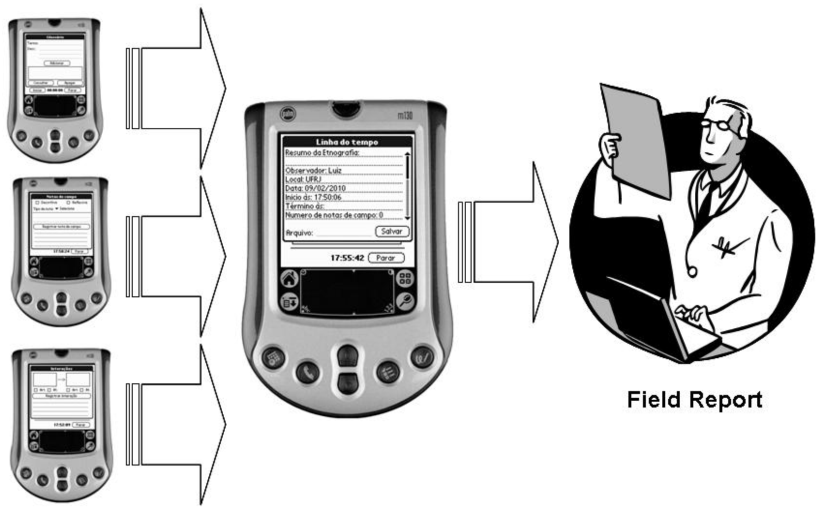


Fig. 3. A schema, inputs and outputs of supporting tool

Several features were designed with the goal to centralize the information collected in the field in a single device in an organized and retrievable manner. These features are grouped together to support before, during and after an ethnographic study in an attempt to leverage the work of the observer in his field research. A snapshot of schema, the input devices and one of the outputs are depicted on Figure 3.

The system can also be considered a pioneer in event log on the field through mobile devices. The literature review indicates the work of Fransson-Hall et. al. [6] with the Portable Ergonomic Observation Method (PEO), and the FIT-System [10] as the first to support the cognitive and ergonomic study, performing the record of aspects of the workplace through portable computers.

The use of the mobile system was tested at the same time the technique of collaborative ethnography, in order to support it. To evaluate the usefulness of the functionality of the system, users answered a usability questionnaire and reported their experiences with the prototype.

6 Experimentation of Collaborative Ethnography for the Knowledge Elicitation in a Complex Environment

This section will be exposed the experiment using the ethnography for collaborative elicitation of knowledge in a complex environment. The trial is intended to indicate that the use of collaborative approach can elicitate a greater amount of information, adding quality to the information elicitated through the discussion of the information throughout the ethnographic process.

Hughes et. al. [14], Crandall, Klein and Hoffman [3] and Machado [18]were important sources for the framework of ethnography. The work of Hughes gave us inputs for the adoption of techniques of rapid and interative ethnography so that if there is a better use, given the time and conditions available for the implementation of the experiment. The work of Crandall et. al. [3] and Machado [18] will guide the study towards the implementation of methodological phases of ethnography and critical issues of what, who, when and how observe.

The experiment was conducted at the Laboratory of Human-System Interface (LABIHS) of Nuclear Engineering Institute (IEN) located in Rio de Janeiro. The LABIHS is a laboratory for experiments based on a compact reactor simulator. This simulator aims to improve safety and operational performance of nuclear power plants and other facilities in industry. LABIHS consists of an advanced control room, an experimenter's gallery room and other auxiliary rooms. The advanced control room consists of nuclear reactor simulator software, graphical user interface design software, a hardware/software platform to run and provide the adequate communication between these software systems, and the operator interface – VDUs and controls needed to operate the simulated process.

The goal of the ethnographic method application on LABIHS was the performance evaluation of the operators during postulated accidents that may occur in a nuclear power plant in order to improve the human computer interface design. We paid particular attention to the tasks dictated by the procedure manual and to the operators' actual activity. This experiment is part of process to search for particular deficiencies in the support of operator response to abnormal system states, in order to redesign the operator interface to improve upon the graphical layout of the information, the navigation across screens, the alarm presentation, acknowledgement and response, and to integrate these with computer-based procedures that dynamically correspond with real-time system information.

The LABIHS control room operating crew, who participated in the study, is composed of three operators – Shift Supervisor, Reactor Operator (RO) and the Secondary Circuit Operator (SCO). The Shift Supervisor is an engineer who has experience in the simulator operation. The RO and SCO are instrumentation technicians who have been trained in LABIHS operation for 2 years before this study, but have no previous experience in the reference plant operation.

As motivation for choosing this environment, we can mention the presence of common elements in complex environments such as large number of displays featuring information with frequent changes of state, need for coordination among distributed team working to resolve issues and events taking joint decision, in real time, with considerably gravity and impact.

The experiment was carried out by observers in three stages: preparation, capture and analysis of data (Figure 4). In the first two stages, the observers made observations, questions to the actors and recorded the information in the mobile system to support the ethnography. These individuals freely observe the actors (primary operator, secondary operator and supervisor) and interacted with each other to clarify problems, views and questions about the tasks being performed.

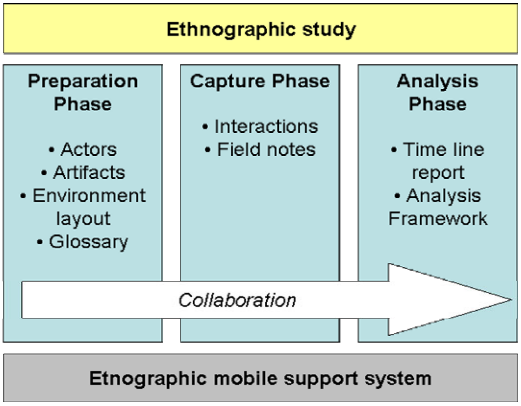


Fig. 4. Experiment framework through ethnographic phases

At the end of the stages, the observers met to discuss the questions and points that they found relevant on the records. After the two stages (preparation and capture) the observers met to analyze the data together and extract the relevant knowledge of the tasks observed in the field. Observers completed a framework with the main points, making a complete record of information elicitated.

	Questions	Observers		Especialists		Average
Ethnography	The use of collaborative factor helped in the preparation phase of ethnography?	4	5	5	5	4,75
	The use of collaborative aggregated in quantity of the information collected?	4	5	4	5	4,5
	The use of collaborative aggregated in quality of the information collected?	3	5	4	5	4,25
	The technique of collaborative ethnography facilitated the discovery of new information?	3	5	4	5	4,25
	In his opinion the collaborative approach turned the observation more effective than in a non-collaborative approach?	3	5	5	5	4,5
	There were easier to discuss and gather information after collaborative ethnography?	5	4	5	5	4,75
	The technique of collaborative ethnography allowed the observer to focus on certain events in the field?	5	5	4	5	4,75
Mobile System	The system facilitated the collaborative ethnography in the preparation phase?	4	4	5	5	4,5
	The system facilitated the collaborative ethnography in the capture phase?	4	4	4	4	4
	The system facilitated the collaborative ethnography in the analysis phase?	4	4	5	4	4,25
	The system facilitated the location of certain key aspects of the records audio and video ethnography from the index?	4	5	5	4	4,5
	The interface of the system has proved to be easy to use during the ethnographic study?	4	2	5	4	3,75
	The interface of the system allowed agility throughout the ethnographic study?	2	2	4	4	3
	The report generated by the system is easy to understand for the viewer?	5	4	5	4	4,5
	The system proved efficient performance throughout the ethnographic study?	5	4	5	4	4,5

Fig. 5. Participants answers to evaluation questionnaire

At the end of the experiment, the evaluation of the strengths and weaknesses was conducted through a questionnaire that assessed both the use of the mobile system and the methodology of observations.

It is important to remember, as one of the limitations of this study, observers had little experience with the practice of ethnography to knowledge elicitation. The observers held only theoretical knowledge about the execution of ethnographic practice, one of which had already run an experiment aiming to elicitate system requirements through ethnography. The lack of experience with mobile devices by observers is another important factor to be considered.

The implementation of the experiment in question was successful and served its original purpose to point out the hypothesis that collaborative ethnography can be more efficient than other non-collaborative knowledge elicitation in complex environments. This assumption can be verified by the subjective evaluation of the respondents (observers and two cognitive task analysis specialists) who indicated by means of the questionnaire (Figure 5), a general agreement with the benefits of incorporation of collaborative factor throughout the ethnographic process on the environment of a nuclear plant.

The answers to the questions are classified according to the Likert Scale: totally agree (5 points), partially agree (4 points), have no opinion (3 points), mildly disagree (2 points) and strongly disagree (1 point). Through the weighted average of these points were composed of the overall scores for each question. Were also encouraged comments from observers and experts on each question, so should be obtained from each of the justification of each individual score.

Another point found in the evaluations was the benefit of using a system of support for ethnography in the field, helping observers to collect, share, organize and analyze information. This factor was crucial to collect relevant information in complex environments, such as interactions between individuals and artifacts, field notes, classified by type of information and automatically creating a timeline containing all the events recorded in chronological order. These records were also important inputs for the analysis of other records often used to collect information in the field, records audio and video recording.

However, we have also verified counterpoints, limitations and opportunities for improvement, not only in the mobile support system, but also the approach of ethnography collaborative. One of the factors mentioned was the lack of direct comparison between the quantity and quality of information elicitated by two different techniques (a collaborative and other non-collaborative), leading to uncertainty in some places, since they will not enjoyed extensive experience in cognitive task analysis studies.

Despite these points, the participants believe that the collaboration between participants stimulated the discovery of knowledge, generating greater amounts of relevant information and adding quality after filtering and improved information through further discussion. The doubts and limitations were pointed out by participants on their observations and can be considered as a result of this work and input for future work.

7 Conclusions and Future Work

In this work we reviewed some important concepts in the area of cognitive systems. We also revised the main approaches for the knowledge elicitation: interviews, observation and ethnography. We confirmed that these approaches, though extensively adopted, still show some inadequacies and inefficiency in the knowledge elicitation process, particularly in relation to knowledge in complex environments.

To deal with these problems, we proposed a collaborative ethnography approach as a more appropriate alternative for the knowledge elicitation. We claim that the ethnography approach has many advantages, such as a detailed analysis of the environment where the operator performs his/her task. When ethnography is combined with a collaborative approach that supports different viewpoints as well as interaction and discussion among the observers, we observed an approach that produces a more efficient and consequential results.

As an important factor to realize ethnographic studies and faster and more organized data collection, we identified the need for an information system that would allow observers collecting the data in a ubiquitous, driven and structured manner, and still supported the technique of ethnography in its main aspect: the collaboration. This prototype has been specified, designed and called the ethnography mobile support system, being used later, in the experimentation of the technique in the environment of a simulated nuclear reactor.

In general, collaborative ethnography proved to be an effective technique for the elicitation of knowledge teams, allowing observers in two sessions to identify large amounts of relevant information such as activities performed, executors, and important information for decision making, interactions, artifacts problems and difficulties in performing the task. Collaboration in the unanimous opinion of the participants was a factor that improved the quantity and quality of records.

The system also showed some limitations, including: the need for expertise in the operation of the mobile device for recording the information fast, lack of agility in the exchange between the interfaces and functionality to chat, need for manual synchronization between the clocks of the devices and record video, to generate a timeline without major distortions. However, the quantity and quality of information have been well evaluated and have been identified opportunities for improvements in the prototype: the incorporation of new capabilities for standardized register of other types of events in the field and for advanced analysis of the information in an automated way.

As future work, we identified opportunities for improvement in the technique of collaborative ethnography, with a larger number of participants, experiments on other types of complex environments and adding new points to be observed in the framework, guiding especially when the observation made by beginners and people inexperienced in the environment in question. For the system to support mobile ethnography have identified areas for improvement so that it allows the registration of information faster, better interfaces and further support the analysis phase, integrating it with packages of statistical analysis.

Acknowledgements

This work was partially supported by CNPq (Brazil) grant No. 304252/2008-5, and LACCIR grant No. R0308LAC004.

References

1. Bell, G.: Insights into Asia: 19 Cities, 7 Countries, 2 Years - What people really want from technology. Technology@Intel Magazine (2004)
2. Bentley, R., Hughes, J., Randall, D., Rodden, T., Sawyer, P., Shapiro, D., Sommerville, I.: Ethnographically-informed systems design for air traffic control. In: Proc. Computer-Supported Coooperative Work Conference, pp. 123–129. ACM Press, New York (1992)
3. Crandall, B., Klein, G., Hoffman, R.R.: Working minds: a practitioner's guide to cognitive task analysis. MIT Press, Cambridge (2006)
4. Christel, M.G., Kang, K.C.: Issues in Requirements Elicitation, Technical Report, CMU/SEI-92-TR-012, ESC-TR-92-012 (September 1992)
5. dos Santos, I.J.A.L., Teixeira, D.V., Ferraz, F.T., Carvalho, P.V.R.: The use of a simulator to include human factors issues in the interface design of a nuclear power plant control room. Journal of Loss Prevention in the Process Industries 21(3), 227–238 (2007)
6. Fransson-Hall, C., Gloria, R., Kilborn, A., Winkel, J.: A portable ergonomic observation method (PEO) for computerized on-line recording of postures and manual handling. Applied Ergonomics 2, 93–100 (1992)
7. Guérin, F., Laville, A., Daniellou, F., Duraffourg, J., Kerguelen, A.: Compreender o trabalho para transformá-lo: a prática da ergonomia. In: Blucher, E. (ed.) Fundação Vanzolini, São Paulo (2004)
8. Guerlain, S., Shin, T., Guo, H., Calland, J.F.: A team performance data collection and analysis system. In: Proc. of the human factors and ergonomics society 46th annual meeting, Baltimore, MD, September 30 - October 4 (2002)
9. Harper, R., Hughes, J.A., Shapiro, D.: Harmonious Working and CSCW: Computer Technology and Air Traffic Control. In: Bowers, J., Benford, S. (eds.) Studies in Computer Supported Cooperative Work: Theory, Practice and Design. North-Holland, Amsterdam (1990)
10. Held, J.: Fitting the mental model: a new technique for computerised event recording. In: Harris, D. (ed.) Engineering Psychology and Cognititve Ergonomics, vol. 4, pp. 293–300. Ashgate, Aldershot (1999)
11. Hollan, J., Hutchins, E., Kirsh, D.: Distributed Cognition: Toward a New Foundation for Human-Computer Interaction. Research ACM Transactions on Computer-Human Interaction 7(2), 174–196 (2000)
12. Hollnagel, E., Woods, D.D.: Cognitive Systems Engineering: New wine in new bottles. International Journal of Human-Computer Studies 51(2), 339–356 (1999)
13. Hollnagel, E., Woods, D.D.: Joint cognitive systems: Foundations of cognitive systems engineering. CRC Press, Inc., Boca Raton (2005)
14. Hughes, J., King, V., Rodden, T., Andersen, H.: Moving out from the control room: ethnography in system design. In: Proc. of the ACM conference on Computer supported cooperative work, pp. 429–439. Chapel Hill, North Carolina (October 1994)
15. Hughes, J., O'Brien, J., Rodden, T., Rouncefield, M.: Ethnography, Communication and Support for Design. Technical Report Ref. CSEG/24/1997, Lancaster University, Lancaster, UK (1997), `http://www.comp.lancs.ac.uk/computing/research/cseg/97_rep.html`
16. Hutchins, E.: Cognition in the wild. MIT Press, Cambridge (1995)
17. Lassiter, L.E.: The Chicago Guide to Collaborative Ethnography. University of Chicago Press, Chicago (2005)

18. Machado, R.G.: Developing of a Collaborative Ethnography Method for Software Requirements Elicitation. Rio de Janeiro, 2008. M.Sc. Dissertation, Graduate Program in Informatics, Federal University of Rio de Janeiro, Brazil (2008) (in Portuguese)
19. May, R.A.B., Pattillo-McCoy, M.: Do You See What I See? Examining a Collaborative Ethnography. Qualitative Inquiry 6(1), 65–87 (2000)
20. Millen, D.: Rapid ethnography: time deepening strategies for HCI field. In: Proc. ACM Symposium on Designing Interactive Systems, pp. 280–286. ACM Press, New York (2000)
21. Myers, M.D.: Investigating information systems with ethnographic research. Communications of the Association for Information Systems 2, Article 23 (December 1999)
22. Nardi, B.: The use of ethnographic methods in design and evaluation. In: Helander, M.G., Landauer, T.K., Prabhu, P. (eds.) Handbook of Human-Computer Interaction, pp. 361–366. Elsevier Science, North-Holland, Holland (1997)
23. Reason, J.T.: Human Error. Cambridge University Press, Cambridge (1990)
24. Roth, E.M., Christian, C.K., Gustafson, M., Sheridan, T.B., Dwyer, K., Gandhi, T.K., Zinner, M.J., Dierks, M.M.: Using field observations as a tool for discovery: analysing cognitive and collaborative demands in the operating room. Cognition, Technology and Work 6, 148–157 (2004)

A Method for Identification and Representation of Business Process Deviations

Lilian Bitton Migon, Marcos R.S. Borges, and Maria Luiza Machado Campos

Graduate Program in Informatics
Computer Science Department, Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
lilianbitton@hotmail.com, mborges@nce.ufrj.br, mluiza@ufrj.br

Abstract. Nowadays, process management represents a fundamental initiative to provide competitive advantages to organizations. The ability to build and operate such processes can provide competitive advantages to these organizations. In this paper, we present an approach for elicitating and discovering problems in business processes that combines the technique of group storytelling with the theory of constraints. This article suggests that group storytelling allows the collection of knowledge to identify the gaps and deviations that exist in business processes, while the theory of constraints provides a language for representing them. We conducted and reported an experimentation of the proposed method in the petroleum field sampling process, with the support of a groupware tool.

Keywords: Group storytelling, Theory of Constraints, Business Process.

1 Introduction

Increasing competition in the business scenario has driven the need for organizations to enhance their productive differential against competitors. According to Gerstner [3], contributing to this differential is the skill with which organizations build, operate, or manage their business processes so as to perform their activities better than their competitors.

Considering the increasing diversity of relationships among organizations, customers, suppliers and competitors, there is a trend that business processes become more complex, amplifying the difficulties of organizations managing them [10]. One of these difficulties is the identification and appropriate treatment of existing problems and deviations of the defined process. It is common to adopt ad hoc solutions, which transfer the problems to other parts of the business process and, additionally, the adoption of strategies of "firefighting", where problems are solved as they arise, as emergency solutions.

In this scenario, in order to enable improvements on business processes management in organizations, we propose a method of identification and representation of deviations of business processes. A knowledge management approach and the group storytelling technique constitute the bases of this method for collecting knowledge on deviations, together with the use of the Theory of Constraints (TOC)

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 49–64, 2010.

for the representation of relationships between deviations, their root causes and main effects that must be prioritized and treated.

The method was tested on a scenario of field sampling of petroleum. To facilitate its application, a computational tool was developed and used to support the collaborative construction of stories and the analysis and structuring of knowledge contained in these stories.

The structure of this paper comprises a review of the group storytelling technique and its use associated with TOC for knowledge management. Then, we present the method of identification and representation of deviations of business processes, and the groupware tool built to support it, as a proposal for improved management of business processes. Subsequently we present some results and conclusions, as well as advantages and limitations observed from the application of the method and tool on the chain for petroleum sampling. Finally, we highlight the importance of collaborative work and use of groupware to improve the administration of business processes and knowledge management within an organization.

2 Problems in Business Process Management

The difficulties faced by organizations to manage their business processes are usually related to three main reasons.

The first reason is characterized by the rapid increase in demand for a product or service generated by the business. Because of the growth in demand, greater is the requirement for performing the process activities in a given space of time, and better should be the coordination between the areas that integrate the process.

The second reason is related to the perception of the deviations in existing business processes and the impact caused by them. Deviations in a business process, in the scope of this research, are alternative execution flows of process activities, altering the pre-established execution flow of this process. In fact, a deviation consists of an anomaly in the execution of a business process, contributing to an imperfect operation of the process and, consequently, to the appearance of changes in the characteristics of its products and services. Deviations in this scenario, therefore, limit the performance of the business process, preventing it from achieving its goal.

A deviation can be caused by an instrumental inefficiency or by other resources considered inadequate for performing activities; by human inefficiency, represented by the misunderstanding of the person responsible for performing a specific activity or control; or by process inefficiency in respect of its rules of conduct and procedures.

The perception of an existing deviation consists in identifying it and understanding its characteristics and its impacts, so that it can be properly treated. But often the deviations and their characteristics are not perceived by members of the process as factors that impact the final result. Thus, their occurrence is "tolerated" and their treatment by the organization is neglected.

In situations where the deviations are perceived as potential problems for the implementation of business process, what is found in organizations is an isolated treatment of these deviations. The treatment comprises isolated and punctual solutions that in general solve the problems individually, transferring them to other phases of the business process. It is like the problems exist independently, and individual solutions were appropriate for their treatment.

Isolated solutions are not integrated into the process as a whole, i.e. they are solutions that do not contain the involvement of participating areas and are unable to solve the problem at all stages of the process. This type of solution tends to transfer problems to other parts of the business process or not to solve the problems completely.

According to Cox and Spencer [2], another aspect that encourages the adoption of isolated solutions for deviations is the tendency of organizations in “firefighting”. In this way, the urgent problems are prioritized, without the adoption of an integrated solution for the treatment of the problem throughout the business process. So, the organization's ability to identify the central problems responsible for most of the deviations of the business process is not developed. In other words, efforts are not concentrated on solving the real cause(s) of problems identified.

Also according to Cox and Spencer [2], the central problems are those responsible for most of the anomalies identified in the process business. It means that concentrating efforts and plan actions on the resolution of central problems contributes to solve the majority of the existing problems.

In other words, efforts should be concentrated in answering an important question "What are the problems that must be focused to enhance the productivity of a business process?". Or, "Where may be obtained significant gains for the organization?".

For the identification of central problems and the appropriate efforts to solve them, the relations between the problems should be clear. Otherwise, the strategy of “ firefighting” is maintained. Summarizing, solving the problem in all phases of the business process requires the ability to identify problems considered central, whose solution will be prioritized by the organization.

The third reason is the lack of a systemic view of these processes. According to Senge [12], it can be characterized by the lack of visibility of the interrelationships and interdependencies between the participating areas of the business process, representing a fragmented operation of the process. Business processes for which a systemic vision is not adopted are configured in a fragmented way, into parts that act alone in activities execution and treatment of deviations, making it difficult to adopt adequate solutions for the business process as a whole. To identify and represent deviations for an integrated business process it is fundamental to enhance the perception of the interdependencies and relationships between the various problems and, therefore, the identification of the central problems cited above.

Considering that the knowledge about the deviations of a business process are distributed among its participants and it is especially derived from their experiences,, the investigation of deviations of business processes represents a potential field for the application of mechanisms for collaboration.

3 Group Storytelling and Theory of Constraints

In this section we present the theoretical approach adopted in this work, describing the two main theories that underlie the method developed for identification and representation of deviations of business processes: group storytelling technique and the TOC. The approach also applies concepts of business process modeling and knowledge management in organizations, but these topics are not discussed in this article.

3.1 The Group Storytelling Approach

The technique of group storytelling consists on using stories as a method of communication for knowledge sharing by a group of people. Another existing definition for group storytelling is: construction technique of stories in which "more than one person contributes, synchronously or asynchronously, locally or in a distributed manner, at various points in the process" [13].

By adopting a group storytelling approach we aim to motivate and to inspire those involved, making use of informal language and narrative that generally attract the interest of the participants, creating entertainment and fun during the process of structuring knowledge [7].

The technique of storytelling exists for thousands of years, with the aim of exchanging information and generating understanding. In organizations it has been mostly informally used to transmit knowledge. Only recently its implementation and recognition as a tool for sharing technical knowledge has been recognized, supporting a rapidly growing number of applications. The involvement of a group of people in the construction of a story can result in a more realistic story or, according to Perret, Borges and Santoro [10], can result in a more complete and solid story.

According to the Cambridge Dictionary [5] story is a description, either true or imagined, of a connected series of events. For Valle et al. [13], "stories are a narration of a chain of events recounted or written in verse or prose." Still, stories convey ideas, not rules of execution of tasks. Stories are memorable [10] and they can help the externalization of tacit knowledge, the kind of knowledge inherent to personal skills, systems of ideas, perceptions and experiences, and, therefore, difficult to be formalized, transferred or explained to someone else. The use of written stories seems to be the most appropriate, because writing is a way to convert tacit knowledge into formal knowledge [8].

The application of the group storytelling technique requires a fairly long time for a story to be created. The time devoted to the construction of stories should ensure that they present a satisfactory level of detail and quality and, at the same time, that their construction does not disturb the progress of the daily tasks of those involved.

One of the works on the use of the group storytelling technique was developed by Perret, Borges and Santoro [10], who discuss the importance of using a groupware to support the technique. The concept of groupware is supported by the personal interaction required for the sharing, creation and explanation of any content that needs interpretation, or in cases where knowledge is tacit. When this tacit knowledge can be registered and explicitly encoded to be shared later, groupware assume a central position at knowledge purchasing or collecting, combining, interpreting and disseminating. Santoro, Borges and Pino [11] used the technique in business process elicitation, trying to extract the workflow from the stakeholder´s stories.

As described above, the group storytelling technique allows the collection of knowledge in a group, stimulating the creation of a collaborative environment in which the stories are narrated and shared [11]. So, this technique supports the collection stage of the method for identification and representation of business process deviations, as described in the next section. However, the collected knowledge requires mechanisms that can be appropriately outsourced, allowing its analysis and understanding by a group of people. The TOC has been investigated as a strategy to provide formal structures to analyze the collected knowledge and also as a language for knowledge representation. It serves as a foundation for important stages of our method.

3.2 The Theory of Constraints

The Theory of Constraints (TOC) was created by Eliyahu M. Goldratt, as a management philosophy that aims to improve production systems [2]. A production system consists of a series of successive steps performed by different resources. All steps or operations must be completed in a specific sequence to obtain the final product. Among the resources that operate in the production system, there are those that limit the global production, representing constraints on system performance. As examples, we can consider physical constraints such as a faulty equipment or lack of material, or managerial constraints, such as procedures, policies and standards [1].

The management of the constraints is an approach that plans and controls the production and sale of products and services, aiming at a continuous improvement. This approach recognizes the powerful role that the constraint, or limiting resource, plays in determining the output or end product of the production system. Thus, the production system is viewed globally, which means to be regarded as a complex whole consisting of interdependent parts, which must act in coordination so that the final product can be generated.

Also, these parties should act in coordination to treat the identified problems, to reduce or even eliminate the impact that the constraints have, and thereby increasing the performance of the production system.

The TOC can be understood from three interrelated perspectives: logistics, performance indicators and thinking processes. The thinking process is also known as a method of solving problems. It is geared to answer three questions: What change? To change what? and How to bring about change? The logic of the thinking process is based on cause and effect relationships and critical view of reality, which seeks to know why things happen and not just that they happen.

There are five tools designed to support the thinking process helping to answer these three fundamental questions: Current Reality Tree (CRT), Evaporating Cloud, Future Reality Tree, PreRequisite Tree and the Transition Tree. According to Cox and Spencer [2], these tools can be used individually or interconnected logically. In our work, we emphasized the use of the CRT.

CRT is the tool chosen to represent the deviations along the business process. It is recognized as a diagram of effect-cause-effect (ECE). ECE diagrams are based on the scientific method to postulate a cause of an observed effect, then testing the cause by raising a second effect. The confirmation of this second effect provides evidence of the veracity of the postulated cause. The ECE diagram consists of a series of relationships postulated and confirmed that lead to the identification of a primary cause or root problem that explains most of the undesirable effects observed.

The purpose of CRT is the definition of the central problems found in a particular production system [2]. According to Noreen, Smith and Mackey [9] the logical links are indicative of sufficiency, i.e., the occurrence of certain undesirable effect or problem requires the occurrence of others (individually or both simultaneously). For Klein and DeBruine [6] and Cox and Spencer [2], the CRT provides mechanisms to:(i) identify the impact of policies, procedures and actions in the organization; (ii) communicate clearly and concisely, the causality of these policies, procedures and actions; (iii) identify the central problem in a situation; and (iv) allow the creation of an environment motivated by the problems.

4 A Method for Identification and Representation of Business Process Deviations

Business process management also contemplates managing the quality of process execution. Amongst other activities, quality management includes awareness of the deviations that exist along the process, helping to understand the differences between the as-is and the should-be process situations. For this, the deviations must be identified and represented so that the participants of the process recognize them as factors that have impact on the outcome of the process. This is a crucial step for the definition of future treatment plans and deviation prevention mechanisms.

The main contribution of this work is to develop a method of identification and representation of business process deviations and a groupware tool to support the method application. The expectation is that the representation of the deviations and their causal relations can help in the treatment of deviations and in the writing of organization standards and internal control procedures.

The method was developed mainly based on three approaches: the business process modeling, the thinking process of the Theory of Constraints and the group storytelling technique. It consists of three stages: collection of knowledge about deviations, analysis of the collected knowledge, and representation of deviations. Four actors participate in the stages of the method: storyteller, reviser, constructor and specialist. The last one should have experience on the business process and will be responsible for reviewing the knowledge after specific activities.

The group storytelling technique is the basis for the collection stage. The thinking process of TOC and its set of tools support all three stages of the method. They help to establish: (i) what additional information should be collected in conjunction with the stories told; (ii) what kind of verification, classification and correlation should be made over the collected information with the aim of identifying process deviations; and (iii) what mechanism or representation language could be used to record the identified deviations.

The modeling of business processes is for the collection stage because the stories should be collected for specific activities or parts of the process. The process model is also used during the presentation of identified deviations, at the representation stage. Thus, there is the assumption of the availability of the process model to identify and to represent deviations, guiding the collection of knowledge and being used as a representation language to support their understanding.

According to the TOC, a constraint is any element or factor that prevents a system to reach a better level of performance with regard to its goals, both physical or managerial [2]. When considering that it affects the perfect operation of the process, mostly in a negative way, the deviation may be understood as a factor that limits or restricts the operation of the process, acting as a constraint.

Still concerning the application of TOC to business processes, there is certainly a strong correlation between a business process and a production system. A production system consists of a series of successive steps performed by different resources. All steps or operations must be completed in a specific sequence to obtain the final product, goods or services [2]. This definition comes close to the definition associated to a business process, considered as a set of activities performed following a predetermined flow, where each activity has someone or something responsible for it, resulting in a product that can be a material good or a service [3]. Figure 1 presents the method with its stages and activities, actors and products.

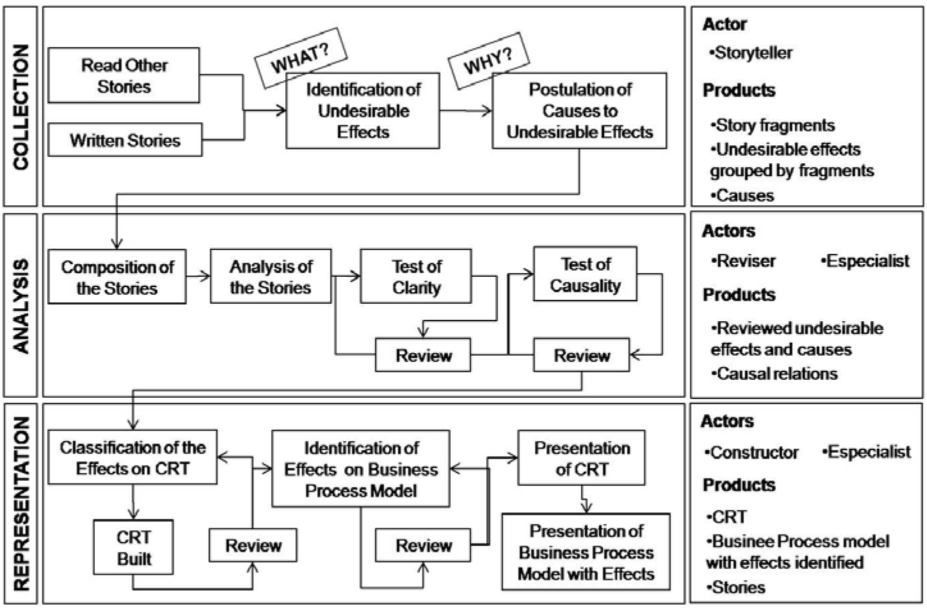


Fig. 1. Method stages and activities

The identification of business processes deviations can be facilitated by the group storytelling technique as this promotes an environment for knowledge gathering and sharing that is crucial to stimulate the externalization and comprehension of facts associated to these deviations. In this environment, the deviations can be extracted from narratives, corresponding to experiences and situations described by the participants. The narratives correspond to fragments of stories that can be organized, so that to correlate causes and undesirable effects associated to abnormalities observed in the business process. An undesirable effect in this context corresponds to a problem or a failure that have an adverse impact on the functioning of the business process and it can be reported directly or indirectly in the story fragments. It corresponds to answering the question "WHAT?" about the reported problem. The cause indicates an explanation for the occurrence of a certain undesirable effect, corresponding to answering the question "WHY?" about the reported problem.

To make these concepts of undesirable effect and cause clearer, let us imagine the following situation: a plastic box that stores a sample of petroleum collected in a producing well is subjected to high temperatures during transport. According to the regulations for petroleum samples, there is a range of temperature adequate for petroleum properties not to be altered in transit. Thus, if a sample is exposed to temperatures outside the recommended range, then there could be a change on its physical-chemical characteristics, what constitutes an undesirable effect identified during the transport process of petroleum samples. In this scenario, a valid cause for the undesirable effect can be identified, that is the exposure to high temperatures.

After the collection stage, the analysis stage of the collected knowledge is initiated. At this stage the stories about the deviations are constructed and tests of clarity and causality are performed in the undesirable effects and causes collected. The test of clarity is responsible for reviewing the effects and causes, ensuring that they are unique and contain just one meaning. The test of causality aims at establishing causal relations between undesirable effects and causes that were reviewed by the test of clarity. These relations are characterized as being of type IF *cause (s)* THEN *effect*. Returning to the example of the plastic box used to store petroleum exposed to high temperatures during its transport, resulting in physical and chemical changes in its composition, the following causality could be established:

IF *a plastic box that stores a sample of petroleum collected in a producing well is exposed to high temperatures during transportation*, THEN *occurrence of change on physical and chemical characteristics of this sample of petroleum.*

Following the construction of causal relations, there is the stage of representation of deviations. At this stage the CRT are built, containing the causal relations identified and the model of business process with the undesirable effects, identified by activity. Hence, the method generates, besides the CRT, another diagram representing deviations in the model of business process.

The proposed method constitutes a relevant solution for three reasons: (a) the first is to detect deviations from the business process for the entire process, helping the future drafting of appropriate rules and procedures geared to the needs of the whole process and not only the needs of specific areas, and the writing of plans for the treatment of deviations; (b) the second aspect is to propose a model for representing deflection of business processes, highlighting features and relations of cause and effect, which could not be understood before in their entirety; and (c) the third aspect is related to the identification of the central problems of the business process, which are responsible for most of the deviations identified and therefore should be properly understood and prioritized by the organizations.

4.1 A Groupware Tool to Support the Method

The groupware tool developed supports the collection and analysis stages and partially the representation stage of the method. It allows: (a) the recording of stories in groups and the association of these stories to groups of undesirable effects and causes; (b) the analysis of the knowledge through clarity and causality tests, specially the activities of grouping effects and causes and creating causal relations; and (c) the classification of undesirable effects to built the CRT and presentation of the products of the method. Figure 2 shows the screen of a story recorded in group about a process deviation and its undesirable effects and causes.

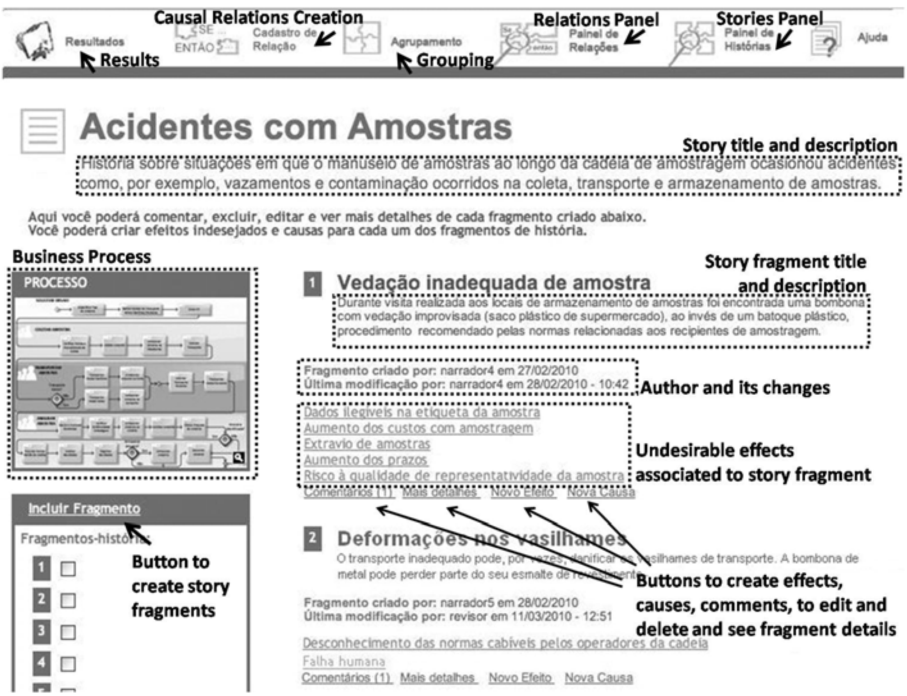


Fig. 2. Story fragments about process deviations and the list of undesirable effects

Soon as the stories are analyzed and completed, the undesirable effects and causes are checked through the test of clarity and test of causality. During these tests, the groupware tool enables the grouping of effects and causes with the same meaning, and the separation of causes and effects that contain more than one meaning. After the test of clarity, the tool supports the causality test, allowing the registration of causal relations between undesirable effects and causes that are after displayed in the CRT generated, in the final of representation stage.

At the end of these tests, the analysis stage is completed and the representation of deviations identified starts. In this stage, the tool classifies the undesirable effects and causes in one of the three possible positions that can fill in the CRT: top, for the purposes of representing the central problems of the business process studied; basis for the basic causes for the identified problems; and problem area, which includes all other effects and causes identified. From this classification, the CRT may be generated, representing the undesirable effects, causes and the causal relations between them.

The tool has two panels that allow the visualization of knowledge collected about the stories, their fragments, undesirable effects, causes and causal relations established. Information displayed on panels is the status of the activities of undesirable effects and causes, enabling to determine which effects and causes were reviewed, grouped and related by causal relations. Through these panels, the user, at any stage of the method, can trace to which story or fragment a specific effect or cause belongs.

The user can also find out who is the author of a fragment, effect, cause or causal relation, or monitor clarity and causality tests. Figure 3 shows one of these panels, called Stories Panel.

To carry out the activities of the method described above in the groupware tool, different profiles of users were designed, representing the roles of storyteller, reviser and knowledge builder.

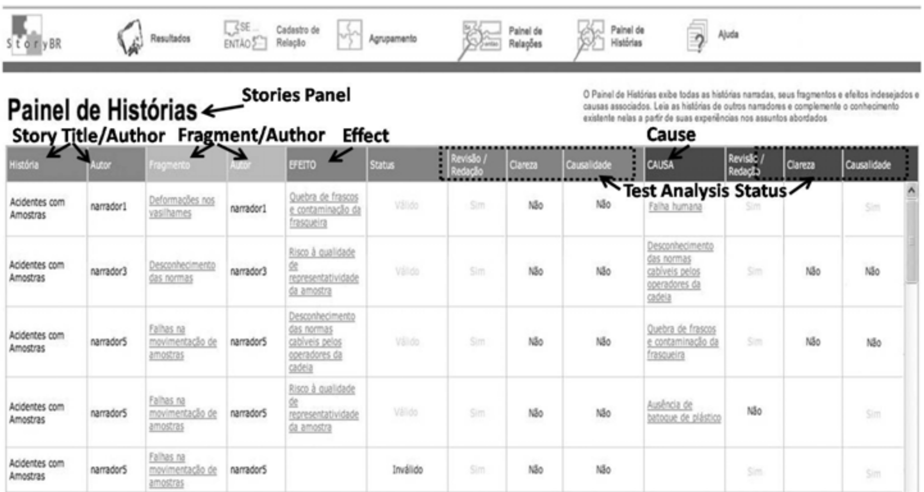


Fig. 3. Stories Panel and traceability of knowledge collected

5 Application of the Method for Identification and Representation of Business Process Deviations

The method and its supporting tool have been applied to the field of study of sampling of petroleum. The growth of the petroleum industry in Brazil, both in volume and geographical extent, brought a large increase in the number of samples handled by companies operating in the sectors of exploration and petroleum production sectors. The observation points toward the loss in quality and the possible loss of control in the processes of the chain of petroleum sampling, mainly for lack of adequate controls and efficient indicators to the new dimension of this activity.

The chain of petroleum sampling represents an example of business process where the organization may face difficulties to manage it, according to following reasons: (a) accelerated growth in demand for petroleum sampling in recent years due to the expansion of exploration activities and production of petroleum; (b) the links in this chain are weakened because the deviations are identified for each area independently and their treatment is done individually; and (c) the chain consists of several distinct areas of knowledge, what difficult the systemic vision of the chain.

According to the method, the model of business process must be displayed during the recorded of the story by the storyteller, so that he can associate a fragment with one or more business process activities.

For the experiment conducted, we selected members of the organization and individuals involved in studies of non-conformities in the chain of petroleum sampling. Eight of these people assumed the role of storytellers and recorded stories about deviations. Another two people were selected to assume the roles of reviser and builder of the method, and acted in the stage of analysis and representation of the method, respectively. The storytellers recorded a total of five stories, which report major problems in the chain of sampling. These stories have been described in a total of 29 fragments, to which 23 undesirable effects and 13 causes were identified. For each cause registered in the collection stage, it was assigned an undesirable effect extracted from a given fragment or piece of story. Thus, to assign a cause to an undesirable effect, the storyteller indicated the existence of a causal relation of the type IF cause THEN effect. So, 13 causal relations were created from the set of causes and effects collected.

The knowledge collected was analyzed by the reviser, with the application of the test of clarity in undesirable effects and causes. After applying this test in which the undesirable effects and causes might be reviewed, we obtained a new set containing 20 undesirable effects and 17 causes. It was noted, therefore, a 13% reduction in the number of undesirable effects and a 31% increase in the number of causes.

The reduction in the number of undesirable effects was caused by the elimination of redundancy that exists in these effects. The level of redundancy of 26% means that about ¼ of effects had the same meaning of other reported effects.

In relation to the 31% increase in the number of causes, if this measure were analyzed together with the redundancy level of 46% obtained for the causes registered, we can conclude that storytellers had difficulties in associating causes to effects identified. This forced the reviser to create new causes for the existing undesirable effects and causal relations.

The test of clarity also separated the effects and causes which had more than one meaning and, therefore, should be separated into one or more effects and causes. 16% of the causes and effects needed to be separated into single effects and causes.

After the test of clarity, the reviser conducted the causality test, in which were created new causal relations. From an initial number of 13 causal relations registered by the storytellers, were created more than 39 causal relations.

The low number of causes and causal relations can be attributed to some factors: (a) members of an organization more easily identify the problems they face in their work environment than the causes of these problems; (b) the functionality of registration of causes in groupware tool may not have developed in an appropriate way to stimulate it; (c) the causes for the effects presented do not belong or are not under the responsibility of the operating area of the storytellers.

One factor that explains the low number of causal relations is that the collection stage does not request explicitly the creation of causal relations by storytellers. The activities demanded by storytellers are storytelling, the identification of undesirable effects from them and the association of causes for the undesirable effects created. Moreover, in the analysis stage this request is explicit as part of the causality test. Therefore, it is expected the reviser instead of the storyteller to create causal relations, to allow built the CRT in the next stage of the method.

The Current Reality Tree was built from the set of effects and causes and causal relations created. Figure 4 shows a part of the CRT originated a kind of sub CRT.

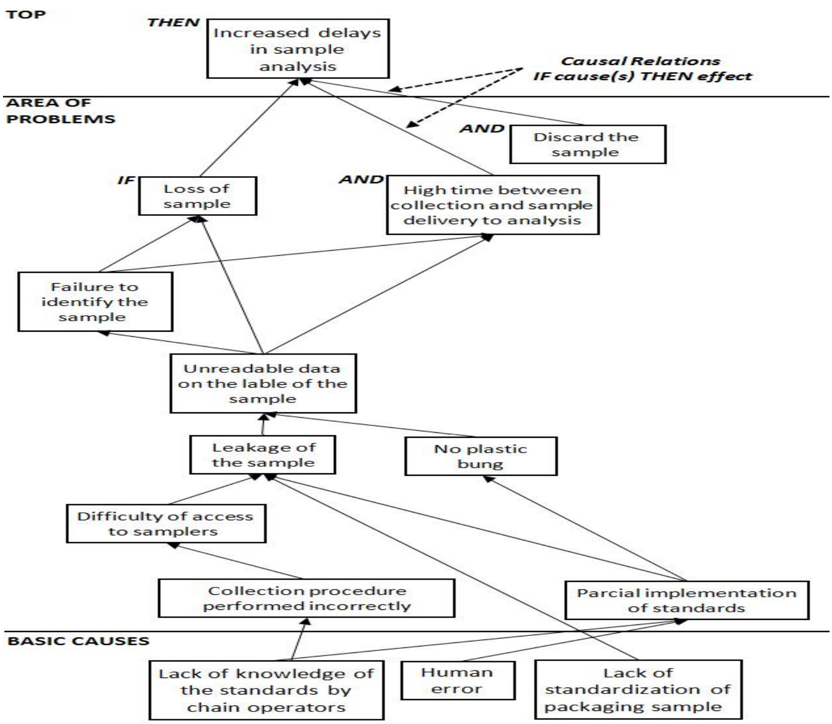


Fig. 4. Sub CRT generated after applying the method

The relationships between problems can be viewed during the analysis of CRT. The effects at the top are responsible for most of the causes and effects described in the CRT, for each path leads directly or indirectly to them. Therefore, these problems should be prioritized by the organizations in the moment of their treatment plans. CRT shows a kind of diagnosis to the problems situated on the top, indicating what are the causes and other effects related to each of the central problems that must be addressed together to ensure that the central problem will be solved. For example, *increased delays in sample analysis* is a central problem related to causes displayed on CRT. One of the causal relations built through story fragments is:

IF *loss of sample*, *high time between collection* AND/OR *sample delivery to analysis* AND *discard the sample*, THEN *increased delays in sample analysis*.

The causes at the base of CRT represent those which are not under the responsibility of the process participants, or causes they are able to address; only reduce their impacts. For example, the basic cause “human error” is always present in the execution of any activity, and whose impact cannot be completely eliminated.

We noticed that 39% of the undesirable effects were created for story fragments from other storytellers and 38% of the causes were associated to effects of other storytellers. These numbers demonstrate a degree of indirect interaction.

The next section presents a discussion about the level of contribution and collaboration as measured by the number of fragments inserted and the comments to other participants´ fragments. Then, Section 5.2 addresses the complexity of the method by analyzing the level of intervention required by the reviser. The data used in the analysis was obtained by the data generated during the experiment.

5.1 Levels of Contribution and Collaboration

The level of contribution in the record of knowledge was measured by the number of story fragments, undesirable effects and causal relations created by storytellers. It was measured on absolute value and average.

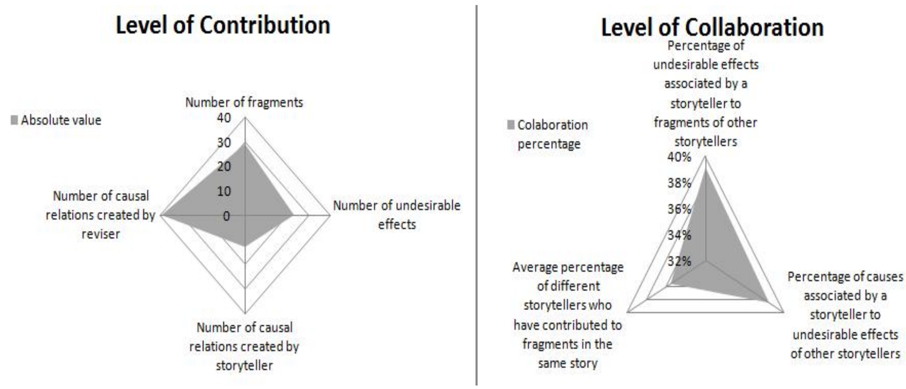


Fig. 5. Level of contribution and level of collaboration of the method

The level of collaboration in the recording of knowledge measured the degree of associated knowledge recorded by others. It was measured by: the rate of undesirable effects associated by a storyteller to fragments of other storytellers; by the percentage of causes associated to undesirable effects of other storytellers; and by the average rate of different storytellers who have contributed to fragments in the same story.

The individual contribution and collaboration varied between storytellers. It was noted that participants with some background or experience with collaborative environments had higher levels of contribution and collaboration. The main difficulty was observed in contribution to attribution of causes to undesirable effects. Figure 5 below shows contribution level and collaboration level taken.

5.2 Levels of Clarity and Intervention

Level of clarity and intervention of the reviser should be explained together, because they influence each other, and have impact on the complexity of the method. The lower the level of clarity is, greater is the intervention required to generate a representation. Consequently, greater is the complexity of using the method.

The level of clarity indicates how knowledge is unique and is structured to its representation. It was measured by the rate of redundancy of undesirable effects and causes; and based on the number of causes and effects that needed to be separated.

The level of intervention indicates the rate of reviser's change on the collected knowledge. It was measured by the percentage change in the number of undesirable effects, causes and causal relations; and by the amount of changes in the associations between business process activities and fragments.

The low number of causes assigned by storytellers demonstrated the difficulty in attributing causes to problems, generating a greater need for intervention, mostly in establishing causal relations to build the CRT. In more complex processes, it is expected that the level of clarity would be even smaller, increasing the intervention and complexity of using the method.

Another result of the experiment indicated that the rate of redundancy may show a certain alignment among storytellers about the business process deviations. In other words, more than one person identified the same undesirable effects to business process. Figure 6 shows the level of clarity and intervention in the experiment.

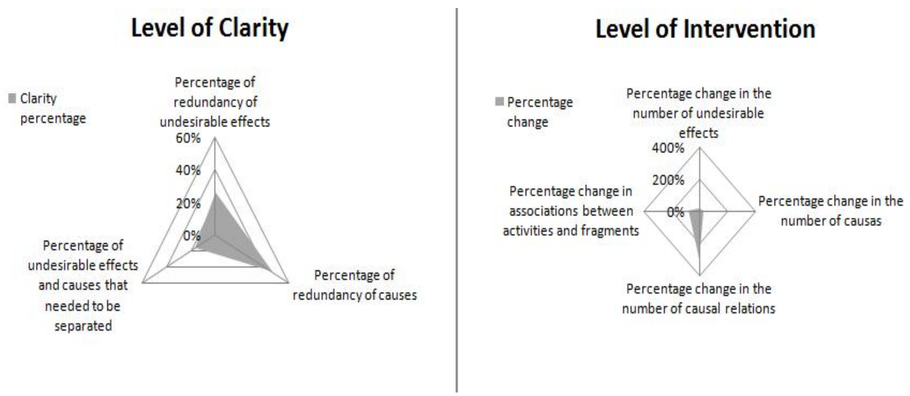


Fig. 6. Level of clarity and level of intervention of the method

6 Conclusions

The paper presented a method for identifying deviations of business processes. By deviation we meant any variation of the prescribed process to deal with situations that occurred during the process enactment. The method assumed a collaborative approach of the actors involved in the process. It is supported by a group storytelling technique to collective recall stories describing scenarios that led to deviations. It also makes use of the Theory of Constraints to support the analyses of such scenarios. The systematic knowledge building from capture to diagnosis generated a path that allowed tracking the reported problem to its impact. As a result, besides the identification of the deviations the method promotes a systemic examination of the business process. Also, motivating people to contribute to the identification of problems may increase the future commitment to their solution.

A groupware tool was developed on the top of an existing group storytelling tool [10] extending its capability to support the next steps of the method. The groupware tool makes possible an asynchronous and distributed capturing of knowledge as well as sharing this knowledge among the participants. The externalization of this knowledge provides an important mechanism to promote organizational learning and a continuous improvement cycle. The approach assumes a constructive attitude of the participants.

Of course, no collaborative tool can generate this attitude; it can only facilitate the collaboration to manifest. It is not different with the tool we developed.

To understand and evaluate the effectiveness of the method, a case study was conducted on a Brazilian petroleum company. The study provided interesting insights about the method and on the use of the supporting tool. In general, the method fulfills its promises as it was able to capture stories reporting problems and their causes. The analysis based on the theory of constraints make explicit the basic causes as well as their relationships with the reported problems and causes.

Some limitations were identified in the method and on the groupware tool that support it. These limitations are indicative to future work. The groupware tool should increase its support to the representation stage of the method and reduce the ambiguity of the knowledge collected. Alternatives to explicitly represent causal relations by storytellers are desired. For instance: (a) to include a step in the method for revising the CRT, generating causal links between the original CRT and the new version and (b) to add an activity to establish causal relations between the story fragments in the capturing step.

We also identified some factors that influenced the application of the method: The first was the time assigned to the collection stage. Second, the participants' ability to tell their stories resulted on very different narratives. Finally, the level of cooperation in the workplace was overestimated. We cannot assume it as a natural attitude, which means it should be stimulated in some way before the exercise.

Depending on the complexity of the business process in which the method is applied, mechanisms to stimulate the adoption of systematic view of its participants may be incorporated to the method to contribute to identify causes for the problems. Consequently, it may reduce ambiguity and the need for intervention of the reviser during the representation stage of the method.

We plan to apply the revised method and a new version of the groupware tool in different business processes and for longer periods. Besides, for the new set of experiments we plan to use a control group using a typical interview approach. By doing this we intend to obtain additional data to confirm, or not, our claims on the effectiveness of the method.

Acknowledgments

This work was partially supported by CNPq (Brazil) grants 304252/2008-5 and 311454/2006-2, and MCT/FINEP grant 01.05.0851.00 Ref. 3542/05. The authors would like to thank those who participated in the experiments.

References

1. APICS: The Association for Operations Management Dictionary, http://www.apics.org/Resources/APICSDictionary.htm
2. Cox, J., Spencer, M.S.: The Constraints Management Handbook. CRC Press, Boca Raton (1998)
3. Davenport, T.H.: Mission critical: realizing the promise of enterprise systems. Harvard Business School Press, Boston (2000)

4. Gerstner, L.V.: Who Says Elephants Can't Dance? Harper Business, New York (2002)
5. Cambridge Online Dictionary, `http://dictionary.cambridge.org/`
6. Klein, D.J., Debruine, M.: A Thinking Process for Establishing Management Polices. Review of Business 16, 31–37 (1995)
7. Lelic, S.: Fuel Your Imagination - KM and the Art of Storytelling. Knowledge Management Magazine 5 (2001)
8. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, New York (1995)
9. Noreen, E., Smith, D., Mackey, J.T.: The Theory of Constraints and its Implications for Management Accounting. North River Press, Great Barrington (1995)
10. Perret, R., Borges, M.R.S., Santoro, F.M.: Applying Group Storytelling in Knowledge Management. In: de Vreede, G.-J., Guerrero, L.A., Marín Raventós, G. (eds.) CRIWG 2004. LNCS, vol. 3198, pp. 34–41. Springer, Heidelberg (2004)
11. Santoro, F.M., Borges, M.R.S., Pino, J.A.: Acquiring knowledge on business processes from stakeholders' stories. Advanced Engineering Informatics 24, 138–148 (2010)
12. Senge, P.M.J., Spencer, M.S.: Fifth Discipline Fieldbook. Doubleday, Massachusetts (1994)
13. Valle, C., Raybourn, E.M., Prinz, W., Borges, M.R.S.: Group Storytelling to Support Tacit Knowledge Externalization. In: 10th International Conference on Human-Computer Interaction, pp. 1218–1222. HCI International, Crete (2003)

Computer Supported Reflection of Good Practice

Till Schümmer, Martin Mühlpfordt, and Joerg M. Haake

Cooperative Systems, FernUniversität in Hagen, Universitätsstraße 1, 58084 Hagen
{till.schuemmer,martin.muehlpfordt,joerg.haake}@fernuni-hagen.de

Abstract. Knowledge exchange and collaboration in problem solving processes are important factors for learning organizations. Involving practitioners with joint interests in discussions of shared challenges and solutions is an important step in the identification and sharing of good practice. A process facilitating such reflection workshops for practitioners and a tool supporting this process are introduced. The approach is based on computer supported collection of challenges and solution ideas. It uses semantic nets for identifying and connecting practitioners with related interests. First trials show promising results.

Keywords: collaborative reflection, good practice, group facilitation, process support, knowledge sharing, organizational learning.

1 Introduction

The exchange of good practice is a central element of learning organizations. Unlike small organizations featuring social relationships between most employees, large distributed organizations face new challenges in implementing such exchange since practitioners are not aware of experiences of remote colleagues. Thus, innovations and mistakes are both repeated in several locations. Here, computer support may facilitate exchange, learning and reuse of challenges and proven solutions across organizational boundaries.

Learning in a learning organization is often performed in cycles, following the knowledge spiral described by Nonaka and Takeuchi [9]: new forms of practice are developed and tried, experiences are described (externalized) and combined with other practices. The combined knowledge is applied and is internalized by other practitioners. Kolb's experience based learning cycle consisting of four steps follows a similar direction: (1) concrete experience, (2) surveying and reflection on the experience, (3) forming abstract concepts, and (4) active experimentation with newly acquired knowledge [6].

Based on this model [14] presents a process for learning social practice in large nongovernmental organizations (NGO). The form of describing externalized practice knowledge plays a central role in this approach: good practice is described as practice patterns resembling design patterns [2]. Practice patterns relate actions in a good practice to the challenge addressed by the practice. When creating such a pattern the practitioners reflect on their experiences and form abstract concepts (cf. [6]). Concrete applications of the pattern (i.e. experimentation, see above) are supported by giving practitioners new impulses for addressing concrete challenges.

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 65–80, 2010.

This paper presents the design and use of a process and tool facilitating workshops of practitioners when creating, retrieving, and relating practice knowledge in the above described way. During a computer supported workshop practitioners are motivated and prompted to describe observations based on their concrete experiences, and to reflect about related challenges and solution ideas. Practitioners relate their experiences according to the experience's subject. This supports that practitioners with shared interests identify each other, discuss their experiences and become involved in a shared reflection process.

The remainder of the paper is organized as follows: the next section discusses existing approaches for supporting such a knowledge exchange process based on a concrete and realistic scenario. Section 3 presents our solution consisting of a process and a collaborative system that supports relating contributions and practitioners by using a semantic net approach. Section 4 describes the implementation, and Section 5 reports experiences with using the process and tool in a real workshop. Finally, Section 6 concludes the paper with a summary, comparison to related work, and plans for further work.

2 Problem Analysis and Related Work

We begin by presenting a real world scenario for good practice identification and sharing before we discuss current approaches for supporting such a scenario.

2.1 Problem Analysis

The PATONGO project (www.patongo.de) aims at improving knowledge exchange among practitioners in the Evangelical Church in Germany through a process facilitating retrieval of challenges and matching solution ideas. The organization of such a process is challenging due to the large number of users (250.000 employees and 1 Mio. voluntaries), high degree of geographical and organizational distribution into parishes, and a broad spectrum of areas of practice. Both, priests as well as voluntary parish leaders need to act and lead in new areas of practice.

While a large part of work is done in local parishes sometimes opportunities for personal meetings and relationships across organizational borders occur sporadically. Meetings among leaders often serve the purpose of informal exchange of experiences. However, due to the wide variety of challenges, practices, and topics people do not always meet others with matching experiences. Thus, it is felt that some coordination ensuring that contacts among complementary people could be established would be helpful. From this, we can identify some challenges that are typical for large distributed organizations featuring a diverse set of challenges and practices:

- Employees and voluntaries often face new challenges, and develop new local good practice.
- While an exchange of good solutions for a concrete challenge may be valuable in a specific situation such exchange rarely happens due to the high coordination overhead.

- Employees and voluntaries often know only local colleagues. Exchange across geographical and organizational boundaries rarely happens although it would be beneficial in order to avoid repetition of mistakes or reinventing the wheel.
- Even when exchange happens at global meetings it remains difficult to find others with matching experience and become engaged in discussions. Thus, new ideas are generated during discussions at random if at all.

Our hypothesis is that collaborative learning support may help to lower these barriers for knowledge exchange. More specifically, we claim that such a learning environment may help to bring practitioners together with similar experiences and interests so that they may benefit from each other.

2.2 Current Approaches

From a collaborative learning perspective two central requirements must be met in order to facilitate knowledge exchange of the kind discussed above: Firstly, in the collaborative process practice knowledge must be externalized respectively explicated, discussed, and varied in such a way that new solution ideas are created for meeting a challenge. Secondly, practitioners looking for solutions to a challenge must be brought in contact with other practitioners experienced with addressing the challenge.

Story telling approaches may be used for reflection about your own practices, in the sense of the reflective practitioner [13]. A story based process for creating practice patterns was created in the PLANET project [8, 4]. Case stories contributed by practitioner are first discussed with other practitioners with similar experience or background aiming at identification of common properties of a solution, context, problem and solution. As a result prototypical design patterns are created and later integrated in a pattern language [2].

In project management, project retrospectives [5] are an established means to detect and document good practice and challenges at certain milestones. However, while project retrospectives usually aim at learning about interaction in the project, our target is organizational learning across organizational boundaries.

Many commercial group facilitation systems such as GroupSystem's ThinkTank[1] or teambits:workshop[2] support group brainstorming and clustering of contributions. Such systems can be used to collect ideas present in a room. ThinkTank also supports the commenting of contributions. This is already a first approach towards our problem of collaborative reflection. However, we see the need for supporting the connection of ideas better in order to unveil the relationships of practices and practitioners. In addition, the process of learning through reflection can benefit from explicit content structures, such as the distinction between challenges, ideas and comments.

Specific creativity techniques, such as the 6-3-5 method [12], go one step further in this direction where idea collection is embedded in a structured group process. In the 6-3-6-method, members of a group each write down three ideas to address a challenge. These written notes are then passed to other members who augment them with their own ideas. The PREP system [10] supports this method with a collaborative

[1] http://www.groupsystems.com
[2] http://www.teambits.de

system. Group members create ideas in the system, which forwards them to another group member. Usage experience from successful teaching and learning applications was reported. This approach does not directly fit our needs: Firstly, it supports only idea creation but not the reflection about experiences. Secondly, in analogy to classical brainstorming the method does not support discussion about the contributions. However, practitioners have a big need for discussions, especially when solution proposals are based on real experience. Atizo (http://www.atizo.com) is an example of an asynchronous distributed system that combines collection of ideas with their discussion.

Semantic analysis of questions and user profiles can be used to distribute only relevant questions and ideas to users. Instead of forwarding questions simply to the next person as in the 6-5-3 method a computer-based system may determine the person with appropriate background knowledge. Social recommender systems for expert finding [1] provide mechanisms that can be used to form groups of people with complementary experiences, based on user profiles. Such mechanisms may be used to redirect questions to matching users. Aardvark (http://vark.com) is an asynchronous system that classifies questions using tags and propagates them to users with matching experience. Continuing this line, the approach presented in [11] uses semantic nets or topic maps to aid assigning questions to users providing answers in the e-learning domain. Compared to statistic approaches, such approaches may produce better quality for even a small number of contributions if those are placed in a semantic net.

In summary, many systems and approaches for collecting ideas addressing a challenge have been developed in recent years. Usually, these are designed for distributed and asynchronous interaction, and they ignore the relationship between reflection and innovation. This is where our approach makes its contribution.

3 The PATONGO-Storm Approach

The PATONGO-Storm approach aims at explicitly supporting the reflection and idea generation process described before. It consists of two components: a process model regulating the social process of conducting an effective reflection and idea generation workshop, and technology supporting this process.

3.1 A Process Model for Reflection Workshops

In our proposed process, we exploit the experience of people with face-to-face collaboration. Thus, we assume that practitioners in an organization meet occasionally, e.g., at an annual meeting or at a congress, and use this opportunity for running a reflection workshop. During such meeting, we use a 2 hour slot where participants interact using the PATONGO-Storm approach.

Fig. 1 illustrates the proposed process. It consists of five phases: initialization, collection, transition, discussion, and summary.

A facilitator begins with an introduction, explains the process to the participants, forms groups consisting of 2-3 people, and explains the tools to be used (see next section).

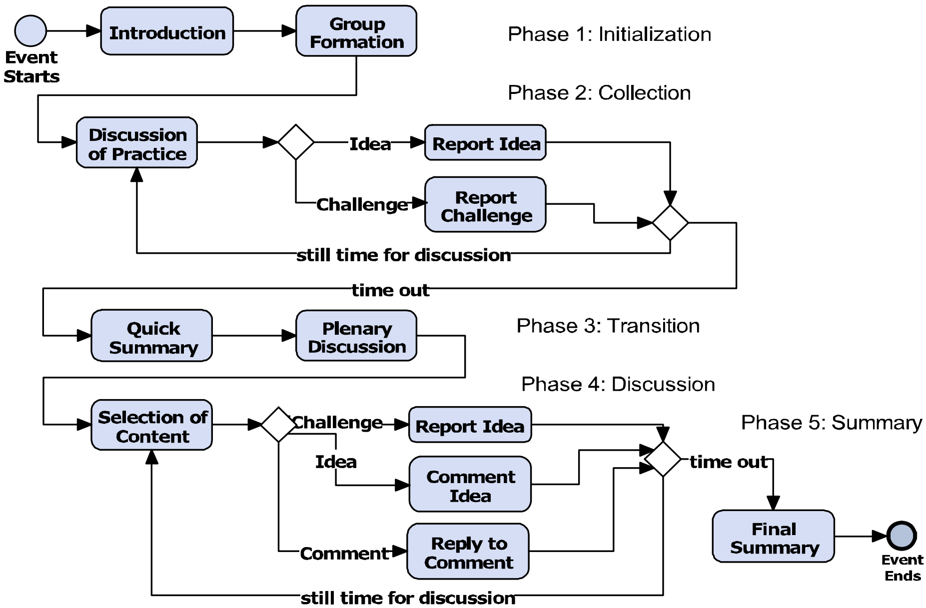


Fig. 1. Interaction process in PATONGO-Storm

The second phase is limited to 45 minutes and starts with an intentionally general starting question (e.g., "Report in your group about a successful project of last year.") used to initiate a face-to-face discussion within each group. This is a form of direct verbal story telling. Discussion is recommended to be limited to one minute per report, after which the group should be able to agree on whether they want to define the core idea behind the practice or specify a challenge becoming visible in the report. In both cases, they should be able to define the title and describe the spirit of the idea or challenge in one or two sentences. In addition, they have to specify keywords. Until time is over groups can continue to discuss and create new ideas or name challenges.

After the time assigned to the second phase, the facilitator closes this phase and enters the third phase of transitioning from intra group work to inter group work taking approximately 10 minutes. For this purpose, the facilitator shows a summary of the contributions so that participants get an idea of their amount and diversity. In addition, the facilitator selects up to three ideas and asks their contributors to explain them. Then, the fourth phase is introduced.

In the fourth phase ideas and challenges are discussed for another 45 minutes. For this purpose, groups with similar interests are determined and ideas of matching groups are shown to each other. Now, each group can select a contribution of another group and respond to it: comments can be added to ideas; ideas can be added to a challenge; or a comment of another group to an idea of this group can be answered.

After the end of the fourth phase, the facilitator uses the last phase of approximately 20 minutes to summarize the results. This process leads to group knowledge externalization and sharing behavior taking place in the 120 minutes time frame of the reflection workshop.

3.2 Technology-Support for Reflection Workshops

In order to support the execution of the above process in an efficient manner, we developed a process execution environment providing support for the stakeholders involved in the process: facilitator, groups, and group members. For this purpose, a networked computer is provided to each group and to the facilitator. The system provides a user interface to the facilitator and each group for performing their activities in each phase. At the beginning of the workshop in the initialization phase, each user registers herself as a team member on her group's computer. Afterwards, the group can assign the names of creators to ideas, challenges, and comments (i.e., creators can be one or more individuals). Once this is done, the system displays a pause sign to redirect the participants' attention to the facilitator.

Once the facilitator starts the second phase, the instructions and actions possible in this phase are displayed on each group's computer. Group members can now discuss their practices, select the type of contribution they want to make (i.e. idea or challenge), and contribute their input to a shared information space.

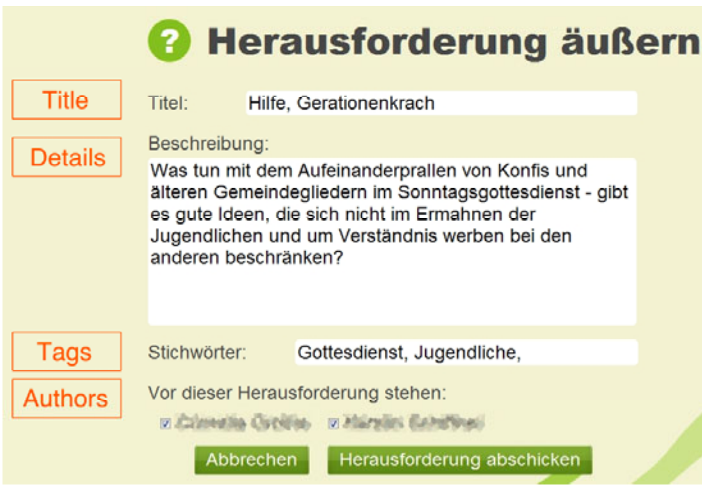


Fig. 2. Expressing a challenge in PATONGO-Storm

Figure 2 shows the German user interface for creation of a challenge. Note that the labels in the left column are only added in the figure for explaining the different sections in the German user interface. The interface was intentionally kept simple: besides a title and textual details, the group only inputs tags (assisted by an auto-complete mechanism) and decides whether or not the individual group members want to identify themselves with the challenge. The system lists all group members' names and the group members can uncheck their name in case of disagreement about the challenge.

Until time is over the groups can continue to discuss and create new contributions. When time is getting close to end of the second phase, the facilitator asks participants to come to an end. Then, the computer screens display a pause sign and the third phase (transitioning) is in progress. The facilitator is now using the system to project a summary of contributions on a screen. In our validation case (see section 5), the system showed the titles of contributions in creation order flying in over a map of the country to give a sense of the amount and variance of contributions (Fig. 3). The numbers at the top left side of figure 3 denote the number of ideas, challenges, ideas for other group's challenges, and comments created.

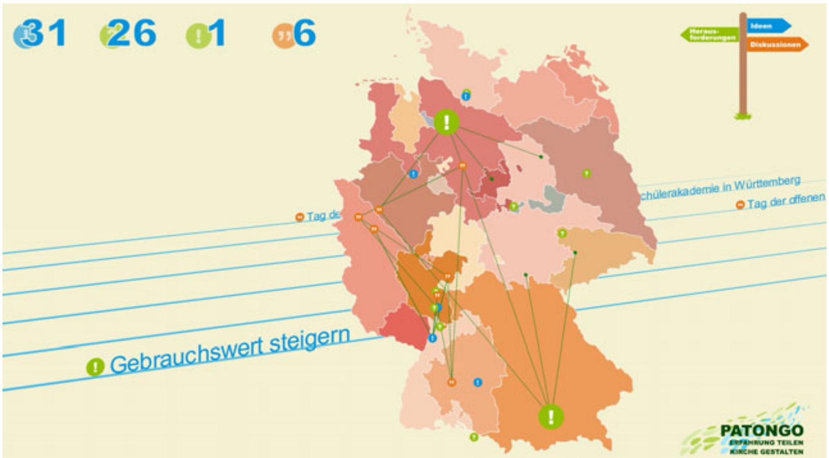


Fig. 3. Map visualization of ideas and challenges used in the transition phase

The map visualization helps the participants to understand the potential for connecting ideas and people. Each contribution is positioned at the contributor's home coordinates and connected with other ideas or challenges that are direct replies to the contribution (note that the visualization was also used in the final phase, thus Fig. 3 also contains comments that were added in the fourth phase).

After the facilitator introduced the fourth phase, this phase is started and ideas and challenges are discussed. Again the instructions and actions possible in the fourth phase are displayed on each group's computer. However, now such actions focus on collaboration among peers sharing similar interests. Based on the keywords used by the group the system determines its focal concepts. Now, related topics or concepts and contributions of other groups are computed. For example, if one group created challenges or ideas related to "prayers for young people" then another group with a focus on "prayers for catechumen" would be a matching discussion partner. Ideas of matching groups are then offered to each group by showing them on the computer screen of the respective group. Fig. 4 shows an example of how ideas and challenges from other groups are presented to this group on the section page of the fourth phase. Reactions of other groups to the ideas and challenges created by this group are presented on the selection page, too. In this way, the system supports awareness on reactions from other groups and helps to spark a computer-mediated dialogue.

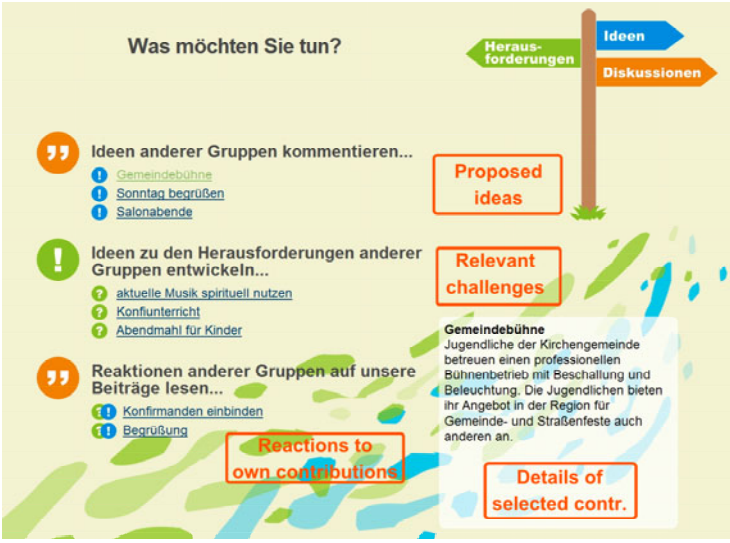


Fig. 4. Contribution selection page in phase 4

When the time of the fourth phase is finished, the facilitator summarizes the results in the final summary phase. The system supports this summary by presenting contributions on a map using the location of the respective creators (using the same map visualization as in the transition phase, see Fig. 3). This provides participants with an idea about the number of contributed challenges and ideas, and about possibilities for networking with other locations of the organization.

4 Implementation

The PATONGO-Storm system is a web application implemented using Ruby on Rails. Fig. 5 shows the distribution architecture of PATONGO-Storm. Besides the Ruby-on-Rails-based web server, PATONGO-Storm uses an additional synchronous communication channel based on the Juggernaut Messaging infrastructure in order to enable central moderation or facilitation of phase transitions, so that all groups transition from one phase to another.

Always two or three practitioners share one client computer that displays PATONGO-Storm in a standard web browser. The presentation computer is connected to a projector so that the contribution map can be shown in phases 3 and 5.

We use the K-Infinity software (www.i-views.de) to place contributions in a semantic net according to their keywords. This allows us to suggest matching contributions to users in the fourth phase of the process. In addition to keywords already included in the current semnatic net users can use new tags. These tags are concurrently introduced in the sematic web by knowledge engineers and editors. Directly after a contribution was stored that includes new tags, the knowledge engineers are informed and asked to relate the new tag with the other tags that are already part of the semantic net. They use a synchronous graphical groupware application calles Knowledge Builder for this task since they need to reach a shared understanding of additions to the semantic network in realtime.

The knowledge engineers also survey the collected contributions and together with the facilitator select three ideas to be introduced in a plenary at the end of the first phase.

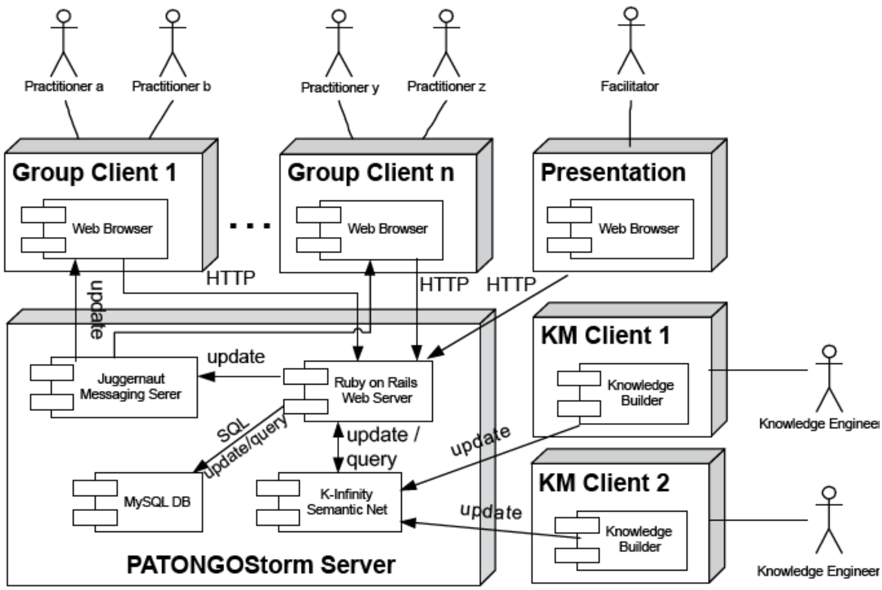


Fig. 5. Distribution architecture of PATONGO-Storm

5 Experiences

The approach was tested in a workshop at the Evangelical Church in Germany. In the following, we report our experiences with using PATONGO-Storm regarding how ideas were exchanged and what kind of networking between participants could be observed.

Setting

This workshop took place in September 2009 during a big congress of the Evangelical Church in Germany. Experienced practitioners and key members from all over Germany met for two days to discuss new forms of churchly practices. One of the workshop offers was a workshop on networking among practitioners using Web 2.0 technology. In this workshop PATONGO-Storm was used.

Subjects

A total of 24 people with age between 27 and 68 years participated. 12 groups of two people were formed who each shared a laptop computer running the PATONGO-Storm tool. One third of the participants were female. All were experienced computer users.

Measures and data collection

We recorded the development of the semantic net and all contributions and communication between groups. Based on this data, we are able to determine the number of activities per user and per group as well as the number of contributions shown to other groups. Both kinds of data could be categorized into the phase it belongs to and the duration of visibility on the respective screen. Based on this data, we could observe typical behavioral patterns during each phase.

We also looked at the development and use of keywords and tags both, for participants and knowledge editors.

In addition, we collected user feedback by using questionnaires.

Procedure

The workshop lasted 90 minutes. After a short introduction on the workshop topic the groups started to work in the second phase for 21 minutes on collecting challenges and solution ideas. In the third phase the collected contributions were visualized on a map of Germany, and three ideas selected by the moderators were discussed in the plenary with their authors.

Then the fourth phase began where the groups had 20 minutes to comment ideas of other groups, to propose solution ideas to challenges of other groups, and to discuss about challenges and ideas. A brief summary of lessons learned and a discussion on concepts for continuing the exchange and collaboration after the workshop followed in the last phase. Finally, feedback about the process and interaction with PATONGO-Storm was collected using questionnaires.

Results

Even though the number of participants is relatively low (24) and does not allow statistical analysis, we analyzed the communication flow between participants in order to assess the impact of our approach on a qualitative level. We focus on how solution ideas were created, integrated into the semantic net, and how discussions around ideas evolved. The short duration of our experiment turned out to be beneficial as time-pressure forced groups to create ideas quickly. This is consistent with the 6-3-5 method introduced before. However, participants reported that they would have appreciated a bit more time for making contributions (in phases 2 and 4), which would lead to the 120 minutes outlined in our process description. Unfortunately, in the overall context of the congress it was not possible to give the workshop a two-hour slot.

Table 1 compares the forms of contributions within the two group work phases of the workshop (i.e., phases 2 and 4). During the second phase (20 minutes) 12 groups created 31 solution ideas and 26 challenges. During the fourth phase of 22 minutes the groups created 85 contributions. In both phases, participants had first to decide and select the type of contribution they wanted to create (cf. the selection shown in figure 1). This step is denoted as “selection” in table 1. During this step groups usually discussed the topic, agreed on the topic, and selected the corresponding type of contribution. In phase 4, the selection step also included the reception of the contributions proposed by the system.

Participants had no problems understanding the types of contribution (idea, challenge, and comment). In the second phase, all contributions were assigned correct types. In the fourth phase, still 92% of contributions in this phase were assigned the correct type. 5 of 32 contributions classified as ideas were inquiries (comments) and 2 comments actually contained ideas. Incorrect assignment of types was caused by our process design, since we assumed that the reaction to a challenge would always be ideas. However, sometimes participants did ask questions instead of posing an idea.

Table 1. Overview about activities of groups

	Type of activity	**Frequency**	**Of which had contributions**	**Of which were shown to others**		**Of which had later reaction**		**Time spent on over-all**	**Time spent on in phase**
Phase 2	Selection	62						10%	21%
	Reporting an idea	32	31	13	42%	10	32%	22%	47%
	Reporting a challenge	26	26	21	81%	14	54%	15%	32%
Phase 4	Read and Select	126						15%	28%
	Create idea for challenge	37	32	32	100%	23	72%	15%	28%
	Comment idea of other group	23	19					7%	13%
	Read response & answer	57	34					17%	31%
	total	**363**	**142**					**100%**	

In the second phase the majority of contributions were ideas (47% ideas, 32% challenges). This is consistent with the results of the questionnaire where participants expressed their wish to share ideas with others.

In the fourth phase groups invested similar time into reception of other group's statement (31%) and creating statements for other groups (28% + 13% = 41%). This emphasizes the importance of the dialogue. Participants confirmed this in the questionnaire: two thirds of the participants stated that they wanted to discuss their idea with other groups in any case. Only one participant explicitly stated that he did not want to discuss the idea.

Table 1 shows that groups used only approximately 13% of the time for commenting other groups' ideas but 30% of the time for reception of contributions and selection, creation of new ideas to other groups' challenges, and reading and reacting to contributions of other groups. As triggered by the process, participants changed their activities during the fourth phase (see contribution type selection in figure 4).

Fig. 6 shows on the horizontal axis the time in minutes in the fourth phase, the vertical axis on the left shows the total time in seconds used by all groups on the respective activity type. The bold line shows how many groups (vertical axis on the right) did at that time not yet receive any comments or ideas on their contributions.

At the beginning (i.e. first 4 minutes) primarily ideas and challenges of other groups were perceived („selection") and ideas were commented. This was followed by a phase of dealing with ideas and challenges of other groups (until minute 9). In the second half of the fourth phase groups entered a dialogue. Interestingly, groups returned occasionally to dealing with other groups challenges (minutes 11 to 15). This is a consequence of our process design since groups can only respond to reactions of other groups that have not yet received a reaction. The number of groups having received no comments yet when they needed to select the next activity is not monotonic decreasing but has a local minimum at minute 10 followed by some small and brief increase.

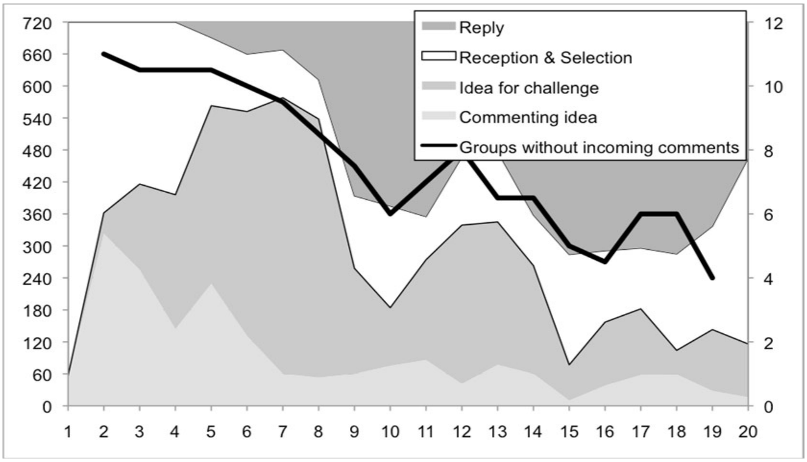


Fig. 6. Change of activities over time in all groups

The increasing share of selection activities after minute 18 was due to an intervention of the moderator who asked participants at this point in time to stop creating new contributions and to come to an end.

Each group received in phase 2 responses from on average 4.5 groups (min=1, max=8) on their contributions. On average each group reacted on contributions from 4 groups (min=3, max=7). Thus, each group interacted with on average 6 groups (min=4, max=9).

Fig. 7 shows on the right side the interactions between participants. The bolder a line is the more messages were exchanged. Interactions within groups are not shown. The many lines crossing borders of the state churches being members of the Evangelical Church in Germany show that much cross-organizational interaction happened. People who usually rarely come to know each other were motivated by the semantic closeness to other participants to enter into discussions. This brings us to the issue of quality of the integration of new contributions into the semantic net, as this is the means for finding similar interested peers.

The participants were asked to attach keywords to ideas and challenges but not to comments. These keywords were interpreted as reference nodes in the semantic net and used to connect new contributions to the semantic net. The initial version of the semantic net was constructed by knowledge editors based on keywords from the church domain and contained 442 keywords. The groups used 46 of them while adding another 120 new keywords. This observation points to a problem of keyword use: since the topics discussed were from as broad spectrum of churchly work the initial version of the semantic net could feature many gaps. 101 of the 120 new keywords were processed by the two knowledge editors during the workshop, i.e. integrated into the semantic net as new reference nodes. On average groups used 3.5 keywords per contribution (min=1, max=9).

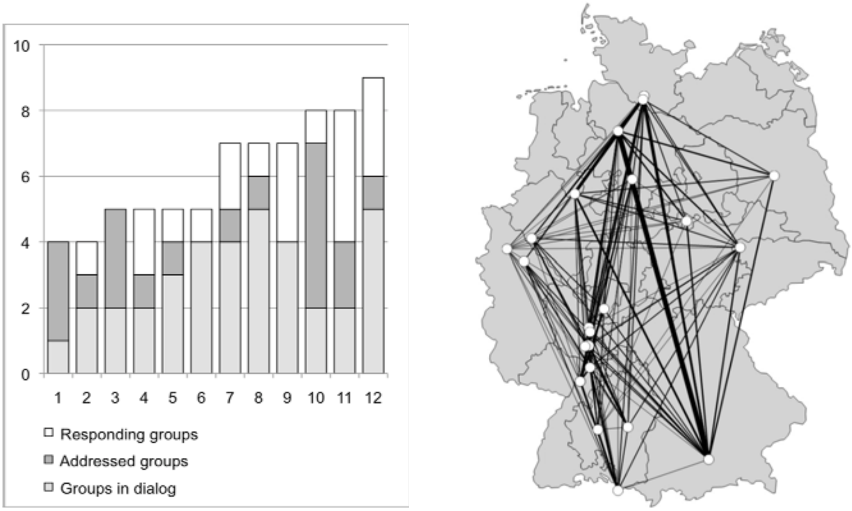


Fig. 7. Interactions between groups (left) and geographical locations of participants (right)

We observed keywords that were just used by the group which created it. For 11 of the 31 ideas and for 9 of 26 challenges created in the second phase no other groups with related topics could have been determined by just using co-tagging (i.e. overlapping keyword sets). In this case, the semantic net approach shows its power: since keywords were connected to related keywords one can determine those groups who are interested in related topics since they have a short distance in the semantic net.

Fig. 8 illustrates a case where two groups would not have met using simple co-tagging approaches. Group A contributed an experience of a resurrection path that this group once installed in their parish. They had an interest in celebrating the resurrection. Another group (Group B) also reported on a good practice for the Easter season (an idea for a special service on Easter Monday). However, Group B used different tags (Easter instead of Easter season). The knowledge engineers then related the tag Easter with the composite tag Easter season and thereby forged a link between the two groups. Later on, other groups benefited from the connection between Easter and Easter season as well.

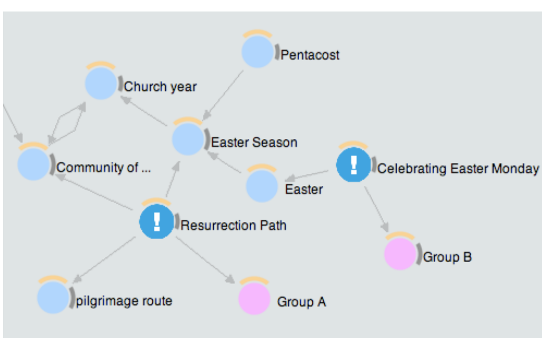


Fig. 8. Excerpt from the semantic net (translated to English)

A third of the contributions used just new keywords. In such cases, the existing semantic net could not help in locating relevant peer groups. However, due to the two knowledge editors these keywords were integrated and detecting related groups became possible. This demonstrates the importance of the role of knowledge editor especially in domains not completely mapped into the semantic net.

Finally, the questionnaire data shows that the participants rated the contents proposed by the system as interesting, and that participants liked to respond to such content. This confirms that in this case the assignment of contributions to groups worked.

6 Conclusions and Future Work

In this paper, we presented the PATONGO-Storm approach facilitating exchange of good practice and joint development of new solution ideas in large distributed organizations. Our approach supports co-located workshops in which a five phase process is used to organize exchange of good practice and joint development of new solution ideas. This process uses a form of description schema for good practice that is inspired from design patterns. It consists of challenges and solution ideas. The process organizes the networking among practitioners based on their personal experiences, which are derived from a semantic net built from all contributions. This net is used for formation of groups with complementary experiences.

The approach was tested in workshops with practitioners from the Evangelical Church of Germany. Interaction analysis shows a lively exchange between groups, even when group members did not know each other before. The use of the semantic net led to good networking among participants from different parts of the organization, and did enable exchange between groups with related interests. Feedback from participants was extremely positive.

Although the Evangelical Church of Germany is a very specific organization, we assume that the approach of collaborative reflection on good practices is applicable in a very large set of different contexts ranging from large non-profit organizations over political bodies up to corporate settings (e.g., large automotive companies reflecting on good practices for improving their production processes). Especially the very diverse professional backgrounds brought in by voluntaries who participated in our observed workshop support this assumption.

Our approach goes beyond related research in that it adds an organizational learning perspective to storytelling and project retrospectives, which otherwise remain focus on learning from individual practice. It goes beyond known creativity techniques focusing on idea generation by adding discussions with practitioners with appropriate background knowledge. Technically, our approach uses semantic net technology to create this semantic fit between challenges, ideas, and people. However, while such techniques are usually employed in asynchronous recommendation, filtering, and search scenarios, we have shown that our approach can be used to support synchronous co-located knowledge creation and exchange. The analysis of the workshop data in section 5 shows that our method in fact led to discussions of ideas and challenges with other groups. Participants found the proposed content and partners interesting and relevant. These findings support our claim that the proposed approach helps to bring practitioners together with similar experiences and interests so that they may benefit from each other.

In future workshops we will focus on two aspects: Firstly, we want to test applicability of our approach to other application domains, e.g. in workshops with computer scientists discussing software engineering practices. We foresee that the approach can be beneficial for most large organizations where members of the organization do no longer have an overview of distributed experience. Secondly, we are currently investigating how the proposed co-located process can be combined with asynchronous and distributed forms of interaction. We currently integrate PATONGO-Storm into a community platform for exchanging practice knowledge in the Evangelical Church in Germany that has been released in May 2010[3]. Our focus is on whether and how PATONGO-Storm may be used as a motivational element in co-located workshops to spark asynchronous discussions in the distributed community.

Acknowledgements

This research was funded by the German Federal Ministry of Education and Research and the ESF (FKZ: 01PF08005A). We thank the partners of the PATONGO project, especially Christina Matschke of the Knowledge Media Research Center and the PATONGO team of the Evangelical Church of Germany. Special thanks are due to our team member Wolfram Schobert for his excellent contributions to the development of PATONGO-Storm.

References

1. Ackerman, M.S.: Augmenting the Organizational Memory: A Field Study of Answer Garden. In: Proceedings of CSCW 1994, pp. 243–252. ACM Press, New York (1994)
2. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S.: A pattern language. Oxford University Press, New York (1977)
3. Kirchenamt der Ev. Kirche in Deutschland (EKD): Kirche der Freiheit – Perspektiven für die Evangelische Kirche im 21 Jahrhundert. EKD, Hannover (2006)
4. Finlay, J., Gray, J., Falconer, I., Hensman, J., Mor, Y., Warburton, S.: Planet: Pattern Language Network for Web 2.0 in Learning, Final Report (2009), `http://www.patternlanguagenetwork.org/reports`
5. Kerth, N.L.: Project Retrospectives: a Handbook for Team Reviews. Dorset House Publishing, New York (2001)
6. Kolb, D.A.: Experiential Learning. Prentice Hall, Englewood Cliffs (1984)
7. McDonald, D.W., Ackerman, M.S.: Expertise recommender: a flexible recom-mendation system and architecture. In: Proceedings CSCW 2000, pp. 231–240. ACM Press, New York (2000)
8. Mor, Y., Winters, N.: Design approaches in technology enhanced learning. Interactive Learning Environments 15(1), 61–75 (2007)
9. Nonaka, I., Takeuchi, H.: The Knowledge Creating Company. Oxford University Press, New York (1995)

---

[3] The German homepage of the community platform is available at http://www.geistreich.de

10. Ning, H., Williams, J., Sanchez, A.: On-line Peer Review in Teaching Design-oriented Courses. In: Proceedings of EISTA 2003, Orlando, FL (2003), http://www.iiisci.org/journal/CV$/sci/pdfs/P990045.pdf
11. Rensing, C., Böhnstedt, D.: Bildung einer Community zur Vermittlung von E-Learning-Erfahrungen auf Basis semantischer Netze. In: Schwill, A., Apostolopoulos, N. (eds.) Lernen im digitalen Zeitalter - DeLFI 2009. LNI, vol. 153, pp. 187–198. GI, Bonn (2009)
12. Rohrbach, B.: Kreativ nach Regeln – Methode 635, eine neue Technik zum Lösen von Problemen. Absatzwirtschaft 12(19), 73–75 (1969)
13. Schön, D.A.: The Reflective Practitioner: How Professionals Think in Action. Basic Books, New York (1983)
14. Schümmer, T., Haake, J.M.: Supporting social practice learning in NGOs. In: Proceedings of WebScience 2008, pp. 17–21. ACM Press, New York (2008)

Establishing On-Line Corporate Training in Distributed, Synchronous eCollaboration: A Field Study

Tero Päivärinta and Bjørn Erik Munkvold

University of Agder, Department of Information Systems,
Serviceboks 422, N-4604 Kristiansand, Norway
{Tero.Paivarinta,Bjorn.E.Munkvold}@uia.no

Abstract. Whereas technologies for eCollaboration are maturing, a need for learning eCollaboration in distributed organizations continues. This paper presents the experiences from establishing corporate training in distributed, synchronous eCollaboration. The training package was delivered solely on-line with synchronous interaction among the participants and the instructors. The study contributes to the scarce body of research on computer-supported collaborative learning in professional and corporate contexts. The reported experiences illustrate several challenges to establish corporate on-line training of eCollaboration: promotion of awareness of the benefits of learning eCollaboration, management of varying pre-skills of employees participating in distributed and synchronous eCollaboration, lack of common conventions for selecting and using tools for particular organizational eCollaboration scenarios, inclusion of synchronous eCollaboration in the corporate eLearning strategy, and alignment of focused eCollaboration competencies to ever-evolving corporate eCollaboration infrastructure, policies and IT operations.

Keywords: eCollaboration, distributed meetings, computer-supported collaborative learning (CSCL), corporate training, e-learning.

1 Introduction

Training has long been acknowledged as a prerequisite for effective eCollaboration [1], [2]. Yet, this implies several challenges. The number of eCollaboration solutions in the marketplace increases by the month [3], [4], and companies need to establish and maintain a portfolio of different eCollaboration technologies that together provide integrated support for various collaborative tasks [5]. To be effective, training in corporate eCollaboration infrastructures need to be conducted in situ, related to actual collaborative tasks in the workplace [6]. This again demands resources for establishing a corporate infrastructure for learning eCollaboration, and facilitating employee participation through motivation and incentives. As will be discussed in this paper, many of these challenges are similar to those experienced in the process of organizational implementation of eCollaboration solutions [1].

The area of Computer Supported Collaborative Learning (CSCL) offers insight in terms of pedagogical issues related to ICT-supported learning in collaborative settings [7]. Yet, most of the research in this area has been conducted in educational settings

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 81–96, 2010.

[8], [9]. Also, the main focus in CSCL research has so far been on asynchronous collaboration for learner interaction and participation [10], [11], [12], [13], reflecting the limited access to synchronous eCollaboration support in educational settings. Research on virtual teamwork emphasizes the importance of including tools for synchronous interaction in the team's technology support repertoire [14], [15]. However, also in this area a majority of the empirical research studies are conducted as quasi-experimental studies in educational settings [16].

This paper presents in-depth experiences from establishing on-line corporate training in eCollaboration, with main focus on support for distributed, synchronous eCollaboration scenarios such as virtual meetings based on on-line textual, audio and video communications. The research setting is a global engineering and construction company, with needs for effective eCollaboration practices to support the distributed projects and cost cuts related to employee travel. A pilot course for online training in eCollaboration was developed, with online, hands-on learning mediated through the same synchronous eCollaboration tools as were focused in the course. The reported experiences address a number of challenges to establish on-line training of eCollaboration in the corporate context, contributing to the scarce body of research on computer-supported collaborative learning in professional and corporate contexts.

The next section presents a brief overview of related research. Section 3 introduces the research setting and the background for the pilot course. Section 4 describes the research process, and section 5 presents the course concept. Section 6 presents the experiences from the pilot course, and reflection upon these. Section 7 discusses the implications from our findings, and relates these to former research. Section 8 presents conclusions and suggestions for further research.

2 Related Research

In reflecting upon the experiences from the eCollaboration course, we have drawn upon research from different areas. The course concept builds upon the collaborative, distributed learning model of eCollaboration through participant interactions across geographical distance [17]. This principle is in line with Vygotsky's theory of social learning, which converges "speech and practical activity" [18] during the learning process. In our pilot course this was obtained by conveying the learning process through the very same eCollaboration media that were to be learned.

The CSCL literature provides insight into various forms of collaborative learning, involving both asynchronous and synchronous communication media [19]. Still, few studies have focused on synchronous eCollaboration as the core element of CSCL [13] and even fewer in a professional setting [12]. According to a recent design theory, synchronous communication (as a complementary to traditional asynchronous media) in online education is theorized to enhance small group and personal level participation, task support exchanges, social support exchanges, class-wide relations among students, and group-wide relations [20]. As such, the theory motivates use of synchronous eCollaboration media for distributed learning. However, the literature lacks reports of how learning of eCollaboration itself should be established in organizations. Effective learning of eCollaboration requires a collaborative setting, as individual learning of such tools will not cover the social dynamics involved in eCollaboration in 'real' settings [21]. (While another research idea has suggested that professionals could learn eCollaboration individually with help of artificial computer simulations [22], experiences from this approach remain undocumented.)

Based on our focus on distributed, synchronous meetings, we also searched the literature on adoption and diffusion of virtual meetings and on-line collaboration in organizations for relevant experiences. At the individual, *psychological*, level, people should be able to see personal value to learn eCollaboration and to feel comfort and trust on using the technology [23], [24]. At the *social* level, the participants and the instructor should trust each other; the setting should provide contextual information and possibilities for social communication, and social presences [23], [24].

We found no research on the process of establishing training of eCollaboration as an on-line corporate initiative. However, experiences from corporate eLearning in general [25] provide relevant insight. Also, case studies on the implementation of eCollaboration infrastructure are relevant in this context. For example, a study of the implementation of distributed meetings in the Boeing company illustrates the mix of organizational and technological challenges involved in this process [1]. At the *organizational* and *technological* level, on-line learning can meet resistance and friction, and challenges to ensure technical expertise, support, and well-functioning infrastructure [1], [25]. In general, the key challenge has proved to be to create self-sustainable eCollaboration work practices [26].

3 Research Setting

The study took place in a global corporation providing engineering and construction services, products and integrated solutions in approximately 30 countries world-wide. The corporation has centralized the support for IT applications and infrastructure services to an internal service organization, which serves the business units through jointly established service level agreements (SLAs). The history of strong growth through a number of acquisitions and mergers since the late 1990s has resulted in slightly varying cultures of eCollaboration in terms of adoption and use of particular technologies (such as data and telephone conferencing vs. instant messaging, or videoconferencing). As well, an internal study on eCollaboration in 2008 revealed great variations among the sites in perceived maturity of using the available tools and a good number of deviations from using the "standard" tools.

Microsoft Net Meeting and telephone conferencing had been widely in use since the 1990s, although many still commented that distributed meetings should be practiced more. Throughout the 2000s the organization has standardized the office and communication tools in general (Table 1). In 2009, a change process was initiated when Microsoft stopped supporting Net Meeting, and this was replaced by MS Office Communicator and MS Live Meeting. Due to corporate network policy and bandwidth concerns, the audio and video functionalities of these tools are not in use. Hence, most distributed meetings take place through telephone conferencing, often complemented with data sharing through one of the applications mentioned above. The high-end videoconferencing rooms are used less, being unavailable in a few sites due to their perceived costliness.

Table 1. The standard eCollaboration tools in the target company

eCollaboration Tool	Purpose
Microsoft Office and Outlook 2003 (2007)	Office tools, e-mail, calendar, resource management, meeting scheduling and invitations
MS Share Point 2003 team sites	Document sharing in teams, discussion forum, wiki, surveys, etc. document-based asynchronous collaboration.
Intranet	Personalized links of importance, blogging (little used), personalized and corporate news, access to shared IT services
MS Office Communicator (replacing Net Meeting)	On-line awareness, contact search and information, instant messaging, application / desktop sharing for smaller ad hoc meetings
Meet Me (British Telecom)	Telephone conferencing
MS Live Meeting 2007 (replacing Net Meeting)	Virtual ad hoc and pre-scheduled meetings for max 15 participants, application / desktop sharing (audio and video disabled).
Tandberg 1700, 6000	High quality videoconferencing

The shift from Net Meeting to the two new communication tools changed the "business model" for eCollaboration in the company. While Net Meeting had been free, Office Communicator and hosting rights for Live Meeting now became separate service offerings to be paid for by the user organizations in line with other special applications (alike Share point team sites and telephone conference hosting rights).

Whereas the office infrastructure was mainly based on the Microsoft 2003 package, Live Meeting changed during the study period from the 2005 version to 2007. As well, some units had got a special permission to use Microsoft Office 2007 software. Together with the fact that the audio and video were closed, this meant that the application infrastructure was not following the vendor standard.

The global financial crisis hit the organization in fall 2008 and the subsequent need for cost cutting, e.g. by reducing travel to meetings, became a motivation for promoting eCollaboration. The Chief Executive Officer stated:

"Telephone and video conferencing, online collaboration and "net meetings" are effective examples of hosting business meetings without travel. A reduced-travel approach to our business will benefit the environment and help to control non-essential expenses." (CEO, Revised travel policy, Oct 23rd, 2008).

A policy to cut business travelling was introduced. At the same time, the financial crisis also stopped further investments in eCollaboration. The management started a re-structuring process, which furthermore lead to difficulties in placing eCollaboration training under any clear area of top management responsibility. With regard to corporate learning programs and training offerings in general, the corporate courses had traditionally involved two types of learning approaches: classroom training and eLearning with plain human-computer interaction. Hence, synchronous eCollaboration tools had not been included in corporate courses prior to this study.

In August 2008, the first author (hereafter referred to as 'the researcher') joined the company for a sabbatical year from academia, spending 30 hours per week there. His first task in fall 2008 was to facilitate general-level consultation of eCollaboration and to give a master level course on eCollaboration and content management for 16 IT professionals located in six sites (in 2 countries within the same time zone). The course was given fully through the corporate eCollaboration tools (Table 1). Travel was not allowed for the students or the instructor. The idea was to simultaneously enhance the students' skills in using the company's standard eCollaboration tools. Twelve weekly 3-hour sessions started through Net Meeting and telephone conferencing and later changed through Office Communicator sessions to Live Meeting. Beyond the organization's standard tools, Microsoft's Round Table cameras were used to test video and voice-over-ip-audio in the two last Live Meeting sessions. A Share Point team site hosted the course materials, presentations, and wiki group works. According to the participant feedback, the course was well-received and popular – the attendance rate being over 80% throughout. The course participants, being IT professionals themselves, reported to have learned eCollaboration tools well during the course, in addition to the academic course content.

In the corporation, varying information about the eCollaboration tools existed, such as several old and redundant versions of user guides, scattered across the Intranet. It took time to learn the portfolio of the tools available, and to acquire access to these tools from the service desk. Altogether, the researcher spent 4 work days to look for, acquire, and learn the eCollaboration tools (table 1) by himself to master these sufficiently to facilitate the master course. This period lasted approximately 2 weeks of calendar time, due to lead time in getting answers from service desks etc. Another person helped to test the tool functionality. Hence, the total preparation process for mastering the eCollaboration tools took 6 effective work-days. This provided another motivation for the course as no regular employee in the company could be expected to use this amount of time for learning eCollaboration by themselves in the middle of hectic workdays. While some tool-centric courses on Live Meeting and Tandberg videoconferencing were offered (mainly by the vendors), these were not fitted to the organization's special work arrangements, e.g., when these media were used in connection to other media. In total, no coordinated learning setting for eCollaboration had yet been established in the company. As commented by an IT expert about eCollaboration adoption in the company (Diary entry Nov 7th, 2008):

"We assume that our engineers are eagles and we just say to them: "Fly!" We just never teach them to do so."

4 Research Process

During the master course mentioned above, an idea of a more practical eCollaboration pilot course was suggested in co-operation by the company's IT personnel and the researcher. The researcher applied and got public funding to enable the research part to develop the concept. The funding for developing the idea made the project possible also in the company context, while other IT-related investments were stopped. The pilot was planned in a team of the researcher and three practitioner representatives of the target organization, located in two sites. The course planning and marketing for volunteer participants took place in March-May 2009. Pre-skills and ideas of the course participants were surveyed through the intranet. The pilot took place over two weeks in May-June 2009.

The researcher wrote a diary of his observations throughout his stay in the company (August 2008 – July 2009). He recorded the comments and impressions from the interactions before and during the course. In addition, the course had its own wiki site where the project team documented their experiences. During the course sessions, the instructors gathered feedback directly from participants in relation to each session and documented the main points to the wiki immediately after. As well, the researcher collected documentation about eCollaboration in general in the company. The other team members also documented their impressions and evaluations of the course on the way. The author also continued to have contact with persons from the company in fall 2009, reflecting on the further actions taken.

5 The Planned Course Concept

The project team decided that the course modules should focus on typical scenarios related to distributed eCollaboration, which integrate use of the tools in many occasions. Hence, the course was tailored for the corporate setting. The user scenarios were divided in four main categories (Table 2): coordination and awareness, distributed meetings through telephone and data conferencing, document-based collaboration and videoconferencing. The categories and related task scenarios were based on the corporate team members' judgement on what should be mastered by a major proportion of the company's employees. These scenarios were covered through the following five course modules:

1. Introduction of eCollaboration practices (2 hours)
2. Organizing and running small, ad hoc distributed meetings (1 hour)
3. Organizing and hosting larger distributed meetings, or meetings outside the corporate IT network (2 hours)
4. Document-based asynchronous eCollaboration (3 hours)
5. Videoconferencing (through dedicated videoconferencing rooms) (2 hours).

Table 2 maps the identified eCollaboration scenarios to the planned course modules. Altogether, the course modules involve 10-12 working hours to be spent flexibly according to the modules. After module 1, the rest could be attended independently from each other.

To overcome the psychological and social thresholds of eCollaboration identified above [23], [24], we organized the course as a short-term eCollaboration setting, highlighted individual hands-on activity in groups during the sessions, pursued to facilitate interest beyond particular tools on integrated eCollaboration scenarios (aiming at "more vested" interest in the setting), and highlighted the focus on eCollaboration by top management. We modularized the course to enable people to select modules based on their interests and needs, and we sought to mitigate the learning curve through instructions for meeting preparations. In addition, we provided a fallback plan and technical support, and well-prepared process plans for each synchronous session. We also had an extra person during the first sessions to communicate through separate channels if any participant had problems. The instructor became a facilitator giving feedback of how the participants performed and learned the eCollaboration scenarios

Table 2. Collaboration scenarios and related tools planned for the pilot course

Collaboration Scenario	eCollaboration tools involved	Mod.
Coordination and awareness - being available for eCollaboration		
On-line awareness, updating availability status	MS Communicator (& Outlook Calendar)	1, 2
Adding contacts for collaboration	MS Communicator	1
Booking a meeting in a group calendar, keeping track about a group's events as a whole	Share Point Group calendar	1, 3
Synchronizing own Outlook with a group's calendar	Outlook, Share Point group calendar	1, 3
Informing others about contact info and interests	Intranet, blog	4
Keeping track of interesting issues and persons in the corporation	Intranet, MyLinks, Favourites	4
Distributed meetings		
Inviting a group of people to a pre-scheduled distributed meeting	Outlook, Live Meeting	1, 3
Participating in a pre-scheduled on-line presentation	Live Meeting, Meet Me, Outlook Calendar	1, 4
Inviting others to a small distributed meeting inside the corporate network spontaneously	MS Communicator	2
Setting up a telephone conference	Meet Me, (Outlook Calendar)	2
Running a small meeting inside the corporate network, contributing to a smaller meeting	Communicator, MeetMe	2
Inviting someone spontaneously to a meeting, including participants outside the corporate network	Live Meeting	3
Contributing to a group decision in a distributed meeting	Live Meeting, MS Communicator, Meet Me, Share Point	1, 2, 3
Hosting a pre-scheduled meeting, need for flexible attendance of the participants	Live Meeting, Meet Me	3
Presenting in a pre-scheduled, bigger meeting	Live Meeting, Meet Me	3
Creating a joint idea / decision in a bigger meeting	Live Meeting, Meet Me, Share Point	3
Basics of document-based collaboration		
Best practice for using a shared document library	Share Point	4
Arranging 3rd party access to a document library	SharePoint	4
Joint and quick group editing of dynamic textual information	Share Point Wiki, Alert functionality	4
Informing the whole group about important issues	Share Point Announcement, Share Point participants, e-mail	4
Group discussion about longer-term matters	Share Point discussion forum, Alert functionality	4
Videoconferencing		
Booking a videoconference	Tandberg, Outlook	5
Participating in a videoconference	Tandberg VC	5
Hosting a videoconference	Tandberg	5
Documenting and sharing results	Tandberg, Share Point	5

in question [23], [24]. The overall aim was to provide the participants an immediate and positive hands-on experience with the tools, which was related to use scenarios they would most likely experience in their future work.

To overcome the foreseen organizational and technical challenges [25], the team pre-tested the technologies and scenarios carefully and pre-clarified technical support for the instructors from the IT services, within the limits of the SLA. Managers expressed their support for the course and recruited participants from their units to the course, thus indicating that the required "strategic alignment" was in place.

6 Experiences from the Pilot Course

6.1 The Course Modules Given

The pilot course was advertised in three subsidiaries in five sites. The volunteer participants were selected due to their estimated frequency of hosting virtual meetings. As planned, 20 participants accessed the first joint session through MS Live Meeting and MeetMe telephone conferencing. A part of them logged in from their PCs, while some gathered to plug in through meeting-room PCs. Module 2 was given as on-line small group sessions for six independent groups. The sessions lasted between 45-75 minutes depending on the group activity and interest. The module involved altogether 16 participants, using MS Communicator and MeetMe. The small group sizes varied between 2 to 5 participants and 1-2 instructors.

Module 3 was given in two phases. First there was a common 1-hour Live Meeting and Meet Me session, attended by 18 participants, learning to participate in a Live Meeting. The participants organised further another LiveMeeting in small groups by themselves. This session focused on hosting Live Meeting. Each group for the latter part contained 3-5 participants. The instructors followed three such sessions, while some groups met on-line without the instructor. We estimated that approximately 12 participants altogether attended the latter session. Drop-out was reported to emerge due to hectic daily schedules.

Module 4 was given by a Share Point expert through MS Office Communicator, due to technical problems to adopt Live Meeting. Nine participants, the instructor (and the researcher) participated in a two-hour session during which the participants conducted exercises on a Share Point team site and Intranet.

A lack of available videoconferencing champions hindered the development of the last module, as well as lack of videoconferencing installations at the site where the researcher was located. The re-structuring of the management required presence of the existing champions in other tasks, while some of them had left the company during spring 2009. The organization reported some use of the vendor's course offerings according to emerging needs.

6.2 After the Pilot Course

The project failed to reach its immediate practical goal as in the period following the pilot course the company moved back to ordinary classroom training for the meeting tools, instead of choosing the hands-on on-line approach. The main reason for the discontinuance was a lack of further resources to develop on-line training after the researcher left the company. The videoconferencing module was never implemented as planned. The IT service organization continued ad hoc training of videoconferencing users in addition to providing access to the vendor-organized and charged training sessions. However, the contact person stated that the company still (in Spring 2010) intends to use on-line training for eCollaboration in the future.

The person responsible for the further training of eCollaboration interpreted the reasons for abandoning the on-line training (e-mail Nov 25th 2009):

"NetMeeting was decommissioned and they [IT service management] wanted a couple of sessions covering a lot of people at the same time – therefore sessions in an

auditorium was just a more sufficient solution… I am sure that this decision was also influenced by experiences from our pilot and the fact that the start up on LiveMeeting was time-consuming – 20 min – if people had not tried it before. But […on-line] try-out sessions [are] not totally on ice… we will get something set up – it is new focus on collaboration in general in our company."

She also mentioned another potential reason for the classroom popularity to be that the "normal" users are not by default "curious IT users", and it may feel safer to have training where one can meet the instructor face-to-face. Evidently, this preference was in the end of the day stronger than continuing the on-line option. Training outside the headquarters was, however, still cut due to the diminished resources allocated to training in general due to the corporate cost-cutting mode.

A manager, who had intended to attend the pilot course but did not find time, later on reflected (Diary entry Oct 28th 2009):

"Nowadays I use Live Meeting for the steering group meetings. After [hosting] three-four such meetings, I, however, still wish to feel more confident to use it… I could still have use for such a course. [Discussion about the hectic daily duties]. [The contact person for the course] gave me some 1-to-1 guidance [on-line], though!"

Hence, reactions to emerging needs may lead to ad hoc training sessions on-line. However, such training gives a less thorough view on the tools or skills to use them to their full potential. Moreover, the costs involved in using an instructor's time for individual training sessions will be hidden from any visible training budgets.

In July 2009, a corporate survey revealed that only a month before closing down the Net Meeting application there were still 1700 active users of it, who had not ordered the new tools. In August 2009, this observation led to a quick decision to organize class-room training sessions for the new meeting tools in the headquarters. The two modules for distributed meetings were used as the basis for class-room teaching (for two classes around 50 participants each). The company also produced recorded materials on the Intranet for individual self-study of the new tools. The classroom teaching received mainly good feedback.

6.3 Reflecting on the General-Level Experiences

The process in general confirmed the need for establishing corporate training of eCollaboration. The participants expressed their improved confidence and skills to collaborate immediately during the hands-on course modules. Detailed experiences of each module were collected and the modules were updated. The pilot also indicated more general-level experiences. In the following, we reflect on these experiences, categorized as individual and psychological issues, social issues, technological and infrastructural issues, and organizational issues [23], [24], [25].

On the ***individual psychological*** level, the main challenge was not, paradoxically, to squeeze the actual participation time required from the individual, which was one of the initial rationales for promoting the on-line mode, or to motivate people to participate as such. We ground this observation on the subsequent popularity and good feedback from the classroom courses, given by the same instructor involved in the on-line pilot and with approximately the same content. Rather, varying attitudes towards the on-line medium in the workplace context may represent a bigger issue. The distributed, on-line mode makes some people significantly more anxious on the idle time and frustrated enough to log-out from the on-line alternative. For example, when first time users in the distributed meeting tools logged in, it usually took 15-20 minutes of extra time and support to get them able to participate fully in the on-line screen-sharing and telephone conferencing sessions. This could frustrate a few of the more experienced users, and ultimately lead to explicit comments about "waste of time" from the most anxious ones. A couple of people quickly left such sessions, with explicit comments. While some participants followed the initial advice to log-in 15 minutes before the scheduled time with the instructors, in order to have time to learn the technicalities, not everyone did so. Another source of on-line frustration for some was if the instructor was only presenting issues which were already familiar to the participant. People were anxious to get their "hands on" the technology. According to the classroom trainer, who had previously also instructed the on-line modules, this was not experienced to the same degree in the classroom teaching.

On the other hand, those who persevered through the idle starting moments of the on-line meetings gave positive feedback on the on-line mode in general. In addition, the sessions where everyone was at least a "second-timer", the "hassle-time" in the beginning was decreased to 0-5 minutes and the third joint session started smoothly. An overall impression was that the participants' psychologies were polarized with regard to their tolerance of the on-line idle time resulting from participation of the beginners. Even when the instructors phrased this phenomenon as a relevant learning experience (the same hassle is frequent in any real-life meeting with first time users) a few people could still not tolerate the delay. At the immediate workplace, the threshold to prioritize other on-going work duties may thus be significantly lower if compared to a local auditorium or meeting room. Hence, one key to make the on-line course participation to succeed is psychological – those who prioritize their learning as important as or more important than other work tasks, will persevere – otherwise the idle moments are regarded as intolerable "waste of time".

Another individual-level issue is self-awareness of the need for training. For example, several course participants expressed first scepticism on participating in the MS Communicator module, as they "*had already been using the tool for months*". During the small-group sessions, however, it quickly appeared that the most had mainly used the tool only with regard to the basic on-line presence awareness and chat functionality, while many other features were untouched. For example, many of the course participants were positively surprised of the possibilities to use the tool as a corporate address book providing very quick contact and search, informing about their particular status, and using the desktop sharing capabilities.

At the ***social*** level a challenging issue was to handle the varying pre-skills of the participants to ensure the flow of the training sessions. This required well-scrutinized modularization of the course. Our starting point was the use scenarios created within the project group (Table 3). However, such conventions vary between the subsidiaries and local organizations within a big corporation. On the other hand, few users of particular tools shared common conventions of where and how to use which tool before the course anyhow. For example, Communicator had been used for big pre-scheduled meetings in cases where Live Meeting would have been a better option from the viewpoint of meeting management.

A few people dropped out from the later sessions, unlike in the master course. The master course participants had longer-term tasks of teamwork, which may have kept them better involved. In the short term pilot course, the participants were less committed to the teams in which they rehearsed the eCollaboration skills.

A few course participants did not adhere to the standard ensemble of corporate eCollaboration tools, while still expecting that the course would cover their preferred tools. This indicates deviations from the corporate policies in units where some persons had access to independent resources – in fact, they were not aware of the "strict" corporate policies on the standard tool package. This caused confusion and delusional expectations on the course among some participants. Although the course conceptualized a first idea of conventions when to select which tool for which purpose, this issue would really need a longer time period to be internalized in the organization towards a more shared eCollaboration culture.

The group interaction among students and with the instructor flowed well, especially in the small group exercises of 2-6 people. The Communicator and Live Meeting sessions in small groups received clearly positive feedback from the participants. The larger presentations appeared slightly less interactive and attractive. However, for these presentations it helped if another meeting facilitator encouraged back-channel communication while the other was doing the presentation.

The ***technological and infrastructural*** issues also caused challenges. Several detail-level encounters with technologies and applications challenged the instructors and students. For example, Live Meeting usernames were restricted to 32 characters. Typically, e-mail addresses were automatically used as usernames, causing trouble for those with e-mail addresses exceeding the 32 characters. A few similar issues caused extra work for the instructors, participants and technology support.

As the technical issues emerged, a related issue was the service-level agreement (SLA) between the instructor organization, the participants' organizations and IT support. The SLAs defined that a bug-fixing could take several hours or days, which did not correspond to the often immediate need to get the eCollaboration tools to work for the meetings and sessions at hand. This applies not only to the learning setting, but eCollaboration adoption in general and distributed meetings in particular.

The physical office arrangements also had an impact. Some participants were located in open landscape offices, while moving to the "silent rooms" during the meetings; some were joining others in meeting rooms. In a few meeting rooms, if the user had not logged-in to the meeting room PC before, the log-in installation could take 15 minutes. A good practice was to move with one's own laptop to the silent rooms where no installation time was needed. However, such practices need to be included in the general-level eCollaboration conventions and best practices.

Network disturbances caused little problems. For example, web-based Live Meeting sessions can always be accessed again after the web connection is recovered. MS Office Communicator is slightly more complex to manage, requiring activity from those still in the meeting to invite the lost person(s) back.

The technical documentation produced by the IT professionals was insufficient to be used as learning guidelines for eCollaboration. The vendor courses, e.g. on Live Meeting, were less suitable for the corporate context, as the corporate use of the tools deviated from that recommended by the vendor. Moreover, while videoconferencing had been widely installed, its education was mainly outsourced to the vendor – but a responsibility for the vendor-provided training courses remained unclear.

The ***organizational*** changes made training of videoconferencing an ad hoc issue, as the experts of this area had left the organization or moved to other duties. Ambiguity of responsibilities and resources hindered the development of the videoconferencing module – while many still regarded the technology as underutilized in relation to its full potential. Further responsibility of eCollaboration training was delegated to a person with a load of other duties as well. Despite of top management statements of the eCollaboration importance, this indicates lack of resource backing from the top management to focus eCollaboration as a strategically important issue. While the person responsible for eCollaboration training continued to give the class-room oriented teaching of the course content in the headquarters, the necessary resources to promote the on-line offer elsewhere remained absent.

7 Discussion

In light of recent reviews on participative, professional on-line CSCL [12], [13], this study represents a rare case of corporate training in distributed, synchronous eCollaboration, mediated through the eCollaboration tools in daily use in the corporation. In this section we discuss the implications of our findings related to the process of establishing a sustainable service for corporate training in eCollaboration.

7.1 Promoting Awareness of the Benefits of eCollaboration

Former research has shown how adoption of eCollaboration practices requires an awareness of potential benefits for the individual users [1], [21], [24]. As well, our experiences show that participation in eCollaboration requires a vested interest in learning effective use of these tools. However, the individual benefits of eCollaboration may often be different from the organizational benefits [21]. Hence, the course setting should encourage each participant through examples and self-reflection to make their own case of self-motivation. This study did not find individual time saving to be a sufficient motivator for course participation by itself. More likely, the "acceptance management" of the eCollaboration setting [21] can be related to personal incentives to use eCollaboration with regard to a personal career and work development plans and frequent needs to host virtual meetings at work.

When participating in on-line training sessions in the work context a clear psychological challenge is to prioritize the on-line sessions higher than the emerging work tasks. On the other hand, if the classroom option exists locally, people seem to prefer this [23], thus decreasing the hands-on impact of learning. Again, the attendance during the professional master course was high whereas the pilot course participants prioritized the course sessions perhaps less. This confirms the importance of longer term vested interest [23] to achieve attendance from one's own desk.

Presence or absence of vested interest should be somehow mapped by the instructor among the participants in order to select whether to give a fully distributed on-line course or a more classroom-oriented instruction. However, when wireless local networks are available, course participants may gather in classrooms and auditoriums with their laptops to try out the tools hands-on.

7.2 Handling Varying Pre-skills in eCollaboration among the Participants

The technology learning curve for virtual meetings has been recognized as a challenge [23], [24], and our study illustrates the recurring start-up problems in the synchronous meeting sessions due to lack of experience from part of the participants. This was found to cause psychological challenges [24] among the more experienced participants. The instructor or facilitator preparation for the phenomenon helped little, as the instructor recommendations to log-in beforehand were often ignored. However, the phenomenon awoke less drastic reactions among the 2008 master course participants, who had greater and longer-term vested interest embedded in the course setting. The pilot course audience was more polarized with regard to their reactions to the idle time. This indicates how a vested interest may mitigate destructive reactions to the start-up problems of the on-line sessions.

The modularity helped tackle the variety of pre-skills in eCollaboration among the participants. The conceptualization of the course through integrated use scenarios of the tools convinced a few participants to take even modules involving media they had thought to master from before.

7.3 Establishing Conventions for eCollaboration Tools and Use Scenarios

Our findings confirm the previously observed challenge related to the variation in conventions of when to use what eCollaboration tool or media among the multiple available communication channels [23]. While the project team suggested an initial list of use scenarios as a basis for organizing the course, such conventions still varied and evolve over time in different parts of the corporation. In the worst case, “corporate standards” for learning eCollaboration may appear less relevant in a few subsidiaries of a global corporation. This requires coordination of the issue by the eCollaboration course champions and in-depth knowledge of the organization and the evolving eCollaboration technologies.

7.4 Including eCollaboration as Part of the Corporate Learning Strategy

At the organizational level, eCollaboration had not yet been defined as an inherent part of the corporate learning strategy, falling somewhat outside of the traditional human-computer e-learning and the pure classroom settings. On the other hand, the master course given solely through the eCollaboration tools was well-received. Instead of providing courses plainly on learning eCollaboration *per* se, an alternative strategy could be to embed eCollaboration as media for professional eLearning. This would also create a setting in which to establish longer-term team commitment and better vested interest [23], giving true motivation to learn the tools aside the actual course content in question. However, this would require a more explicit positioning of synchronous eCollaboration as a new element in the eLearning strategy, which has not been common in contemporary CSCL research [12]. This approach highlights the need to systematically “train the trainers” to educate also through synchronous on-line eCollaboration rather than only through asynchronous eLearning or classroom activities.

Despite of the top management’s expressed support of eCollaboration, the organization simultaneously cut the learning, development, and management resources to enhance eCollaboration. During the management re-structuring in winter 2008-09 responsibilities for enhancing eCollaboration in general and videoconferencing in particular remained vague, if not absent. This was paradoxical, as both the organization's own practitioners and previous research had for long recognized that corporate adoption of eCollaboration [21] and the distance education approach as well [25] require explicit facilitation with well-defined management responsibilities. On the other hand, evaluation of investments in eCollaboration in general has for long been identified as a difficult challenge [21]. It was still a challenge to come up with clearer business cases to catch the awareness of top management about the related needs and challenges. These findings resonate well with previous cases of implementation of eCollaboration [1]. In Boeing, the challenge was the transition from a pilot of distributed meeting service to establishing this as a 'regular' IT service, as the IT organization first did not want to service this. In Statoil a key challenge was to establish a sustainable support team of eCollaboration experts/facilitators. These persons tended to quickly be 'promoted' to other duties, making the support function rather vulnerable [1].

7.5 Aligning eCollaboration Competencies to Corporate IT Infrastructure, Policies and Operations

Alignment of distance learning competencies to corporate IT infrastructure, policies and operations has previously been identified as a challenge in professional contexts [25]. Such decisions as the drastic decommissioning of Net Meeting to replace or upgrade the eCollaboration application portfolios should be coordinated with targeted efforts to improve eCollaboration competencies.

The IT security and bandwidth restriction policies, and cost-saving by delaying the latest proprietary offerings, had lead to the tailored implementation of the Microsoft application infrastructure. These issues posed challenges to the learning programs of eCollaboration, as the organization-specific adjustments decrease the validity of the standard vendor course offerings, increasing the need for more detailed, organization-specific course concepts. Finally, the centralized IT services and the contemporary SLAs posed a challenge with regard to immediate bug-fixing and on-line technological support. All these issues need to be coordinated in collaboration among the eCollaboration training champion(s) and the IT service representatives.

8 Conclusion

While technology for eCollaboration is maturing, a need for learning eCollaboration in distributed organizations continues. This paper has presented a project to enhance corporate learning of synchronous, distributed eCollaboration through the very same media to be learned. The study contributes to the scarce body of professional CSCL research by reporting practical experiences from an on-line pilot course on eCollaboration. Moreover, the article specifies learning from the experiences in the form of psychological, social, technological and organizational issues which are discussed in light of the previous literature, and suggested to contribute to success of corporate eCollaboration learning programs. The study addresses challenges in the corporate context to promote awareness of the benefits to learn eCollaboration, to handle varying pre-skills of employees intended to participate in synchronous and distributed eCollaboration, to establish common conventions for how to select and use tools for particular organizational eCollaboration scenarios, to include synchronous and distributed eCollaboration media in the corporate eLearning strategy, and to align learning of eCollaboration competencies to ever-evolving corporate eCollaboration infrastructure, policies and IT operations. We believe that these experiences are of interest for other organizations and e-Learning professionals with similar intentions, as well as for researchers studying and conceptualizing such practices.

References

1. Munkvold, B.E.: Implementing Collaboration Technologies in Industry: Case Examples and Lessons Learned. Springer, London (2003)
2. Orlikowski, W.: Learning from Notes: Organizational Issues in Groupware Implementation. In: Mantel, M., Baecker, R. (eds.) Proceedings of the 1992 ACM conference on Computer-supported cooperative work, pp. 362–369. ACM, New York (1992)
3. Riemer, K.: The Market for E-Collaboration Systems - Identification of System Classes Using Cluster Analysis. In: Österle, H., Schelp, J., Winter, R. (eds.) Proceedings of the Fifteenth European Conference on Information Systems, pp. 346–357. University of St. Gallen, St. Gallen (2007)
4. Mittleman, D.D., Briggs, R.O., Murphy, J., Davis, A.: Toward a Taxonomy of Groupware Technologies. In: Briggs, R.O., Antunes, P., de Vreede, G.-J., Read, A.S. (eds.) CRIWG 2008. LNCS, vol. 5411, pp. 305–317. Springer, Heidelberg (2008)
5. Munkvold, B.E., Zigurs, I.: Integration of e-Collaboration Technologies: Research Opportunities and Challenges. International Journal of e-Collaboration 1(2), 1–24 (2005)
6. Munkvold, B.E.: Experiences from Global e-Collaboration: Contextual Influences on Technology Adoption and Use. IEEE T. Prof. Commun. 48(1), 78–86 (2005)
7. Dillenbourg, P.: What Do You Mean by Collaborative Learning? In: Dillenbourg, P. (ed.) Collaborative-Learning: Cognitive and Computational Approaches, pp. 1–19. Elsevier, Oxford (1999)
8. Hórreo, V.S., Carro, R.M.: Studying the Impact of Personality and Group Formation on Learner Performance. In: Haake, J.M., Ochoa, S.F., Cechich, A. (eds.) CRIWG 2007. LNCS, vol. 4715, pp. 287–294. Springer, Heidelberg (2007)
9. Stahl, G.: Groupware Goes to School. In: Haake, J.M., Pino, J.A. (eds.) CRIWG 2002. LNCS, vol. 2440, pp. 7–24. Springer, Heidelberg (2002)
10. Smith, P.J.: Workplace Learning and Flexible Delivery. Rev. Educ. Res. 73(1), 53–88 (2003)
11. Maushak, N., Ou, C.: Effect of Synchronous Communication on Graduate Students' Online Collaboration. In: Proceedings of the Education International Conference (SITE), pp. 515–520. AASE, Chesapeake (2005)
12. Maor, D., Volet, S.: Interactivity in Professional Online Learning: A Review of Research Based Studies. Australas. J. Educ. Tec. 23(2), 269–290 (2007)
13. Hrastinski, S.: What is Online Learner Participation? A Literature Review. Comput. Educ. 51, 1755–1765 (2008)
14. Malhotra, A., Majchrzak, A.: Virtual Workspace Technologies. Sloan Manage. Rev. 46(2), 11–14 (2005)
15. Nunamaker, J.F., Reinig, B.A., Briggs, R.O.: Principles for Effective Virtual Teamwork. Commun. ACM 52(4), 113–117 (2009)

16. Pinsonneault, A., Caya, O.: Virtual Teams: What We Know, What We Don't Know. International Journal of e-Collaboration 1(3), 1–16 (2005)
17. Alavi, M., Marakas, G.M., Yoo, Y.: A Comparative Study of Distributed Learning Environments on Learning Outcomes. Inform. Syst. Res. 13(4), 404–415 (2002)
18. Vygotsky, L.S.: Mind in Society. The Development of Higher Psychological Processes. Harvard University Press, Cambridge (1978)
19. Hrastinski, S.: The Potential of Synchronous Communication to Enhance Participation in Online Discussions: A Case Study of Two e-Learning Courses. Inform. Manage. 45, 499–506 (2008)
20. Hrastinski, S., Keller, C., Carlsson, S.A.: Towards a Design Theory for Synchronous Communication in Online Education. In: Chatterjee, S., Rossi, M. (eds.) Proceedings of the Second International Conference on Design Science Research in Information Systems and Technology (DESRIST 2007), pp. 208–224. CGU, Pasadena (2007)
21. Grudin, J.: Groupware and Social Dynamics: Eight challenges for developers. Commun. ACM 37(1), 92–105 (1994)
22. Angehrn, A.A., Nabeth, T.: The L2C Project: Learning to Collaborate Through Advanced SmallWorld Simulations. In: Nejdl, W., Tochtermann, K. (eds.) EC-TEL 2006. LNCS, vol. 4227, pp. 452–457. Springer, Heidelberg (2006)
23. Mittleman, D.D., Briggs, R.O., Nunamaker, J.F.: Best Practices in Facilitating Virtual Meetings: Some Notes from Initial Experience. Group Facilitation: A Research & Applications Journal 2(2), 5–14 (2000)
24. Hughes, S.C., Wickersham, L., Ryan-Jones, D.L., Smith, S.A.: Overcoming Social and Psychological Barriers to Effective On-Line Collaboration. Educ. Technol. Soc. 5(1), 86–92 (2002)
25. Berge, Z.L.: Obstacles to Distance Training and Education in Corporate Organizations. Journal of Workplace Learning 14(5), 182–189 (2002)
26. Briggs, R.O., Davis, A.J., Murphy, J.D., Steinhauser, L., Carlisle, T.F.: Transferring a Collaborative Work Practice to Practitioners. In: Haake, J.M., Ochoa, S.F., Cechich, A. (eds.) CRIWG 2007. LNCS, vol. 4715, pp. 295–302. Springer, Heidelberg (2007)

Integrating Spatial Data and Decision Models in an E-Planning Tool

Pedro Antunes[1], Claudio Sapateiro[1], Gustavo Zurita[2], and Nelson Baloian[3]

[1] University of Lisbon, Faculty of Sciences,
Campo Grande, Lisbon, Portugal
paa@di.fc.ul.pt, claudio.sapateiro@estsetubal.ips.pt
[2] Universidad de Chile, Economy and Business School,
Diagonal Paraguay 257, Santiago de Chile, Chile
gzurita@ing.puc.cl
[3] Universidad de Chile, Engineering School,
Blanco Encalada 2120, Santiago de Chile, Chile
nbaloian@gmail.com

Abstract. We review several decision models to derive six fundamental requirements to collaborative spatial decision-making: perceiving changes in spatial data; retaining interpretation mindsets; externalizing actions and expectancies in spatial data; organizing divergent and convergent working modes; supporting the recognition of situation-action elements; and managing task/pattern structures. A review of the current state of the art shows limited support to some of these requirements, in particular task/pattern and divergent/convergent support. An e-planning prototype was developed to demonstrate the impact of these requirements in collaborative spatial decision-making. Results from a preliminary experiment indicate the prototype enables people to contribute, explain, exteriorize and share their ideas in relation with spatial references.

Keywords: Geocollaboration, Decision Models, e-Planning.

1 Introduction

This research was motivated by the design of a collaborative tool supporting e-planning. E-planning is the label adopted by a broad research agenda addressing the interaction between information technology and planning, including various key concerns such as territorial management, policy making, governance, citizenship and participation [1].

The main vision driving the tool design was supplying various stakeholders – architects, urban designers, city planners and public administrators – with a collaborative tool capable to advance new perceptions and ideas regarding city planning. City planning is a complex process challenging design with a variety of technical and human requirements. In our view, the best approach to a wicked situation like this one is starting highly focused on a very specific challenge and only moving forward when that challenge is sufficiently understood and conveniently resolved.

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 97–112, 2010.

In our case study, the main challenge is integrating spatial data with the decision model. City planning involves various types of geographically related data. This data is traditionally managed with Geographical Information Systems (GIS).

The integration of spatial data with decision models is not new. Actually, it has lead to an emerging category of GIS designated Collaborative Spatial Decision Making (CSDM) [2, 3]. CSDM may also be regarded as a combination of GIS with Decision Support Systems (DSS) and Group Support Systems (GSS) [4, 5], although falling outside the typical DSS/GSS categorization [6]. According to the state of the art, CSDM concerns the provision of the following functionality [7]: collecting spatially-related data, identifying locations according to a set of criteria, exploring relationships, displaying and analyzing data, and exporting data to other systems and tools.

While this functionality is essential to integrate spatial data with decision-making activities, it does not address some specific problems: (1) it does not explicitly consider models of the decision-making process, which means its potential users will have to informally manage the process; (2) in complex contexts, decisions are highly dependent on collaboration, which requires adding support to coordination, awareness and collaborative visualization into the CSDM functionality; and finally, (3) decision making also brings new types of spatially-related data, such as talks, discussions, negotiations, and brainstorms, which should be seamlessly integrated with the remaining data.

We may express with more accuracy that our main challenge is modeling spatial data within the context of a broader model of the decision-making process, understood as a collaborative endeavor. The paper is organized as follows. We start with a review of several decision-making models to highlight the main model constructs that inform CSDM design. We then review several CSDM tools to highlight present omissions and opportunities. In Section four we describe the e-planning tool developed to explore the integration between spatial data and decision models. In section five we discuss the tool's evaluation. We conclude the paper with a synthesis and discussion of the obtained results.

2 Overview of Decision-Making Models

2.1 Conceptual Views

In Figure 1 we present three conceptual views of the decision-making process. They may be regarded as meta-models, since they serve to build other models. The first view regards the decision process as a production system having three components: inputs, process and outputs [8]. The process component concerns social interaction with support from technology in three main forms: decision aids; managing the decision process; and adoption of emerging structures to enhance decisions. This conceptual view is highly prevalent in the research field [9].

The second view regards decision making as a composition of data management, model management and dialogue management [10]. Of most importance in this view is model management, which is responsible for controlling the strategic, tactical and operational decisions of the decision makers through technology support.

The third model was originally proposed by Seligmann et al. [11] and later on adopted by Vreede et al. [12] to conceptualize the different aspects that set up the technological support to the decision-making process. The way of thinking concerns thinking about the application domain, while the way of controlling concerns the design approach that follows problem conceptualization. Design is then dependent on two other constructs: the way of working, i.e. how people carry out their activities; and the way of modeling, i.e. the representations necessary to support the way of working. These models highlight that decision models coexist in a complex context characterized by process inputs and outcomes, competing data and dialogue models, and a difficult balance between design constraints and ways of working.

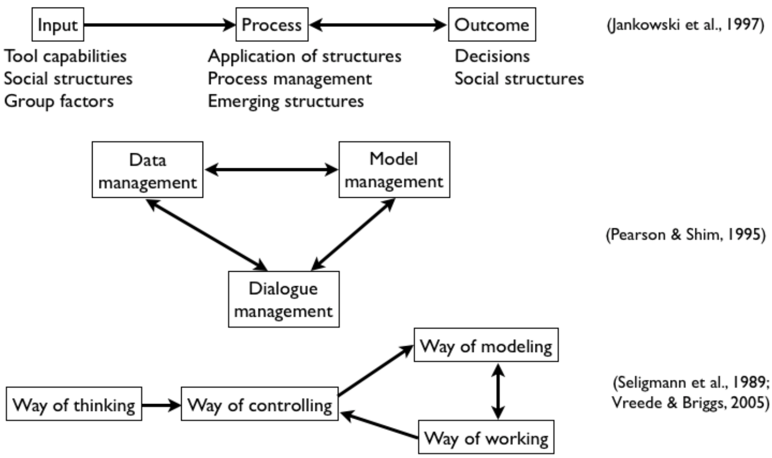


Fig. 1. Conceptual views

2.2 Models of the Cognitive Process

The models of the cognitive process regard decision-making as a cognitive function. One of the most famous models was proposed by Card et al. [13]. It regards the cognitive function as a machine where stimuli ignite perceptual activities, followed by cognitive and motor activities, which in turn originate new stimuli. In this model, decision-making is a cyclic endeavor continuously supported with feedback information.

This model has been highly influential, the reason why many other models tend to reflect the same information processing view, with most differences centered on the cognitive task. For instance, the Contextual Control Model [14] adopts a similar cyclic view, although with the addition of disturbances, which are fundamental to understand human behavior facing the unexpected. The Reference Model of Cognition [15, 16] extends the cognitive component with interpretation and planning components. The Step Ladder Model [17] also extends the cognitive component with identification, interpretation, task definition and planning components.

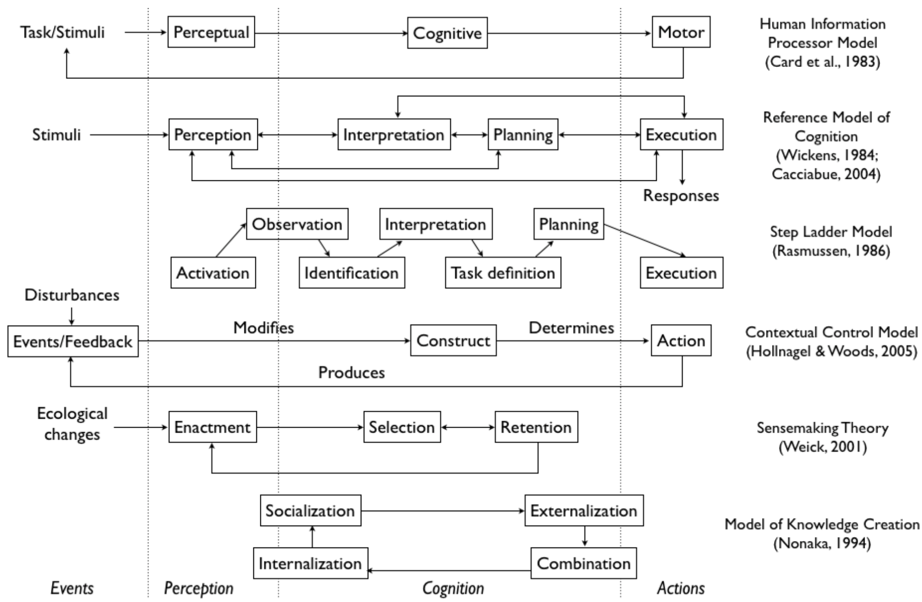


Fig. 2. Models of the cognitive process

Two cognitive approaches that depart away from the perceptual-cognitive-motor mechanics are the Sensemaking theory and the Model of Knowledge Creation. The Sensemaking theory [18] seeks to understand how humans deal with information through their equivocal perceptions and cognitive biases. Ecological changes enact perception according to commitment and interpretation mindsets. Some cues are selected, made intelligible and retained. Retention is important to understand how organizations learn. Perhaps the main conceptual change proposed by sensemaking, when compared to the previous models, is that it moves away from reproducing an information-processing machine towards a more ambiguous cycle, highly dependent on retention.

The Model of Knowledge Creation [19] seeks to understand how humans utilize tacit and explicit knowledge. Knowledge is transformed from tacit (in the mind) to explicit (in the world) through a cycle of data socialization, externalization, combination and internalization. This model highlights the main differences between individual (internalization and externalization) and group (socialization and combination) functions.

In Figure 2 we present a visual representation of the reviewed models. We note that this representation is necessarily incomplete. It primarily serves to highlight that the decision-making process seems to be grounded on four main theoretical constructs: events, perception, cognition and actions. The main differences posited by these models seem to be centered on the cognition construct. In the next section we will further analyze this particular construct.

2.3 Models of the Decision Process

Likewise the cognitive process, many models have been proposed to explain the decision process. One highly influential perspective is the Subjective Expected Theory [20, 21]. This theory considers that rational decision makers, when facing a set of alternatives and outcomes, define utility functions to determine which choices should be elected. This theory is the basis for what has been designated normative approach to decision making under uncertainty [20].

Other theories have followed the normative approach, e.g., Analytic Hierarchy Process (AHP) [22]. AHP commends four major steps in making decisions: break down the problem into a hierarchy of decision elements; collect data regarding these elements using pair-wise comparisons; estimate the relative weights of decision elements; and aggregate the relative weights to obtain a set of ratings for the decision alternatives.

Simon [23, 24] criticizes the normative approach for its perfect utility-maximizing rationality, emphasizing that in real-world organizations decision makers do not find the perfect conditions necessary to frame problems. Simon proposed the Problem Solving Model with three main elements: representing the problem, finding alternatives and selecting alternatives (often designated intelligence, design and choice). Two other distinctive concepts in this model are heuristics and the notion of satisficing. Heuristics explain why decision makers often simplify the problem space by applying means-ends analysis, compromises, time constraints and even rules of thumb. The notion of satisficing explains that often the decision makers do no aspire to maximize utility but instead seek to find out a solution that satisfies reasonable conditions.

The Recognition Primed Decision Making theory (RPDM) [25] introduced the naturalistic perspective over decision making [26]. This perspective distinguishes itself from the previous approaches by trying to understand how time pressure, uncertainty, ill-defined goals and other factors affect the decision makers. Instead of trying to define how to make decisions, the naturalistic perspective seeks to understand how decisions are actually made. RPDM thus stresses three fundamental components of decision-making: experience the situation, recognize and classify, and react. This theory also brings forward the concept of situation awareness as a mechanism to apprehend expectancies, cues, goals and actions.

Besides the rational-versus-organizational-versus-naturalistic debate briefly described above, many other theories seek to explain more specific conditions underlying the decision-making process. For instance, the Cooperative Decision Making model [27] emphasizes the importance of negotiating conflicts. The Participatory Decision Making model [28] distinguishes between divergent and convergent collaboration modes. The Soft Systems Methodology (SSM) [29] proposes a conceptual approach to problem solving based on action research, coping iteratively with problem complexity while at the same time avoiding reductionism. SSM highlights action as a fundamental driver for problem solving, instead of analysis and structure. And finally, Collaboration Engineering [30] synthesizes decision-making as a collection of behavioral patterns that may be "engineered" to respond to contextual situations.

In Figure 3 we may observe the impact of the Problem Solving Model and it's threefold construct (intelligence, design and choice) on understanding the decision-making process. We also find a relative consensus that this logical construct is considered cyclical and not necessarily prescriptive.

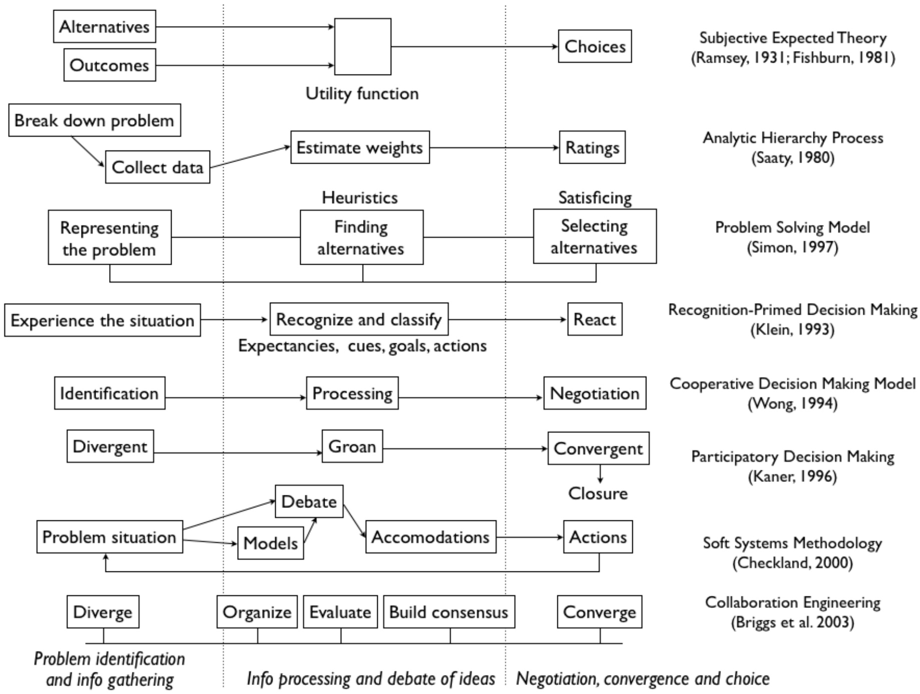


Fig. 3. Models of the decision process

2.4 Discussion and Synthesis

What can we distill from the above models? First, decision-making involves a constant flow of events, perception, cognition and action. Second, the cognitive activity seems to be organized according to three main patterns: (1) problem identification and information gathering; (2) information processing and debate of alternatives; and (3) negotiation, convergence and choice. And third, decision-making also involves data management, dialogue management and model management. Model management is fundamentally concerned with structuring the main patterns previously identified. Furthermore, model management is intimately related with the way of thinking, controlling and working. From this integrated perspective we may now derive some fundamental requirements to CSDM:

Perception support. Stimuli, disturbances, events and ecological changes are necessary to stimulate perception. CSDM should therefore associate changes in spatial data with adequate perceptual mechanisms, e.g., dynamic visualization, strategic and tactical views of spatial data and associated events.

Retention support. Retention is a fundamental driver of sensemaking. It serves to construct personal and organizational memory and contributes to enact responses whenever recognizable situations emerge. CSDM should maintain a repository of the interpretation mindsets and enacted responses in context with spatial data.

Externalization support. Externalization is essential to knowledge creation, since knowledge is constructed by articulating tacit knowledge into shared expectancies, cues, goals and actions. CSDM should therefore provide support for integrating tacit knowledge with spatial data.

Divergent/convergent support. Decision-making seems to be organized according to intertwined cycles of divergent and convergent activities, where divergent activities favor problem identification and information gathering, and convergent activities promote the negotiation and selection of alternatives. CSDM should support these working modes.

Recognition support. Recognition and classification play a fundamental role in the naturalistic approach to decision-making. Recognition prepares the ground for empirical decisions based on situation awareness and pattern matching. CSDM should therefore provide situation-action support by linking spatial data with expectancies, cues, goals and actions defined by the decision makers.

Task/pattern management. The decision-making process seems to be organized according with patterned activities like divergence, convergence, data organization, option evaluation, etc. Several theories posit these patterned activities are iterative/cyclic and may not follow a prescriptive or recommended structure. Thus, although CSDM should carefully avoid prescribing rigid structures, they should also support the way of controlling by implementing task/pattern management.

3 Literature Analysis

We adopted the following approach to analyze CSDM literature. First, we started by collecting papers published in journals, conferences and workshops on the subject of enabling working on spatial data while simultaneously making collaborative decisions [31]. This task allowed us to identify a set of 61 papers. We then applied a set of criteria to reduce our review to: (1) papers published from 2000 to 2009; (2) papers focused on the analysis, design, development and evaluation of CSDM applications; and (3) excluding papers centered on literature review, framework and theory development, infrastructure support to CSDM and specific application scenarios. A total of 18 papers were found to fulfill these criteria. We then elected 10 papers as most representative of current CSDM. Table 1 summarizes the elected papers.

From this overview we may draw some observations and comments. Our first observation is that none of the reviewed CSDM tools addresses task/pattern management. Actually, only [2] refers to the importance of managing decision-making tasks, although such functionality is not implemented in the prototype.

Another issue is related with divergent/convergent support. Most reviewed CSDM tools support either convergent or divergent activities, with only three cases supporting both modes [36-38]. But more interestingly, the tools supporting both modes do so in a transparent way, i.e. the users may converge and diverge according to factors such as network connectivity or interaction with private and public spaces. These tools do not explicitly define if work is divergent or convergent according to the specific task at hand.

Table 1. Most representative papers of current CSDM

	Convertino et al. 2005 [32]	**Rinner 2006 [33]**
Perception	Workspace metaphor, multiple views, filters, activity awareness indicators, change icons	Hypermap metaphor
Retention	Historical records	Retains geo-argumentative relations
Externalization	Has chat, editor and interactive map	Argumentation model
Divergent/convergent	Convergent (tactical planning)	Divergent (early phase)
Recognition	Annotations and visual landmarks	Annotations and visual landmarks
Task/pattern manag.	Only supports roles	
	MacEachren et al. 2006 [34]	**Bortenschlager et al. 2007 [35]**
Perception	Desktop metaphor	Regular updates
Retention		
Externalization	Speech and gesture recognition	Overlays
Divergent/convergent	Convergent (large whiteboard)	Convergent (using mobile devices)
Recognition	Incident markers	
Task/pattern manag.		
	Convertino et al. 2008 [36]	**Capata et al. 2008 [37]**
Perception	Sidebar, telepointer, role indicators	Object push
Retention		
Externalization	Notes, scribbles, symbols	Geographical features
Divergent/convergent	Both (shared and private workspaces)	Both (using mobile devices)
Recognition		
Task/pattern manag.	Only supports roles	
	Brewer et al. 2000 [38]	**MacEachren et al. 2004 [2]**
Perception	Depict change over time, gestures, member behavior, flash regions	Split views, member behavior, avatars, pointing gestures
Retention	Activity logging	
Externalization		Drawing and selection tools
Divergent/convergent	Both	Convergent
Recognition		
Task/pattern manag.		Defines exploration, analysis, synthesis and presentation tasks, but does not implement
	Cai 2005 [39]	**Torino et al. 2001 [40]**
Perception	Change propagation	
Retention		
Externalization	GIS workspace, group summary	Stands and seeds (markers)
Divergent/convergent	Convergent (large displays)	Divergent (using shared database)
Recognition	Marking	Conflict detection (with markers)
Task/pattern manag.		

Most tools do not support retention, with few exceptions supporting activity logging and historical records [32, 33, 38]. Perception has received significant attention, with multiple mechanisms being available. The recognition support is apparently less rich. Several tools support annotations and markers [32, 33, 39, 40] but miss more strategic features linking spatial data with expectancies, cues, goals and actions. Finally, externalization combines GIS features with common groupware functionality like chat, text editing and argumentation. This review clearly indicates our research and development efforts should be centered on the support to: (1) task/pattern management; (2) explicit convergent/divergent collaboration modes; (3) retention; and (4) recognition.

4 E-Planning Tool

Like most CSDM tools reviewed in the previous section, the e-planning tool has a workspace allowing visualizing and interacting with a map. This map may be complemented with spatially related visual objects like sketches, drawings and free-hand text, collaboratively produced by the users to enhance their **perception**.

Unlike the other tools, we also support **task/pattern management**. This is implemented with multiple workspaces, targeted to specific tasks/patterns (see Figure 4). The set of tasks/patterns was derived from the Problem Solving Model: (1) gathering, (2) debate, and (3) choice. We stipulate that all users operate in the same workspace, but they may collectively change the current workspace whenever necessary. This avoids a prescriptive approach to making decisions. Mini-maps allow visualizing the three workspaces and also serve to select the current workspace.

Externalization is supported with sketching, drawing and text writing in the workspace. **Retention** is based on logging changes to the visual objects present in the workspace, allowing the users to move back and forth the timeline. To support **recognition**, the tool allows selecting annotations (sketches, drawings, text) from one workspace and dragging them to another workspace (using the mini-maps).

We define the gathering workspace is **divergent** and the debate and choice workspaces are **convergent**. This allows users' free whiling and divergent thinking while gathering information, but requests the users' focus while debating and choosing options.

The tool runs on tablet computers and may be used in several physical configurations, including a set of interconnected tablets, one large whiteboard or a combination of both. Our prototype uses SMARTech's SmartBoard. The prototype adopts a fully replicated architecture and is heavily based on pen-based gestures to interact with the user interface [41]. When a replica is started in a tablet, it automatically establishes an ad-hoc network with the other tablets and synchronizes all spatial data.

The tool's user interface is shown in Figure 4. The current workspace is shown on the left handside. The participants may use the pen to sketch and write over the map. The mini-maps are shown to the right. They support two functions. One is moving the group's focus of attention to a different workspace (the dark background color indicates what workspace is currently selected). As previously mentioned, three different workspaces are supported. The one on the top is the gathering workspace, the one in the middle is the debate workspace, and the lower one is the choice workspace.

Another important functionality is indexing the data elements created over the map. Each index entry has a set of sketches consecutively made by one user. In Figure 4, the gathering and debate workspaces show two index entries each. These indexes simplify the selection and edition of individual data elements using gestures. The mini-maps support vertical scrolling but do not use a scrollbar to preserve space.

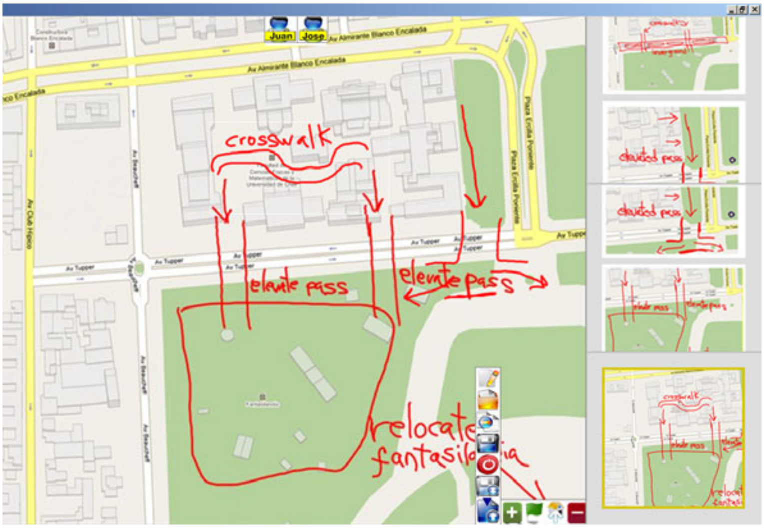


Fig. 4. The e-planning tool

The data elements may be copied from one workspace to another. The copy and paste operations are performed at a granularity that considers index entries, not individual sketches. The origins and destinations of the copy/paste operations are either the current workspace or the mini-map. The pasted data elements maintain their spatial references. It is possible to scroll and zoom over the map and related data elements. The icons located on the center-top of the screen provide awareness on who is currently using the tool.

We now describe in more detail the functionality associated with each workspace. The **gathering workspace** operates in a divergent mode. This means the workspace is private and the data elements sketched over the map are not shared with the group. This collaboration mode allows users to prepare their ideas before sharing them with the others. When necessary, a user may share a particular data element with other users. Selecting the corresponding index entry and dragging it to the users' icons shown at the center-top of the screen accomplish this.

The **debate workspace** serves to explore and refine ideas. This is a convergent task involving all users. Initially this workspace only shows the map, but allows users copying index entries from their individual gathering workspaces using pen-based gestures. In this way the users may share, organize and refine the set of common ideas.

The **choice workspace** operates in a very similar way. It is initially empty and may be populated by copying index entries from the debate workspace. This workspace is intended to develop a final visual representation of the decisions made by the group.

5 Case Scenario and Evaluation

The main challenge addressed by this research was supporting decision making within the spatial context. We regarded this challenge from a design science perspective, i.e., seeking to extend human and organizational capabilities through innovative artifacts [42]. It thus makes sense to also evaluate the proposed solution using a design science approach. According to Hevner et al. [42], design solutions must be justified/evaluated with the twofold purpose to improve artifacts and develop theory. One possible evaluation approach consists in evaluating the utility of the design artifact using controlled experiments. In this section we report a preliminary field trial with the e-planning tool.

We asked a team of three people to perform an e-planning task considering the plans of the municipality of Santiago to merge the area occupied by the Faculty of Engineering of the Universidad de Chile with the nearby-located park, now separated by an avenue. The task was divided in two sessions during which the team had to use the e-planning tool to generate ideas on how to create a continuous area from the faculty to the park.

During the first session, two team members worked in the field using their Tablet PCs, documenting their ideas in the gathering workspace. In the second session, they joined the third member in the office. The third member used a SmartBoard while the other two members kept using their Tablet PCs (Figure 5).

After synchronizing the applications, the members started exposing their solutions and discussing using the debate workspace. New alternatives were generated and indexed in this workspace. Finally, in the choice workspace all members collaboratively developed the final solution, which in fact was a merge of the two proposed solutions.

Fig. 5. Using the e-planning tool in the second working session

In the end of the experiment we asked the participants to analyze the prototype usage in the predefined scenario and come up with comments and observations regarding its utility. The discussion confirmed the divergent collaboration mode is beneficial to the initial decision stages, where the decision makers seek to generate ideas. Externalization was considered adequately implemented by the prototype. It was explicitly noted the sketches helped exteriorizing and sharing tacit knowledge.

The pen-based gestures were considered easy to use, although more traditional interaction modalities based on mouse and keyboard were also requested. The choice workspace was perceived as the most helpful one because it is focused on bringing the group towards the task goals and, at the same time, allows importing information from the other tasks in a flexible way. The debate workspace was also perceived as very helpful to organize ideas through sketches and concept maps.

Overall, the prototype was perceived as relevant to e-planners because it enables people to contribute, explain, exteriorize and share their ideas in relation with spatial references. Nevertheless, the participants suggested improving the recognition abilities, considering a major challenge the implementation of adequate awareness mechanisms. Finally, the participants considered the learning curve was adequate, commenting they were adapted to the prototype during the second session.

6 Synthesis and Discussion

The main contributions of this work include an integrated perspective of the relationships between spatially related data and decision-making. To build this integrated perspective, we analyzed an extensive pool of models explaining the cognitive behaviors associated with decision-making. We then distilled six requirements: perception, retention, externalization, recognition, divergent/convergent collaboration modes, and task/pattern management. Perception emphasizes a cognitive view over the decision-making process. It brings forward the need to convey spatially-related data in a way that stimulates decision makers to perceive and enact cognitive functions like identification, interpretation, selection, task definition, planning, externalization, action, etc.

Retention addresses the withholding of personal and organizational experiences, resulting from the confrontation between events and actions, interpretations, choices and other constructs. Retention is a fundamental driver for sensemaking, and sensemaking is a fundamental driver for making decisions. In the CSDM context, we understand the retention requirement as the need to preserve decisions, decision constructs and spatial data in a coherent framework that promotes learning and recall.

Externalization brings forward the view that decision-making is a collective endeavor and knowledge must be transformed from tacit to explicit. This signals that decision makers should be able to collaboratively manage spatially related data. Recognition is closely associated with a naturalistic view over decision-making where emergence, time pressure and uncertainty give the fundamental context to understand choices. In the CSDM context, this requirement renders the collaborative creation of annotations, visual marks and other spatially-related visual elements that contribute to react to evolving situations.

The divergent/convergent view brings forward the understanding that teams must devise strategies to optimize collective tasks. Often the best strategy is focusing on the same task, while in other cases is having the participants working independently. We regard flexible management of collaboration modes a fundamental requirement of CSDM.

The consideration for task/pattern management highlights the view that decision makers should be able to control the tasks necessary to reach their goals. In the CSDM context, this means that task/pattern management should be explicitly available, although avoiding prescribed procedures.

Our review of the state of the art shows that existing CSDM tools offer adequate levels of perception, retention and externalization support. However, divergent/convergent and task/pattern support seems to be underdeveloped. Of course we had to verify if these requirements would have some concrete impact on CSDM design. We developed an e-planning tool with that goal in mind. We codified the requirements into concrete functionality. In particular, we structured the tool in three working spaces specifically dedicated to support problem representation, finding alternatives and selecting alternatives. Each working space maintains the decision-making elements according to their spatial context. This functionality implements the task/pattern management requirement.

Of the three workspaces, one supports divergent activities while the other two support convergent activities. This decision was drawn from theory recommending the adoption of divergent activities during the preliminary decision phases and convergent activities during the later decision phases [28]. Divergence was implemented with private workspaces, while convergence relies on shared workspaces supporting concurrent data management. This functionality implements the divergent/convergent requirement.

The perception, recognition and externalization requirements were resolved with a set of visual elements the users may create and manipulate using pen-based gestures. These elements are spatially related since a map always exists in the workspaces. A team evaluated the tool in a laboratory experiment and considered it useful and usable.

Nevertheless, we should discuss some limitations we find in this study. One limitation is that we do not attempt to define a unifying decision-making theory. It may perhaps be attempted in a future work, and significant groundwork has already been done, for instance bringing together the cognitive and decision-making models. But this endeavor requires additional work to demonstrate the validity of its constructs. What we have done instead was focusing on design-oriented goals, deriving a set of requirements from the various theories. This approach is much more simple to validate: we just have to build a tool and justify its utility, as recommended by Hevner et al. [42].

Nevertheless the justification of the tool will require future work. More scenarios, experiments, participants and inquiries are necessary to validate it. We should also consider in the future moving out from the laboratory to the field, which will require developing further the prototype to improve its overall stability.

We also recognize that the perception, recognition and externalization requirements were underdeveloped when compared with the other requirements. Indeed our main focus was on the requirements we perceived as most neglected by the CSDM literature. But in retrospect we perceive that many contributions to better implement these requirements could be done in the future. Particularly, we may further explore the perception and recognition requirements in the context of team situation awareness [43]. This perspective may bring forward new technological mechanisms capable to improve the perception of the dynamics often associated to spatially-related data.

Acknowledgements. This paper was supported by the Portuguese Foundation for Science and Technology (PTDC/EIA/102875/2008) and Fondecyt.

References

1. Curwell, S., Deakin, M., Cooper, I., Paskaleva-Shapira, K., Ravetz, J., Babicki, D.: Citizens' Expectations of Information Cities: Implications for Urban Planning and Design. Building Research and Information 33, 55–66 (2005)
2. MacEachren, A., Brewer, I.: Developing a Conceptual Framework for Visually-Enabled Geocollaboration. International Journal of Geographical Information Science 18, 1–34 (2004)
3. MacEachren, A.: Moving Geovisualization toward Support for Group Work. In: Dykes, J., et al. (eds.) Exploring Geovisualization, pp. 445–462. Elsevier, Amsterdam (2005)
4. Nyerges, T., Montejano, R., Oshiro, C., Dadswell, M.: Group-Based Geographic Information Systems for Transportation Site Selection. Transportation Research C 5, 349–369 (1997)
5. Armstrong, M.: Requirements for the Development of Gis-Based Group Decision Support Systems. Journal of the American Society for Information Science 45, 669–677 (1994)
6. Arnott, D., Pervan, G.: A Critical Analysis of Decision Support Systems Research. Journal of Information Technology 20, 67–87 (2005)
7. Crossland, M., Wynne, B., Perkins, W.: Spatial Decision Support Systems: An Overview of Technology and a Test of Efficacy. Decision Support Systems 14, 219–235 (1995)
8. Jankowski, P., Nyerges, T., Smith, A., Moore, T., Horvath, E.: Spatial Group Choice: A Sdss Tool for Collaborative Spatial Decision-Making. International Journal of Geographical Information Science 11, 577–602 (1997)
9. Fjermestad, J., Hiltz, S.: An Assessment of Group Support Systems Experimental Research: Methodology and Results. Journal of Management Information Systems 15, 7–149 (1999)
10. Pearson, J., Shim, J.: An Empirical Investigation into DSS Structures and Environments. Decision Support Systems 13, 141–158 (1995)
11. Seligmann, P., Wijers, G., Sol, H.: Analysing the Structure of IS Methodologies. In: Proceedings of the 1st Dutch Conference on Information Systems, Amersfoort, The Netherlands (1989)
12. Vreede, G., Briggs, R.: Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks. In: Proceedings of the 38th Hawaii International Conference on System Sciences, Hawaii (2005)
13. Card, S., Moran, T., Newell, A.: The Psychology of Human-Computer Interaction. Lawrance Elrbaum, Hillsdale (1983)
14. Hollnagel, E., Woods, D.: Joint Cognitive Systems: Foundations of Cognitive Systems Engineering. CRC Press, Boca Raton (2005)
15. Wickens, C.: Engineering Psychology and Human Performance. Charles Merrill, Columbus (1984)
16. Cacciabue, P.: Guide to Applying Human Factors Methods. Springer, London (2004)
17. Rasmussen, J.: Information Processes and Human-Computer Interaction. An Approach to Cognitive Engineering. North Holland, Oxford (1986)
18. Weick, K.: Making Sense of the Organization. Blackwell, Oxford (2001)
19. Nonaka, I.: The Knowledge-Creating Company. Harvard Business Review, 96–104 (1991)

20. Fishburn, P.: Subjective Expected Utility: A Review of Normative Theories. Theory and Decision 13, 139–199 (1981)
21. Ramsey, F.: Truth and Probability. In: Ramsey, F. (ed.) The Foundations of Mathematics and Other Logial Essays, Harcourt, Brace & Co., New York (1931)
22. Saaty, T.: The Analytical Hierarchy Process. McGraw-Hill, Inc., New York (1980)
23. Simon, H.: Administrative Behavior: A Study of Decision-Making Processes in Administrative Organizations, 4th edn. Free Press, New York (1997)
24. Simon, H.: Decision Making and Problem Solving. Interfaces 17, 11–31 (1987)
25. Klein, G.: A Recognition-Primed Decision (Rpd) Model of Rapid Decision Making. In: Klein, G., et al. (eds.) Decision Making in Action: Models and Methods. Ablex, Norwood (1993)
26. Lipshitz, R., Klein, G., Orasanu, J.: Taking Stock of Naturalistic Decision Making. Journal of Behavioral Decision Making 14, 331–352 (2001)
27. Wong, S.: Preference-Based Decision Making for Cooperative Knowledge-Based Systems. ACM transactions on information systems 12, 407–435 (1994)
28. Kaner, S.: Facilitator's Guide to Participatory Decision-Making. New Society Publishers, Philadelphia (1996)
29. Checkland, P.: Soft Systems Methodology: A Thirty Year Retrospective. Systems Research and Behavioral Science 17, S11-S58 (1981)
30. Briggs, R., Vreede, G., Nunamaker, J.: Collaboration Engineering with Thinklets to Pursue Sustained Success with Group Support Systems. Journal of Management Information Systems 19, 31–64 (2003)
31. MacEachren, A., Cai, G., Sharma, R., Rauschert, I., Brewer, I., Bolelli, L., Shaparenko, B., Fuhrmann, S., Wang, H.: Enabling Collaborative Geoinformation Access and Decision-Making through a Natural, Multimodal Interface. International Journal of Geographical Information Science 19, 293–317 (2005)
32. Convertino, G., Ganoe, C., Schafer, W., Yost, B., Carroll, J.: A Multiple View Approach to Support Common Ground in Distributed and Synchronous Geo-Collaboration. In: Proceedings of Third International Conference on Coordinated and Multiple Views in Exploratory Visualization, London, UK, pp. 121–132. IEEE Computer Society, Los Alamitos (2005)
33. Rinner, C.: Argumentation Mapping in Collaborative Spatial Decision Making. In: Balram, S., Dragićević, S. (eds.) Collaborative Geographic Information Systems. Idea Group Publishing, Hershey (2006)
34. MacEachren, A., Guiray, C., Brewer, I., Chen, J.: Supporting Map-Based Geocollaboration through Natural Interfaces to Large-Screen Displays. Cartographic Perspectives 54, 4–22 (2006)
35. Bortenschlager, M., Leitinger, S., Rieser, H., Steinmann, R.: Towards a P2P-Based Geo-collaboration System for Disaster Management. In: GI-Days 2007 - Young Researchers Forum (2007)
36. Convertino, G., Mentis, H., Bhambare, P., Ferro, C., Carroll, J., Rosson, M.: Comparing Media in Emergency Planning. In: Proceedings of the 5th International ISCRAM Conference, Washington, DC (2008)
37. Capata, A., Marella, A., Russo, R.: A Geo-Based Application for the Managemnt of Mobile Actors During Crisis Situations. In: Proceedings of the 5th International ISCRAM Conference, Washington, DC (2008)
38. Brewer, I., MacEachren, A., Abdo, H., Gundrum, J., Otto, G.: Collaborative Geographic Visualization: Enabling Shared Understanding of Environmental Processes. In: Proceedings of IEEE Symposium on Information Visualization, Washington, DC, p. 137 (2000)

39. Cai, G., MacEachren, A., Brewer, I., McNeese, M., Sharma, R., Fuhrmann, S.: Map-Mediated Geocollaborative Crisis Management. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) ISI 2005. LNCS, vol. 3495, pp. 429–435. Springer, Heidelberg (2005)
40. Touriño, J., Rivera, F., Alvarez, C., Dans, C., Parapar, J., Doallo, R., Boullón, M., Bruguera, J., Crecente, R., González, X.: Copa: A Ge-Based Tool for Land Consolidation Projects. In: Proceedings of the Ninth ACM International Symposium on Advances in Geographic Information Systems, Atlanta, Georgia, pp. 53–58. ACM Press, New York (2001)
41. Zurita, G., Antunes, P., Baloian, N., Baytelman, F., Farias, A.: Visually-Driven Decision Making Using Handheld Devices. In: Zaraté, P., et al. (eds.) Collaborative Decison Making: Prospectives and Challenges. Frontiers in Artificial Intelligence and Applications, pp. 257–269. IOS Press, Amsterdam (2008)
42. Hevner, A., March, S., Park, J., Ram, S.: Design Science in Information Systems Research. Management Information Systems Quarterly 28, 75–105 (2004)
43. Salmon, P., Stanton, N., Jenkins, D., Walker, G., Young, M., Aujla, A.: What Really Is Going On? Review, Critique and Extension of Situation Awareness Theory. In: Harris, D. (ed.) HCII 2007 and EPCE 2007. LNCS (LNAI), vol. 4562, pp. 407–416. Springer, Heidelberg (2007)

Supporting the Decision Implementation Process

André Campos, Adriana S. Vivacqua, and Marcos R.S. Borges

PPGI – Graduate Program in Informatics,
DCC-IM – Department of Computer Science, Institute of Mathematics,
UFRJ – Federal University of Rio de Janeiro
andrecampos@ufrj.br, avivacqua@dcc.ufrj.br, mborges@nce.ufrj.br

Abstract. As the environment becomes competitive, organizations must develop the ability to quickly adapt to changes, becoming flexible and responsive. The decision making process should be quicker and more assertive, leading to action. Organizations need to make the right choices and implement them as decided. It is important that organizations be aware of the implementation of decisions and its impact. Monitoring the implementation of the decision and analyzing its results is the main way of assessing the decisive process itself. Extensive work has been done on decision making, but not on decision implementation. The goal of this research is to increase adherence of the implementation to the decision made. Given that large sums of money are spent on the decision making phase, implementations that do not adhere to the decisions may lead to undesired results, frustrating decision makers. In this paper, we present a method and a system to support decision implementation. With this research, we seek to contribute to the decision making process, specifically during implementation phase.

Keywords: decision making, decision implementation, decision follow-up.

1 Introduction

The capacity to make decisions and implement them contributes to increase the flexibility and responsiveness of organizations. Decision making must be quick and assertive, as organizations need to make the right choices and implement them faithfully, as quickly as possible [1]. Organizations need to constantly modify and refine their goal-achieving mechanisms by reorganizing their internal structures, their relationships, and the control and decision making processes [2]. This involves responding to changes, making appropriate decisions at the right time, and still using the expert knowledge in the best possible way to deal with uncertainties [3]. The success of organizations depends on the development of new capabilities to deal with complex problems and decisions [1].

However, even though the decision-making process has been studied and improved over the years, the process of implementing decisions has not received the same level of attention. This paper deals with the gap between the decision and its implementation, which is characterized by the implementation turning out differently from what had been decided.

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 113–120, 2010.

To be considered a success, the decision cycle should provide a link between the activities of implementation and decision making [4]. A successful implementation is one that meets the expectations of decision makers [5], and accomplishes what was decided [6]. Therefore, an implementation that does not adhere to what was decided can be considered a failure. Low decision-implementation adherence is treated in this work as an implementation flaw.

1.1 Contextualizing the Problem

There is little evidence and study on the effectiveness of decisions, that is, if these have reached expected results, even though there is there is a relationship between the process of decision making and the achievement of expected results [7]. The implementation stage is frequently neglected and deserves more attention [8]. According to Nutt [6], there has been much speculation about the factors that influence the successful implementation of the decision, but few studies about these factors, and about half of the decisions in organizations result in failure.

According to Miller [9], putting the decision into practice is a critical success factor for an organization. She also notes that the successful implementation of a decision depends on specific management actions, and it is important that the implementers understand how the decision was made by decision makers. Borges et al [4] point to a gap between the end of the meeting, (i.e., the decision-making itself) and the implementation, which can result in the implementation of the decision in a different way than what was decided. Failure to implement a decision may lead to tangible and intangible losses for an organization, depending on the importance of the decision. The goal of this paper is to present a decision method and process to increase adherence of implementations to decisions made.

The remainder of this paper is organized as follows: in the next section we present the problem of decision support and decision implementation. In Section 3, we present our decision implementation model, followed by SAID, its instantiation, in Section 4. We finish with a brief discussion and next steps in Section 5.

2 Decision Support and Implementation

Decision support systems have been studied since the 1950s, and, given the increase in computational power and competitiveness the world experienced in the following decades, this field has experienced an ever-increasing growth, and organizations have been investing a significant portion of their resources into decision support systems.

Fig. 1 presents a simplified view of the decision process: it starts with the definition of the problem, as decision makers or stakeholders perceive there is an issue to be addressed and convene to discuss and make decisions about it. After the problem has been defined, a number of alternatives can be defined and decisions need to be made regarding the best one. Selection of the option and of the strategy to implement it is the next step, which initiates an implementation cycle. After the implementation strategy has been defined, the decision is communicated to the implementation team, which goes on to implement the decisions and communicate progress and results back to decision makers. When necessary, decision makers may make alterations to the strategy, in response to feedback from implementers.

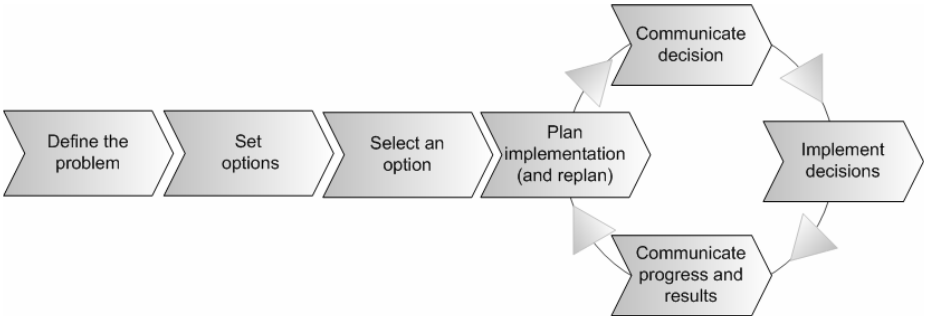


Fig. 1. Decision Making and Implementation Process (macro level)

Despite the efforts, half of the decision processes fail to implement decisions [6] [5], or implement actions that are not in accordance with what was decided [4]. An implementation is considered failed when it does not adhere to what was decided by decision makers. This highlights the importance of the implementation issue, addressed in this paper. It is also important to understand the potential causes of implementation failure.

2.1 Causes of Implementation Failure

Inadequate or poor quality information exchange between decision-makers and implementers and lack of mechanisms to coordinate implementation are frequent causes of implementation failure [4].

Communication is an important human skill in many domains. Good communication is widely recognized as a fundamental condition for cooperation in personal life and organizations [10]. When details of the implementation tasks are distributed among a group of individuals, there is always a risk of failure when sharing this information [11], which, in turn, affects implementation. Mechanisms that provide clearer communication between users and system designers would contribute towards greater success of projects in this domain [11]. Scenarios of poor communication between decision maker and implementer are [4]:

- The decision is not accompanied by sufficient information for its implementation;
- There are different contexts among decision makers and implementers.

There is a relation of cause and effect between these two scenarios and implementation results. When the decision is not accompanied by sufficient information or there are different contexts between actors, the implementation may turn out different from what was decided.

In view of these aspects, we can reasonably expect that a more effective communication channel between decision maker and implementer would increase the alignment between decision and implementation. The problem of low compliance of implementation to decision stems from poor communication between decision maker and implementer. Another consequence of insufficient communication is the lack of feedback between decision makers and implementers, which reduces the possibility for corrective measures during implementation of decisions.

2.2 Communication Flow during Implementation

The implementation of decisions is, in general, an activity performed by a group of people, especially when it comes to the implementation of complex decisions. This activity may involve multiple tasks, with some degree of interdependence between them. Therefore, besides the need to send information to decision makers, implementers might need to exchange information among themselves. In addition, decision makers need to be able to monitor the implementation progress [4].

Different scenarios are possible: decision makers and implementers may be geographically distant or simply physically separated by the nature of their duties; implementers may be involved with one or several implementations, as a group or individually. In the process of implementing a decision there are three important communication flows:

- decision-maker to implementer;
- implementer to decision-maker;
- implementer to implementer.

This communication may happen in various ways, and may be accompanied by artifacts such as documents or digital images. There is also a need to organize and follow activities, i.e., coordinate the work. It should be noted that this communication generates knowledge, and, as the implementation progresses, useful information may be captured for future decision making.

3 A Method for Decision Implementation

Given the collaborative nature of decision implementation and its potential to contribute to knowledge management, our proposed method addresses not only the increase in compliance of implementation to decision but also knowledge capture, taking into account both the collaborative and knowledge management aspects of the problem. The proposal focuses on communication and coordination between parts. The proposed method does not focus on the decision-making phase, looking instead at the implementation itself. It includes a process for evaluating the implementation to see whether it happened and its level of adherence to the decision.

The implementation process is initiated with a decision, and finished with a review. The model includes an information storage element, which stores both steps and results of implementation as well as the final evaluation, so that this knowledge can be reused for future decision-making and implementation processes. These previous implementations fuel not only the decision phase, but also the implementation itself. The implementation phase is strongly dependent on communication, coordination and cooperation actions, which generate information that is stored for future use.

The decision implementation process shown in Fig. 2 shows activities and their interrelations, as well as information flow between them and the parties responsible for each step. Initially, a decision implementation plan is built, defining the steps that should be taken to implement the decision, checkpoints and indicators that must be informed during the implementation process. After that, roles and tasks are assigned to implementers, which will be in charge of conducting the implementation work.

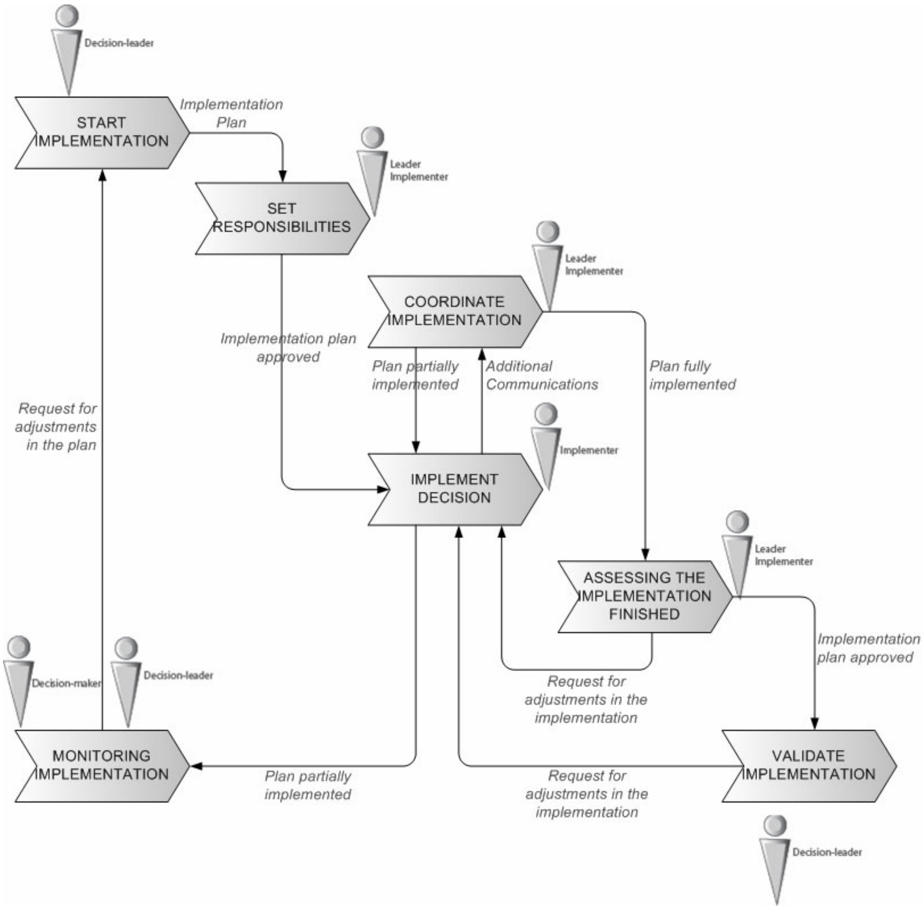


Fig. 2. Decision implementation process

They start working on their activities and informing the checkpoints and indicators related to their work. A supervisor follows the implementation and the information provided through indicators, controlling the percentage of concluded work for each activity. When all activities are concluded, and evaluation is conducted, so any mistakes identified can be corrected before the final evaluation. When the decision is considered to have been adequately implemented, it is passed on to decision makers for analysis. During this phase, they might still find problems and request corrections, but, after the implementation has been validated, it is considered finalized (concluded). With a concluded implementation, decision makers may evaluate the final outcome and provide information on the level of adherence between the decision and the implementation (according to their subjective evaluation), and leave notes, marks or keywords to facilitate retrieval and reuse at a later date.

4 System Support for Decision Implementation

Given the defined processes, we developed a system to support decision implementation, seeking adherence to the method. The Decision Implementation Support System (SAID in Portuguese) instantiates the method and processes involved in implementing and following up decisions made. The system allows the definition of profiles and access rights. One of five roles can be assigned to each user:

- System Administrator: manages the system, user accounts, and assigns access to system functionality.
- Decision Leader: records the implementation plan, accompanies the plan's implementation, and validates the implementation after its conclusion.
- Decision Maker: can monitor the implementation, helping the Decision Leader locate inconsistencies or errors.
- Implementation Leader: defines and assigns responsibilities, coordinates deployment, monitors checkpoints, and assesses the results, before the final evaluation is conducted by the Decision Leader.
- Implementer: is the one who actually implements the decision, performing tasks, and providing information at checkpoints defined by Decision Leader.

The main system screen shows the implementation plan. It allows the lead decision maker to register plans for decision implementation and follow it. Each plan may contain several activities, which must follow a certain order. The lead decision maker can also add other users to the decision maker team, who will then be able to send and receive communications to those involved with that plan. For each activity, progress indicators and checkpoints can be included, which will be informed by implementers during the process. An activity can only be recorded as completed if all checkpoints are properly completed. These indicators and checkpoints were designed to help Decision Makers and Implementers follow the work that's being done and identify problems before they become too serious.

4.1 Initial Results

This system has been implemented to support the Coordination of the Management of Information Technology (CGTI in Portuguese) of the Oswaldo Cruz Foundation (Fiocruz), a research and development organization that promotes health and social development.

SAID has only been used for a short period of time, but through preliminary interviews with implementers of a particular decision we have been able to get some initial impressions. At this point, users believe that SAID can become an important communication mechanism in the implementation process, which can already be observed in this initial trial period. Coordinators emphasized the importance of the tool as it leads to the establishment of a contract to be fulfilled by implementers, and implementers highlighted the importance of setting clear goals to be achieved, including progress indicators, as their experience showed that decision makers, in general, were not clear as to what was expected from them.

System trials will continue, and more formal evaluations have been started. Already more than 30 decisions regarding Fiocruz IT have been made and need to be implemented, in the year 2010. We expect to gather a vast amount of data from this usage, which will lead to further improvement in the model and in the system.

5 Discussion

In this paper, we presented a model for decision implementation and a system to support it. Implementation failure is a frequent and costly problem in organizations and more should be done to address it. The system is under testing and experimentation by decision and implementation teams at Fiocruz.

In designing and detailing the process, we observed that there was a relation between it and the 3C model. To further analyze the method, we detailed each step and correlated it to the collaboration, coordination and cooperation. This analysis indicates that the implementation process involves firstly, coordination between parts, communication between individuals and, to a much smaller degree, cooperation (working together) among them.

Our initial analyses show that implementation is related mainly to communication and coordination. Cooperation is needed during the implementation phase, when two or more participants must work closely on interrelated tasks, and on the coordination, again when more than one person must work together to coordinate tasks. The analysis also shows that communication is most important for coordination of the efforts (to understand what others are doing and arrange their joint work), for monitoring (through backchannels, indicators and checkpoints, lead implementers keep track of what is going on) and for assessment of the completed implementation (when implementers communicate what is done and decision makers discuss it.) It also figures prominently in the beginning, when the implementation starts (and initial descriptions and information on what and how should be implemented have to be communicated to the team.) It is important to note that communication may be undertaken in many ways, whether through direct face to face interaction, chat, email or by leaving annotations on artifacts. We did not distinguish these, but this would be an interesting study.

Coordination figures, not surprisingly, in the implementation coordination step, but also in the assessment and validation steps. This happens because during assessment, corrections and alterations may become necessary, demanding both communication and coordination from the parts to make them happen. Monitoring does not so much demand coordination, but rather alter coordination schemes. Through monitoring, necessary adjustments are perceived and changes in the working processes, roles and tasks are identified and made.

This initial analysis has already raised some interesting points, which lead us to further analysis of the process, both on a theoretical and practical level. When are different types of communication appropriate and what types of coordination happen (simple monitoring versus alterations to the structures) are some of the questions that could be addressed through further study, shedding more light on the implementation problem. We believe this is an important tissue, which we have just begun to address. We will continue to explore and refine our processes and techniques to further our understanding of the problem and possible solutions.

Acknowledgments

This work was partially supported by CNPq (Brazil) grant No. 304252/2008-5, and LACCIR grant No. R0308LAC004.

References

1. High Performance Organizations in a Wicked Problem World, http://www.innovationlabs.com/high_performance.pdf
2. Miles, R., Snow, C.C.: Organizational Strategy, Structure and Process. Stanford Business Books, Stanford (2003)
3. Wu, S., Kotak, D.: Agent-Based Collaborative Project Management System for Distributed Manufacturing. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 1223–1228. IEEE Press, New York (2003)
4. Borges, M.R.S., Pino, J.A., Araujo, R.M.: Bridging the Gap Between Decisions and Their Implementations. In: de Vreede, G.-J., Guerrero, L.A., Marín Raventós, G. (eds.) CRIWG 2004. LNCS, vol. 3198, pp. 153–165. Springer, Heidelberg (2004)
5. Hickson, D., Miller, S.J., Wilson, D.: Planned or Prioritized? Two Options in Managing the Implementation of Strategic Decisions. Journal of Management Studies 40, 1803–1836 (2003)
6. Nutt, P.C.: How Decision Makers Evaluate Alternatives and the Influence of Complexity. Management Science 44, 1148–1166 (1998)
7. Dean, J.W., Sharfman, M.P.: Does Decision Process Matter? A Study of Strategic Decision – Making Effectiveness. Academy of Management Journal 39, 368–396 (1996)
8. Papadakis, V., Barwise, P.: Research on Strategic Decisions: Where Do We Go From Here? In: Papadakis, V., Barwise, P. (eds.) Strategic Decisions, pp. 289–302. Kluwer Academic Publishers, MA (1998)
9. Miller, S., Wilson, D., Hickson, D.: Beyond Planning Strategies for Successfully Implementing Strategic Decisions. Long Range Planning 37, 201–218 (2004)
10. Hall, T., Wilson, D., Austen, R., Jagielska, D.: Communication: the Neglected Technical Skill? In: Proceedings of the ACM SIGMIS CPR Conference on Computer Personnel Research, pp. 196–202. ACM Press, New York (2007)
11. Cramton, C.D.: The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration. Organization Science 12, 346–371 (2001)

A Framework and an Architecture for Context-Aware Group Recommendations

Tim Hussein, Timm Linder, Werner Gaulke, and Juergen Ziegler

University of Duisburg-Essen, Germany
firstname.lastname@uni-due.de
http://www.interactivesystems.info

Abstract. In this paper, we propose a generic framework to generate context-aware recommendations for both single users as well as groups. We present the the concept of *context views* and a corresponding architecture implementing the framework as well as exemplary recommendation workflows for group recommendations.

Keywords: Context, Recommender Systems, Adaptation.

1 Introduction

Recommender Systems [3,9] are a well-established means for improving the user-experience in e-commerce, collaborative environments, or e-learning. Although existing approaches are quite successful, we see potential for improvement especially regarding two aspects:

1. Most recommender systems aim at single users, not groups.
2. Recommenders usually don't take the particular usage context into account.

We think that the usage context contains valuable information to improve recommendation quality, because recommendations, especially those for groups, highly depend on the current context (for instance the social or historic context as well as time or location). As a solution, we propose a generic framework for generating recommendations for both single users and groups with respect to the context at hand, where context can be virtually anything from external circumstances (time, place, etc.) to information derived from system interaction (browsing history, downloads, etc.).

Jameson and others [7,8] distinguish between four sub-tasks for group recommendations: (a) Acquisition of preferences, (b) Recommendation Generation, (c) Presentation and (d) Achieving a consensus. Although (c) and (d) may be important for many systems, we focus on group preference acquisition and recommendation generation (and combine this with context-awareness).

2 Context Views

A truly adaptive system should consider both the users' current state and context information to select the most important entities and concepts with regard to the

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 121–128, 2010.

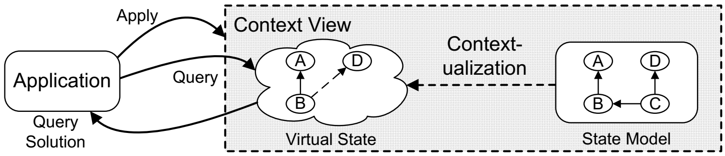


Fig. 1. A client triggers a context view on the state and queries the resulting virtual state. The contextualization process transforms the elements of the state. In this contextualization process, C is not important (and thus removed), whereas a virtual connection between B and D is generated (dashed) as a contextualization result. Virtual new entities could be added as well (although this is not the case here).

current situation. *Context views* apply contextualization operations to a state resulting in a *virtual state* representing a certain view on the state. The name context view refers to the view concept in database theory, where a sequence of operations leads to a *virtual table* that can thereupon be queried as if it were a regular one (Figure 1). Each view shifts the focus to other information from the state model thus enabling different forms of adaptation.

Context, in our understanding, can be virtually anything from external circumstances (location, weather, etc.) to internal resources (click-stream, state of the application, etc.). This rather broad understanding of context goes along with the often cited context-definition by Dey & Abowd [2]. Applying the *view* concept to context-adaptivity, we define a context view as a sequence of contextualizing operations leading to a virtual state. Contextualizing operation can be any means that are able to analyze initial states (with respect to the context), draw conclusions and create virtual states thereupon. In the *context* of a certain project, a company's chat application could perform operations like this:

```
(1) Identify all users working on the same project & attending the same chat.
(2) Mark past chat protocols of sessions, in which at least 2 users working
    on that particular project participated.
```

This example highlights the project context, but any other aspect like the current location could be used as well. The contextualization does not affect the original state at all. Technically, the contextualization process is spawned with a copy of the original state. From a more abstract point of view this means, that only events change the current state. Reasoning, on the other hand, leads to a cognitive model *reflecting* reality but does not change it per se.

A more detailed conceptual explanation of context views including an introduction of the underlying conceptual framework for context-adaptive applications has been published by the authors in an earlier publication [4].

3 Architecture

In this section, we present the architecture of a Java-based client-server architecture called *Hybreed Context Views*[1], implementing the context view concept.

[1] See `http://www.hybreed.orgformore(especiallytechnical)information`

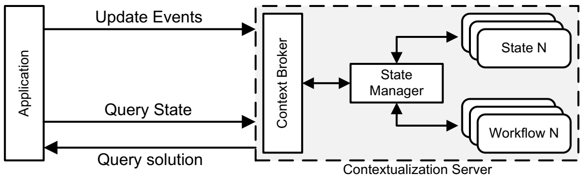


Fig. 2. Overview of the client-server architecture. The access type depends on, whether the server is embedded in an application or deployed on a remote destination.

The core of Hybreed Context Views, the so-called *contextualization server*, makes use of standardized web-service protocols (SOAP/WSDL) and serves the following purposes (see Figure 2 for an illustration):

- Managing the distinct user states.
- Aggregation of the user states to a global state.
- Applying context views to a state, either user or global, to obtain a *virtual state*. Each context view contains a workflow of contextualizing operations.
- Handle state queries. Clients can query a user global state, or (in most cases) trigger the creation of a virtual state to be queried.

A state is updated each time the client registers an event with the server. Such an event may be that the user's location has changed, or that he has clicked on a certain item. Moreover, a client can query a *virtual state*, which is created on-demand and is identified by the name of a context view. Finally, a client may ask for a *virtual global state*, which is a contextualized view on the global state of all active users; i. e. this virtual global state is generated by executing a workflow on the combined state of all users. By storing the states of all users on a central server, it is possible to make statements about a group of users and the group's context as a whole, especially in a collaborative scenario. Furthermore, since the actual contextualization process takes place server-side, the whole application logic is also placed there, leading to thin client implementations. In fact, the client in our implementation is basically limited to a thin wrapper around a SOAP-based communication protocol, i. e. a web service proxy without any functionality of its own. This makes it possible to benefit from sophisticated contextualization techniques even when performance considerations play a big role, for instance in a mobile environment on a smart phone. In the following sections the main functionalities of the contextualization server are illustrated in more detail.

State Management

One of the main purposes of the Hybreed Context Views framework is the management of states and virtual states of individual users and the entire group of active users. Whenever a client notices a change in the user's current state, it informs the server that the respective state needs to be updated (Figure 3).

Fig. 3. In case of a client-side state update, the server updates the state model

Depending on the configuration, the event update may trigger follow-up actions on the server like inference operations or subsequent sensing activities.[2] In this way, the server always maintains a model of the current state, which in our implementation is stored as an RDF/OWL model using the Jena Framework[3]. Such an event-update message looks like this (simplified):

```
<sensor-id>AppSensor</sensor-id>
<statement>
 <subject>UserX</subject>
 <predicate>starts</predicate>
 <object>ChatApp</object>
</statement>
```

In order to retrieve and deduce statements about a group in its entirety, we introduce a so-called *global state*, which combines the states of all users into one that contains the sum of all this information, for instance by merging the sets of RDF statements of all active users. This is done each time the global state is requested and at least one of the user-specific states has changed; otherwise, we can fall back to a previously created and cached global state[4]. Another important consideration besides the performance aspect is, of course, multi-user synchronization; since multiple users can and will be active in the system at the same time and each one might want to access that global state while the other one is possibly just causing a change in it by updating his own state, it is imperative to synchronize the accesses in order to maintain thread safety. In our implementation, this is achieved by multiple locking mechanisms based upon the multiple readers / one writer approach.

Simple State Queries and Context-Aware Queries

Clients can now query their state to ask for non-contextualized information from the state layer (Figure 4). Following the database metaphor, this is like executing a query on a raw, materialized table within a database. In our case, all query strings must follow the SPARQL query format and can be SELECT, ASK, DESCRIBE or CONSTRUCT queries[5]. By querying the state, clients

[2] Example: Imagine, a system senses a user's current location by resolving his/her IP address to geographical coordinates. If a user now sends the event that the IP address has changed, the server should refresh the location as well.

[3] `http://jena.sourceforge.net`

[4] Preference aggregation by averaging single ratings is out of the scope of this paper.

[5] `http://www.w3.org/TR/rdf-sparql-query`

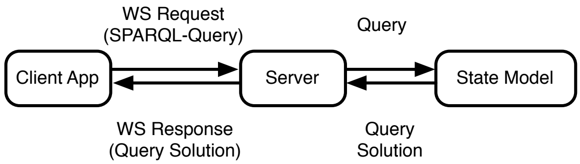


Fig. 4. A simple request

can not only request the information that they have previously supplied to the system. Instead, they can also query for information that was automatically inferred by the internal sensing engine (cf. "Components" section below), like a geographic location that was deduced from the IP address supplied by the client by contacting an appropriate external web service.

In context-adaptive scenarios, we can additionally provide the name of a context view as a request parameter in order to obtain a *virtual state*. This concept works for single users' states as well as for global ones.

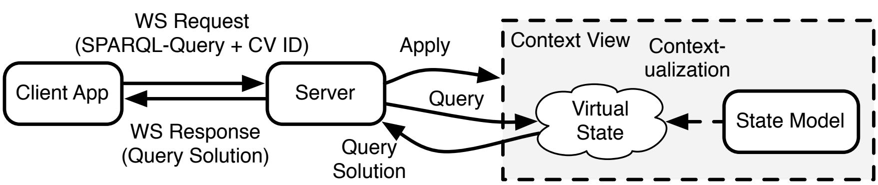


Fig. 5. Context-aware requests add a reference to a Context View (CV ID). The server then applies the appropriate Context View and queries the resulting virtual state.

Components

In figure 6, we see the main components of Hybreed Context Views. The ContextBroker and StateManager classes are the core of the system and are responsible for processing the users' requests as well as managing the states of the individual users. Since the framework can also be integrated into an existing Java application without necessarily exposing a web service endpoint, such an application can directly communicate with the central ContextBroker instance.

When an event is registered with the ContextBroker and is subsequently sent to the StateManager component, it passes the event to an implementation of the SensingEngine interface that is provided by the application. In essence, a sensing engine processes events, optionally inferences new information from these events and stores this information in the current user's state, which, in our case, is represented by a RDF model. As soon as the client intends to query a specific virtual state, the ViewEngine that is injected into the StateManager triggers a contextualization process: This class is provided with a set of context views by the application, each of which is identified by a unique name and links to

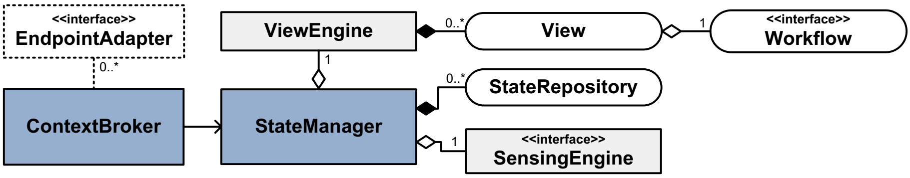


Fig. 6. Class diagram: Overview of the Hybreed Context Views architecture

an implementation of the Workflow interface that describes how a state is to be turned into a contextualized, virtual state (Exemplary workflows for context views are described in the next section).

4 Algorithms for Group Recommendations

The contextualization algorithms are not part of the Context Views framework; instead, a *workflow* interface is provided that can be implemented by any suitable contextualization framework (Figure 6). For demonstration purposes, we implemented a *Hybreed RecFlows*[6] library containing different recommendation algorithms ranging from item-based and user-based collaborative filtering, simple top-n or rule-based algorithms to different spreading activation techniques. Each algorithm is implemented in several variations and can be used in conjunction with a Context Views contextualization server. We now describe some of the algorithms in detail.

Rule-based. The example in Section 2 used a simple rule-based approach to highlight the most relevant elements. Personalization by means of fixed rules often is a good trade-off between administration and implementation effort on the one hand and good results on the other.

Spreading Activation. Spreading Activation was introduced in the 1970s [1] and has successfully been used in several research areas in computer science, most notably in information retrieval or e-commerce [6]. The basic concept behind Spreading Activation is that all relevant information is mapped on a graph as nodes with certain "activation levels". Relations between two concepts are represented by a link between the corresponding nodes. If for any reason one or more nodes are activated, their activation level rises and the activation is spread to adjacent nodes (and the ones related to them and so on). Thereby the flow of activation is attenuated the more it strides away from the initially activated node(s). In the end, several nodes are activated to a certain degree that are semantically related to the elements originally selected.

This technique can directly be adapted to context views as all relevant information already is encoded in a graph. If, for instance, user A and B attend a chat, the nodes representing A and B could be initially activated. The activation

[6] RecFlows stands for "Recommendation Workflows".

would now spread through the graph, increasing adjacent nodes. As a result, exactly those elements that A and B have in common (for instance a project both are working on) would receive the most activation.

Collaborative Filtering. Collaborative filtering [3] methods are supposed to be the most widely implemented recommendation techniques. Transferred to context views and collaborative work, users could be regarded as belonging to the same group, if they have certain features in common. In this case, instead of items in a shopping cart, recently opened documents could for instance be the foundation to compute similarity upon, resulting in something like "users who opened Roadmap for Project X also opened Work Packages of Project X".

Hybrid approaches. Hybrid recommender systems can be applied as well (in fact we did just that in a previous publication [5]). For instance could both algorithms mentioned above be combined by first applying collaborative filtering to identify certain elements and, in a second step, using the the results as initial nodes in a spreading activation process for refining.

These approaches only sketch the basic ideas in order to demonstrate the generic applicability of context views regardless of the algorithm. In reality, things like privacy or rights management have to be considered as well, but the principles and ideas of context views should be clear now.

5 Summary

The definition "any information that can be used to characterize the situation of an entity [is context]" [2] leads to the implication that for instance the membership in a certain group can be regarded as context as well – which affects the notion of context-adaptive systems in turn: Not only should context-adaptive systems exploit external observations like time, location, etc., but also take any other information into account "that can be used to characterize the situation".

We introduced *context views*, which can be used to identify the most important elements of a situation with regard to the particular contextual perspective. A system can thereupon use them for adaptation purposes. In Section 3, we described an architecture for this conceptual framework: We implemented a contextualization server that can make use of arbitrary personalization techniques to generate context-aware group recommendations as a service. After that, we explained how several well-established techniques can be used for context views, either solely or in combination.

Context views can be used in group-based scenarios, but in fact they are meant as a more general concept for arbitrary context-adaptive systems. However, in this paper, we focused on the derivation of the concept and its implementation as well as its applicability for group-based work. This paper focused on the construction of shared preference models (identified as (c) according to Jameson & Smyth [8]). Although (a) and (b) are possible with this framework and architecture as well, we skipped the details for the sake of simplicity, and will publish our experience with these approaches in a different publication.

Acknowledgements. The research presented in this paper is part of the CONTici project funded by the German Research Foundation (Deutsche Forschungsgemeinschaft).

References

1. Collins, A.M., Loftus, E.F.: A spreading activation theory of semantic processing. Psychological Review 82(6), 407–428 (1975)
2. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. In: CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness. ACM Press, The Hague (2000)
3. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM 35(12), 61–70 (1992)
4. Hussein, T., Gaulke, W., Linder, T., Ziegler, J.: Improving collaboration by using context views. In: Proceedings of CAICOLL: First Workshop on Context-Adaptive Interaction for Collaborative Work (2010)
5. Hussein, T., Linder, T., Gaulke, W., Ziegler, J.: Context-aware recommendations on rails. In: Workshop on Context-Aware Recommender Systems (CARS-2009) in conjunction with the 3rd ACM Conference on Recommender Systems (ACM RecSys 2009), New York, NY, USA (2009)
6. Hussein, T., Ziegler, J.: Adapting web sites by spreading activation in ontologies. In: Proceedings of International Workshop on Recommendation and Collaboration. ACM, New York (2008)
7. Jameson, A.: More than the sum of its members: Challenges for group recommender systems. In: Proceedings of the International Working Conference on Advanced Visual Interfaces, pp. 48–54. Gallipoli, Italy (2004)
8. Jameson, A., Smyth, B.: Recommendation to groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization, pp. 596–627. Springer, Berlin (2007)
9. Resnick, P., Varian, H.: Recommender systems. Commun. ACM 40(3) (1997)

Concurrent Modeling in Early Phases of the Software Development Life Cycle*

Petra Brosch[1], Philip Langer[2], Martina Seidl[1], Konrad Wieland[1], Manuel Wimmer[1], and Gerti Kappel[1]

[1] Business Informatics Group, Vienna University of Technology, Austria
{brosch,seidl,wieland,wimmer,gerti}@big.tuwien.ac.at
[2] Department of Telecooperation, Johannes Kepler University Linz, Austria
philip.langer@jku.ac.at

Abstract. Software engineering deals with the development of complex software systems which is an inherently team-based task. Therefore, version control support is needed to coordinate the teamwork and to manage parallel modifications. If conflicting modifications occur, in standard approaches the developer who detected the conflict is responsible for the conflict resolution alone and has to resolve the conflict immediately.

Especially in early project phases, when software models are typically employed for brainstorming, analysis, and design purposes, such an approach bears the danger of losing important viewpoints of different stakeholders and domain engineers, resulting in a lower quality of the overall system specification. In this paper, we propose conflict-tolerant model versioning to overcome this problem. Conflicts are marked during the merge phase and are tolerated temporarily in order to resolve them later in a collaborative setting. We illustrate the proposed approach for the standardized modeling language UML and discuss how it can be integrated in current modeling tools and version control systems.

Keywords: team-based modeling, model versioning, conflict tolerance.

1 Introduction

Models are used in nearly all engineering disciplines. Also in software development, model-driven engineering (MDE) recently gained high momentum. In MDE, the powerful abstraction mechanisms of models are not only used for documentation purposes, but also for compiling executable code directly out of models [1]. Like any other software artifacts, models are developed in teams and evolve over time, consequently they also have to be put under version control. Following a pessimistic approach, version control systems (VCS) lock artifacts for exclusive modification whereas in optimistic approaches, VCS support

* This work has been partly funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) and FFG under grant FIT-IT-819584 and the fFORTE WIT - Women in Technology Program of the Vienna University of Technology, and the Austrian Federal Ministry of Science and Research.

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 129–144, 2010.

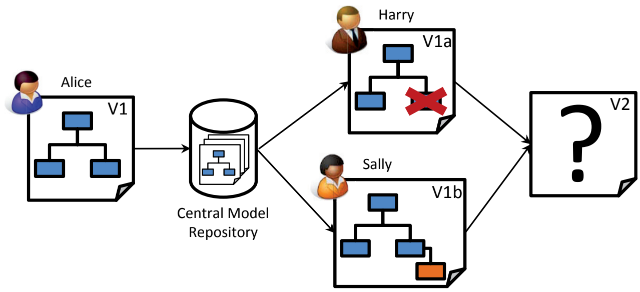


Fig. 1. Optimistic Versioning Process

distributed, parallel team-work. This comes along with the price of merging conflicting changes. Fig. 1 shows a typical versioning scenario with conflicting modifications. The modeler Alice creates a new version of a model (V1) and commits this model to the central model repository. The modelers Harry and Sally check out the current version and perform their changes in parallel. Harry deletes an element which is extended by Sally at the same time. Assume that Harry is the first who finishes his work and he is also the first who checks in his version into the repository. Afterwards, Sally tries the same, but the VCS rejects her version V1b, because her changes are conflicting with Harry's changes. In standard VCS, Sally is responsable for resolving the conflict immediately. Finally one conflict-free version is checked in with the consequence that some modifications may be inevitably lost as they are removed in an undocumented manner. Such an approach is adequate for code when the specification of the system is already established and the code has to be executable at any point in time. In contrast to code, models are often used in an informal manner for collecting ideas and discussing design alternatives in brainstorming periods. Models may serve as sketch in early project phases [2]. Models are then used to manage and improve communication among the team members by establishing common domain knowledge. In this context, it is desirable to keep all (or at least many) of the modifications, even if they are conflicting. The conflicts may help to develop a common understanding of the requirements on the new system. To support versioning in early project phases, we propose a versioning system which temporarily tolerates conflicts enabling a creative design process without destroying the model's structure to use common modeling tools for editing.

This paper is structured as follows. In the next section, we discuss a representative scenario why a collaborative approach to model versioning is needed. In Section 3 we present which kinds of conflicts may occur in model versioning and elaborate on how they are handled in state-of-the-art VCS. In Section 4 our approach of tolerating conflicts when merging different versions of a model is presented. The necessary consolidation phase is discussed in Section 5. Before we conclude with a critical discussion and future work in Section 7, we survey related work in Section 6.

2 Motivating Example

The example depicted in Fig. 2 describes a merge scenario which demonstrates typical problems when developing models in a distributed team following the standard versioning process.

V1. Alice creates a new model in terms of a UML Class Diagram [3] containing the two classes `Person` and `Passport`. She adds the attributes `name` and `bday`—representing the birthday of a person—to the class `Person` and `passNo` to the class `Passport`. Finally, she defines an association between these two classes. After she has finished, she checks in the new model (called V1) into the central repository of the VCS. Now, the modelers Harry, Sally, and Joe want to continue working on this model and therefore, they check out the current version V1 of the model from the central repository to perform their changes.

V1a. In Harry's opinion, a passport is always owned by exactly one person. Therefore, he modifies the association to express a containment relationship (noted by the black diamond). In addition, he changes the attribute `bday` of the class `Person` to `birthday` and adds two new attributes, namely `hash`—a checksum for validating the passport—and `citizen` to the class `Passport`.

Resolution. Because Harry is the first to check in, he has no conflicts to resolve. His version is the new version within the repository.

V1b. In parallel, Sally also understands a passport as a part of a person. Hence, she inlines the attribute `passNo` in the class `Person` and deletes the class `Passport`. Sally commits her changes. The VCS rejects her modification, because they are conflicting with the already performed changes of Harry. A so called "Delete/Update conflict" occurs: the deleted class `Passport` has been changed by another modeler (namely, Harry). Consequently, Sally is responsible to resolve this conflict.

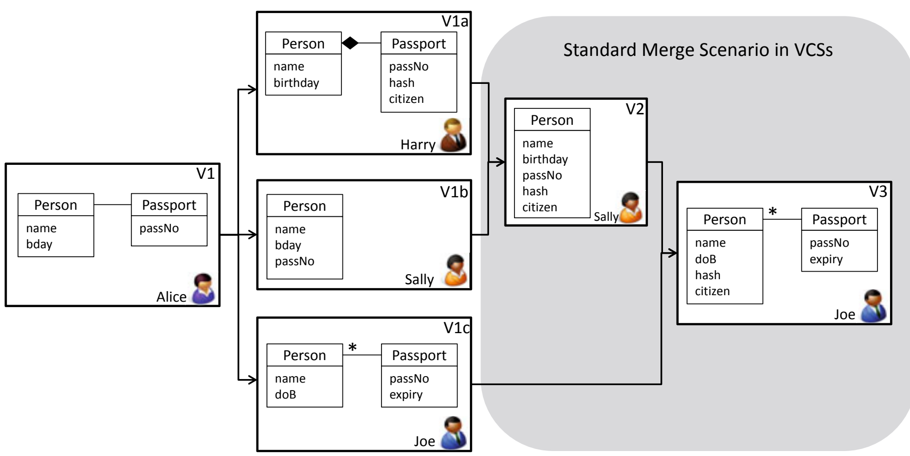


Fig. 2. Motivating Example: Merging in Current VCSs

Resolution. In Sally's opinion, the class `Passport` is still unnecessary in the model, although Harry has updated this class by adding new attributes. Therefore, she decides to inline the added attributes of Harry, namely `hash` and `citizen`, in the class `Person` and to still abandon the class `Passport`. Version V2 now consists of the class `Person` including all attributes added or updated by Alice, Harry, and Sally.

V1c. Joe has also checked out the initial version of Alice (V1) and performs his changes in parallel to Harry and Sally. He renames the attribute `bday` to `doB` in the class `Person` and adds to the class `Passport` a new attribute called `expiry`. Furthermore, Joe is of the opinion that one passport may belong to several persons (i.e., a parent and its children) and therefore, he sets the multiplicity of the association to unbound indicated by the asterisk in the model.

Resolution. Now, Joe tries to check in his version V1c into the central repository. Different conflicts are reported by the VCS, because Joe's version is now conflicting with the current version V2 in the repository. Recall that V2 is the version of the model covering Harry's and Sally's modifications consolidated by Sally. Now an "Update/Update conflict" is reported because both, Harry and Joe, have updated the attribute `bday` in different ways. Joe decides to take his version namely `doB`. He also decides not to delete the class `Passport` as afore propagated by Sally.

The new version V3 of the model is put into the repository after Joe has resolved the conflicts. Although version V3 is a valid model, it contains several flaws resulting from the conflict resolutions. The final version of the example model does not reflect all intentions of the participating modelers, because there is only one single modeler responsible for the merge of two versions. For this modeler it is difficult to follow the motivation behind the changes of the others, especially, if more than two modelers perform their changes in parallel.

The version V3 of the example described above does not cover the idea of Harry to consider a passport as integral part of one person expressed by an aggregation. In V3 the attributes `hash` and `citizen` have become part of the wrong class, because over the evolution of the model the information that these attributes are referring to the passport get lost.

Note that the shown scenario is only exemplarily and that other merge sequences are possible. In the example, we assume that the modeler who is responsible for the conflict resolution, does her/his best. Currently, only limited tool support is available for such tasks. Usually only one modeler is responsible for the conflict resolution during the merge and it is very likely that this modeler has not the same information as the other modelers which have checked in parallel versions. Consequently, information is lost during conflict resolution.

3 Conflicts in Model Versioning

As we have seen in the previous section, when merging differently evolved versions of one model, various kinds of conflicts might occur [4]. These kinds of

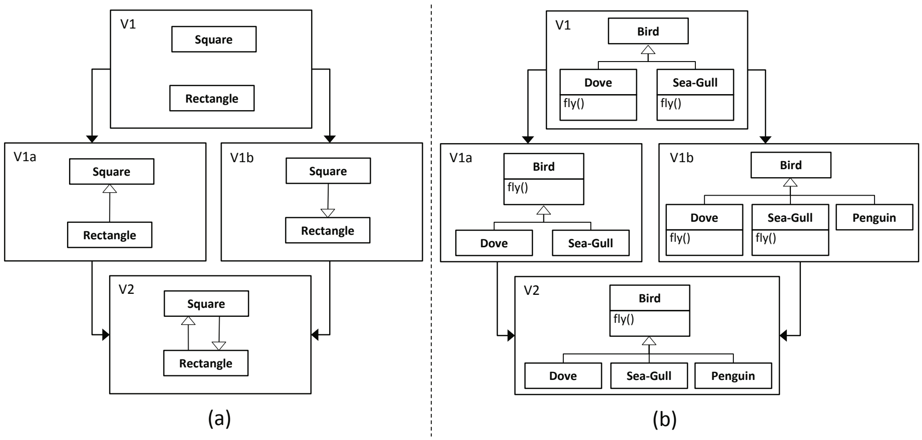


Fig. 3. Example Conflicts: (a) Syntactic Conflict, (b) Semantic Conflict

conflicts are shortly discussed in the following subsection. Furthermore, we elaborate on how state-of-the-art model versioning approaches are handling conflicts.

3.1 Kinds of Conflicts

Overlapping changes like Update/Update or Delete/Update conflicts occur due to concurrent changes on one model element. Such kinds of conflicts are easy to detect. In the motivating example (cf. Fig. 2) two Delete/Update conflicts occur, because Sally deletes the class `Passport` which is modified by Harry and Joe. The example also contains several Update/Update conflicts caused by the parallel modifications of one element, like of the attribute `bday` in the class `Person`.

Syntactic conflicts in a model may be regarded as violation of its metamodel, i.e., the specification of the modeling language. Fig. 3(a) shows an example where the modifications of two modelers lead to a so called "inheritance cycle" which violates the metamodel constraint forbidding such cycles.

Semantic conflicts occur if the meaning of the merged model is incorrect. Since the semantics and the correct interpretation of a model is difficult to express in a formal way, the detection of such problems is challenging and usually requires human assistance although domain knowledge as encoded in upper ontologies might be supportive. Fig. 3(b) shows a Class Diagram containing the class `Bird` which has two subclasses `Dove` and `Sea Gull`. Both classes provide the method `fly()`. Then one modeler performs a refactoring and shifts the method `fly()` into the class `Bird`. At the same time, another modeler introduces a novel class `Penguin` which is also of type `Bird`. When standard merging the modifications of both modelers, we have probably a undesired situation at hand, because a penguin, which is able to fly, contradicts reality.

3.2 State-of-the-Art of Model Versioning

In order to deal with conflicts, according to [5,6], three different approaches are possible. First, techniques could be established to *avoid conflicts* completely. This is accomplished by either versioning in a *pessimistic manner*, i.e., if an artifact is modified, it is locked for all other modelers. Modeling could also be done synchronously, i.e., two modelers are notified immediately, if they are working on the same artifact. Conflict avoidance poses several restrictions on the way how people may work and introduce new complexity in management. This approach does not come along with our goal to develop a model versioning system.

The second approach is to *resolve conflicts* immediately to keep only consistent versions of a model in the repository. Those modelers who are not involved in the conflicting scenario should never notice that conflicts have occurred. In traditional VCS the modeler who is checking in later, bears the full responsibility to resolve the conflicts. This resolution step may be supported with the help of specific rules or policies [7]. Problems have been discussed in the example shown in the previous section.

The third approach is to *tolerate conflicts*. Temporary inconsistencies are accepted in the merged version of a model. The possibly destroyed model has to be repaired by the modelers later and therefore it is taken as basis for discussion leading finally to a model of potentially higher quality. Especially in early phases of software development, such an approach is beneficial, where conflicts could be regarded as possibility to increase the quality of a model through extensive discussions. For the feasibility of this approach, adequate presentation and visualization of the conflicts is indispensable for their resolution. In the remainder of this paper we elaborate on the technical realization of conflict-tolerant versioning.

4 Conflict-Tolerant Merging of Models

The overall goal of our conflict-tolerant merge is to incorporate all changes concurrently performed by two modelers into a new version of a model. The merge implements our major premiss that neither model elements nor changes get lost and that irrespectively of any occurring conflicts the merge process is never interfered by forcing users to immediately resolve inconsistencies. Consequently, model elements are never truly deleted and conflicts are annotated at the affected model elements. Note that in this paper we only focus on overlapping changes and syntactic conflicts (cf. Sec. 3).

4.1 The Merge Algorithm at a Glance

In this subsection, we present an overview on the conflict-tolerant merge algorithm, which is shown in Algorithm 1. Fig. 4 depicts a UML Class Diagram including all classes used in this algorithm. The diagram contains a class `Model` representing a versioned software model. A `Model` contains one root `ModelElement`, which again might contain several child elements. Each model element has an

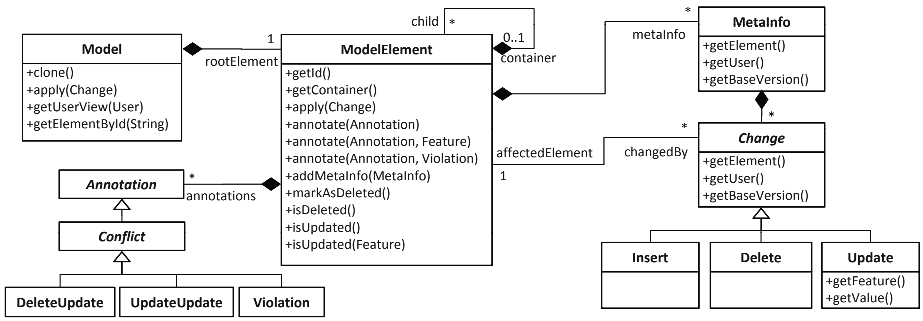


Fig. 4. Types and Operations Used in Algorithm 1

`ID` and knows its container model element. Model elements may be enriched with several `Annotations` and can be further described by `MetaInfos` which contains information about the users who recently changed the model elements, the `Changes` themselves, and the base version.

The algorithm consists of two phases. In the *fusion* phase (cf. Algorithm 1, line 2 to 28), the algorithm iterates over all changes, checks for conflicts, and applies them to a new merged version irrespectively of conflicts. The algorithm considers three kinds of atomic changes. Model elements may have been *inserted*, *deleted*, or *updated*. A model element is considered as updated, if a specific feature value, e.g., its name, or its container has been changed. According to our premiss, we do not truly remove model elements but only mark them as deleted to retain all model elements and all its historic feature values.

When merging atomic changes, two kinds of conflicts may occur. If a feature of a model element has been concurrently modified by both modelers, an *Update/Update* conflict occurs. A *Delete/Update* conflict appears whenever one modeler deletes a model element which has been updated concurrently by another modeler. These conflicts do not block the fusion phase. We immediately add an annotation to the involved model elements indicating the conflict. For Update/Update conflicts, where feature values are changed in parallel, we set the latest value to the element and save the other value by adding a new meta information object to the modified model element.

In the *validation* phase (cf. Algorithm 1, line 29 to 35), the merged model is validated. Validation means to reveal violations of rules and constraints defined by the modeling language. Because values of model element features might have been updated by multiple users, we first have to derive a view of the model for the specific user who performs the check-in. In this view, the feature values of the specific user are set in the model. Thus, we enable the user to actually validate her changed model incorporating the not directly overlapping changes of other modelers. Otherwise, violations might be indicated that are recently caused by other modelers. Like in the fusion phase, we annotate all model elements subject to violations in this phase. Therefore, we iterate through all revealed violations and annotate the elements involved in the violation.

```
Input: originModel, revisedModel, headModel
Output: mergedModel
// create a copy of the headModel as base for the new mergedModel
mergedModel = headModel.clone()
// Fusion: Merging all changes
changeSet = calculateChanges(originModel, revisedModel)
for c ∈ changeSet do
    element = mergedModel.getElementById(c.getElement().getId())
    if c instanceof Insert then
        // containing element to which new element has been inserted
        container = merged-
        Model.getElementById(c.getElement().getContainer().getId())
        if container.isDeleted() then
            // container has been marked as deleted
            // -> add DeleteUpdate annotation to container
            container.annotate(new DeleteUpdate(c))
        end
        mergedModel.apply(c)
    else if c instanceof Delete then
        if
        mergedModel.getElementById(c.getElement().getId()).isUpdated()
        then
            // deleted element has been updated
            // -> add DeleteUpdate annotation
            element.annotate(new DeleteUpdate(c))
        end
        element.markAsDeleted()
        element.addMetaInfo(createMetaInfo(c))
    else if c instanceof Update then
        if element.isDeleted() then
            // updated element has been marked as deleted
            // -> add DeleteUpdate annotation to container
            element.annotate(new DeleteUpdate(c))
        end
        feature = c.getUpdatedFeature()
        if element.isUpdated(feature) then
            element.annotate(new UpdateUpdate(c), feature)
            element.addMetaInfo(createMetaInfo(c))
        end
        mergedModel.apply(c)
    end
end
// Validation: Validate the view of User checking in revisedModel
mergedModel_UserView = mergedModel.getUserView(revisedModel.getUser())
violations = validate(mergedModel_UserView)
for violation ∈ violations do
    // annotate all involved elements violating a model constraint
    for element ∈ violation.getInvolvedElements() do
        element.annotate(new Violation(c), violation)
    end
end
```

Algorithm 1. Conflict-tolerant Merge

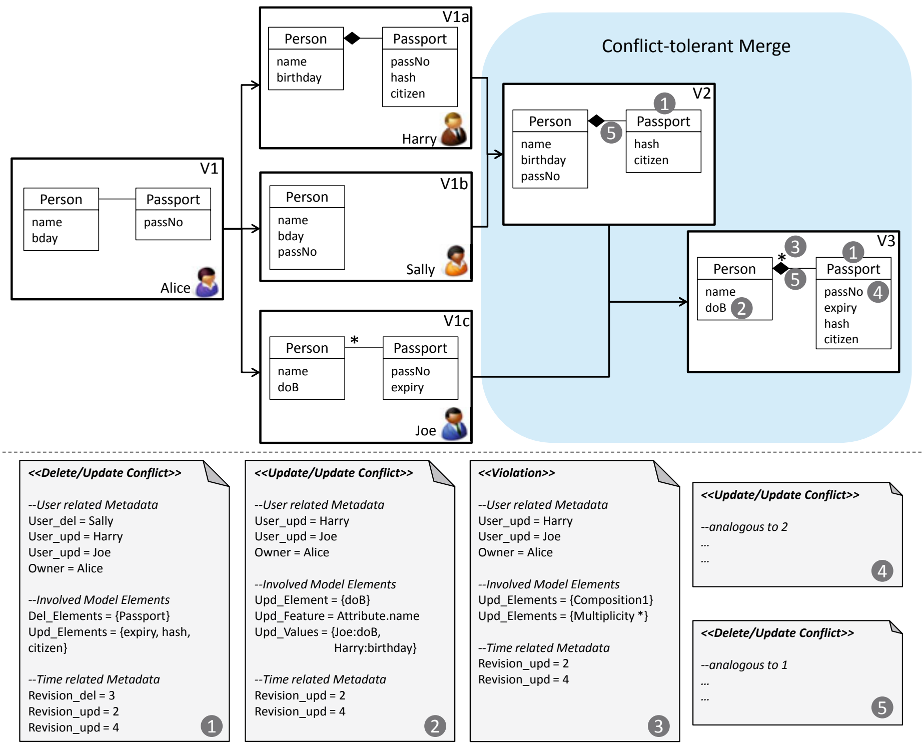


Fig. 5. Conflict-tolerant Merging of the Motivating Example and Annotated Metadata

4.2 Merging the Motivating Example

In this subsection, the afore presented merge algorithm is applied to the motivating example (cf. Fig. 2) in order to illustrate its function in more detail. In contrast to the standard merge depicted in Fig. 2, where conflicts are resolved immediately comparing changed models pairwise, Fig. 5 illustrates the merge results obtained using the *conflict-tolerant merge* and the annotations of the elements marked with the corresponding number.

Merging V1a with V1b into V2. As mentioned afore in Section 2, Sally checks in after Harry has committed his changes to the common repository. His changes comprise two inserts, namely the one of attribute `hash` and of attribute `citizen` in class `Passport`, and two updates. He updated the name of attribute `bday` to `birthday` in class `Person` and changed the aggregation type of the reference connecting `Person` and `Passport` from `unspecified` to `composition`. Consequently, the classes `Passport` and `Person` as well as the reference have to be considered as updated when Sally performs the merge of her version `V1b` with the latest version `V1a` of Harry.

The input for the conflict-tolerant merge algorithm is (i) `V1` as the common base revision (`originModel` in Algorithm 1), (ii) Sally's working copy `V1b` as version to check in (`revisedModel` in Algorithm 1), and (iii) Harry's version `V1a`, which is the most recent version in the repository (`headModel` in Algorithm 1).

Fusion Phase. As a first step in the merge algorithm, a clone of the most recent version `headModel` is created (line 1). This new version saved in variable `mergedModel` acts as basis for incorporating all changes of Sally. Next, the algorithm determines all changes performed by Sally (line 2). This is accomplished by adopting an ID-based model comparison. All changes are saved to `changeSet`. This set comprises a move of the attribute `passNo`, which is actually an update of this attribute's container, a deletion of class `Passport`, and a deletion of the reference connecting `Person` and `Passport`. Subsequently, the algorithm iterates over each change in the set. For the first change, i.e., the update of the container relationship of `passNo`, the variable `element` takes the value of the attribute `passNo` in line 4. The current change `c` is of type `Update` and the element `passNo` has not been deleted or updated by Harry, so the change concerning `passNo` is just applied in line 26 without annotating any conflict. However, for the next change in `changeSet`, i.e., the deletion of `Passport`, the change `c` is an instance of `Delete`. The element `Passport` has been changed in `headModel` by Harry. Thus, the class `Passport` is annotated with a `DeleteUpdate` conflict in line 13 and marked as deleted in line 15. Please note that this class has not actually being deleted. In line 16 we create and add meta information to `Passport` saving informations like who has performed the deletion and which changes are conflicting with that deletion. Like for class `Passport`, the reference from `Person` to `Passport`, which has been updated by Harry and deleted by Sally, is annotated with a `DeleteUpdate` conflict in the last iteration over the change set.

Validation Phase. Since there were only these three changes performed by Sally, the algorithm proceeds to the validation phase starting in line 29. Now, a user specific view is derived from `mergedModel` for Sally. This view prioritizes her changes over contradicting changes performed by other modelers while still incorporating all changes that are not overlapping with her changes. Hence, this view contains the class `Person` without the class `Passport` because her deletion is prioritized over the update of Harry because she checked in later. However, the independent changes by Harry renaming the attribute `bday` to `birthday` and moving the attribute `passNo` are incorporated because they are not in contradiction with any change of Sally. This view is now validated in line 30 in the algorithm. But since there are no language violations in this model view, the algorithm terminates and the merged version in `mergedModel` is published into the repository (cf. `V2` in Fig. 5).

Merging V2 with V1c into V3. According to Fig. 2, Joe checks in his version `V1c`, which has to be merged with the head version `V2` in the repository. Hence, the input for the merge algorithm is again (i) `V1` as the common base version (`originModel` in Algorithm 1), (ii) Joe's working copy `V1c` as version to check in (`revisedModel` in Algorithm 1), and (iii) V2, which is the most recent version in the repository (`headModel` in Algorithm 1).

Between V1 and V2, Harry and Sally performed several changes, which are encompassed in V2. Recall that Harry updated the attribute `bday` as well as the type of the reference from `Person` to `Passport`. Moreover, he added two attributes to `Passport`. Sally removed the class `Passport` and also the reference between `Person` and `Passport`.

Fusion Phase. When Joe checks in his version, a clone of the most recent version V2 is created and his changes are calculated and saved in `changeSet`. This set now contains three changes, namely the update of the name of the attribute `bday` to `doB`, the update of the multiplicity of the reference from `unspecified` to `unbound`, and the addition of the new attribute `expiry` in class `Passport`. In the first iteration over `changeSet`, the variable `element` contains the changed attribute `bday` (line 4) and `c` is an instance of `Update`. Hence, the condition in line 17 is fulfilled. Because the element has not been deleted since their common base version V1, no conflict annotations have to be added. In line 21 the feature currently updated by `c` is determined. Since Joe renamed the attribute, `feature` points at the name feature of attributes. Unfortunately, the name of this attribute has also been changed by Harry. Thus, an `UpdateUpdate` annotation for the name feature of the `element` is added in line 23. Finally, the change is applied. Please note that although the name update of the attribute is applied to the merged version, we do not delete the attribute names "bday" and "birthday". The previous values are saved in the meta information of an element by the `apply` method. The second iteration concerns the update of the reference multiplicity. Again, the block in line 17 to 27 is executed because `c` is an instance of `Update`. Now the updated element, i.e., the reference, has been deleted by Sally. Consequently, a `DeleteUpdate` annotation is added to the reference. As mentioned before, Harry also updated the same reference. He updated a different feature than Joe did. Joe updated the multiplicity whereas Harry updated the aggregation type. Therefore, no conflict annotations are necessary and the update concerning the multiplicity is just applied to the new merged version. In the final iteration, the variable `c` contains a change of type `Insert`. For inserts the variable `element` is `null` because there is no element at this point of time in the merged version contained by the variable `mergedModel` with the ID of the added element. Hence, the container of the added element is fetched in line 6 to check for concurrent deletion. Please note that the container must be available in the model saved in variable *mergedVersion* because the comparison will only report an insert for the added element at the highest position in the containment tree and no more inserts for its implicitly added children. In this case, the container is the class `Passport`, which has been deleted by Sally. Consequently, a `DeleteUpdate` conflict is annotated in line 8 and the insert is applied in line 10.

Validation Phase. After all changes have been handled, we may move on to the validation phase. Like before, a user specific view is derived from the model saved in `mergedModel`. This view corresponds to the model V3 depicted in Fig. 5.

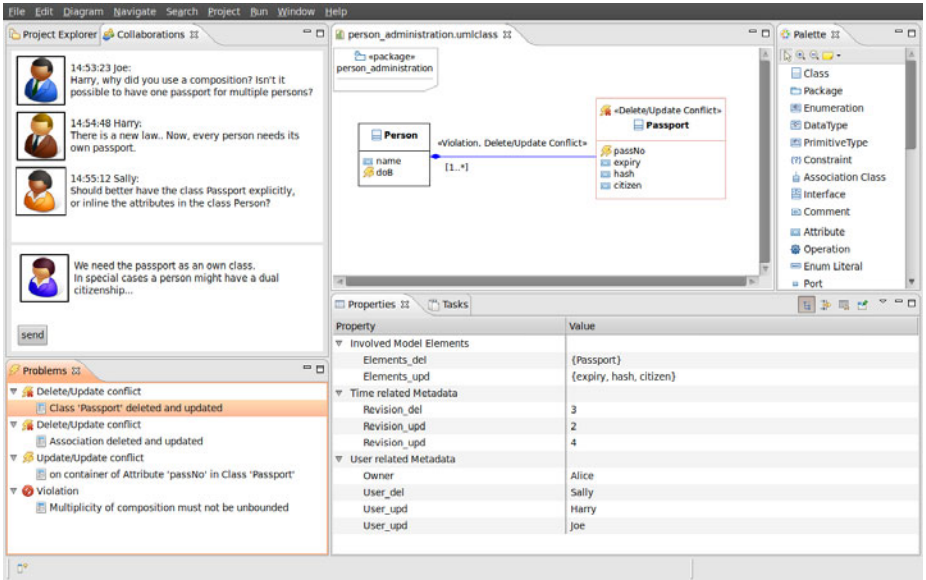


Fig. 6. Screenshot

When this model is validated, a language violation is reported. References of type `composition` may not have an unbound multiplicity (notated as $*$). Consequently, each element involved in this violation is marked. In our example, only the reference is involved and annotated.

5 Consolidation

After all developers have finished to contribute changes to the repository, the all-encompassing head revision in the repository might contain several conflicts and inconsistencies. At a certain point in time during the software development project, a consolidated model version which reflects a unified view on the modeled domain has to be found by all participants. The *consolidation phase* is supported by an adequate visualization of all conflicts in the unconsolidated model. This view serves as a basis to discuss existing issues and different points of view. Fig. 6 depicts a screenshot of the user interface for consideration, which shows all necessary components to resolve the conflicts. On the left side a chat window is provided for discussing the problems to solve. At the bottom left, the conflict list is presented. Each conflict may be selected in order to directly navigate to the involved model elements. Different icons for each type of conflict are used in the conflict report as well as in the model to provide an overview of occurred conflicts. The model itself is displayed and may be edited in the model editor situated in the middle of the window.

By clicking on one of the afore mentioned icons or by selecting a model element in conflict, the metadata describing the conflicting situation is shown in the property view at the bottom of the window. The metadata created during the conflict-tolerant merge (cf. Section 4) contains information on involved users and model elements of a specific conflict as well as the respective revisions the conflict has been introduced.

Coming back to our running example, Harry, Sally, Joe, and Alice collaborate to resolve all existing conflicts. Supported by the conflict report and the metadata, they find a consolidated version which is depicted in Fig. 7.

Delete/Update: Class Passport, Reference Person to Passport. First of all, they have to decide whether the class `Passport` is needed. Sally did not know that in special cases a person might own several passports of different countries. Since such a situation may only be modeled reasonably with an own class `Passport`, they decide to keep this class. Consequently, they also decide to retain the reference from class `Person` to `Passport`.

Update/Update: Attribute passNo. Now, it is evident how to resolve the Update/Update conflict concerning the container of the attribute `passNo`. They agree to keep `passNo` in class `Passport`.

Violation: Reference cardinality. Harry and Joe are responsible for resolving the metamodel violation, because Harry has introduced the compositional relationship and Joe has set the multiplicity to "unbound". They discuss whether more than one person might be registered in one single passport or if the existence of a passport inherently depends on the existence of the associated person. Finally, they agree to keep the composition without the `unbound` multiplicity.

Update/Update: Attribute bday. Harry and Joe have concurrently renamed the attribute `bday`, which has been introduced by Alice. Both agree, that `bday` and `doB` may lead to misunderstandings and, therefore, decide to use `birthday`.

After Harry, Sally, Joe, and Alice have finished the resolution of all conflicts, they all accept the final and consolidated version of the model, which is then saved in the repository as new version *V4*.

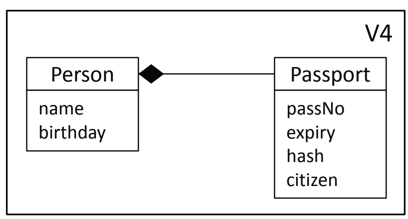


Fig. 7. Consolidated Version

The presented example illustrates that it is highly beneficial to conjointly discuss each conflict because there are different ways to resolve them due to the different viewpoints of the involved modelers. Conflict resolution is an error-prone task when only one modeler has the full responsibility to resolve conflicts. Our presented approach counteracts this problem. We retain all information necessary to make reasonable resolution decisions is kept and collaboration and discussion is fostered. The resulting final version (cf. Fig. 7) reflects all intentions much better than this could have been achieved by one modeler on her own. In summary, such a consolidation leads to a unification of the different viewpoints and finally to a model of higher quality accepted by all team members.

6 Related Work

With respect to our approach of tolerating conflicts between different model versions, we basically distinguish two areas of related work. First, dedicated systems for versioning software artifacts, and second, approaches dealing with tolerating inconsistencies in software engineering are discussed in the following.

Versioning Systems. In the last decades a lot of research approaches in the domain of software versioning have been published which are profoundly outlined in [8] and [6]. Most of them mainly focus on versioning of source code as they deal with software artifacts in a textual manner. Still, dedicated approaches aiming at the versioning of software models exist. For example, Odyssey-VCS [9] supports the versioning of UML models. EMF Compare [10] is an Eclipse plug-in which is able to match, to compare, and to merge models conforming to the Eclipse Modeling Framework (EMF). CoObRA [11] is integrated in the Fujaba tool suite and logs the changes performed on an artifact. The modifications performed by the modeler who did the later commit are replayed on the updated version of the repository. Conflicts occur if an operation may not be applied due to a violated precondition. Unicase [12] is an Eclipse-based CASE-tool comprising a repository for versioning models. The provided three-way merge technique makes use of editing operations similar to CoObRA.

Although all mentioned versioning systems explicitly support models and different conflict detection mechanisms, they are not aiming at tolerating conflicts. Instead, the conflicts have to be resolved at check-in time by one modeler, only, as it is known from code versioning systems regardless of the development phase.

Tolerating Inconsistencies in Software Engineering. During the late 80ies to the late 90ies, several works have been published which aim at managing inconsistencies (conflicts may be seen as a certain kind of inconsistencies) in the software engineering process. One of the most interesting communalities of these works is that the authors considered inconsistencies not only as negative result of collaborative development, but rather see them as necessary means for identifying aspects of systems which need further analysis or which need to reflect different viewpoints of different stakeholders [13]. Originally, the need for inconsistency-aware software engineering emerged in the field of programming languages, especially when very large systems are developed by a team. Schwanke and Kaiser [14] have been one of the first who proposed to live with inconsistencies by using a specially adapted programming environment for identifying, tracking, and tolerating consistencies to a certain extent. A similar idea was followed by Balzer [15] for tolerating inconsistencies by relaxing consistency constraints. Instead of forcing the developer to resolve the inconsistencies immediately as they appear, he proposed to annotate them with so called *pollution markers*. Those markers comprise also meta information for the resolution such as who is likely capable to resolve the inconsistency and for marking code segments which are influenced by the detected inconsistencies. Furthermore, Finkelstein et al. [16] presented the ViewPoints framework for multi-perspective development allowing inconsistencies between different perspectives and their management by employing a logic-based approach which allows powerful reasoning even in cases where inconsistencies occur [17].

The presented approach of this paper is in line with the mentioned approaches for tolerating inconsistencies in software engineering. In particular, we also aim at detecting, marking, and managing inconsistencies. Concerning the marking of conflicts, we also have a kind of pollution markers as introduced by Balzer, however, we are strongly focusing on the parallel development of models, thus we have additional conflicts such as update/update or delete/update. Furthermore, we are not only marking, but we have also to merge a tailored version, in which it is possible to mark the conflicts and inconsistencies. Our goal to support this approach without adapting the implementation of the modeling environment and versioning system. This is mainly supported by the powerful dynamic extension mechanism of UML by using a dedicated conflict profile.

7 Critical Discussion and Future Work

In this paper, we proposed a novel paradigm for optimistic model versioning. Instead of forcing the modelers to resolve merge conflicts immediately, our system supports deferring the resolution decision until a consolidated decision of the involved parties has been elaborated. This approach is based on the assumption that conflicts are not considered as negative results of collaboration, but as chance for indicating improvements. Only by learning that different views exist, all intentions and requirements are covered in the system under development. This is of particular importance in early phases of the software life cycle when a general agreement has not been established yet. In order to set up a common understanding, i.e., for the specification of requirements, graphical modeling languages like UML are the tool of choice. Models provide a powerful mean to sketch and collect first ideas during brainstorming and design phases. At this time, design flaws are cheap to correct, which might be in later phases not the case anymore.

Due to their graphical representation and inherent extension mechanisms, models may be easily extended with annotations indicating conflicting modifications. These conflicts should be discussed later on within the team and resolved in a consolidation phase when a common agreement has been established. This approach demands a novel kind of thinking from the modelers. They must be able to work with temporarily incorrect models containing different viewpoints. In order to avoid distraction by the simultaneously included variants of one model, the user interface design is of particular importance, especially if the models grow big and include many conflicting modifications. Therefore, filters are indispensable to focus on the important aspects and to hide irrelevant details. Furthermore, strategies and tool support are necessary to guide the consolidation process, e.g., by a dedicated conflict browser.

We are currently enhancing our proof-of-concept implementation to perform extensive case studies—first with students of our model engineering course (about 150 students) and later in cooperation with our industrial project partner—in order to evaluate the effectiveness of our approach. By this we hope to gain further insights on how people collaborate in early phases of the software development life cycle and how they are handling conflicts in a collaborative setting.

References

1. Bézivin, J.: On the Unification Power of Models. Journal on Software and Systems Modeling 4(2), 171–188 (2005)
2. Fowler, M.: UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley Longman Publishing Co., Inc, Amsterdam (2003)
3. UML Specification 2.2, http://www.omg.org/spec/UML/2.2
4. Altmanninger, K., Brosch, P., Kappel, G., Langer, P., Seidl, M., Wieland, K., Wimmer, M.: Why Model Versioning Research is Needed!? An Experience Report. In: Joint MoDSE-MCCM 2009 Workshop @ MoDELS 2009 (2009)
5. Edwards, W.K.: Flexible Conflict Detection and Management in Collaborative Applications. In: 10th Annual ACM Symposium on User Interface Software and Technology (UIST 1997), pp. 139–148. ACM, New York (1997)
6. Mens, T.: A State-of-the-Art Survey on Software Merging. IEEE Trans. on Software Engineering 28(5), 449–462 (2002)
7. Munson, J.P., Dewan, P.: A Flexible Object Merging Framework. In: ACM Conference on Computer Supported Cooperative Work (CSCW 1994), pp. 231–242. ACM, New York (1994)
8. Conradi, R., Westfechtel, B.: Version Models for Software Configuration Management. ACM Computing Surveys 30(2), 232 (1998)
9. Murta, L., Corrêa, C., Prudêncio, J.G., Werner, C.: Towards Odyssey-VCS 2: Improvements over a UML-based Version Control System. In: 2nd Int. Workshop on Comparison and Versioning of Software Models @ ICSE 2008. ACM, New York (2008)
10. Brun, C., Pierantonio, A.: Model Differences in the Eclipse Modeling Framework. UPGRADE, The European Journal for the Informatics Professional (2008)
11. Schneider, C., Zündorf, A., Niere, J.: CoObRA - A Small Step for Development Tools to Collaborative Environments. In: Workshop on Directions in Software Engineering Environments @ ICSE 2004 (2004)
12. Kögel, M.: Towards Software Configuration Management for Unified Models. In: Proc. of the 2nd Int. Workshop on Comparison and Versioning of Software Models @ ICSE 2008. ACM, New York (2008)
13. Nuseibeh, B., Easterbrook, S.M., Russo, A.: Making Inconsistency Respectable in Software Development. Journal of Systems and Software 58(2), 171–180 (2001)
14. Schwanke, R.W., Kaiser, G.E.: Living With Inconsistency in Large Systems. In: Workshop on Software Version and Configuration Control, pp. 98–118 (1988)
15. Balzer, R.: Tolerating Inconsistency. In: 5th Int. Software Process Workshop (ISPW 1989), pp. 41–42. IEEE Computer Society, Los Alamitos (1989)
16. Finkelstein, A., Gabbay, D.M., Hunter, A., Kramer, J., Nuseibeh, B.: Inconsistency Handling in Multperspective Specifications. IEEE Trans. on Software Engineering 20(8), 569–578 (1994)
17. Hunter, A., Nuseibeh, B.: Managing Inconsistent Specifications: Reasoning, Analysis, and Action. ACM Trans. on Soft. Eng. and Meth. 7(4), 335–367 (1998)

The Integration of Collaborative Process Modeling and Electronic Brainstorming in Co-located Meetings

Thomas Herrmann and Alexander Nolte

Department Information and Technology Management,
Institute for Applied Work Science, University of Bochum, Germany
thomas.herrmann@iaw.rub.de
alexander.nolte@iaw.rub.de

Abstract. Within a workshop, tools and methods have been employed to support process modeling together with creative ideation for identifying the relevant elements of the process. The process-design workshop was part of a project which aims at implementing new ways of data input and transfer for the coordination of services for elderly people. We have chosen tools and methods which seemed to be appropriate to support an efficient process design which integrates creativity and the differing perspectives of the participating stakeholders. This workshop led to a case study which revealed strengths and weaknesses of our approach and helped us to identify further recommendations and requirements for the integration of collaborative modeling and creativity support.

Keywords: Collaborative Modeling, Creativity Support, Brainstorming, Interactive Large Screen.

1 Introduction

The modeling of processes, such as business processes or workflows, in the context of software engineering is a complex task. Especially if the task allocation between several roles is represented within these models as well as the usage of technical resources, a huge number of elements and several model parts have to be integrated into a larger framework. We have collected experience with several cases where such kinds of complex models have been developed (Herrmann, 2009 [9]). Because of the complexity of the process models it is reasonable to draft them collaboratively by involving various stakeholders which serve as domain experts. Collaborative modeling (Renger et al., 2008 [13]), also called group model building (Rouwette et al., 2000 [14]) has been widely discussed in the literature. Renger et al. [13] provide a framework to analyze collaborative modeling. With respect to this framework and our experience, the collaboration can be described by referring to the following dimensions / aspects:

- Number of modelers / stakeholders who work together to achieve a first draft, to improve it, to finish the layout, to help with the quality improvement, etc.
- The cooperation between modelers and domain experts, facilitators etc. (for a more detailed differentiation of roles see Renger et al. [13]).
- Sequenced modeling vs. modeling in parallel.

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 145–160, 2010.

- Delegation of modeling to modeling experts (who contribute as chauffeurs) vs. modeling contributions by everybody.
- Modeling of existing processes vs. modeling of new processes which usually don't have a pendant in reality.

Within the range of these possibilities we focus on a specific constellation which is determined by the goals of a project which pursues the development of services. These services aim at supporting elderly people to manage an autonomous life at their own home for as long as possible. The services have to be newly designed since it is intended to employ a new technology for submitting the data which is needed to order these services and to coordinate them efficiently. Developing the socio-technical processes is the background of a case study which is described in section 2. Drafting new processes has to employ the creativity of several stakeholders and experts.

With respect to the support of creativity we expect the numbers of creative ideas to probably increase if they are developed by a group of different stakeholders who have differing backgrounds and expertise. This complies with Csikszentmihalyi's [4] observation that "an idea or product that deserves the label 'creative' arises from the synergy of many sources and not only from the mind of a single person". We call this phenomenon "collaborative creativity" (Mamykina et al., 2002 [10]). Collaboration can be considered as a process where people work together and usually know each other, and at least have opportunities to give feedback to each other's ideas and work. Fischer et al. outline that collaborative creativity (in their words "social creativity") draws advantage from including different people with different backgrounds (spatial, temporal, cultural etc.) and that conceptual collision can enrich the collaboration (Fischer et al., 2004 [7]).

Within a group of different people who don't know each other very well, a number of creativity barriers have to be taken into account which are described in the ideation literature and which also apply to our setting. Working in groups may prove less effective for various reasons (Diehl and Stroebe, 1987 [6]; Santanen, 2005 [16]): Production blocking may occur because people wait for a turn to speak, especially if a process is linearly drafted. While waiting they may forget some ideas before they can report them; they may not generate new ideas while listening to others, or while trying not to forget their own thoughts. Free-riding occurs when people stop generating own ideas, but rely on others who make contributions. A further problem is the fear of being evaluated by others (evaluation apprehension). A general obstacle is that people stay within the boundaries of a certain kind of ideas which was voiced at the beginning of a brainstorming session (cognitive inertia, cf. Briggs and Reinig, 2007 [2]). This problem of cognitive inertia can be reduced by strategies of varying promptings (Santanen et al., 2004 [15]). In the case of the combination of process modeling with creative ideation, it is a special requirement that interventions with varying prompts can be easily employed. Therefore various ThinkLets, as described by Briggs and de Vreede [3] should be flexibly employable.

Within our case study we have organized a workshop where several stakeholders participated in drafting a new process model. We have used a special facilitation collaboratory (ModLab) where laptops and a large interactive screen can be coupled. Based on preceding experiences with other workshops we found that we had to employ methods and tools which help to meet the following challenges:

- Sequenced process modeling and brainstorming had to be combined.
- People cannot have the complete process in mind but need to jump between different areas or phases of the process when they firstly start to draft it.
- Manifold options for collaboration are reasonable such as: people think in solitude about their possible contributions to the teamwork; take inspirations into account by observing what others are contributing; they vary the intensity of their communication; parallel drafting of process parts alternates with phases of deliberately drafting a process step-by-step.
- The stakeholders should have the possibility either to directly contribute parts of the process diagram or to delegate this task to an expert modeler.

When we tried to choose appropriate tools and methods we became aware of the following limitations:

- Brainstorming tools and process modeling tools are mostly separated.
- Most electronic brainstorming systems (EBS, cf. Nunamaker et al., 1991 [12]) are text based – while process modeling is focused on the collection of graphical elements of different types.
- Linear walkthroughs, which are often recommended, (cf. Yourdon, 1979 [18]) are an appropriate means for careful inspection but are suboptimal for creative processes since they urge people to be passive and to wait for longer time periods and therefore lead to production blocking.

In the following section we describe our case study and the selected tools and methods. Section 3 outlines the strength and weaknesses of our approach. This serves as a basis to derive recommendations and requirements of how tools and methods have to be refined (section 4).

2 Case Study – A Workshop for Creative Process Design

Our case study focuses on the first in a series of workshops which was planned to develop a process for a newly-created service that offers accompaniment for elderly people during their weekly shopping. This service is one among several others that will support elderly people to manage an autonomous life at their own home for as long as possible. The workshop is related to the overall goal of a three year long interdisciplinary research project that aims to establish a service agency and it's IT-infrastructure. The service agency will coordinate the interplay between several service providers and elderly people (customers). Furthermore it manages the communication between the various stakeholders. Together with the coordination process a specific type of micro systems-technology – a digital pen – will be introduced to order the service. It looks and feels similar to a normal pen but it has an integrated camera, an advanced image microprocessor and a GSM-module for mobile communication. The camera tracks the customer's writings on a special paper when he or she fills in an order form. After ticking a special "send" box on this form, the data will be directly sent via a mobile phone net to the service agency.

This technology is used as a simple solution for elderly people to transfer data electronically in their everyday life instead of using a personal computer. However to make this technology acceptable, the order form must not be very complex and the achieved benefit has to become directly apparent to the users.

Therefore, the goal of the workshops was not only to develop a process for the coordination of several service providers and the communication between them and the customers. By contrast, the usage of this special micro-system pen requires on the one hand that the structure of the form complies with the limited functionality of such an input device and contains all necessary data for the coordination and planning of the services. On the other hand, entering the data with such a form must not be more complex as people are used to when they fill in a form.

In part 2.1 of this section we describe our preceding experience with other methodological variations. We will outline our approaches which shaped the facilitation of workshops, describe their greatest flaws and the requirements that we derived from them. The requirements served as a basis for the selection of the facilitation method and tools that were employed and evaluated in the case study.

2.1 Predecessors

2.1.1 The Socio-technical Walk-through (Predecessor 1):

A frequent design task is to appropriately integrate an IT-infrastructure into an existing socio-technical process. To understand this process and to plan its modification, we usually conduct walkthroughs in a series of workshops as described by the STWT method (Herrmann, 2009 [9]). We start by interviewing relevant process participants and process owners before running the first workshop. These interviews are used to create a high level model of the process. Afterwards we bring relevant stakeholders, such as process participants, domain experts, and IT-specialists together.

The workshop starts by presenting the high level model of the process. The facilitator explains it and walks through the process by asking the participants what they do at a certain point, what they need in order to perform the task at hand or who has to carry out the task. In the course of this communication, the facilitator transfers the contributions of the participants into elements according to the modeling notation. He is supported by a modeler (also called chauffeur, cf. Regner et al., 2008 [13]) who operates the modeling tool. All contributions are gathered sequentially step-by-step by repeated walkthroughs.

If an entirely new process or even new parts of it have to be created, this enforced sequentiality of the method leads to production blocking and therefore proves as its most serious disadvantage. Especially during phases where many contributions arise at the same time, people probably have to wait too long for their turn to speak and therefore forget some ideas before they are able to report them, or they may not generate new ideas while listening to others.

2.1.2 Card-Based Brainstorming (Predecessor 2):

If it is necessary to gather ideas for the tasks, roles, resources etc. of a new process, brainstorming with physical cards is obviously a possible approach. Every participant receives a stack of cards. The facilitator presents a brainstorming question as a prompt that addresses an important aspect of the process and asks the participants to write their ideas onto the cards. After a while the cards are gathered and put upon a wall to

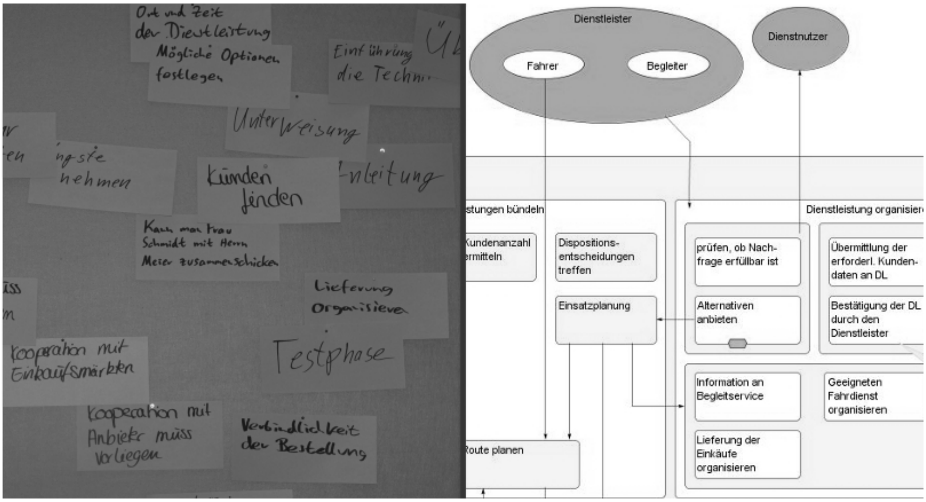


Fig. 1. Clustered brainstorming cards (left) compared to a SeeMe model (right)

be explained and clustered in relation to their content or to their position in the process. After the clustering, the facilitator takes digital pictures, transfers the written cards into graphical representations and inserts them into a process model. This model is displayed in the following workshop in order to continue its development.

Finding a suitable brainstorming procedure that complies with process design is a subtle task since there are many different approaches for ideation that might contribute appropriately (Briggs and de Vreede, 2007 [3]). The card based brainstorming has the advantage that a participant can work with the cards, modify them, exchange them, and personally arrange them in a geometric order before s/he passes the cards to the facilitator. During this phase, several participants can work in parallel. The disadvantages are that someone has to transcribe the cards before they can be integrated into a graphical process model – this causes a break and prevents a smooth interplay between brainstorming and the refinement of the process model. Furthermore, the visualization of the model that evolves from the transcription of the cards has an entirely different look and feel than the original cards (see Fig. 1). This forces the participants to completely reorientate. Furthermore they don't have enough evidence whether the original collection of their ideas complies with the final outcome of a process model.

2.1.3 Electronic Brainstorming (Predecessor 3):

As card based brainstorming does not well support the seamless integration of the results into the process model but appears to be useful for ideation, we tried an electronic brainstorming tool. The participants contribute with a laptop to the brainstorming. Mindmeister [11] – the tool we used – is web based and provides different user interfaces for the participants and the facilitator. The facilitator presents a brainstorming question that appears on a large screen in front of the participants. They enter their contributions on their laptop and send them to the screen with a simple user interface that consists of only one text-entry field. The facilitator stops the brainstorming after a couple of minutes and clusters the contributions according to different aspects of the process. After the workshop, contributions are converted into a graphical process model which is used in the following workshop for the further development.

In contrast to card-based brainstorming (Predecessor 2) this approach does not lead to the necessity to transcribe the contributions, furthermore it is possible to work in parallel and to contribute anonymously. However, someone still has to transfer the brainstorming items into a process model. Furthermore the problem of the previous predecessor remains that it is still difficult for the participants in the following workshops to compare the mindmap structure with the process model.

2.2 Derived Requirements from the Predecessors

Our experience with these predecessors combined with the challenges that were outlined in section 1 led to the following requirements for the integration of electronic brainstorming with co-located collaborative modeling:

- The ability to contribute in parallel to the process diagram helps to prevent production blocking and thus is an integral part of an environment to promote the creative development of a new process.
- The complexity threshold for the participants to contribute to the brainstorming has to be as low as possible to enable them to focus on their creative task.
- Participants have to be able to decide whether they want to think in solitude about their contributions or to observe the contributions of others as an inspiration.
- Brainstorming contributions have to be directly put into the process model in order to make the orientation for the participants easier.
- The brainstorming results should appear as elements of the process model while the look and feel of real brainstorming cards as well as their character of preliminarity is still maintained.

These requirements serve as heuristics for the development of features that support the integration of brainstorming into a co-located collaborative modeling environment. Additionally, the tools have to be integrated into an existing socio-technical infrastructure that will be described in the following section.

2.3 Technical Infrastructure and Environment

The tools we developed are applied in a facilitation collaboratory (ModLab) at the University of Bochum, Germany. Its centerpiece is a large, high-resolution interactive screen (4.80m x 1.20m; 4320x1050 pixels). The whole surface is touch sensitive and allows seamless operations over the whole width of the screen. Additionally, a wifi-network enables communication between the audience and the large screen with suitable devices like laptops or smart phones.

We employ the SeeMe modeling method (Herrmann and Loser, 1999 [8]) for the development and visualization of socio-technical processes. SeeMe is comparable and compatible with many other modeling methods but has some distinctive features: explicit indicators for incompleteness and uncertainty, rough as well as complete specification of relationships, multi-perspective decomposition of elements, indication of space for free decision-making. The advantage of such a semi-formal representation is that it is suitable when expressing the contingent relationships of social structures as well as the formal specifications of a technical solution.

Fig. 2. The ModLab – University of Bochum

The related SeeMe editor [17] serves as a graphical modeling tool for the development of SeeMe models. It was firstly developed to be chauffeured by a modeler in a collaborative environment with different stakeholders, users and domain experts while a facilitator manages the communication. As it is developed by ourselves we are able to enhance it to meet the requirements that we derived from the predecessors as well as to fit it into the ModLab infrastructure. We will explain these enhancements in the following sub-section.

2.4 Tools to Integrate Electronic Brainstorming into Collaborative Modeling

After having reviewed the predecessors, we developed a prototype that covers the requirements presented in section 2.2. The solution includes two different user interfaces, one for the facilitator and one for the participants. The one for the facilitator is an enhancement of the SeeMe editor with a special UI for the purpose of moderating a brainstorming session (see Fig. 3). It is designed to be operated entirely on an interactive large screen and enables the facilitator to determine one or multiple areas inside the model where the brainstorming cards will be collected (see the frame at the tip of the needle in Fig. 3). We will describe the facilitation method and the handling of the interface in section 2.5.

Fig. 3. The SeeMe Editor – enhanced for the purpose of brainstorming

The interface for the participants however has an entirely different look and feel (see Fig. 4). It is a small website that can be operated on any kind of device that supports a web-browser so that all participants can contribute to the brainstorming with their own devices like a laptop or smart phone. The UI itself is designed to be as simply to use as possible. To contribute to the brainstorming participants just have to enter their contribution into the *text* field and press the *Send card* button. Optionally they can enter a name or a pseudonym to personalize the contribution or indicate an element type of the SeeMe modeling language if it seems to be necessary. The brainstorming topic on top of the interface as well as a chronological listing of their own contributions on the right serves as a feedback for the participants.

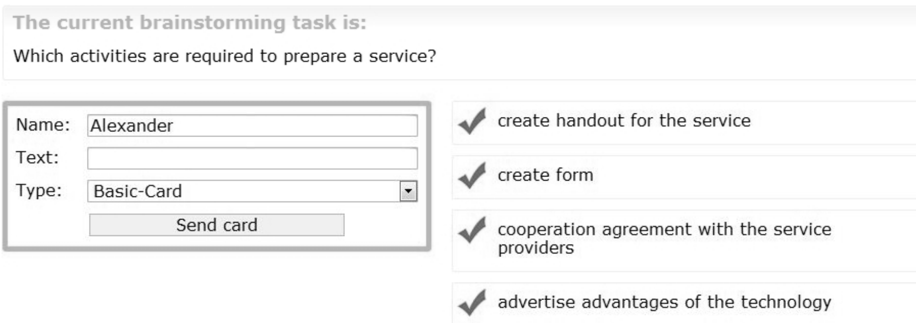


Fig. 4. The brainstorming web interface

The awareness for contributions is intentionally distributed between the web-interface and the large screen: While just working with the web-UI the participants decide to stay with their own contributions and not to be distracted by others' ideas. If they want to be inspired by the ideas of others they can decide to switch their attention to the large screen. This constellation allows different modes of collaboration.

The communication between the participant's web interface and the SeeMe editor on the interactive large screen is managed by a server backbone. The web interface runs on a Jetty web server. The contributions are passed to an XMPP / Jabber server via AJAX and are handed over to the SeeMe editor. We establish an XMPP instant messaging server as the center of the communication to gain more flexibility for the integration of further client input systems like an iPhone-App or digital pictures of physical brainstorming cards. The infrastructure in its whole enables the transition of written contributions into elements of a graphical modeling tool. The following section presents the integration of this technical infrastructure into a facilitated brainstorming procedure.

2.5 A Socio-technical Procedure for the Integration of Brainstorming and Process Modeling

The whole workshop was planned to last 3.5 hours. We invited 11 individuals to serve as participants. Their heterogeneity covered aspects such as gender (5 female, 6 male), age (range: 26 to 57 years), status (students, postdocs, research assistants, full professors, practitioners) and professional background. The participants contributed their experience from several perspectives based on their professional background. Some of them were involved as academics in the research on process design. Others serve as domain experts and service professionals who work in nursing homes or as service providers and understand the needs of elderly people. They played a decisive role in the workshop by providing firsthand experience with services for elderly people through their everyday experience. Furthermore as we plan to use a special kind of micro-technology, we invited two IT-experts who are particularly experienced in the field of mobile computing and client-server-architecture.

When preparing the workshop we created a very high level model of the process that was displayed on the large screen to serve as the focus of the workshop. It included three main activities (*service preparation*, *coordination of service requests* and *service provision*) that covered the coordination and communication aspect of the process. Furthermore we added one entity (*required user data*) to prepare the planning of the IT-infrastructure.

As it was unsure whether the participants would be able to handle the technology from the start, we began the workshop with a warm-up. The facilitator opened an empty modeling area and created a brainstorming field (see Fig. 3) with enough space for the expected contributions. Then he brought up the following brainstorming question: "Where do you want to go on vacation next summer?" Additionally he provided an URL that the participants were supposed to enter into the browser of the laptop in front of them. This URL led them to the brainstorming web interface (see Fig. 4) and they were told to start contributing to the question.

After the facilitator felt that all participants would be able to handle the web interface he told them to stop and opened the model that was created during the preparation of the workshop. He created a brainstorming area around the activity *service preparation* and provided the associated brainstorming question: "Which activities are required to prepare a service?" (See *Brainstorming 1* in Table 1) Afterwards the participants were told to press the reload button on their browser, choose the type *activity* and start with the brainstorming.

It lasted for about six minutes before the facilitator told the participants to stop as he felt that there were no more ideas to be expected. Afterwards the facilitator gave the opportunity to explain unclear contributions and clustered the collected items according to topics that emerged during the discussion. If duplicates appeared they

Table 1. Brainstorming Facts

	Brainstorming 1	Brainstorming 2	Brainstorming 3
Task	Activities that are needed to prepare a service	Data that is required from the user	Activities that are needed to coordinate a service
Area of the model	Service preparation	Required user data	Coordination of service requests
Number of elements	39	46	44
Time	6 Minutes	5 Minutes	8 Minutes
Duplicates	None	5	1

were merged and marked with a comment. The facilitator managed the discussion as well as the clustering by dragging the contributions on the interactive large screen. Afterwards, the clusters and their chronological order within their parent-activity *service preparation* were discussed. The facilitation of this discussion led to moving the clusters of elements to their appropriate destination within the process model.

The following two brainstorming tasks (see *Brainstorming 2* and *Brainstorming 3* in Table 1) were similarly facilitated. Only the brainstorming prompt and the area of the model, to which the collected items were assigned to, had to be newly specified.

3 Evaluation of the Case Study

To evaluate the case study, one of the workshop participants played also the role of an observer. His goal was to document firsthand experience with the tools by keeping track of how the participants interacted with them. Furthermore, he observed the interplay between the tools and the socio-technical environment. He tracked how the environment affected the behavior of the participants and whether they were able to focus on the brainstorming task while using the tools. Finally his previous involvement into the development of the tools enabled him to distinguish between influencing factors of the software and effects that were caused by the procedure of the workshop. To gain an insight into the facilitator's experience during the workshop we conducted a qualitative interview with him afterwards where he gave a report of his experience with the tools and with the procedure of the workshop. He also compared his experience with the facilitation of previous STWT workshops and described advantages and disadvantages of the employed procedure.

The following results of our case study provide an insight into the most obvious strengths and weaknesses of the technical infrastructure and of its usage during the workshop. Additionally, these findings serve as a basis to derive further requirements which will be outlined in section 4.

The participant's interface was easy to use
As already stated in section 2.4 we wanted the interface of the participants to be simple enough so that they could focus on their main task. According to the numbers (see Table 1) it worked quite well as 11 participants contributed 129 brainstorming items in just 19 minutes. On average, every participant contributed an idea at least every 90 seconds. The training phase had an important role in this context because all little problems and questions could be solved so that no problems – as far as it could be observed – disturbed the real brainstorming.

The facilitator's interface still needs to be improved
During our post-workshop interview the facilitator stated that the creation of a brainstorming area was difficult to handle while keeping track of the communication. He suggested that it would be more advantageous for him if he could just pin the needle (see Fig. 3) to an existing element that he wants to use as a brainstorming area. Furthermore the facilitator still had to enter the brainstorming prompt by himself using a keyboard during the workshop. He suggested that it would be easier if he could prepare the prompts before the workshop. During the workshop he could refer to these prepared prompts and just add them to the relevant areas. Some of these needs could

have been assumed before the workshop took place – however, because of our experience with similar cases we know that under real conditions where the facilitator has to be fully attentive and may be stressed, new requirements become apparent and priorities for improvement may change.

Clustering the contributions after the brainstorming
The interactive large screen enabled the facilitator to move elements to their designated destination with a simple drag of his fingertip. He handled this task while still being able to keep track of the communication. Merging of duplicates, however, was not supported by a suitable function but was managed by deleting one element and adding a comment to the remaining element that it occurred twice or manyfold.

Easily overviewing and moving around the brainstorming items was supported by the fact that they were small enough since the participant's interface suggests that they limit their contributions to a length of about 40 characters (see Fig. 4). Problems occur if elements have to be dragged over longer distances e.g. more than 3 meters. Furthermore the facilitator was not sufficiently supported to move a whole set of selected elements and to allocate them to a specific destination or element of the model. It was quite awkward that all contributions had to be moved one after another.

The participants could not change or enhance their own contributions
Since the contributions' length was limited some of them were quite vague and needed to be explained. This made it necessary to change some of them or to add a comment for further specification. For this task it would be suitable if participants change their own contribution or add a comment to it by themselves during the discussion. Since this was not supported by the web interface (see Fig. 4), the facilitator had to add changes or comments by himself what he found quite awkward since he had to use a keyboard to do the necessary typing. Therefore, it was sometimes difficult for him to keep track of the communication. However, it is not trivial to find an appropriate design for an interaction mode which supports users to enhance their own contributions – it is not reasonable to generally allow users to submit large text items since this is not feasible for the ongoing procedure. It may be disturbing if users change or enlarge their contributions while the facilitator is working with them or tries to focus the participants' attention on another part of the process.

Gathering the contributions as graphical elements made the post-processing considerably easier
Before the workshop started we doubted that the participants would be quickly able to verify whether their text based contributions (on the web interface) were successfully transferred into graphical elements on the large screen. According to our observations the participants did not try to do this verification very often. Furthermore, when they tried and struggled to find their contributions in the cloud of a larger number of other contributions they were still able to rely on the history of their contributions on their own web interface. Therefore, this verification problem did not seem to have a negative impact on their productivity during the brainstorming. With card-based items, new ways of two-dimensional, geometrical types of clustering can be pursued which are so far not explored in the context of current text-based EBS (Dennis and Williams, 2003 [5]).

The post-processing of the collected brainstorming items – such as clustering or allocation to the process model – was considerably easier through the use of graphical elements. It was advantageous that the items could be easily combined with any other elements of the process modeling language and thus easily be integrated into the final process model.

The production blocking effect could successfully be avoided

The most significant advantage compared to the linear approach of the socio-technical walkthrough (Predecessor 1) was that – according to our observations – we were able to avoid the production blocking effect. As already stated in section 2.1.1 the effect seemed to be primarily caused by the enforced sequentiality of the contributions. The participants had to wait for their turn to speak what may have hindered them to develop or report their ideas. Ideas may get lost before they are captured. Furthermore the participants sometimes adjusted their original idea by integrating the contributions of others. This effect may limit the divergence which can be produced by a brainstorming. By enabling parallel contributions through the web interface, the participants were no longer forced to wait and thus no longer negatively affected by the contributions of others as they could stay focused on the web interface. The web interface provided all necessary information like the brainstorming question and the participant's own contributions. It even provided the information that a contribution had been conveyed by displaying a tick in front of it (see Fig. 4).

According to the observed behavior of the participants and to the self-awareness of the observer, the contributions of others did not necessarily have negative effects on each other. They sometimes served as a prompt for the participants to foster their creativity. It caused them to rethink their original idea and to come up with a more suitable or even completely new one. When a participant felt the need for further inspiration s/he could pay attention to the contributions of others by looking at the large screen. However, we were not able to figure out if they could take into consideration every contribution they wanted to since some were still hard to read. So whether they all were suitable to cause a prompt is widely coincidental at the moment.

We were told by the participants that the advantage of our setting was the ability to choose individually whether they wanted to refer to the contributions of others or not. This leads to the assumption that the production blocking effect can be avoided as long as the participants are not forced to pay attention to the contributions of others. However, effects of distraction cannot completely be prevented as we were working within a synchronous co-located situation where all kind of influencing factors might cause production blocking.

Support of convergence by aligning brainstorming results within a process model

During the clustering phases it worked quite well to use a process model as a scaffold where the brainstorming contributions were sorted in. It helped to keep the interplay between the different brainstorming areas / process elements in mind while clustering the brainstorming items. In the course of the brainstorming itself the model provided further context so that the contributions did not drift too far apart. It was useful that a high-level model of the process had been presented at the beginning of the workshop. This model helped to divide the problem into manageable tasks and to consider various aspects separately before converging them into one consistent process model. However, it remains difficult to find a suitable degree of abstraction for the high-level model. On the one hand the model may be too precise and consequently narrows the space for creative solutions. On the other hand, a model with a too high level of abstraction can compromise the appropriateness of the contributions.

Additionally, we found several other enhancements that could improve the flexibility of the brainstorming. However, they only have the status of first ideas for further improvement or research:

- The brainstorming would gain more flexibility if the participants could freely choose to which part of the model they contribute their items. This free choice could be structured by the facilitator if he has the possibility to indicate those parts of the model that can be selected as brainstorming areas at a certain moment.
- The facilitator could have the option to offer the participants the possibility to freely indicate whether a contribution represents a specific element of a process modeling language such as an activity, a role, a document etc. In our case study, it was only possible to contribute pre-specified types of elements (see Table 1).
- Collaborative parallel clustering could considerably accelerate the process as the clustering remains one of the most time consuming factors at the moment.

All in all it becomes obvious that the necessary activities of the facilitator represent a decisive bottle neck with respect to efficiency when a brainstorming is integrated into the drafting of a process model. This observation complies with observations in literature which examines the challenges facilitators face during group sessions in general and brainstorming sessions in particular (cf. Bostrom et al., 1993 [1]).

4 Conclusions: Recommendations and Technical Requirements

The empirical basis of our research consists of a series of workshops which were intertwined with the continuous improvement of socio-technical design including the method of facilitation and the technical features of brainstorming tools. If needed, the experience with the workshops is completed by interviews with those participants who seem to have the most valuable insights. Therefore, the methodology is clearly explorative and the value of the results is to explain which features of electronic brainstorming are important for collaborative process modeling and should be taken into account for the development and testing of further prototypes.

From the analysis of our case study we derive a number of recommendations and requirements which should be met by the combination of tools with a facility method which are employed in similar workshops:

The brainstorming tool should be seamlessly integrated into the process modeling tool. The contributed brainstorming items should immediately appear as one of the basic elements of the modeling notation. This has the advantage that all kinds of geometrical operations or functions to combine elements of the process modeling language can immediately be applied on the collected brainstorming results. Sometimes – also in our case study – it can be cognitively disturbing to decide which element-type might be appropriate for the representation of a brainstorming contribution. Therefore the participants must have the choice to use a neutral element which is later converted into a meaningful symbol of the modeling language.

It is reasonable to combine a large interactive touch screen with laptops or other small input devices. While the input devices are used to collect the brainstorming contributions, the large screen is mainly useful to allow the facilitator a smooth interaction with the participants and with the graphical model itself (e.g. by highlighting elements, indicating places where a new element should be inserted, changing the location of elements etc.). With the possibilities of an interactive large screen it is much easier to bring the complete collection of process items as well as the emerging of the whole process model to the participants' attention.

The interactive large screen should be enhanced with the possibility for multi-user multi touch. Groups of up to five people should be able to work simultaneously on the large screen to carry out simple tasks. Typical examples which were also relevant in our case are voting (e.g. to indicate the most relevant process elements which should be taken into further consideration) or first steps to structure huge sets of collected items (e.g. to put them into semantic clusters or to align with a time line).

The tools for collecting the brainstorming input should be as simple as possible and be offered as a web-interface. This enables any kind of portable device like laptops or smart phones (provided a web-browser is installed) – to which the participants are used to – to be employed without any preceding configuration work.

The possibilities for input must be enhanced:

1. It should be possible to switch easily between several brainstorming tables (where each one represents a brainstorming question) to make an input. The ThinkLets-approach as presented in Briggs and de Vreede [3] describes a set of partially possible variations for brainstorming sessions within collaborative modeling.
2. It is reasonable that a brainstorming can still be completed after the facilitator has already started to work with the collected items.
3. Participants should be able to add comments to their contributions if it turns out that they have not really been understood by other stakeholders in the workshop.
4. Participants should be able to support the modeler e.g. by typing (instead of telling somebody to change the name of an element one could propose this by entering the new name immediately by oneself).

The facilitator should have more possibilities to change the questions or prompts which s/he gives to support the ideation process. Brainstorming questions or criteria for the type of answers being asked for should be easily varied – either by preparing a controlled variation of prompts or by changing them ad-hoc. It could be observed that it is sometimes difficult for the participants to pay attention to the items which are already collected during a brainstorming since single items disappear in the cloud of a larger number of contributions. This could be overcome by giving the facilitator the possibility to zoom in certain items which could serve as inspiring examples.

Usually brainstorming and the clustering of the results are separated phases. However in the course of collaborative process modeling it can be reasonable to offer flexible transitions between these phases: It is reasonable to test features which enable the participants to associate their brainstorming contributions immediately with others' elements or with certain parts of the process model if it has already emerged.

All in all the integration of brainstorming into co-located collaborative modeling proved to be quite useful. It integrates the ideation process into the process modeling. The integration of brainstorming features into a modeling tool worked quite well considering the usability benefits for the participants. However several problems and aspects remain that provide challenges for further research with respect to the support of the facilitator, the clustering of items and their convergence into a consistent process model.

Acknowledgements

We would like to thank Kai-Uwe Loser, Michael Prilla and Marc Turnwald for their various contributions, ideas and feedback that led to the development of the method as well as to ensure the quality of the software. Furthermore we want to thank Tillmann Neben in particular for the development of a first draft and Thorsten Langer for the further improvement of the software.

References

1. Bostrom, R., Anson, R., Clawson, V.: Group facilitation and group support systems. Group support systems: New perspectives, 146–168 (1993)
2. Briggs, R.O., Reinig, B.A.: Bounded Ideation Theory: A New Model of the Relationship Between Ideaquantity and Idea-quality during Ideation. In: Proceedings of the 40th HICSS, pp. 16–26. IEEE Computer Society, Los Alamitos (2007)
3. Briggs, R., de Vreede, G.-J.: ThinkLets: Building Blocks for concerted Collaboration. Briggs and de Vreede, Nebraska (2009)
4. Csikszentmihalyi, M.: Creativity: Flow and the psychology of discovery and invention. Harper Collins, New York (1996)
5. Dennis, A., Williams, M.: Electronic Brainstorming. Group creativity: Innovation through collaboration, 160–178 (2003)
6. Diehl, M., Stroebe, W.: Productivity loss in brainstorming groups: Towards the solution of a riddle. Journal of personality and social psychology 53, 497–509 (1987)
7. Fischer, G., Scharff, E., Ye, Y.: Fostering Social Creativity by Increasing Social Capital. In: Huysman, M., Wulf, V. (eds.) Social Capital and Information Technology, pp. 355–399. MIT Press, Cambridge (2004)
8. Herrmann, T., Loser, K.-U.: Vagueness in models of socio-technical systems. Behavior & Information Technology: Special Issue on Analysis of Cooperation and Communication 18(5), 313–323 (1999)
9. Herrmann, T.: Systems Design with the Socio-Technical Walkthrough. In: Whitworth, A., de Moore, B. (eds.) Handbook of Research on Socio-Technical Design and Social Networking Systems, pp. 336–351. Idea Group Publishing, Hershey (2009)
10. Mamykina, L., Candy, L., Edmonds, E.: Collaborative Creativity. Communications of the ACM 45, 96–99 (2002)
11. Mindmeister, `http://www.mindmeister.com`
12. Nunamaker, J.F., Dennis, A.R., Valacich, J.S., Vogel, D., George, J.F.: Electronic meeting systems. Communications of the ACM 34, 40–61 (1991)
13. Renger, M., Kolfschoten, G., de Vreede, G.-J.: Challenges in collaborative modelling: a literature review and research agenda. International Journal of Simulation and Process Modelling 4, 248–263 (2008)

14. Rouwette, E., Vennix, J., Thijssen, C.: Group model building: A decision room approach. Simulation & Gaming 31, 359–379 (2000)
15. Santanen, E.L., Briggs, R.O., de Vreede, G.-J.: Causal Relationships in Creative Problem Solving: Comparing Facilitation Interventions for Ideation. Journal of Management Information Systems 20, 167–198 (2004)
16. Santanen, E.L.: Resolving Ideation Paradoxes: Seeing Apples as Oranges through the Clarity of ThinkLets. In: Proceedings of the 38th HICSS, pp. 16–26. IEEE Computer Society, Los Alamitos (2005)
17. SeeMe Editor, `http://www.seeme-imtm.de`
18. Yourdon, E.: Structured walkthroughs. Prentice Hall PTR, Upper Saddle River (1979)

Integrating Context-Enriched Explanations and Communication in an Adaptive Collaboration Environment

Syed Sajid Hussain[1], Dirk Veiel[1], Joerg M. Haake[1], and Stephan Lukosch[2]

[1] Faculty of Mathematics and Computer Science, FernUniversität in Hagen, 58084 Hagen, Germany
{sajid.shah,dirk.veiel,joerg.haake}@fernuni-hagen.de
[2] Faculty of Technology, Policy, and Management, Delft University of Technology, PO box 5015, 2600 GA Delft, The Netherlands
s.g.lukosch@tudelft.nl

Abstract. Distributed collaboration among teams involves dynamically changing situations. Making collaboration environment fit the needs of these situations is supported by system-initiated adaptation based on user and team context. Such adaptations may confuse the users, because they cannot remember all adaptation policies. In this paper we propose an approach to present context-enriched explanations to help them understand the adaptation behavior. Also, we describe a social network analysis strategy to identify socially related and current situation relevant communication partners. Integrated on demand communication is facilitated among these partners for discussion and learning about adaptation policies.

Keywords: Context, adaptation, collaboration environment, process support, context-enriched explanation, communication, socially related relevant partner.

1 Introduction

Collaboration in modern organizations has become ubiquitous. Distributed teams solve complex problems using multidisciplinary knowledge and skills of team members. Such collaboration poses enormous challenges: it occurs on the spur and in-situ, is often ill-structured, and involves a variety of dynamically changing situations [1]. Collaboration Environments (CE) provide a variety of tools and services to support communication, coordination and collaboration among team members. However, these tools and services are not coherently integrated to cope with these challenges insofar as users have to manually select tools and services, configure and adjust them as they are required.

Context-based adaptation approach aims to bridge the gap between the actual needs of collaborating end-users and the functionality provided by their CE [2]. Context-based Adaptation and Collaboration Technology (CONTact) [3,4,5,6] is a service oriented context-based adaptive shared workspace CE. It supports automated system-initiated adaptation of CE based on user and team interaction context. CONTact provides a

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 161–168, 2010.

framework and a runtime system allowing context-based adaptive tools to register to it, and use context information to reconfigure their behavior according to changing context as defined by the adaptation policies.

End-users facing system-initiated adaptation may be confused because they (especially less experienced users) cannot remember all adaptation policies. Therefore as prescribed by the meta-design framework [7], *integrated* and *situated* system support is needed to help them understand the adaptation behavior. In [8] we have proposed a theoretical process model guiding the interaction among end-users when dealing with context-based adaptations in CONTact in accordance with the meta-design framework. This process model guides us as designers of CE about the in-use state of adaptations, thereby providing requirements for supporting end-users consuming and improving adaptations. In this paper, we present an approach to implement context enriched explanations to help avoid confusion and to facilitate understanding adaptation policies in CONTact CE using our process model. Moreover, we describe an approach of social network analysis to identify relevant communication partners. Explanations include hyperlinks to such communication partners to help address further need for on demand discussion, clarification and learning from one another.

The rest of this paper is organized as follows: In section 2, we analyze the problem to identify requirements. Section 3 presents our approach of extending the domain model to present context-enriched explanations and communication with relevant partners. Section 4 concludes this paper summarizing the contributions.

2 Problem Analysis

In the following, we use a sample scenario to describe how users can get confused after a context-based adaption has been applied to the CE identifying requirements our system has to deal with. Further requirements stem from the integration of our approach into the current prototype of the CONTact CE.

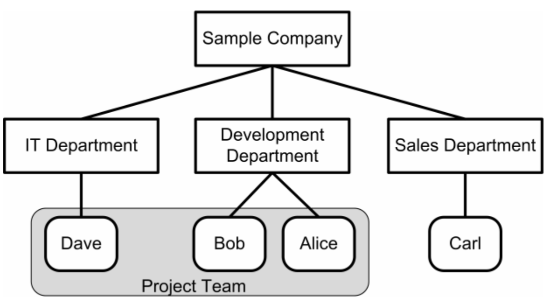


Fig. 1. Organization chart of our *Sample Company*

Figure 1 shows the organization chart of our Sample Company. Alice, Bob, and Dave are members of our Project Team. Alice is the head of the Development Department and manages the project. Bob is responsible for the more technical part of the project and, because of that, the personal point of contact for Dave who is an expert in managing networks and their infrastructure. The Project Team has to prepare a product specification and presentation for the customer. Now assume, few days before a presentation for the customer, Dave updates his part regarding the network topology and the corresponding constraints. Next morning, while driving to work, he recognizes that one important fact is missing. At the office, he opens the shared workspace, where the presentation is stored and notices that it is locked.

He gets confused because someone has removed his access rights. Consider there may be multiple adaptation policies applicable to this collaboration state (e.g., shared editing, versioning, read only view, and locking). Also, the presentation may be locked by a user for editing, as well by the system because of an expired deadline. Therefore, despite knowing that the system can adapt itself to the current collaboration state by applying context-based adaptations, Dave is not able to figure out what adaptation caused the locking of the presentation and why. What Dave does not know is that Alice is presenting the current version of the presentation to Carl of the Sales Department, to get valuable feedback from him. This sample conflict leads to the question, how a system can help avoiding or getting out of this confusion. As described in [8] we have proposed a theoretical process model guiding the interaction among end-users when dealing with context-based adaptations. This process model provides end-user support requirements while dealing with adaptations. Support for adaptation understanding is required to help Dave understand the changed system configuration of (in our sample scenario removed the access rights). Such understanding can be supported by presenting an explanation that takes the current situation into account.

R1: *The CE has to support situation-aware explanation to help understand the adaptation.*

As it is a normal human behavior to discuss with relevant communication partners. Therefore, further discussion can be facilitated by supporting integrated means of communication with users who are able to explain the adaptation policies and in some cases the vocabulary used in the formulation of the situation-aware explanation.

R2: *The CE has to provide integrated communication support for further explanation and discussion.*

These requirements have to be fulfilled by the current CONTact CE prototype. In the following section, we briefly introduce CONTact architecture and identify additional requirements caused by the needed integration.

2.1 CONTact Architecture

In CONTact CE every user has her workspace. Collaborative applications register themselves to the workspace. The applications specify the adaptations they support, and the context information they can provide. Adaptation policies (implemented as rules) specify under which conditions (based on properties of the context) what adaptations are to be performed. The runtime system senses the user and team context by collecting respective information from the applications, and if a condition of a specific adaptation policy is met, it is executed. This causes the applications to adapt.

CONTact uses four-layered framework for context-based adaptations consisting of the knowledge layer, the state layer, the contextualization layer, and the adaptation layer as described in [2]. The knowledge layer contains all relevant conceptual and factual knowledge about the domain. Our collaboration domain model for describing CE and collaborative situations to manage the context and handle context-based adaptations is described in detail in [5]. The state layer contains information about the current situation including information about physical environment, computing environment, resources and user model. In the contextualization layer, contextualization rules define which subset of the state is currently relevant. Upon this, adaptation policies defined in the adaptation layer are selected. From the set of policies, the relevant policies are identified using the contextualized state. To support **R1** at the CONTact CE, we have to extend the current prototype leading to further requirements:

R3: *The adaptation rule has to support an explanation block.*

R4: *The explanation block has to support mechanisms to add valuable information about the current situation to the explanation at execution time.*

R5: *CONTact client has to present the situation-aware explanation on demand.*

Requirement **R2** leads to the following requirements:

R6: *CONTact has to find possible socially related and relevant partners that may explain the current situation to the confused user.*

R7: *CONTact has to offer means for communication with socially related and relevant partners.*

```
BEGIN:VCARD
VERSION:2.1
N:Dylon;Bob
FN:Bob Dylon
ORG:Sample Company
TITLE:Head of Project Team
TEL;WORK;VOICE:(02331) 4567-778
ADR;WORK;PREF:;;Sample Street 1;
  Sample City;;12345;SampleLand
LABEL;WORK;PREF;ENCODING=QUOTED-
  PRINTABLE:Sample Street
  1=0D=0A=12345
  Sample City=0D=0A= SampleLand
X-MS-OL-DEFAULT-POSTAL-ADDRESS:2
URL;WORK:
  www.sample-comany.com/staff/bob
EMAIL;PREF;INTERNET:
  bob@sample-company.com
X-MS-IMADDRESS:
  bob@sample-company.com
REV:20100402T090910Z
END:VCARD
```

Fig. 2. An excerpt of *Bob*'s VCard

```
<rdf:RDF
 xmlns:rdf=
  "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs=
  "http://www.w3.org/2000/01/rdf-schema#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/">
<foaf:Person rdf:ID="Bob">
<foaf:name>Bob</foaf:name>
<foaf:mbox rdf:resource=
 "mailto:bob@sample-company.com"/>
<foaf:knows>
 <foaf:Person>
  <foaf:name>Alice</foaf:name>
  <foaf:mbox rdf:resource=
   "mailto:alice@sample-company.com"/>
 </foaf:Person>
 <foaf:Person>
  <foaf:name>Dave</foaf:name>
  <foaf:mbox rdf:resource=
   "mailto:dave@sample-company.com"/>
 </foaf:Person>
</foaf:knows>
</foaf:Person>
</rdf:RDF>
```

Fig. 3. An excerpt of *Bob*'s FOAF profile

3 Approach

Our approach consists of presenting context-enriched explanations and integrated communication to conform to the identified requirements. We extend our domain model and integrate information about the social relations between users (cf. **R6**) and their communication details (cf. **R7**).

3.1 Extending the Domain Model

We extend our domain model for collaboration by adding the ontology *vCard* [9]. We use the *vCard* ontology to address **R7** because it is standardized and supported by different tools (e.g., Microsoft Outlook). An excerpt of *Bob's* information in *vCard* is shown in Figure 2.

Additionally, we support a basic implementation of the *Friend of a Friend* (*FOAF*) [10] pattern using the user's buddy list to address **R6**. An excerpt of *Bob's* FOAF profile (presented in Figure 3) shows that *Bob* knows *Alice* and *Dave*.

These two contributions enable us to describe the current collaboration situation of the users, their corresponding contact information, and the relations between among them focusing on the usage of the buddy list. Figure 4 shows the current contextualized state of the *Project Team* currently available at the system (i.e. they are online). The *knows* relation between *Alice* and *Bob*, and *Bob* and *Dave* are established by using the aforementioned *FOAF* information. The corresponding *vCard* information is not shown to preserve the readability. The arrows in light grey will be removed after applying the adaptation rule (shown in Figure 5).

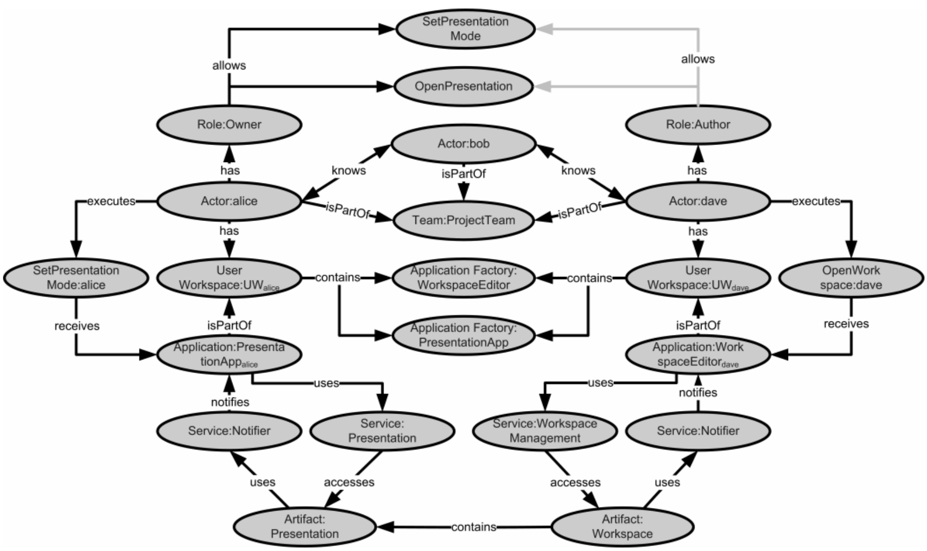


Fig. 4. Current contextualized state of the collaboration situation

3.2 Context Enriched Explanations

Now, the context representation contains information that enables us to address requirement **R1**, i.e. to enrich explanations of applied adaption rules to make user better understand the intentions, terminology, and consequences of the corresponding adaption rule and the system. Currently, an adaption rule consists of a block of conditions to be fulfilled, before the corresponding action block will be executed. The two system variables used in the following adaptation rule (`${focus}` and `${time}`) are initialized before executing it. To support situation-aware explanations we propose to add an explanation block (cf. **R3**) that is able to use the bound variables from the conditions block to add situation specific information to the explanation (cf. **R4**).

The adaption rule shown in Figure 5 is the source of the confusion of *Dave* in our sample scenario (cf. section 2). After this adaptation rule is applied, only the presenter (*Alice*) has rights to execute actions on the corresponding presentation artifact.

In our scenario, the above adaptation rule is triggered by *Alice (Actor:alice)* executing the action *SetPresentationMode:alice* on the *Artifact:Presentation*. The system variable `${focus}` is initialized with *SetPresentationMode:alice* before executing the adaptation rule. The function `getArtifactsInContext` returns a set of artifacts which are in the context of the action *SetPresentationMode:alice*. To be able to handle only artifacts of type *Presentation*, we apply the function `getArtifactsOfType` to filter the found artifacts. By executing the function `getActorOfAction`, we retrieve the actor starting the presentation. The function `getActorsInContext` then calculates all actors which have access to the presentation artifacts in the context of *SetPresentationMode:alice*, i.e. in Figure 4 *Actor:alice and Actor:dave*. To avoid removing the access permissions of the presenter, we remove the presenter from the list of actors.

```
RULE "disallow actions on presented artifact"
 WHEN
  // ${focus} = SetPresentationMode:alice
  // ${time} = current timestamp
  ?artifacts: getArtifactsInContext(${focus})
  ?presentations: getArtifactsOfType(?artifact,
    "Presentation")
  ?presenter: getActorsOfAction(${focus})
  ?actors: getActorsInContext(?presentations)
    - ?presenter
 DO
  revokeRightsForActions(?actors,
    ?presentations, "OpenPresentation",
    "SetPresentationMode")
 EXPLANATION
  Because ?presenter presents ?presentations
  at ${time},you are not allowed to open
  or present it.
END
```

Fig. 5. Rule: disallow actions on presented artifact

After all these functions were applied and the returned set of actors is not an empty, the corresponding action block is executed. The function `revokeAccessRights-ForActions` removes the rights for the actors to execute the given actions on the artifact. Applying this to our sample scenario will remove *Actor:dave*'s permission to execute *OpenPresentation* and *SetPresentationMode* on the *Artifact:Presentation*.

The next step of executing the adaption rule is to create the corresponding explanation using the bound variables from the conditions block. In our sample scenario the corresponding explanation looks like this: *"Because **Alice** presents **Presentation** at **2010-04-01 09:15:00**, you are not allowed to open or present it."* This explanation is presented to the user on demand, i.e. when he or she presses the explanation button of the CONTact client. Figure 6 illustrates the situation after Dave pressed the explanation button.

3.3 Communication with Socially Relevant Partner

We address requirement **R2** (*The CE has to provide communication support for further explanation and discussion.*) by adding possible communication channels to

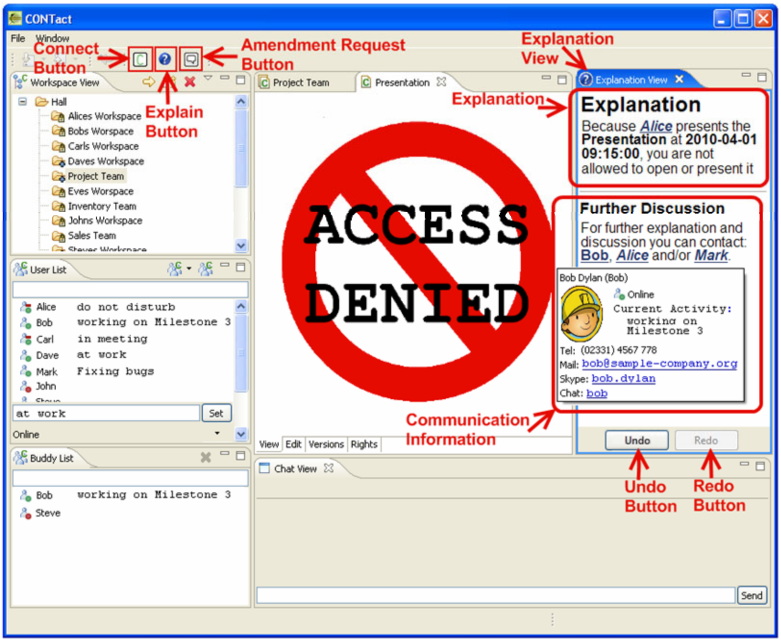


Fig. 6. A screenshot showing *Dave*'s view of CONTact client

socially related and in the current situation relevant partners to the explanation view (cf. **R5**). Creating this information takes the current context into account.

We assume that socially related persons add each other to their buddy list (cf. *FOAF* [10]). Adding someone to the buddy list leads to updates of the context representation by adding *knows* relations between the corresponding buddies (cf. **R6**). As shown in Figure 4, *Alice* knows *Bob*, and *Dave* knows *Bob*, and *Bob* knows both of them. In our assumption *knows* means that the two persons are socially related and are used to talk to each other. Hence, *Dave* is not directly connected to *Alice*, because *Dave* is used to talk to *Bob*, e.g., about the project, but not to *Alice*.

The applied adaptation rule triggered by *Alice* (cf. Figure 5) leads to the confusion of *Dave*. To help getting *Dave* out of this situation, we use the context representation to try to find possible socially related and relevant persons that may explain the current situation to him (cf. **R6**). Therefore, we try to find a path between the actor causing the confusion and the confused user. As shown in Figure 4, *Alice* and *Dave* are not directly connected, but they are related to the same person (*Bob*). So a path can be built (*Alice* – *Bob* – *Dave*). We suppose that *Dave* should ask *Bob* for further explanations because he is directly socially connected to him, and he is in the context of the current situation being in the same *Project Team*. In the case that *Bob* is not present *Alice* should be contacted, because they share the same "buddy". In worst case, none of them is accessible at the moment, so we suppose that *Dave* contacts *Mark*, the adaptation policy designer. Applying these assumptions in the sample scenario the ordered list of persons to be asked looks like this: *Bob*, *Alice*, *Mark*.

Using this ordered list of persons, the explanation view creates and shows the following additional information: "For further explanation and discussion you can contact **Bob**, ***Alice*, and/or *Mark***." To enable Dave to contact these persons, the explanation view uses the context representation to retrieve possible communication channels for them (cf. **R7**). When *Dave* clicks on a person to communicate with, the corresponding available communication channels are presented to him. Applying this mechanisms lead to the explanation view shown in Figure 6 on the right hand side.

4 Conclusions

This research in progress describes the implementation of context enriched explanations to help users understand the adaptation behavior of an adaptive collaboration environment. Furthermore, we present an approach of social network analysis to facilitate integrated on demand communication. Such communications among socially relevant partners helps learning about adaptation policies. Functional tests have demonstrated that the identified requirements are indeed met. In future, we aim to validate our approach by usability inspection methods and field studies.

References

1. Sari, B., Katzy, B., Loeh, H.: Coordination routines behind knowledge intensive work processes. In: 13th Intl. Conf. on Concurrent Enterprising (2007)
2. Haake, J.M., Hussein, T., Joop, B., Lukosch, S., Veiel, D., Ziegler, J.: Context modeling for adaptive collaboration. Technical Report 2/2009, Universität Duisburg-Essen (2009)
3. Lukosch, S., Veiel, D., Haake, J.M.: Towards a context-adaptive integrated shared work environment. Technical Report 1, FernUniversität in Hagen (2008)
4. Lukosch, S., Veiel, D., Haake, J.M.: Enabling context-adaptive collaboration for knowledge-intense processes. In: 11th Intl. Conf. on Enterprise Inf. Systems (2009)
5. Veiel, D., Haake, J.M., Lukosch, S.: Extending a shared workspace environment with context-based adaptations. In: Carriço, L., Baloian, N., Fonseca, B. (eds.) CRIWG 2009. LNCS, vol. 5784, pp. 174–181. Springer, Heidelberg (2009)
6. Veiel, D., Haake, J.M., Lukosch, S.: Facilitating team-based adaptation of shared workspaces. In: Intl. Symposium on Collaborative Technologies and Systems (2010)
7. Fischer, G., Giaccardi, E.: Meta-design: A Framework for the Future of End-User Development. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) End User Development. Human-Computer Interaction Series, vol. 9, pp. 427–457. Springer, Netherlands (2006)
8. Hussain, S.S., Haake, J.M.: Process support for context-based adaptations in a collaboration environment. In: Intl. Symposium on Collaborative Technologies and Systems (2010)
9. Representing vCard objects in RDF/XML, `http://www.w3.org/TR/vcard-rdf`
10. FOAF vocabulary specification, `http://xmlns.com/foaf/spec/`

A Tool for Training Students and Engineers in Global Software Development Practices

Miguel J. Monasor[1], Aurora Vizcaíno[2], and Mario Piattini[2]

[1] University of Castilla-La Mancha,
Campus Universitario s/n,
02007, Albacete, Spain
MiguelJ.Monasor@gmail.com

[2] Alarcos Research Group, Institute of Information Technologies & Systems,
Escuela Superior de Informática, University of Castilla-La Mancha,
Paseo de la Universidad 4,
13071, Ciudad Real, Spain
{Aurora.Vizcaino,Mario.Piattini}@uclm.es

Abstract. Global Software Development (GSD) is an emerging trend in which virtual teams work on the same projects at a distance. Despite the advantages of this shift, the collaboration between distant members becomes more difficult. Team members interact by using collaborative tools, and this collaboration is affected by time, cultural and language differences. These drawbacks lead to the need to train students and software engineers in the new collaborative skills required.

These skills can only be trained by involving learners in practical experiences, but this is not always possible since it necessitates collaboration with distant institutions (universities/firms). We have focused our work on the development of a tool with which to train these skills through the use of a virtual training environment for GSD that avoids this difficulty by placing learners in virtual GSD scenarios in which they will develop the skills needed to work on global software projects.

Keywords: Global Software Development, Engineering Education.

1 Introduction

Collaboration between virtual teams is one of the main challenges of Global Software Development (GSD) [1]. This relatively recent trend allows team members to work on the same projects in different countries by interacting through communication and collaboration tools. Face-to-face contact is not possible in these scenarios, and although GSD provides benefits such as the lower cost, the higher availability of skilled workforce or the broader area of commercialization, it also entails new problems mainly derived from distance. These problems particularly affect team members' communication, which becomes more complex, especially with the appearance of cultural and language differences [2] and time zone differences [3].

The use of a non-native language in communications creates additional drawbacks, as interlocutors are not always able to express their ideas, and the presence of different

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 169–184, 2010.

cultures, terminology and languages often cause misunderstandings and a lack of trust [2], [4]. Cultural differences may also cause problems related to legal issues and knowledge-transfer [3].

These problems are present in the different stages of the software life cycle in different ways, and affect different collaborative tools and processes.

Moreover, software factories managers often complain about the poor skill level of team members in the use of communication tools which leads to delays, a lack of trust and misunderstandings [5]. However, finding solutions to these difficulties is not easy and traditional software engineering education does not deal with these topics.

Software Engineering Education must, therefore, be focused on training students and software engineers in the problems that GSD entails [6]. However, the training of cultural differences and communication and collaboration difficulties requires practice. Since these skills are better learned by doing, the challenge consists of integrating theory into practice rather than simply learning theoretical concepts.

Many current proposals in educational environments confront this issue by coordinating practical experiences with distant learners from different cultures and languages. This entails complex problems for the instructors who must coordinate their efforts with distant institutions. Students also have problems, as they usually encounter scheduling problems when interacting with other distant learners.

The training of GSD activities requires new theoretical contents and training methods in order to avoid the great deal of coordination with distant members and institutions that is implied.

One solution that we propose to this problem is the use of a virtual training environment that can simulate realistic GSD scenarios in which learners are introduced into the context of a problem that they will solve by interacting with Virtual Agents (VAs). This interaction takes into account traditional communication tools (instant messaging and e-mail) and deals with cultural and language differences since VAs of different countries play a specific role in the scenario.

The textual interaction with VAs will allow learners to solve common project issues related to collaboration with multicultural and multidisciplinary members with regard to communication problems, information sharing and documentation.

This paper is organized as follows: Section 2 explains the related work along with the tools proposed and the skills required in GSD according to a systematic review carried out previously. Section 3 describes the virtual learning environment developed. In Section 4 we discuss the results obtained. Finally, Section 5 provides some concluding remarks and outlines our future work.

2 Related Work

As an initial step in our research, we carried out a study in order to discover the main skills needed by software engineers in GSD. We also performed a Systematic Literature Review that allowed us to discover the main strategies and proposals reported in the field of GSD training and education, along with the main tools applied in this area. The results of our study are summarized below.

2.1 Skills Required in GSD

Both students and software engineers must acquire specific skills that will allow them to carry out an effective development in order to confront the problems of globalization which are not part of their conventional education. The skill most commonly reported in literature is that of the use of computer-mediated communications [7], [8] since traditional face-to-face meetings are no longer common in GSD and the appropriate use of communication and collaboration tools is essential in these environments. Furthermore, members of different cultures take part in this interaction, signifying that members must know how to communicate effectively by using a common terminology and language, and by taking into account the different customs of the participants [9], [10].

Members must be familiar with both formal and informal means of communication. On the one hand, they must know how to write formal documents, contracts and emails in a common language [2], and on the other hand they must know how to interact by using the telephone or instant messaging services.

This interaction must be oriented towards gaining the team's trustï[9], as this is one of the common problems when interacting with distant members. In order to achieve this, participants must be versed in the concepts of conflict resolution [7] and negotiation [2] which allow them to argue and minimize problems correctly during the interaction. They must also improve their improvisation skills [11], which are essential if fluent communication is to be achieved, and which also helps to improve the team's trust and teamwork skills [12]. It is therefore important for them to have experience in working with a multidisciplinary team [13], in which different degrees of knowledge and abilities are present during communication that may cause problems in reaching an understanding and comprehension difficulties. Learners therefore need to know how to manage the ambiguity and uncertainty that are present in GSD environments [8].

Finally, learners of education programs must know the traditional methods and processes used in distributed projects [5] and acquire realistic experiences in the use of traditional knowledge management tools, document management and version control systems [14], [15]. Only experience in realistic scenarios will allow them to develop leadership skills and learn how to effectively manage their time [16].

2.2 GSD Education Proposals

Literature deals with GSD education through different kinds of proposals which aim to train the skills required by globalization. We have classified these proposals by considering the following trends:

1. Traditional theoretical classes, adapted as a response to the needs for adjustments in software engineering education [17], [18]. These courses are focused on the collaborative processes, technical issues and cultural dimensions of GSD in different areas of Software Engineering [19]. The courses are usually organized in collaboration with other distant universities [20]. However, coordination and collaboration difficulties with the different institutions [21] appear. Moreover, the participation of students with different skills and backgrounds must be considered [22].

2. Practical experience training courses, in which students put theory into practice and learn by solving typical problems that can be found in real environments. Learners interact with members of distant institutions by using email, telephone and instant messaging, and learn from their partners' skills and culture [11], [2] and by tackling processes with a close similarity to those applied in industry, including language, time and cultural problems [13] [8], [23], [10].
3. e-Learning approaches, which consist of web-based courses involving discussion boards, mail systems, chat and content management [24], [9]. Some of these use or adap WebCT, FirstClass OLAT or BlackBoard platforms, and are especially focused on improving communicative skills. We also discovered an approach in the context of collaboration that focused on artifact sharing [25].
4. Training courses in companies, which take advantage of their software engineers' real experience in order to apply the concept of a "learning network" [26] [16], in which experts in specific software development activities of the company, combine their training activities with their work as engineers. Learners can take advantage of real work experiences by maintaining contact with specialized professionals.

   Although this approach is less commonly reported in literature, [5] presents a training initiative in which a multinational organization provided a training course related to best practices of communication, trust, cultural differences and coordination.
5. Blended learning environments, applied in companies or universities which use learning platforms designed to support the development of real projects by using of collaborative tools similar to those used in real environments [14], [27], [28].

2.3 Training Tools in GSD

One of the results of our systematic literature review was the discovery of a number of tools or environments oriented towards the training of GSD activities.

Genesis [29] is a collaborative environment that can be used in educational environments and which supports formal and informal communications and the definition, enactment and control of software processes using workflow. It also uses an artifact management system, called OSCAR [30] which supports collaborative software development via artifact sharing.

Jazz [28] is another collaborative platform based on Eclipse that supports functionalities such as: source code repository, chat, web interface, reports generation, and work items. Students can use this platform to generate work items containing the relevant information related to any problem, along with the associated chat conversations.

In the same line, XPairtise [31] supports distributed pair programming practices providing coding and testing functionalities. It also supports communications through chat, shared whiteboards and a graphical Shared Editor in which pairs can cooperate by sharing their ideas. Using this Eclipse plugin, inexperts engineers can invite experts to a pair programming session who will help them to solve certain problems. The role of the "spectator" is also supported, who can watch the interaction among the pair and participate in the session chat, which can be used to teach a group of learners in a specific problem domain.

iBistro [14] is also based on the ‘learning by doing’ approach and is an environment that can be used to learn project management, software development and social skills. iBistro enables distributed members to collaborate during the software development and addresses miscommunications and information problems in informal meetings. This is achieved by allowing students to capture structures and retrieve knowledge from the meetings by using the audio, video, sketches, notes and the drawings generated. A minute generator tool stores the contextual information and allows the meetings to be represented and analyzed.

iBistro is oriented towards tacking some of the problems in GSD, and also provides intelligent support mechanisms such as that of computer supported group formation, and the ability to find stakeholders and experts in certain areas.

[32] presents a lab course based on the collaborative virtual learning environment CURE [33]. It basically uses virtual places for collaboration. These virtual places may contain pages with different contents, and communication channels such as chat, threaded mailbox, etc., and users, who can interact with other users, stand in the same virtual place.

In [34], the authors propose a framework oriented towards offshoring practices. This framework uses CodeBeamer, which is a collaborative platform that offers integrated support in project management, requirements management and code management and supports asynchronous communication through a wiki system.

In [27], the authors present a platform that integrates CURE and CodeBeamer, which allows students to develop a large software system by collaborating during all the phases of the software life cycle.

A Web-based collaborative platform is presented in [7], in which students can work with their partners in order to achieve the training module’s scopes. This is done by using communication and content management tools, including a discussion board, a file sharing repository and a project calendar. Instructors can manage the training modules by defining instructions, milestones, and deliverables.

A further collaborative environment is presented in [35], and provides learners with a set of tools such as a chat, a scribble tool, an application sharing tool, graphics tools for designing UML documents, etc. A Web portal is used to allow students to manage the groups and projects in which they are involved, and to share their personal information. They can also access their partner’s schedule and thus agree possible meeting times.

Finally, [12] proposes a framework for training learners in some of the difficulties involved in GSD which consists of tools for project scheduling and tracking, configuration management and for performing technical reviews.

However, these approaches do not completely satisfy the requirements of universities and companies as they create certain problems such as:

- a great deal of coordination with other institutions, which implies a high workload for the instructors
- dependency on other learners’ availability
- time limitations
- difficulty in reproducing realistic scenarios

- high economic costs and infrastructure requirements
- high maintainability requirements

It is therefore necessary to offer instructors an appropriate environment and training materials in order to provide learners with realistic experiences which are adjusted to the reality of companies' current requirements [28] by avoiding the aforementioned problems.

3 Our Virtual Training Environment

The Virtual Training Environment presented here places learners in realistic virtual training scenarios in which they must interact with VAs in order to perform certain typical GSD activities. Since the interaction is carried out through VAs, learners can train in cultural and language differences at any time without depending on the availability of real partners.

Our environment helps instructors to manage learners and their activities and to maintain the training scenarios. Learners can access training scenarios, take part in virtual meetings with VAs and access artifacts for that scenario.

The Meetings Simulator allows learners to textually interact with VAs which will answer their questions in relation to a problem. We can thus simulate meetings with any kind of stakeholder involved in the project by defining new VAs, and learners can play any role in the project according to the design of the training scenario.

Our environment basically provides three main components:

- **Learners' interface:** this allows learners to communicate with instructors, manage their assigned tasks and execute simulated meetings through the meetings simulator. They can also submit deliverables, access documents or UML diagrams and answer tests associated with the scenario. A Website is therefore created for each learner in order to allow the deliverables for each scenario to be submitted.
- **Instructors' interface:** this allows instructors to assign tasks and monitor learners' actions. They can also organize teams and send notices and emails to individuals or groups. It also provides an editor which allows new training scenarios to be managed and created or existing ones to be modified. This editor permits instructors to define new VAs with specific cultures and personalities for their scenarios. We thus intend to minimize the instructors' effort by providing mechanisms that enable easy customization and provide a wide set of training scenarios.
- **Central server:** Both, learners' and instructors' interfaces access the central server, which offers a set of services required by the interfaces and manages the required information stored in its database regarding learners and training GSD scenarios.

3.1 Definition of Training Scenarios

Our virtual environment works with training scenarios consisting of a set of schedules, documents, exams and virtual meetings. One or more VAs can take part in a virtual meeting, and can play a specific role in the GSD project (e.g. customer, requirements analyst, developer, project manager, etc.). A virtual meeting is also guided by a specific VA called a Virtual Colleague (VC) which has been designed to help learners during their training. This VA therefore plays the role of campaigner, which has been successfully used in other learning environments, as is reported in [36] and [37]. The VC will correct the learners' interventions by providing rationale and explaining the consequences of their actions, particularly with regard to cultural and language mistakes, but is also focused on GSD activities. The VC will also guide the learner towards following a logical sequence during the meeting.

3.1.1 Components of a Training Scenario

As is shown in Fig. 1, a training scenario is made up of one or more virtual meetings and a set of artifacts that constitute the "scenario content".

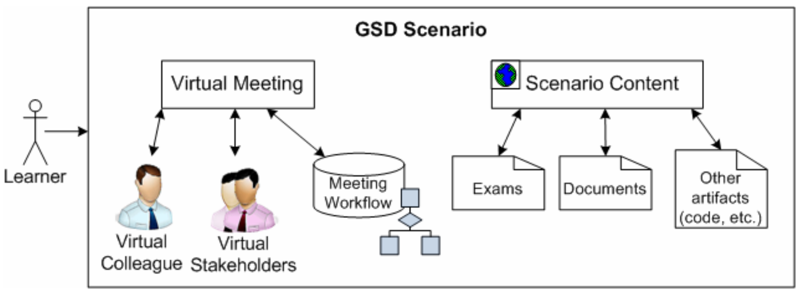


Fig. 1. GSD scenario definition

Scenario Content: Each scenario will have specific documents that learners will need or will have to complete, such as a requirements specification report. A scenario also contains exams, tests, schedules, source code, etc.

Virtual meetings: in which VAs and a VC will participate in the conversation guided by the *Meeting Workflow* by following a logical sequence according to the learners' actions.

3.1.2 Meetings Workflow

The Meeting Workflows define the course of the conversation as a set of phases, each of which defines a small part of the conversation. Each phase is defined by a specific piece of conversational knowledge, context specific language and cultural knowledge which are used for that phase in the conversation.

The phases can also store information about their priorities, which can serve to evaluate the learners' actions and the correctness of their decisions. Finally, the phases also define any gesture that VAs could make as regards the context of the conversation. For example, a VA can emulate different emotions, such as anger, anxiety, annoyance, nervousness, distress, excitement, enthusiasm, happiness and disgust.

The phases are arranged by forming a sequential diagram that defines the Meeting Workflow in which the students will influence the execution path of the meeting as a result of their textual responses. Furthermore, these phases can be *simple* or *composed*. Composed phases contain other workflows with the aim of structuring the conversation with a high granularity level, and can contain information and conversational knowledge that is inherited from the phases contained in it.

This design of the virtual meetings avoids speech repetitions and out of context interventions, making it possible to simulate profound and insightful conversations in which the VC can provide immediate feedback depending on the context of the conversation.

The different parts of the conversational knowledge of the meeting are stored in the phases related to the context of the conversation. This knowledge is stored as XML text based on patterns that can be interpreted by a chatbot engine which is used by the VAs.

The instructor´s interface allows *Meeting Workflows* to be created and edited through an editor which permits the phases and conversational knowledge required to be introduced. Apart from the conversational knowledge, the instructor can also associate the cultural and language knowledge with the context that the VC will use to provide feedbacks to the learner.

3.2 Managing Cultural and Language Problems

We have effectively managed cultural problems by designing our virtual meetings on the basis of the existing literature of Hall [38] and Hosfstede [39], and by considering the specific problems for the cultures involved in the meeting. With regard to the language problems it is necessary to study the possible problems that could appear for the languages involved in the meeting for each scenario. For our first scenario we have considered problems related to the use of English as a lingua franca [40], [41], since this is the language usually used in GSD.

The phases of the Meeting Workflow contain information that the VC can use to detect inappropriate interventions by the learners. More specifically, a phase can contain:

- A list of expressions regarding cultural problems, which contains the appropriate and inappropriate use of titles, presentations and greetings, how to start and finish conversations, requests, means of negotiation, etc. The following example shows how we correct a learner who does not use appropriate titles:

  Cultural Problem:

  Type: qualification

  Pattern: "? Edwards" **Trigger**: "? <> Mr."

  Definition: You should refer to Mr. Edwards by using his title (Mr.).

- List of expressions regarding language problems, such as of the overuse of certain verbs of a high semantic generality (do, have, make, put, etc.) or the use of false friends. The following example shows a pattern with which to correct the use of a false friend:

  Language Problem:

  Type: false friend

  Pattern: "politic"

  Definition:"Politic" is a false friend in Spanish. Do you mean policy?

- Rules regarding grammatical inaccuracies. Third party dictionaries and grammatical correctors for the target language are used for this purpose. These engines detect any mistake during the conversation and take into account typical mistakes. For example; a common mistake in the Spanish culture consists of changing the termination of a Spanish word in the hope that it will be correct in English. The VC will use these to report errors such as: avoidance of passive forms, incorrect plural formations, the absence of the third person, the use of redundant prepositions, etc.

Each of these entries also has an associated explanation that the VC will use to help the learner. These entries also have a score that will be used to evaluate the learner's interaction. This also serves to show statistics with regard to the number of mistakes made in each category at the end of the meeting.

The composed phases store cultural and language knowledge, signifying that any phase contained in it will use this information. We therefore avoid redundant cultural and language information in different phases, since general knowledge for this context can be stored in the parent phase.

4 Description of a Scenario

The definition of a training scenario requires a great deal of knowledge about the stage of GSD being dealt with, so it is first necessary to study the existing literature on this subject to discover the specific problems that may appear in each stage in order to design a representative training scenario.

The first training scenario we have developed is related to the Requirements Elicitation (RE) stage, since it is a highly communicative process, which is particularly affected by poor communication, and cultural and language differences that frequently causes problems related to ill-defined requirements and misinterpretations that may occur during interactions with customers and team members.

Our RE scenario is focused on the problems that Spanish learners confront when interacting using English. More specifically, the Meeting Workflow designed for this scenario places learners in a virtual meeting in which they will play the role of analysts in order to elicit a set of requirements for a virtual customer from the United States.

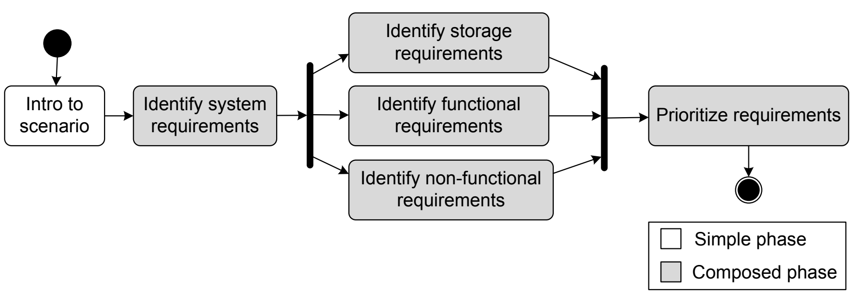


Fig. 2. Example of Requirements Elicitation Workflow

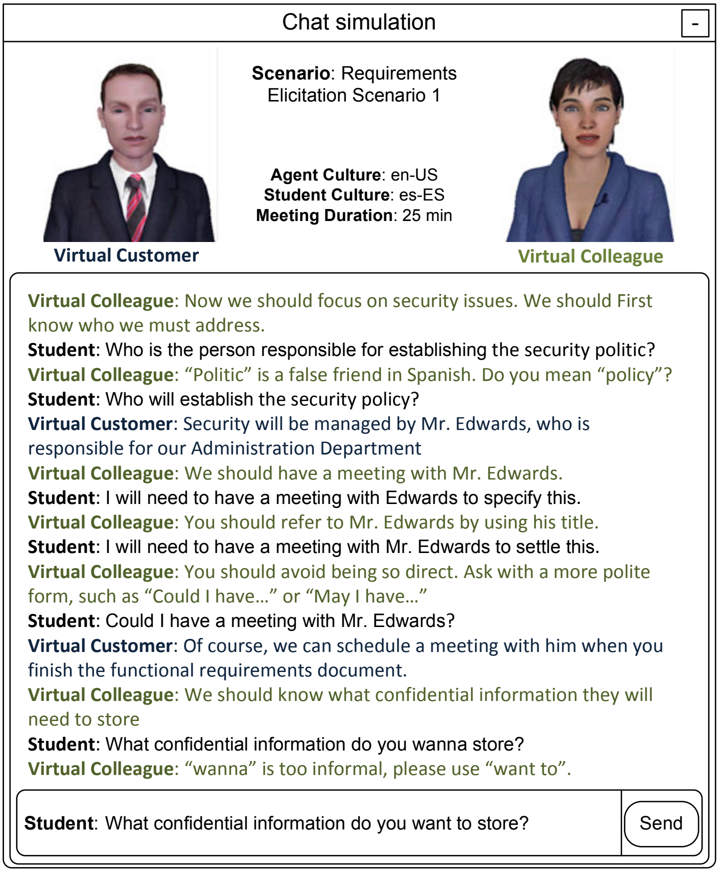


Fig. 3. Virtual meeting example

One of the Meeting Workflows proposed for this scenario is shown in Fig. 2, and starts with an introduction to the problem to be solved and to the VAs involved. This will be explained by the VC. After the introduction, the learners will actively interact with the VAs through the following phases of the meeting, which are organized by using composed phases that group the workflows and consider the type of requirements elicited (system requirements, functional requirements, non-functional requirements, and storage requirements).

Fig. 3 shows an example of a dialog between a Spanish learner and the virtual customer, in which the VC guides and corrects the learners' mistakes according to the aforementioned Meeting Workflow.

This is an exaggerated case in which the VC acts many times. However, when the learners' behavior is appropriate the VC's intervention is not necessary. The details of the Meeting Workflow phase with regard to this conversation are shown in Fig. 4, in which we have designed a composed phase (Identity storage requirements) which contains the definition of some of the cultural problems that learners may confront

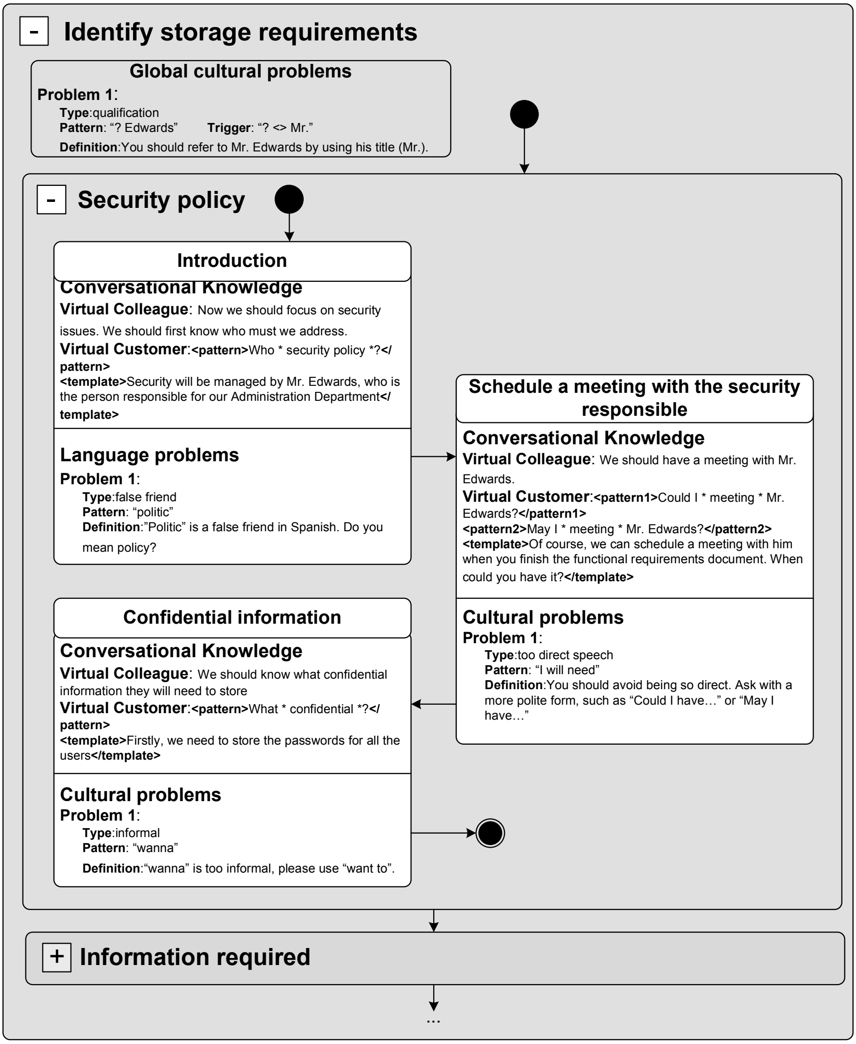


Fig. 4. Virtual scenario definition

during this phase. For example, in this case it detects that the learner is referring to Mr. Edwards inappropriately. The first sub-phase (Security policy) in turn contains a sub-workflow in which we can see simple phases containing the conversational knowledge along with language and cultural problems specific to these phases.

For each learner intervention, our system will review the text introduced, checking the patterns defined for cultural and language problems. For example, in the "Introduction" phase, if the learner uses the word "politic", the VC will correct him/her by saying: ""Politic" is a false friend in Spanish. Do you mean policy?".

If any conflict were to exist between the cultural or language knowledge of a phase and its parent, the system would give priority to the information corresponding to the child phase, since it contains more specific information for that context than the parent. The scenario concludes when the learner completes all the virtual meetings associated with the scenario, finalizes the requirements elicitation document and fills in a questionnaire.

5 Discussion

Our proposal permits communication skills in GSD problems to be trained through typical communication and coordination channels and avoids the need for coordination with other institutions, thus reducing the instructors' workload and scheduling problems. It also avoids the difficulty of finding team members from different cultures with the appropriate skills and knowledge to carry out the GSD activities.

Learners do not depend on their partners' activities and can work at any moment without depending on another learner's availability. They can also play different roles in the projects, and thus become aware of the different kinds of problems from different perspectives.

VAs will provide learners with opportunities for self-reflection and self-correction by explaining the consequences and rationales of their actions with regard to team ethics and cultural differences. Instructors will also be able to provide the learners with feedback, since they can monitor the learners' activities and communicate with them.

Since it is not possible for instructors to have a profound knowledge of all the stages and problems of GSD [16], one of our future works will aim to provide a wide set of training scenarios oriented towards other stages of GSD such as software design, software construction or software testing in which different types of documentation can be managed. Fig. 5 shows the phases that we plan to implement in a complete training scenario along with the virtual meetings that could take place and the artifacts associated with each phase.

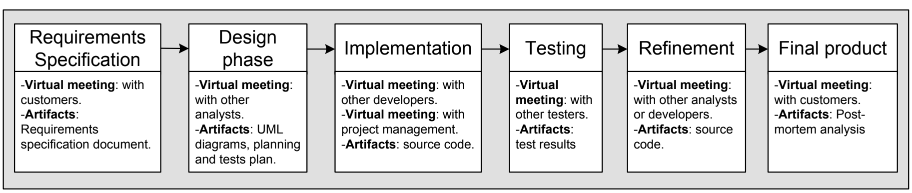


Fig. 5. Phases of a complete training scenario

Instructors would thus be provided with realistic training scenarios that reproduce the complexity of GSD environments and, since our simulator permits the customization of the existing training scenarios, the instructors could adapt them to their specific needs.

6 Conclusions and Future Work

Training the collaboration skills required in GSD environments is a difficult task. In this paper we have presented an environment that simulates the complexity of real

GSD projects by providing training scenarios that are especially focused on cultural and language differences.

Although our first training scenario is focused on the Requirements Engineering stage, we plan to develop further scenarios once we have completed the evaluation of this one. We also plan to adapt the scenarios to other pairs of languages and cultures apart from those of English (United States) and Spanish (Spain).

In order to facilitate the collaboration artifacts management, we also provide a repository, signifying that the artifacts are under version control and can be made available to the instructor. Our proposal, based on VAs, avoids the problems of other existing approaches related to coordination with other learners or institutions and minimizes the instructors' effort and the costs of infrastructure and maintenance. It is therefore easy to provide learners with a wide set of GSD problems in which they can train by using communication tools (chat, email).

Although we have presented a training scenario oriented towards training learners in collaboration through chat, our environment also permits the definition of scenarios in which the interaction can be made through emails. In addition, although our research is focused on GSD, this proposal is also extensible to outsourcing, offshoring or distributed software development education.

In our future work we intend to define training scenarios in which more than a learner can be involved. Thereby, each learner would play a different role in the scenario and they could interact during the meeting helped by the VC. The idea in this case consists of providing learners with an introduction to the problem and a set of artifacts and tasks that they can discuss during the interaction with their partner in order to collaboratively solve a GSD problem.

Finally, as part of our future work we plan to validate our final model by comparing the performance of the members of a company involved in real GSD projects trained with our model, with other members with similar skills who have not used it.

Acknowledgments. This work has been funded by the PEGASO/MAGO project (Ministerio de Ciencia e Innovación MICINN and Fondos FEDER, TIN2009-13718-C02-01). It is also supported by MEVALHE (HITO-09-126) and ENGLOBAS (PII2I09-0147-8235), funded by Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha), and co-funded by Fondos FEDER, as well as MELISA (PAC08-0142-3315), Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia in Spain.

References

1. Ågerfalk, P., Fitzgerald, B., Olsson, H.H., Conchúir, E.Ó.: Benefits of Global Software Development: The Known and Unknown. In: Wang, Q., Pfahl, D., Raffo, D.M. (eds.) ICSP 2008. LNCS, vol. 5007, pp. 1–9. Springer, Heidelberg (2008)
2. Damian, D., Hadwin, A., Al-Ani, B.: Instructional design and assessment strategies for teaching global software development: a framework. In: Proceedings of the 28th international conference on Software engineering, pp. 685–690. ACM, Shanghai (2006)
3. Damian, D., Zowghi, D.: Requirements Engineering challenges in multi-site software development organisations. Requirements Engineering 8, 149–160 (2003)

4. Lutz, B.: Linguistic Challenges in Global Software Development: Lessons Learned in an International SW Development Division. In: Fourth IEEE International Conference on Global Software Engineering (ICGSE 2009), pp. 249–253. IEEE Computer Society, Limerick (2009)
5. Prikladnicki, R., Pilatti, L.: Improving Contextual Skills in Global Software Engineering: A Corporate Training Experience. In: IEEE International Conference on Global Software Engineering (ICGSE 2008), pp. 239–243. IEEE Computer Society, Bangalore (2008)
6. Herbsleb, J.D., Moitra, D.: Global software development. IEEE Software 18, 16–20 (2001)
7. Ocker, R., Rosson, M.B., Kracaw, D., Hiltz, S.R.: Training Students to Work Effectively in Partially Distributed Teams. Trans. Comput. Educ. 9, 1–24 (2009)
8. Richardson, I., Moore, S., Paulish, D., Casey, V., Zage, D.: Globalizing Software Development in the Local Classroom. In: Proceedings of the 20th Conference on Software Engineering Education & Training, pp. 64–71. IEEE Computer Society, Los Alamitos (2007)
9. Toyoda, S., Miura, M., Kunifuji, S.: A Case Study on Project-Management Training-Support Tools for Japanese/Chinese/Indian Offshore Development Engineers. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part II. LNCS (LNAI), vol. 4693, pp. 1222–1229. Springer, Heidelberg (2007)
10. Gotel, O., Kulkarni, V., Scharff, C., Neak, L.: Working Across Borders: Overcoming Culturally-Based Technology Challenges in Student Global Software Development. In: Proceedings of the 2008 21st Conference on Software Engineering Education and Training, pp. 33–40. IEEE Computer Society, Los Alamitos (2008)
11. Richardson, I., Milewski, A.E., Mullick, N., Keil, P.: Distributed development: an education perspective on the global studio project. In: Proceedings of the 28th international conference on Software engineering, pp. 679–684. ACM, Shanghai (2006)
12. Favela, J., Peña-Mora, F.: An Experience in Collaborative Software Engineering Education. IEEE Softw. 18, 47–53 (2001)
13. Burnell, L.J., Priest, J.W., Durrett, J.R.: Teaching Distributed Multidisciplinary Software Development. IEEE Softw. 19, 86–93 (2002)
14. Braun, A., Dutoit, A.H., Harrer, A.G., Brüge, B.: iBistro: A Learning Environment for Knowledge Construction in Distributed Software Engineering Courses. In: Proceedings of the Ninth Asia-Pacific Software Engineering Conference, pp. 197–203. IEEE Computer Society, Los Alamitos (2002)
15. Adya, M., Nath, D., Malik, A., Sridhar, V.: Bringing global sourcing into the classroom: experiential learning via software development project. In: Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce, pp. 20–27. ACM, St. Louis (2007)
16. Lutz, B.: Training for Global Software Development in an International "Learning Network". In: Proceedings of the International Conference on Global Software Engineering, pp. 140–150. IEEE Computer Society, Los Alamitos (2007)
17. Lago, P., Muccini, H., Beus-Dukic, L., Crnkovic, I., Punnekkat, S., Vliet, H.V.: Towards a European Master Programme on Global Software Engineering. In: Proceedings of the 20th Conference on Software Engineering Education & Training, pp. 184–194. IEEE Computer Society, Los Alamitos (2007)
18. Bmegge, B., Dutoit, A.H., Kobylinski, R., Teubner, G.: Transatlantic project courses in a university environment. In: Proceedings of the Seventh Asia-Pacific Software Engineering Conference, pp. 30–37. IEEE Computer Society, Los Alamitos (2000)
19. Liu, X.: Collaborative global software development and education. In: Conference Collaborative global software development and education, p. 371

20. Petkovic, D., Thompson, G.D., Todtenhoefer, R.: Assessment and comparison of local and global SW engineering practices in a classroom setting. In: Proceedings of the 13th annual conference on Innovation and technology in computer science education, pp. 78–82. ACM, Madrid (2008)
21. Bellur, U.: An Academic Perspective on Globalization in the Software Industry. In: Proceedings of the 30th Annual International Computer Software and Applications Conference, vol. 1, pp. 53–54. IEEE Computer Society, Los Alamitos (2006)
22. Lago, P., Muccini, H., Babar, M.A.: Developing a Course on Designing Software in Globally Distributed Teams. In: IEEE International Conference on Global Software Engineering (ICGSE 2008), pp. 249–253. IEEE Computer Society, Bangalore (2008)
23. Petkovic, D., Thompson, G., Todtenhoefer, R.: Teaching practical software engineering and global software engineering: evaluation and comparison. SIGCSE Bull. 38, 294–298 (2006)
24. Swigger, K., Aplaslan, F.N., Lopez, V., Brazile, R., Dafoulas, G., Serce, F.C.: Structural factors that affect global software development learning team performance. In: Proceedings of the special interest group on management information system's 47th annual conference on Computer personnel research, pp. 187–196. ACM, Limerick (2009)
25. Boldyreff, C., Kyaw, P., Lavery, J., Nutter, D., Rank, S.: Towards Collaborative Learning via Shared Artefacts over the Grid. In: Conference Towards Collaborative Learning via Shared Artefacts over the Grid
26. Gotel, O., Kulkarni, V., Scharff, C., Neak, L.: Students as Partners and Students as Mentors: An Educational Model for Quality Assurance in Global Software Development, vol. 16, pp. 90–106. Springer, Heidelberg (2009)
27. Bouillon, P., Krinke, J., Lukosch, S.: Software Engineering Projects in Distant Teaching. In: Proceedings of the 18th Conference on Software Engineering Education & Training, pp. 147–154. IEEE Computer Society, Los Alamitos (2005)
28. Meneely, A., Williams, L.: On preparing students for distributed software development with a synchronous, collaborative development platform. In: Proceedings of the 40th ACM technical symposium on Computer science education, pp. 529–533. ACM, Chattanooga (2009)
29. Gaeta, M., Ritrovato, P.: Generalised Environment for Process Management in Cooperative Software Engineering. In: Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment, pp. 1049–1053. IEEE Computer Society, Los Alamitos (2002)
30. Boldyreff, C., Nutter, D., Rank, S.: Active Artefact Management for Distributed Software Engineering. In: Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment, pp. 1081–1086. IEEE Computer Society, Los Alamitos (2002)
31. Schümmer, T., Lukosch, S.: Understanding Tools and Practices for Distributed Pair Programming. Journal of Universal Computer Science 15, 3101–3125 (2009)
32. Schümmer, T., Lukosch, S., Haake, J.M.: Teaching distributed software development with the project method. In: Proceedings of the 2005 conference on computer support for collaborative learning: learning 2005: the next 10 years!, pp. 577–586. International Society of the Learning Sciences, Taipei (2005)
33. Haake, J.M., Schümmer, T., Haake, A., Bourimi, M., Landgraf, B.: Supporting Flexible Collaborative Distance Learning in the CURE Platform. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004) - Track 1, vol. 1, p. 10. IEEE Computer Society, Los Alamitos (2004)

34. Berkling, K., Geisser, M., Hildenbrand, T., Rothlauf, F.: Offshore Software Development: Transferring Research Findings into the Classroom. In: Meyer, B., Joseph, M. (eds.) SEAFOOD 2007. LNCS, vol. 4716, pp. 1–18. Springer, Heidelberg (2007)
35. Swigger, K., Brazile, R., Harrington, B., Peng, X., Alpaslan, F.: Teaching Students How to Work in Global Software Development Environments. In: Conference Teaching Students How to Work in Global Software Development Environments, pp. 1–7
36. Uresti, J.A.R., Boulay, B.d.: Expertise, Motivation and Teaching in Learning Companion Systems. Int. J. Artif. Intell. Ed. 14, 193–231 (2004)
37. Chou, C.-Y., Chan, T.-W., Lin, C.-J.: Redefining the learning companion: the past, present, and future of educational agents. Comput. Educ. 40, 255–269 (2003)
38. Hall, E.T.: Beyond Culture. Anchor Press (1976)
39. Hofstede, G., Hofstede, G.J.: Cultures and organizations: software of the mind, New York, NY, USA (2005)
40. Grzega, J.: Reflection on Concepts of English for Europe: British English, American English, Euro-English, Global English. Journal for EuroLinguistiX 2, 44–64 (2005)
41. Axtell, R.E.: Do's and Taboos of Using English Around the World. Wiley, Canada (1995)

Awareness Support in Global Software Development: A Systematic Review Based on the 3C Collaboration Model

Igor Steinmacher[1], Ana Paula Chaves[2], and Marco Aurelio Gerosa[3]

[1] Coordination of Informatics - Federal University of Technology - Parana,
BR 369 - km 0,5 - CEP: 87301-006 - Caixa Postal: 271, Campo Mourao-PR - Brazil
igorfs@utfpr.edu.br

[2] System Analysis and Development - Faculdade Integrado de Campo Mourao
Avenida Irmaos Pereira, 670 - CEP:87301-010 - Campo Mourao-PR - Brazil
anachaves@grupointegrado.br

[3] University of Sao Paulo (USP)
Rua do Matao, 1010 - CEP 05508-090 - Sao Paulo - SP - Brazil
gerosa@ime.usp.br

Abstract. The developers' physical distribution in Global Software Development (GSD) imposes challenges related to awareness support during collaboration. In this paper, we present a systematic review of the literature that describes studies that improve awareness support in a GSD scenario, identifying which of the dimensions of the 3C model, namely communication, coordination, and cooperation, are supported by these studies. Results indicate that coordination is far the most explored dimension, while awareness support in communication is very poorly studied. The research also identified a high number of tools introduced in the GSD domain and some new research opportunities.

Keywords: Awareness, Global Software Development, 3C Collaboration Model, Communication, Coordination, Cooperation.

1 Introduction

The Software Development industry has been using the benefits brought by CSCW in order to obtain competitive advantages in terms of cost and quality using qualified professionals distributed from all around the world [50]. This new approach, called Global Software Development (GSD), is based on geographically dispersed teams working collaboratively in a software project. Besides its advantages, GSD brings new challenges such as contextual, cultural, organizational, geographical, temporal, and political differences [35]. With the increasing number of organizations adopting GSD, researches and related literature also increased [49,38]. Within these researches, there is a great number of studies related to awareness support in distributed development environments. This occurs because awareness is essential when teams are distributed and there is a need to collaborate in order to achieve a common goal.

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 185–201, 2010.

In this paper, we report a systematic literature review on awareness support within the GSD scenario. Its purpose was to identify awareness studies that brought improvements to collaboration in GSD. For the purpose of analysis the improvements were classified into the three dimensions of the 3C Collaboration Model [27]. According to this model, commonly used in the CSCW literature, the collaboration is analyzed from the communication, coordination, and cooperation points of view. The systematic review also identified aspects upon which researchers have focused more intently, thus allowing analysis and identification of current challenges and opportunities for future works.

This paper is organized as following: in Section 2 we present the concepts of awareness and the 3C model; in Section 3 we present the systematic review, including its planning, conduction, and analysis; in Section 4 we classify and summarize the improvements and the opportunities identified on this review; and, in Section 5, we discuss our findings and the limitations of our review.

2 Awareness and the 3C Collaboration Model

Awareness was defined by Dourish and Belloti [18] as “an understanding of the activities of others, which provides a context for one’s own activities.” Its objective is to allow a group of people working collaboratively to realize how and which of their contributions are relevant to the group activities. Awareness is concerned to support activities that involve two or more individuals, resources or services, voluntarily or involuntarily involved in any collaborative activity.

In GSD environments, collaborative teams are geographically dispersed. Thus, physical, temporal, and cultural distances make the difficulty of providing awareness more evident. The participants of a collaborative work often do not know other participants in person, work in different timezones, do not speak the same language and do not share the same culture. These, among other factors, hinder the information sharing [10], increase the possibility of conflicts [53], and inhibits informal interaction [7] among team members.

The problems addressed by GSD affect communication, coordination, and cooperation among team members. This occurs due to the relationship between these elements and awareness. This relationship is reflected in the 3C collaboration model, which was originally proposed by Ellis et al.[21] and later extended by Fuks et al.[26]. This model defines collaboration as the union of communication, coordination, and cooperation efforts, as represented by Figure 1.

Communication generates commitments that are managed by coordination. Moreover, during communication people negotiate and make decisions. Coordination arranges task for cooperation, helps managing conflicts, and organizes people to prevent loss of communication and of cooperation efforts. Cooperation is the joint operation of members of the group in a shared space, seeking to execute tasks, and generate and manipulate cooperation objects. To obtain success, cooperation demands more communication, generating a cycle that indicates the iterative nature of collaboration. Awareness is the element that intermediates each of the 3Cs, offering feedback to users actions and giving them information about other participants of a collaborative work [27,26].

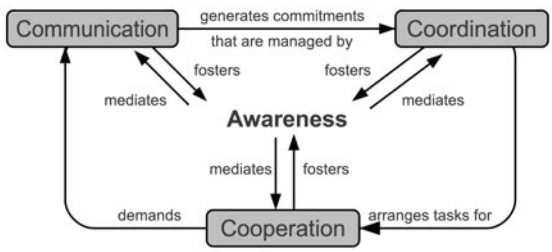


Fig. 1. 3C collaboration model proposed by [21] and adapted by [26]

The relationship among the 3Cs may be used as a guidance to analyze a groupware application domain. A chat, for example, which is a communication tool, requires communication (exchange of messages), coordination (access policies), and cooperation (logging and sharing). Therefore, despite their separation for analysis, there is a constant interplay between them. Figure 2 presents some applications positioned in the triangle formed by the 3 dimensions.

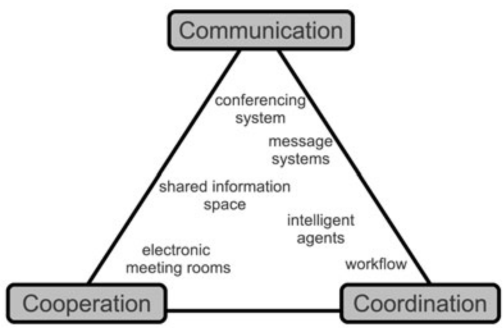


Fig. 2. Applications spread in the triangle formed by the 3C collaboration model [5]

The 3C collaboration model has often been used in the literature to classify collaborative tools [5,45,44]. Organizing collaborative tools according to this model facilitates the analysis since it allows one to realize the problems related to each dimension separately, to compose a complete solution [27]. Moreover, as they are interrelated concepts, dealing with them separately may reveal factors that otherwise would be forgotten.

In this paper, studies that improve awareness support in GSD were analyzed and categorized according to the 3C dimension they mainly support. We adopted the following criteria to classify the studies:

Communication: when the study brings 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;

Coordination: when the study brings improvements to the support offered for people managing themselves, or being aware of the activities and its effects to the collaboration;

Cooperation: when the study brings improvements to the shared space or to the way users interact with shared artifacts synchronous or asynchronously.

3 Systematic Review

Kitchenham [40] summarizes the stages in a systematic review into three main phases: Planning the Review, Conducting the Review, and Reporting the Review. In this section, we present these three steps for our systematic review.

3.1 Review Planning

Review planning includes the identification of the review objective and the development of a protocol. The definition of a review protocol specifies the methods that will be used to undertake a systematic review and aims to reduce the possibility researcher bias [40]. This section summarizes our review protocol.

Formulating the research questions is the most important activity during protocol definition [40]. The research questions guide the systematic review. In our systematic review the research questions were:

Q1: What are the awareness studies carried out in order to improve the Global Software Development scenario?

Q2: Which of the 3Cs are these studies supporting?

The keywords were defined based on terms related to GSD and to awareness, as presented in Table 1.

Table 1. Keywords defined based on research questions

Reference	Category	Keywords
C1	Global 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

Category C1 has more keywords and reflects the fact that GSD area is maturing, and there are many variations of the same term [49]. The three dimensions of the 3C collaboration model (communication, coordination, and cooperation) were not included in the query string, because there are studies related to awareness which might not explicitly present one of these words, however they can be classified according to them. The query string was defined as a combination of C1 and C2 using the logical connectors "AND" and "OR", as presented below:

(Awareness) AND ("Distributed software development" OR "Global software development" OR "Collaborative software development" OR "Global software engineering" OR "Globally distributed work" OR "Collaborative software engineering" OR "Distributed development" OR "Distributed teams" OR "Global software teams" OR "Globally distributed development" OR "Geographically distributed software development" OR "Offshore software development" OR "Dispersed teams" OR "virtual teams")

The query string defined was used to retrieve the candidate studies. The following search sources were used to obtain them:

- Science@Direct (http://www.sciencedirect.com);
- El Compendex (http://www.engineeringvillage.com);
- IEEE Digital Library (http://ieeexplore.ieee.org/); and
- ACM Digital Library (http://portal.acm.org).

After obtaining the studies by running the query string on the selected sources, papers were analyzed to check their relevance to this systematic review. The analysis was made in order to check if the study dealt with awareness on the GSD domain. It is worth noticing that only studies written in English and with online full paper available were considered.

The process used to include or exclude a study was based on [40,49] and followed the following steps. The first three steps were performed by two researchers, independently. When at least one of them included a paper as relevant, it was classified as a relevant study. All the steps were reviewed by a third – more experienced – researcher, responsible for checking the information generated.

1. The first analysis was made by reading papers titles, excluding those that were considered clearly irrelevant to the research questions.
2. The included studies were then analyzed based on the reading of papers abstracts and keywords, considering research questions.
3. Studies included in the previous step were further analyzed based on the reading of introduction, conclusion, and specific parts related to the contributions.
4. All studies selected so far were read by the researchers and documented on a proper form. Those studies which, despite addressing awareness issues, did not focus on GSD domain, were dismissed. We also discarded studies related to a same tool or environment, keeping just the most recent one. Papers included after this step were considered our primary studies.

The process of information extraction was based on obtaining information concerning the main contribution of the studies, thus allowing a categorization of the results. All papers were categorized based on the classification used in [38]. The categories used in our review were:

(i) case studies;
(ii) theoretical studies (also including conceptual/theoretical frameworks);

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

Additionally, studies were categorized according to which of the 3C dimensions the study was supporting. This categorization was made by identifying the dimensions supported and evaluating them from 0 to 3, according to the level of support the study 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 (no focus, but brings indirect improvement);
0: Does not support (when no support or improvement is presented).

3.2 Review Conduction

The review was conducted according to the plan presented on the previous section. After executing the process defined in Section 3.1, a total of 42 primary studies were selected. More details about the process, the studies included and their classification can be found at http://www.igor.pro.br/awarenessRS/.

As can be observed in Table 2, confirming the observations of [49], the lack of standard terminology in GSD resulted in a large number of papers to start with, but only a few were selected. A high number of papers of unrelated areas (like computer networks, ubiquitous computing, e-learning, and psychology) also contributed to the large number of studies discarded at the beginning. After the title analysis, 143 papers (including duplicated ones) were selected; during the second analysis (based on abstract and keywords), 38 papers were dismissed; then, after the third selection (introduction and conclusion reading), we dismissed other 26 studies. During first and second analysis, we have also discarded 22 duplicated papers. Thus, we came to a number of 57 studies selected for an in-depth analysis.

In the deeper analysis, performed by reading the full papers, 15 papers were considered not relevant to the review or presented a same tool or environment (in this case only the latest study was considered). The main reason for dismissing papers at this step was that they were not presenting their contributions to the GSD domain area. In the next section, we present the results based on the data collected from these 42 papers selected as primary studies.

Table 2. Distribution of studies found

Source	Papers Found	Excluded per analysis step				Relevant Studies	Primary Studies Selected
		Title	Abstract, Keywords	Introduction, Conclusion	Repeated, Duplicated		
IEEE	37	11	4	6	2	14	**10**
ACM	325	256	20	9	5	35	**26**
Science Direct	229	203	9	10	2	5	**3**
El Compendex	86	64	5	1	13	3	**3**
Total	677	534	38	26	22	57	**42**

3.3 Review Report

Figure 3 shows the number of relevant studies by year, including the studies related to a same tool or environment to avoid any bias. The first conclusion is that the subject of awareness in GSD evidently is an area which was not widely studied until a few years ago, and that only recently appeared in a greater number of publications. As one can notice, the last three years present the greatest part of papers of our sample. It is important to highlight that the search was performed in early 2010, so it is possible that more studies could have been published in 2009 that have not been indexed to date.

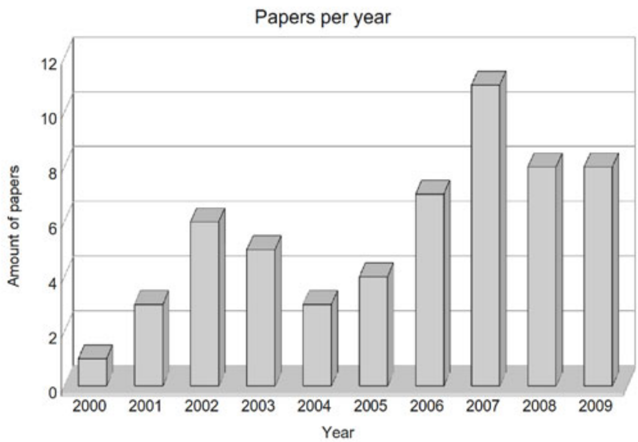


Fig. 3. Amount of relevant studies per year

Figure 4 presents the distribution of studies according to their categories. All studies were classified in, at least, one category. In this figure, the large number of tools presented in the literature may be noted. Out of a total of 42 primary studies, 33 studies (79%) presented a new tool. It is worth pointing out that 21 studies (64%) *only* presented a tool without any experimental analysis. Special attention should be given to one study [47] that presented a *tool* based on a proposed *theoretical study* and was further evaluated by an *experiment*.

The other 9 papers (not classified as tools) were categorized as case studies and/or theoretical studies. Five of them were classified only as theoretical studies: three [19,55,8] presented conceptual frameworks for awareness support; one study presented computer support interaction patterns for dispersed team members [54]; and the other one [31] brought an awareness analysis for Open Source communities.

Three papers were classified only as case studies: one [2] studied people work rhythm within a company, in order to find ways to make people aware of remote colleagues availability, providing a shared sense of time; and the other two [14,10] studied consequences of awareness gaps in broken code builds and communication, respectively. Only one study [9] was classified as both theoretical

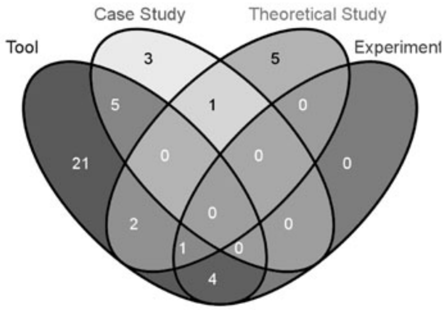


Fig. 4. Venn diagram for types of primary studies found

and case study. And, as one can notice, for this review no study was classified as literature review.

Figure 5 shows the classification of primary studies regarding the 3C collaboration model in two different analyses. The first analysis, depicted by Figure 5(a), is based on a Venn diagram representing the number of awareness studies that presented support (at any level) to each 3C-dimension. So, a first and clear conclusion that can be made is that communication is scarcely studied, presenting just 9 related studies and just 2 focusing *only* on communication. We can also see that a great focus is given to coordination and cooperation, as 40 out of 42 studies (95%) presented some support to one of these dimensions, and 21 (50%) support both dimensions concurrently.

Figure 5(b) presents the distribution of studies according to the 3C model and to the level of awareness support provided. It is clear that coordination is by far the main focus of awareness studies on GSD domain due to the number of studies that mainly support it (evaluated with 3 according to the scale presented in Section 2): a total of 28 studies out of 42 (67%). When we verify studies that support coordination evaluated with 3 or 2, this number grows to 35 (83%).

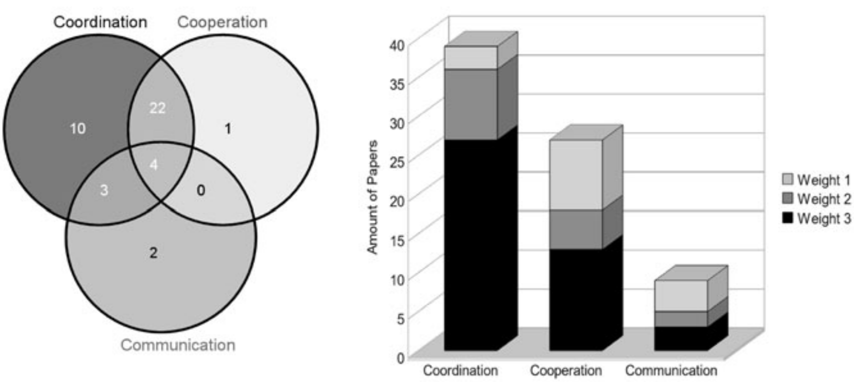


Fig. 5. (a) Venn diagram for 3C model classification; (b) Amount of studies evaluated according to 3C model

On the other hand we have communication, mainly supported by only 4 studies (9%). Additional details about the studies classified according to the 3C model dimensions can be found in the next section.

4 Studies Discussion and Classification

This section discusses the improvements and opportunities identified in this review. The open opportunities presented were raised according to the authors experience and based on the issues that frequently motivate GSD studies.

4.1 Communication

As it is possible to observe in Figure 5, awareness supporting communication is poorly explored within the GSD domain. Although many researchers use communication issues as motivation, only four awareness initiatives were classified as mainly supporting communication.

Three of these studies focused on providing the users with some kind of context for the conversation. Cheng et al. [11] presented a tool called Jazz, a collaborative distributed development environment that included communication facilities, such as a chat tool that allows developers to include links to transcripts of older chats, and team event notifications (e.g. code check-ins and check-outs from source control). Jacovi et al.[36] presented a tool that allows people to know what are the subjects being discussed on chats. Fitzpatrick [25] introduced a tickertape tool responsible for bringing CVS commit messages to members of a project, allowing them to start a private or a group chat within the context of the CVS message.

In the study presented by Calefato et al. [7] the Jazz environment was improved by presenting its integration with FriendFeed (a social network system), bringing personal interests to workspace and offering informal and social communication by using microblogs and forums within their development environment.

Some studied papers [29,37,55,32] presented limited means to allow communication on GSD environments (e.g. chats and forums) without any specific contribution to provide an easier or more effective communication in a GSD scenario.

According to [38], the software life cycle requires a great deal of communication using different tools and formats in order to avoid misunderstandings and delays. In order to avoid these problems and improve communication, awareness initiatives are needed to avoid ambiguity and misunderstandings, as cultural differences imply different vocabulary which may lead to mistakes. Using contextualized information and semantics extraction to improve communication can also be fruitful research areas. Privacy and security issues regarding access to sensitive user data [30] during communication is also a topic that should appear on awareness studies.

4.2 Coordination

Within a GSD environment, awareness is regarded as a means by which team members can become aware of the work of others that is interdependent with their current tasks, therefore enabling better coordination of teams [14]. This maybe is the justification for why 38 out of 42 (90%) awareness studies present some level of support for coordination and 28 (67%) mainly support it.

In order to provide awareness for coordination, some studies focused on presenting information regarding the sequence of activities of an ongoing project. One example is the tool called TeamSCOPE [37], that presents a calendar of activities and a log relating artifacts and activities, allowing people to coordinate themselves. Godart et al. [29] introduces the tool ToxicFarm, which offers a workflow view, allowing one to be aware of the activities, their owners and their dependencies. Biuk-Aghai [4] presents a visualization approach that aims to support users in obtaining a greater understanding of structural and behavioral aspects of virtual collaboration, leading to increased awareness of the activities of the virtual team.

Another approach used to provide awareness to support coordination is gathering information on source code version management repository to make users aware of changes on artifacts that affect their work. Cook et al. [12] presents CAISE, a tool that notifies developers regarding dependent codes, user dependence and impacts of code changes, based on commits. FASTDash [3] is a visualization tool that seeks to improve activity awareness using a representation of the shared code (extracted from SVN/CVS) that highlights team members' current activities. Many other studies revealed that code repositories are a rich source of information for awareness generation [51,48,11,15,13,47].

Also using information from source code version management repository, but studying social network analysis techniques, De Souza et al.[16] introduces Ariadne, a tool that extracts information from code repositories and analyzes sociotechnical dependencies, thus helping to find coordination problems using social network visualization. Tesseract [52] and SmallBlue [20] are other tools that also present sociotechnical network analysis to improve awareness.

Expertise search is another mechanism studied to provide awareness to support coordination. Expertise Browser [43] is a tool to assist users in identifying experts for specific artifacts or tasks, making them aware of how experienced they are and the amount of experts for that artifact or task. SmallBlue [20] is an expertise search tool that can be used to identify experts, see dynamic profile information, and get information about the social distance to the experts; it supports someone to find the right people to work for a given task or area of interest. Minto [42] presents Emergent Expertise Locator, that uses emergent team information extracted from source code repositories to propose experts.

Cataldo et al. [8] study the coordination and propose a framework based on product features in order to support coordination within distributed environments, providing information about members activities and their relations with product features.

Most part of the studies that support coordination presented features based on historical information extracted from source code repositories, but, based on [47] it is also necessary to get recent information, once key information items used to gain awareness are the items that change on a daily, hourly or minute-by-minute basis, according to [3]. Another open research topic is the social network coordination [14], trying to maintain awareness on the emerging and unplanned interactions that appear during the development cycle.

4.3 Cooperation

Activities within a GSD environment require awareness information to help distributed developers to edit shared artifacts, reducing negative impacts of distribution. In this sense, two studies [34,33] use code annotation to present changes being concurrently made by other developers in a shared artifact. Dekel and Herbsleb [17] also use code annotation to provide awareness on a tool called eMoose, that allows developers to write informal comments in the code, stores them in a central database, and spreads them to other developers using the annotated module.

A well explored way to provide awareness to support cooperation is warning and preventing conflicts on shared (cooperative) artifacts. Lighthouse [13] 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. Estublier and Garcia [22] present a study based on cooperative policies to control concurrent engineering in order to avoid conflicts and propose awareness support considering different concurrent models. Palantìr [53] is another tool that supports cooperation by making users aware of direct or indirect conflicts on source code and helping them to reach a solution. Holmes and Walker [33] and Ignat [34] also proposed studies to avoid conflicts during cooperative handling of code artifacts.

A different awareness study is presented by Everitt et al. [23], who propose Designers' Outpost, a tool that allows users writing on Post-it notes and adding them to an electronic whiteboard, and organize information by physically moving Post-its around on the board. The tool provides synchronous remote cooperation and supports awareness regarding changes on the electronic whiteboard.

Despite the amount of studies presenting support to cooperation, it is an area that requires further development. One example is that there is no study presenting ways to suggest pieces of code to complete a given function or method based on similar codes extracted from other developers' code. Another possible opportunity is to focus on providing awareness support to cooperation in development phases other than coding; for example, present support to clients and analysts cooperate during requirement extraction and specification phases.

4.4 Summary

This subsection summarizes and classifies the awareness features of the studied tools. This classification was made to (i) further organize the high number of tools found on this review and (ii) provide a quick reference to GSD environment developers and researchers regarding which awareness features have already been investigated. Table 3 presents the references for the studies classified according to which 3C model dimensions they support. The table does not include frameworks or architectures [39,41,46,24,6].

Table 3. Identified awareness support for each 3C dimension

	Coordination	**Cooperation**	**Communication**
Conflict indication	[3,13,22,53,48,51,34,12]	[22,34]	
Artifact change indication	[29,37,3,11,13,15,48,33]	[29,15,22,53,34]	
Activity control (workflow, logs, agenda, worklist)	[29,37,28,25,4,51]		
Presence/status indication	[29,37,56,11,32,28,23]		
Context/subject-aware message exchange			[37,11,36,28,25]
Historical log		[37,47,22]	
Historic based expert search/recommendation	[47,42,43,20]		
Social/socio-technical network	[16,52,51,20]	[47,16]	
Source code annotation		[17,33,34]	
Collaborative artifact synchronous handling		[13,28,23]	
Screen Sharing	[56,11]	[11,48]	
Informal/social communication			[7]

In addition to the analysis of opportunities made on each subsection of this section and the ones summarized in Table 3, it is possible to highlight other possible research topics that may be explored on all 3C dimensions. Firstly, we have not found studies dealing with awareness to overcome issues related to cultural, political, geographical differences, although this is frequently presented as motivation on GSD studies. Another research opportunity is the definition of policies to provide awareness within a GSD environment, maintaining the privacy of team members and organization as well as information security. Another opportunity that was very poorly explored on all three dimensions is to provide awareness not only to the coding phase, but also to the other software engineering phases. Several concepts and techniques already well investigated by the CSCW community in other contexts may be adapted or extended to the GSD domain.

5 Conclusion

In this paper, we presented a systematic review on the use of awareness to support GSD projects. It can be considered a starting point to establish issues upon which subsequent researches may be focused on or upon which developers may consider while designing GSD environments. 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 real effects that individual smaller studies are unable to detect [40].

We have presented our findings in two phases: in the first phase an initial quantitative data was presented, including number of studies per year, types of primary studies, and their classification according to the 3C model; in the second stage, we have analyzed and discussed the data extracted from the primary studies, enabling us to show some conclusions about the current state of art and practice of the awareness support in the GSD domain and the contributions and challenges identified. In general, most part of the primary studies (79%) focuses on introducing a new tool with some awareness support to GSD. The main focus is given to studies gathering information from source code version management repositories, used to provide awareness, supporting both coordination and cooperation.

During this review we found a lack of studies and tools offering solutions that could provide awareness regarding recent (or real time) context. We did not find any study linking awareness in GSD and ubiquitous computing. Merging these areas should be a promising research topic, since awareness is already being discussed in ubiquitous computing for some time [1]. We also did not find any clue on how to use awareness regarding the physical location of a team member, for example, how to treat different cultures, national laws or organization restrictions. The closer study we found was a case study presented by Begole et al. [2], aiming on finding ways to coordinate teams in different timezones according to their temporal rhythmic activities patterns on a day-by-day and weekly basis.

In terms of the classification according to the 3C collaboration model, the main conclusion is that most of the literature focus on the support for coordination and the support for communication is very poorly explored, being a fruitful research topic. Coordination, even appearing as the main focus of awareness within the GSD area, still presents some opened opportunities like those presented in [47] and [14] and summarized in Section 4.2.

The approach of conducting a systematic review based on a collaboration model widely used by CSCW community may also be considered itself as a contribution, since it has not been yet used and may be adapted by others groupware researchers while conducting systematic reviews for other CSCW domains.

5.1 Limitations

Systematic review is a powerful method to search for primary studies within a given domain [40]. But as any other method, it also presents some limitations. This review may have missed some papers that address the use of awareness to support GSD, since we did not perform our search into every possible source. The four most relevant digital libraries were selected based on previous studies [38,49] and on the subject under review. The findings of this review may also have been affected as the classification is a human process and it is based on some criteria that could be considered subjective. In order to reduce this possible threat, 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.

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. 2(4), 263–277 (2007)
2. Begole, J., Tang, J.C., Smith, R.B., Yankelovich, N.: Work rhythms: analyzing visualizations of awareness histories of distributed groups. In: ACM Conf. on Computer-Supported Cooperative Work, pp. 334–343. ACM, New York (2002)

3. Biehl, J.T., Czerwinski, M., Smith, G., Robertson, G.G.: Fastdash: a visual dashboard for fostering awareness in software teams. In: SIGCHI Conf. on Human Factors in Computing Systems, pp. 1313–1322. ACM, New York (2007)
4. Biuk-Aghai, R.P.: Visualizing structural and behavioural aspects of virtual collaboration. In: 10th IEEE International Workshops on Enabling Technologies, pp. 279–284. IEEE CS, Washington (2001)
5. Borghoff, U.M., Schlichter, J.H.: Computer-Supported Cooperative Work: Introduction to Distributed Applications. Springer, New York (2000)
6. Bouthier, C., Bitcheva, J., Canals, G.: A context-adaptive group awareness tool. In: Symposium on Applications and the Internet Workshops, pp. 238–241. IEEE CS, Washington (2005)
7. Calefato, F., Gendarmi, D., Lanubile, F.: Embedding social networking information into jazz to foster group awareness within distributed teams. In: 2nd International Workshop on Social Software Engineering and Applications, pp. 23–28. ACM, New York (2009)
8. Cataldo, M., Herbsleb, J.D.: End-to-end features as meta-entities for enabling coordination in geographically distributed software development. In: 2nd Workshop on Software Development Governance, pp. 21–26. IEEE CS, Washington (2009)
9. Cataldo, M., Wagstrom, P.A., Herbsleb, J.D., Carley, K.M.: Identification of coordination requirements: implications for the design of collaboration and awareness tools. In: 20th anniversary ACM Conf. on Computer-Supported Cooperative Work, pp. 353–362. ACM, New York (2006)
10. Chang, K.T., Ehrlich, K.: Out of sight but not out of mind?: Informal networks, communication and media use in global software teams. In: Conf. of the Center for Advanced Studies on Collaborative Research, pp. 86–97. ACM, New York (2007)
11. Cheng, L.T., Hupfer, S., Ross, S., Patterson, J.: Jazzing up eclipse with collaborative tools. In: OOPSLA Workshop on Eclipse Technology eXchange, pp. 45–49. ACM, New York (2003)
12. Cook, C., Churcher, N., Irwin, W.: Towards synchronous collaborative software engineering. In: 11th Asia-Pacific Software Engineering Conf, pp. 230–239. IEEE CS, Washington (2004)
13. da Silva, I.A., Chen, P.H., van der Westhuizen, C., Ripley, R.M., van der Hoek, A.: Lighthouse: coordination through emerging design. In: OOPSLA Workshop on Eclipse Technology eXchange, pp. 11–15. ACM, New York (2006)
14. Damian, D., Izquierdo, L., Singer, J., Kwan, I.: Awareness in the wild: Why communication breakdowns occur. In: 2nd IEEE International Global Software Engineering, pp. 81–90. IEEE CS, Washington (2007)
15. de Lucia, A., Fasano, F., Oliveto, R., Santonicola, D.: Improving context awareness in subversion through fine-grained versioning of java code. In: 9th International Workshop on Principles of Software Evolution, pp. 110–113. ACM, New York (2007)
16. de Souza, C.R., Quirk, S., Trainer, E., Redmiles, D.F.: Supporting collaborative software development through the visualization of socio-technical dependencies. In: ACM Conf. on Supporting Group Work, pp. 147–156. ACM, New York (2007)
17. Dekel, U., Herbsleb, J.: Pushing relevant artifact annotations in collaborative software development. In: ACM Conf. on Computer-Supported Cooperative Work, pp. 1–4. ACM, New York (2008)

18. Dourish, P., Belloti, V.: Awareness and coordination in shared workspaces. In: ACM Conf. on Computer-Supported Cooperative Work, pp. 107–114. ACM, New York (1992)
19. Dustdar, S., Gall, H.: Process awareness for distributed software development in virtual teams. In: 28th Euromicro Conf., pp. 244–250. IEEE CS, Washington (2002)
20. Ehrlich, K., Lin, C.Y., Griffiths-Fisher, V.: Searching for experts in the enterprise: combining text and social network analysis. In: ACM Conf. on Supporting Group Work, pp. 117–126. ACM, New York (2007)
21. Ellis, C., Gibbs, S., Rein, G.: Groupware: some issues and experiences. Commun. ACM 34(1), 30–58 (1991)
22. Estublier, J., Garcia, S.: Process model and awareness in SCM. In: 12th International Workshop on Software Configuration Management, pp. 59–74. ACM, New York (2005)
23. Everitt, K.M., Klemmer, S.R., Lee, R., Landay, J.A.: Two worlds apart: bridging the gap between physical and virtual media for distributed design collaboration. In: SIGCHI Conf. on Human Factors in Computing Systems, pp. 553–560. ACM, New York (2003)
24. Farshcian, B.A.: Integrating geographically distributed development teams through increased product awareness. Inf. Syst. 26(3), 123–141 (2001)
25. Fitzpatrick, G., Marshall, P., Phillips, A.: CVS integration with notification and chat: lightweight software team collaboration. In: 20th anniversary ACM Conf. on Computer-Supported Cooperative Work, pp. 49–58. ACM, New York (2006)
26. Fuks, H., Raposo, A., Gerosa, M., Pimentel, M., Lucena, C.: The 3c collaboration model. In: Kock, N. (ed.) The enciclopedia of e-Collaboration, pp. 637–644. Texas A&M International University, USA (2007)
27. Fuks, H., Raposo, A., Gerosa, M., Lucena, C.: Applying the 3C model to Groupware Development. Int. J. Cooperative Inf. Syst. 14(2-3), 299–328 (2005)
28. Geyer, W., Richter, H., Fuchs, L., Frauenhofer, T., Daijavad, S., Poltrock, S.: A team collaboration space supporting capture and access of virtual meetings. In: ACM Conf. on Supporting Group Work, pp. 188–196. ACM, New York (2001)
29. Godart, C., Molli, P., Oster, G., Perrin, O., Skaf-Molli, H., Ray, P., Rabhi, F.: The toxicfarm integrated cooperation framework for virtual teams. Distrib. Parallel Databases 15(1), 67–88 (2004)
30. Godefroid, P., Herbsleb, J.D., Jagadeesany, L.J., Li, D.: Ensuring privacy in presence awareness: an automated verification approach. In: ACM Conf. on Computer-Supported Cooperative Work, pp. 59–68. ACM, New York (2000)
31. Gutwin, C., Penner, R., Schneider, K.: Group awareness in distributed software development. In: ACM Conf. on Computer-Supported Cooperative Work, pp. 72–81. ACM, New York (2004)
32. Hansen, K.M., Damm, C.H.: Instant collaboration: using context-aware instant messaging for session management in distributed collaboration tools. In: 2nd NordCHI, pp. 279–282. ACM, New York (2002)
33. Holmes, R., Walker, R.J.: Promoting developer-specific awareness. In: International Workshop on Cooperative and Human Aspects of Software Engineering, pp. 61–64. ACM, New York (2008)
34. Ignat, C.L.: Annotation of concurrent changes in collaborative software development. In: 4th International Conf. on Intelligent Computer Communication and Processing, pp. 137–144. IEEE CS, Washington (2008)

35. Ivcek, M., Galinac, T.: Aspects of quality assurance in global software development organization. In: 31st International Convention MIPRO, pp. 150–155 (2008)
36. Jacovi, M., Soroka, V., Ur, S.: Why do we ReachOut?: functions of a semi-persistent peer support tool. In: ACM Conf. on Supporting Group Work, pp. 161–169. ACM, New York (2003)
37. Jang, C.Y., Steinfield, C., Pfaff, B.: Virtual team awareness and groupware support: An evaluation of the teamscope system. Int. J. Hum.–Comput. Stud. 56(1), 109–126 (2002)
38. Jimenez, M., Piattini, M., Vizcaño, A.: Challenges and improvements in distributed software development: A systematic review. Adv. Softw. Eng., 1–14 (2009)
39. Kirsch-Pinheiro, M., de Lima, J.V., Borges, M.R.S.: A framework for awareness support in groupware systems. Comput. Ind. 52(1), 47–57 (2003)
40. Kitchenham, B.: Procedures for performing systematic reviews. Technical Report TR/SE-0401, Keele University and NICTA (2004)
41. Mangan, M., Borges, M., Werner, C.: A middleware to increase awareness in distributed software development workspaces. In: 10th Brazilian Symposium on Multimedia and the Web 2nd Latin American Web Congress, pp. 62–64. IEEE CS, Washington (2004)
42. Minto, S., Murphy, G.C.: Recommending emergent teams. In: 4th International Workshop on Mining Software Repositories, p. 5. IEEE CS, Washington (2007)
43. Mockus, A., Herbsleb, J.D.: Expertise browser: a quantitative approach to identifying expertise. In: 24th International Conf. on Software Engineering, pp. 503–512. ACM, New York (2002)
44. Muhammad, A., Enríquez, A.M.M., Decouchant, D.: Awareness and coordination for web cooperative authoring. In: Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A. (eds.) AWIC 2005. LNCS (LNAI), vol. 3528, pp. 327–333. Springer, Heidelberg (2005)
45. Neale, D.C., Carroll, J.M., Rosson, M.B.: Evaluating computer-supported cooperative work: models and frameworks. In: ACM Conf. on Computer-Supported Cooperative Work, pp. 112–121. ACM, New York (2004)
46. Nutter, D., Boldyreff, C.: Historical awareness support and its evaluation in collaborative software engineering. In: 12th IEEE International Workshop on Enabling Technologies, pp. 171–176. IEEE CS, Washington (2003)
47. Omoronyia, I., Ferguson, J., Roper, M., Wood, M.: Using developer activity data to enhance awareness during collaborative software development. Comput. Supported Coop. Work 18(5-6), 509–558 (2009)
48. O'Reilly, C., Bustard, D., Morrow, P.: The war room command console: shared visualizations for inclusive team coordination. In: ACM Symposium on Software Visualization, pp. 57–65. ACM, New York (2005)
49. Prikladnicki, R., Damian, D., Audy, J.L.N.: Patterns of evolution in the practice of distributed software development in wholly owned subsidiaries: A preliminary capability model. In: IEEE International Conf. on Global Software Engineering, pp. 99–108. IEEE CS, Washington (2008)
50. Robinson, M., Kalakota, R.: Offshore Outsourcing: Business Models, ROI and Best Practices. Mivar Press, Georgia (2004)
51. Sarma, A., van der Hoek, A.: Towards awareness in the large. In: IEEE International Conf. on Global Software Engineering, pp. 127–131. IEEE CS, Washington (2006)

52. Sarma, A., Maccherone, L., Wagstrom, P., Herbsleb, J.: Tesseract: Interactive visual exploration of socio-technical relationships in software development. In: 31st International Conf. on Software Engineering, pp. 23–33. IEEE CS, Washington (2009)
53. Sarma, A., Redmiles, D., van der Hoek, A.: Empirical evidence of the benefits of workspace awareness in software configuration management. In: 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 113–123. ACM, New York (2008)
54. Schümmer, T., Lukosch, S.: The absent participant: more patterns for group awareness. In: Conf. on Pattern Languages of Programs, pp. 1–14. ACM, New York (2006)
55. Storey, M.A.D., Čubranić, D., German, D.M.: On the use of visualization to support awareness of human activities in software development: a survey and a framework. In: ACM Symposium on Software Visualization, pp. 193–202. ACM, New York (2005)
56. Tee, K., Greenberg, S., Gutwin, C.: Artifact awareness through screen sharing for distributed groups. Int. J. Hum–Comput. Stud. 67(9), 677–702 (2009)

Awareness Checklist: Reviewing the Quality of Awareness Support in Collaborative Applications

Pedro Antunes[1], Claudio Sapateiro[1], Jose A. Pino[2], Valeria Herskovic[2], and Sergio F. Ochoa[2]

[1] Department of Informatics of the Faculty of Sciences, Campo Grande, Lisbon, Portugal
paa@di.fc.ul.pt, claudio.sapateiro@estsetubal.ips.pt
[2] Computer Science Department, Universidad de Chile, Santiago, Chile
{jpino,vherskov,sochoa}@dcc.uchile.cl

Abstract. A proposal of a method to assess awareness support is made. This proposal is intended for the use of collaborative applications developers at any time during development. It consists of a checklist. It is made with the inclusion of design elements obtained by the analysis of Quality Assurance ideas applied to collaborative systems. The proposal is illustrated with its use in two cases.

Keywords: Awareness Inspection, Collaborative Applications.

1 Introduction

Awareness in its various types has always been considered a distinctive feature of collaborative systems when compared with other kinds of information systems [17]. Moreover, numerous studies have found awareness to be a very important component of a collaborative system [61, 27, 34]. Users' mobility increases the need for awareness since the collaboration environments typically change very often in this case.

We are particularly interested in assessing the awareness support in collaborative systems. An approach to do this study is by asking users about it. Questionnaires can be used for that purpose [53]. Alternatively, observation of people using the system can be useful to do this inquiry. Analysis of logging interactions [55] or video recordings [41] can then provide some answers to the evaluation of awareness support. Nevertheless, all these approaches require the participation of users.

Participation of users is not always possible or available at the time of evaluation [40]. For that case, we propose an awareness checklist which may be useful to system developers to assess their applications at various development stages. It can be argued a system's users are the best evaluators of it, which is true, but an alternative way may be required as a substitute or complement for the users' evaluation. The construction of the awareness checklist followed a process consisting of the following steps:

- Definition of awareness types.
- Definition of the design elements contributing to awareness that will be subject to the evaluation.
- Definition of correlations between design elements and awareness types, with help from experts in collaborative systems development.

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 202–217, 2010.

- Construction of the awareness checklist and summary tables.
- Validation of the awareness checklist in case studies.

The paper continues with a review of related work (Section 2); it starts with quality assurance, following with its relation to collaborative systems and then, with awareness. Section 3 deals with the awareness types. Then, Section 4 presents the proposed checklist. The use of this checklist in two cases is illustrated in Section 5. Section 6 concludes the paper with a summary of the obtained results.

2 Related Work

2.1 Quality Assurance

Quality Assurance (QA) establishes the extent to which quality is being controlled in an organization [43]. QA typically applies control measures to an input-process-output production system, uncovering nonconformities in the system, avoiding wasted resources, while doing so at the least possible cost [39].

Hinckley [39] provides an insightful view over QA progress. Initial QA measures were based on loose judge inspections made by skilled craftsman in the production line. Later on, the adoption of gage instruments and standards has led to improved inspections and greater consistency. The emergence of Statistical Quality Control (SQC) brought a higher concern with predictable production models, adopting production samples and statistical methods to guide process adjustments [42].

Six-Sigma [4] has been developed to make drastic improvements in QA based on standards, measurement and analysis systems, and continuous quality improvement. Total Quality Management (TQM) also deals with a continuous optimization of business performance [5]. But its emphasis has shifted away from the technical towards broader organizational factors such as team development, learning and culture.

Of course most concerns with QA extend beyond the traditional industrial organizations and apply to software development. For instance, the Cleanroom Software Engineering approach adopts SQC to maintain software development under statistical control [54]. However, one main limitation of this approach is the process requires stable software specifications, a requirement that is hard to ensure in the software development field.

The Software Quality Function Deployment (SQFD) [35] method adopts the Six-Sigma's scorecard with a particular focus on customer needs. The origins of SQFD are rooted in the need to improve the quality of software design using precise control points throughout the development process and constant traceability of the customer requirements [9]. Thus the QA chart adopted by SQFD correlates customer-required quality functions with the product's engineering characteristics.

Formal Technical Reviews (FTR) [25] have been widely adopted in software engineering [2]. They involve several people in a formal meeting during which a software artifact is presented, discussed and approved. FTR seek to identify defects and discrepancies in the software against plans, specifications, standards and best practices. They cover the whole software development life-cycle [50].

Johnson [46] analyzed the impact of software reviews on quality, showing that defects can be one or two orders of magnitude less costly to remove when found in initial development stages than after distribution to the customers. Moreover, software reviews were considered effective for discovering certain soft, but nevertheless costly, defects such as logically correct but poorly structured code.

2.2 Collaborative Systems and Quality Assurance

Collaborative systems bring together two main organizational assets: technology and humans. The development of collaborative systems has for long been considered a special branch of software development concerned with: group characteristics and dynamics; communication, coordination and collaboration; conflict resolution and decision making; social context of work; and positive and negative effects of technology on tasks, groups and organizations.

QA is essential to ensure the quality of collaborative systems development. The problem now is that QA must assess a very wide range of factors related with multiple stakeholders (customers, managers, individual workers, formal and informal work groups), various domains of concern (business processes, goals, tasks, group well-being, culture) and multiple technology components (addressing various aspects of collaboration such as awareness). All in all, what distinguishes collaborative systems QA is indeed the need to evaluate its impact with an eclectic perspective.

Research shows that QA activities are difficult to accomplish when collaborative systems are involved. First, these systems are difficult to assess due to the complexity, cost and time involved [38]. Second, the assessments tend to be informal [1]. Finally, collaborative systems involve conflicting views that consider technology and its impact in organizations [38]. Nevertheless several assessment methods have been proposed; e.g. Herskovic et al. [38] identifies twelve methods and classifies them according to various criteria such as development status, scope, time span of the assessment and who participates in the assessment. Of these twelve methods, six require the participation of end users in several ways, like focus groups and observations. However, participation of end users in QA turns the process costly and quite difficult to manage.

Of the remaining six methods, three require modeling and analyzing the system functionality at a very low level of detail. And finally the remaining methods adapt the FTR approach to the specific characteristics of collaborative systems assessment. The methods are: Groupware Heuristic Evaluation (GHE) [3], Groupware Walkthrough (GW) [56] and Knowledge Management Approach (KMA) [62]. GHE defines a procedure for inspecting how a collaborative system conforms with eight heuristics that codify best practices in collaborative systems development [3]. GW entails stepping through task sequences to conceptually explore task goals, actions necessary to perform tasks, knowledge needed to accomplish tasks, and possible performance failures [33, 56]. Finally, KMA involves using a checklist to assess how the system helps knowledge circulation [62].

2.3 Quality Assurance and Awareness

We will now delve into the three FTR methods mentioned above to unravel how they address the quality of awareness support. As previously mentioned, GHE systematizes QA activities around a set of heuristics [3]. These heuristics define a checklist with qualities that a collaborative system should have. Some of these heuristics point towards the importance of awareness: (1) *Provide the means for intentional and appropriate gestural communication, (2) Provide consequential communication of an individual's embodiment, (3) Provide consequential communication of shared artifacts, (4) Management of tightly and loosely-coupled collaboration, (5) Allow people to coordinate their actions, and (6) Facilitate finding collaborators and establishing contact.*

GW involves stepping through task sequences to conceptually explore the actions users will perform. In order to formalize the analysis of the work context, Pinelle and Gutwin [56] defined the Mechanics of Collaboration, a set of seven collaboration primitives that makes up group dynamics [33], that include *monitoring* as an explicit concern with awareness.

KMA differs from the other techniques. Instead of focusing on the essential features of collaboration support, KMA seeks to evaluate how organizations are able to manage their knowledge while using collaborative systems [62]. It focuses on analyzing situations where knowledge does not flow correctly. A checklist is provided with a set of questions that expose missing links, black holes and points of congestion in information flows. Awareness is indirectly considered in this approach.

All in all, we observe concern with awareness is present in these FTR methods but diluted among many other issues. Thus we find here an opportunity to develop a FTR method specifically concerned with reviewing the quality of awareness support.

2.4 Other Methods to Evaluate Quality of Awareness Support

Convertino et al. [12] developed a laboratorial method to assess activity awareness in controlled settings. This is the only work we found that explicitly develops a QA technique for awareness in collaboration systems. The method is based on collaboration scenarios drawn from field studies and assessed during laboratory experiments using questionnaires, interviews and observations. Unfortunately this approach requires significant time and effort to prepare and run the experiments. Furthermore, it requires a mature definition of the system functionality, which makes it difficult to apply at early design stages.

QA of awareness has also been a major issue in a quite different research field: cognitive systems engineering. The main reference in this area is the work by Endsley et al. on situation awareness [22-24]. Situation awareness is the capability to understand a series of events at three different levels [24]: in level 1, training and experience direct attention to critical elements in the environment; level 2 integrates elements that aid understanding the meaning of critical elements; and level 3 considers understanding the possible future scenarios. Endsley developed the Situational Awareness Global Assessment Technique (SAGAT) [20] to assess the users' situation awareness. SAGAT uses questionnaires to inquire users about perception, comprehension and projection issues in situations where working activities have been interrupted [21]. The main application areas of SAGAT deal with complex activities like piloting. Other techniques, like thinking aloud, filling mini situation reports and probing questions have been used to assess situation awareness [65]. All these techniques involve end users in the assessment process.

Still regarding the cognitive perspective, Zhang and Hill [66] developed a pattern-based approach to situation assessment. The approach uses spatial relationships in synthetic workspaces to represent the situation. Situation assessment is based on two major steps: data organization for perception (e.g. clustering) and matching against situation templates, which have to be predefined.

2.5 Summary

Figure 1 summarizes our analysis of the related literature. The discussion on Quality Assurance brings forward the TQM movement, which originated new assessment methods based on participation and collaboration. Of those methods, FTR take a prominent place in software development. Collaborative systems are a specialized sector within software development, which has lead to specialized FTR methods such as GHE, GW and KMA. Our analysis of these methods uncovered there was little coverage of awareness. This opens up the opportunity to develop a FTR method specifically focused on awareness assessment. Beyond the FTR context, we have only found one technique in the literature whose major concern is awareness assessment. However, the proposed approach requires significant effort and time to accomplish; and it is difficult to apply at early design stages.

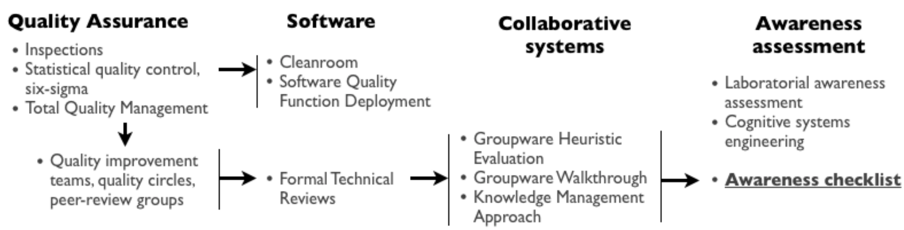


Fig. 1. Summary view of awareness assessment

3 Awareness Elements

The *time/place* relationship is the most prevalent subject related to collaborative applications. The time/place map proposed by Johansen et al. [45] is founded upon the discussion by DeSanctis and Gallupe [15] on the support to remote and local groups. The distinctions between same-place, different-place and any-place do not only highlight spatial issues but also the actual extent members have to access the group. In particular, the members located at different places are conditioned by infrastructure factors like network connectivity, data distribution, throughput, bandwidth and message delays. Some variations of the time/place map have been elaborated to encapsulate these factors [57]. They expand the place dimension to three categories, considering co-located, virtual co-located and remote places.

Social theorists have also regarded the degree of communication afforded by technology as a fundamental constraint to collaboration. Studies of media richness [13] and media naturalness [48] show that communication mediated by technology loses several important features such as nonverbal cues, rapid feedback and arousal. In this line of reasoning, the notion of place is fundamental to adapt the medium to the group and task. The time/place differences define collaboration awareness as the perception of temporal and spatial structures in a group of peers [26, 59].

Several authors extend the notion of place, linked above to infrastructural issues, to the notion of space [18]. Spaces provide additional context to places such as physical location, topology and mobility. We may identify five types of space. The first one is the *geographical space*, which introduces geographical relationships such as location, distance and orientation. Dix et al. [16] further characterized location as either being Cartesian or topological.

Then we have the *physical space*, which mainly concerns mobility. Mobility has been categorized in wandering, visiting and traveling [49]. Dix et al. [16] proposed another taxonomy: fixed, mobile, autonomous, free, embedded and pervasive. Hazas et al. [37] discuss location awareness as the means to determine physical location using various types of sensing technology such as GPS and RFID. Hazas et al. [37] also make the distinction between physical and semantic locations such as rooms, floors and buildings.

Cheverst et al. [10] studied the relationships between physical spaces, mobility, location awareness and location services to derive important requirements such as flexibility, visibility and context-sensitivity. Davis also [14] analyzed the challenges posed by mobility and information access, including the removal of time/space constraints to communication and knowledge work, improved access to decision makers and increased ability to receive and process information.

The third type of space we consider is the *virtual space*. Rodden [58] developed the notion of virtual space as a collection of computer-supported interactive spaces. Many collaborative applications offer various types of virtual spaces, including virtual meeting rooms, media spaces and Collaborative Virtual Environments [60].

Virtual spaces have a conceptual topology, they are interactive, shared, malleable, populated and may be navigated. Interaction involves the dissemination of interaction and navigation information to the group members, thus constructing what Rodden has coined context awareness [58]. Rodden also proposed a conceptual model of context awareness in virtual spaces using focus and nimbus. Focus and nimbus are subspaces that map the attention and presence of elements in spaces. Also related with context awareness, we find the distinction between private and public spaces, the former pertaining to things and actions belonging to one single individual and the latter shared among a group [28, 11].

Navigation in virtual spaces is not necessarily spatial but may also be logical. For instance, the rooms-metaphor defines navigation in virtual spaces like discussion forums [29] that are not spatially organized but rather organized according to a set of interests. Virtual spaces may assume complex structures, such as clusters, stacks, lists, tables and rooms [30]. Users should then be able to navigate these structures and obtain context awareness. Collaborative visualization, as an enabler of collaboration, is naturally a major challenge to consider in virtual spaces [6, 11]. Collaborative visualization involves data exchange, shared control and dynamic interaction [52].

Another type of space we identify is the *social space*. Dourish [18] and Brewer and Dourish [7] proposed social spaces as adequate to understand broader issues related to social practice and context. In this respect, social places combine geographical, physical and virtual affordances with social interaction, cultural meaning, experience and knowledge. Dourish [19] also proposed the notion of embodied interaction to account for the embedded relationships between social and the other spaces.

The final type of space we consider is the *workspace*. According to Snowdon et al. [60], a workspace is a container of places with ongoing activities. We may distinguish two different aspects of workspaces. On the one hand, workspaces may organize activities according to logical sets. A group editor is a good example of this type of workspace, as it serves to organize activities like writing and revising, while maintaining a coherent view of the whole [47]. On the other hand, workspaces also introduce geography as an important context for working activities [52].

Liechti [51] studied the relationship between context and workspace and proposed peripheral awareness as the capability to understand the activities being carried out by others nearby one's place. Gutwin and Greenberg [32] expanded this view to account for the whole space, defining workspace awareness as the understanding of another person's interactions in a shared workspace using a basic set of questions: who, what, where, when, and how.

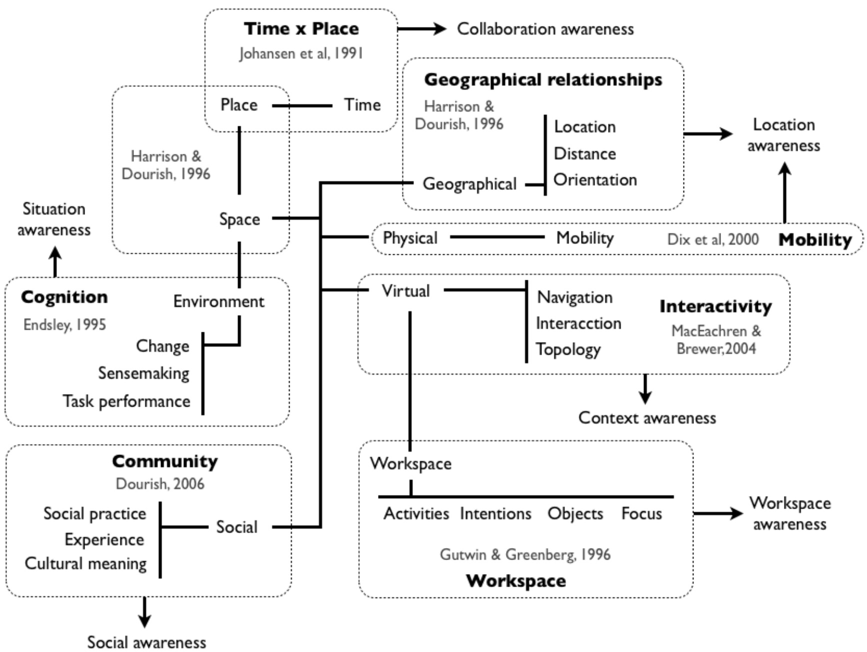


Fig. 2. Overview of main awareness elements

According to Gutwin et al. [31], workspace awareness is a specialization of a more broad concept designated *situation awareness*. Endsley [23, 24] defined situation awareness as the understanding of what is going on in the working environment with the purpose of performing tasks effectively. Endsley defined three levels of situation awareness: perception of elements in the current situation, comprehension of current situation and projection of future status.

Jensen [44] combined situation awareness with sensemaking, a theory developed by Weick [63, 64] to understand the relationships between environmental changes and organizational responses. Sensemaking is defined as the capability to create order and make retrospective sense of what occurs through the articulation of several cognitive functions like perception, interpretation and anticipation of events [64]. Cecez-Kecmanovic [8] highlighted that sensemaking emerges from individual, coordinated and collaborative efforts.

Figure 2 presents an overview of the awareness elements that we have identified: time x place, space (geographical, physical, virtual and social), workspace and situation awareness, as well as their main aspects and the types of awareness they support.

4 The Awareness Checklist

In Section 3 we laid out a comprehensive overview of the main awareness elements we find in collaborative systems. We identified seven types of awareness: time x place, geographical space, physical space, virtual space, social space, workspace, and situation awareness. We also uncovered several design elements that influence or contribute to awareness support. The total number of design elements discussed in the previous section is 77. To make this a manageable list, we organize the design elements in the 14 categories shown in Table 1.

Table 1. Main design elements influencing awareness

	Design categ.	**Design elements**
1	Accessibility	Same place, different place, any place, co-located, virtually co-located, remote
2	Communication	Synchronous, asynchronous, network connectivity, message delivery, network management
3	Spatiality	Cartesian locations, topological locations, distances, orientation, focus/nimbus
4	Mobility	Wandering, visiting, traveling, fixed, mobile, autonomous, independent, embedded, pervasive
5	Physicality	Physical constraints, physical places, physical topology, physical attributes
6	Navigation	Viewports, links, radar views, teleports
7	Virtuality	Private, group, public, data access privileges, concurrency control, floor control, version control, virtual constraints, virtual places, virtual topology, virtual attributes
8	Membership	Participants, roles, activities, privileges, group history
9	Attention	Eye-gaze orientation, body orientation, voice filtering, portholes/peepholes
10	Task	Who, what, where, when, how, task history
11	Interaction	Feedback, feedthrough, backchannel feedback
12	Interdependence	Parallel activities, coordinated activities, mutually adjusted activities, loosely coupled, tightly coupled
13	Internalization	Events, actions, resources, critical elements, meaning, future scenarios
14	Externalization	Individual cognition, distributed cognition, team cognition

In Table 2 we define the relationship between design and awareness elements. These relationships are derived from the analysis presented in Section 3. However, during this research, we observed that these relationships are more complex than what Table 2 implies. For instance, the different-place design element has main influence on "time x place" awareness. However, a different-place design also influences negatively workspace awareness, especially because communication channels tend to be a limiting factor. Therefore we may say that accessibility directly influences "time x space" awareness and indirectly influences workspace awareness.

Table 2. Main relationships between design and awareness elements

	Type of awareness	Design categories
1	Time x place	Accessibility, communication
2	Geographical space	Spatiality
3	Physical space	Mobility, physicality
4	Virtual space	Navigation, virtuality
5	Social space	Membership, attention
6	Workspace	Task, interaction, interdependence
7	Situation	Internalization, externalization

To find out these indirect relationships, we requested five experts in collaborative technology to define the relationships between the 77 design elements and the seven types of awareness. These experts were supplied with a table having the strong relationships shown in Table 2 and were requested to define additional moderate and weak relationships. To calculate the correlations, the strong, moderate and weak relationships were empirically given the values 4, 2 and 1, respectively. The accumulated correlations obtaining a value equal or below 2 were zeroed.

Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	Acessibility						Communication					Spatiality				
Awareness categories \ Design elements	Same place	Different place	Any place	Co-located	Virtually co-located	Remote	Synchronous	Asynchronous	Netw ork connectivity	Message delivery	Netw ork management	Cartesian locations	Topological locations	Distances	Orientations	Focus/nimbus
Time x Place	6,1	6,1	6,1	6,1	6,1	6,1	7,3	7,3	7,3	7,3	7,3		1,1	1,5		
Geographical space			2,4			2,4						14,3	14,3	14,3	14,3	14,3
Physical space			1,1			1,8						1,3	1,3	1,7	1,7	
Virtual space					1,5		1,3		1,7	1,7						2,2
Social space					0,7		1,8	1,2		1,5	0,9					1,5
Workspace	0,6	1,3	0,9	0,6	0,7	1,1	1,8	1,8	0,9	1,3						
Situation		1,3	1,1		0,9	1,3	1,5	1,5	1,0	1,3			1,0			

Column #	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
	Mobility									Physicality				Navigation			
Awareness categories \ Design elements	Wandering	Visiting	Travelling	Fixed	Mobile	Autonomous	Independent	Embedded	Pervasive	Physical constraints	Physical places	Physical topology	Phyisical attributes	View ports	Links	Radar view s	Teleports
Time x Place												1,4		1,4	1,4		
Geographical space	2,0	1,2	2,8		2,4										2,7		
Physical space	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	4,8	10,8	10,8	10,8	10,8				
Virtual space														10,9	10,9	10,9	10,9
Social space	0,7	0,7	0,7			0,5								1,5		1,5	1,5
Workspace	0,7	0,5	0,5		0,4											1,1	
Situation	0,6				0,9	0,6		0,4		1,3	1,3			1,3		1,3	1,0

Column #	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
	Virtuality											Membership					Attention			
Awareness categories \ Design elements	Private objects	Group objects	Public objects	Data access privileges	Concurrency control	Floor control	Version control	Virtual constraints	Virtual places	Virtual topology	Virtual attributes	Participants	Roles	Activities	Privileges	Group history	Eye-gaze orientation	Body orientation	Voice filtering	Portholes, peepholes
Time x Place																				
Geographical space																				
Physical space																				
Virtual space	4,0	4,0	4,0	4,0	4,0	4,0	4,0	4,0	4,0	4,0	4,0						1,6			
Social space	0,4	0,5	0,5	0,4	0,4							5,9	5,9	5,9	5,9	5,9	7,4	7,4	7,4	7,4
Workspace		0,7	0,5			0,6	0,4	0,4	0,4	0,4	0,4	2,0	1,8	2,0	1,1					
Situation		0,5	0,3		0,3							1,5	1,5	1,5		1,0				1,3

Column #	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
	Task						Interaction			Interdependence						Internalization						Externaliz.		
Awareness categories \ Design elements	Who	What	Where	When	How	Task history	Feedback	Feedthrough	Backchannel	Individual activities	Parallel activities	Coordinated activities	Mutually adjusted activities	Loosely coupled	Tightly coupled	Events	Actions	Resources	Critical elements	Meanings	Future scenarios	Individual cognition	Distributed cognition	Team cognition
Time x Place	0,9		1,5	1,8			2,4	2,4		0,9	0,9	1,2	1,2	0,9	1,2	1,2	1,2					2,4		
Geographical space			2,4				4,8	3,6										1,8						
Physical space		1,4		1,1	1,1													1,1						
Virtual space																1,1	1,5							
Social space	1,5	1,0					2,5	2,5	2,5	0,7	1,0	1,0	1,0	0,7	1,0	0,7	1,2		0,7	1,0			2,5	2,5
Workspace	3,7	3,7	3,7	3,7	3,7	3,7	7,4	7,4	7,4	3,7	3,7	3,7	3,7	3,7	3,7	1,3	0,9	0,9				1,9	1,5	1,5
Situation	1,7	1,7	1,7	1,7	1,7	1,3	1,7	2,1	2,1			1,7	1,7	0,8	1,7	4,3	4,3	4,3	4,3	4,3	4,3	8,5	8,5	8,5

Fig. 3. Correlations matrix with moderate and weak relationships expressed by the experts

The correlations were then normalized in two ways: (1) normalize the impact of each design category in the awareness score, avoiding that design categories with a higher number of design elements have more impact on the awareness scores; and (2) normalize the awareness scale so the sum of all correlations for a given awareness category is 100%. The correlations matrix is shown in Figure 3.

We constructed the awareness checklist based on the elements summarized in Tables 1-2 and Figure 3. The checklist is also inspired on the House of Quality (HoQ), a basic QA map used by many organizations to correlate software implementations to quality items [36]. In our case, we correlate 77 design elements with seven awareness categories. The correlations adopt a qualitative classification that is also common in the HoQ: strong positive (+2); positive (+1); uncorrelated (0); negative (-1) and strong negative (-2).

The checklist is shown in Figure 4. After completion, it automatically reports the applications' positive and negative scores (Figure 5). The scores are determined in the following way:

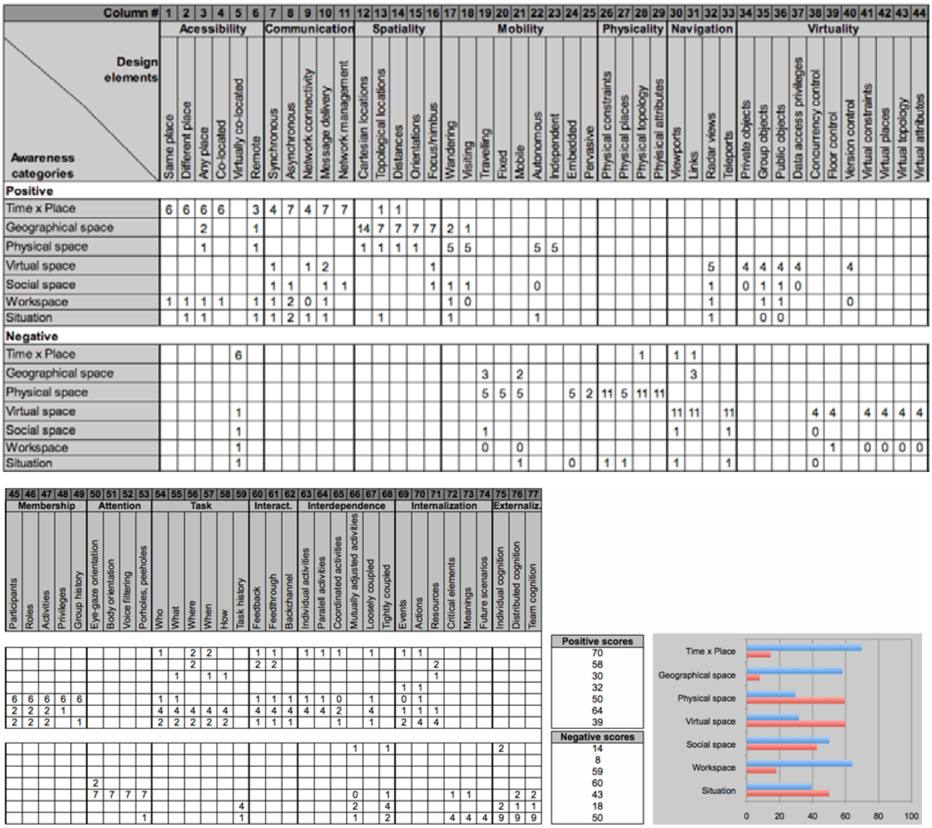
1. For each awareness category, every design element in the checklist that received a positive assessment (+2 or +1) is multiplied by the corresponding correlation expressed in the correlations matrix for that awareness category.
2. The same operation is executed for the negative assessments (-2 or -1).
3. For each awareness category, the positive score is obtained by adding the adjusted results obtained in step 1, multiplied by a 0.5 factor. This allows normalizing the scores on a [0-100] scale.
4. For each awareness category, the negative score is obtained by adding the adjusted results obtained in step 2, multiplied by a -0.5 factor, which again normalizes the scores on a [0-100] scale.

Row #		Design elements	Assessment
1	Acessibility	Same place	Θ
2	Acessibility	Different place	Θ
3	Acessibility	Any place	Θ
4	Acessibility	Co-located (face-to-face)	Θ
5	Acessibility	Virtually co-located (high-quality video/audio links)	⊗
6	Acessibility	Remote	O
7	Communication	Synchronous (same-time)	O
8	Communication	Asynchronous (different-time)	Θ
9	Communication	Network connectivity	O
10	Communication	Message delivery	Θ
11	Communication	Network management	Θ
12	Spatiality	Cartesian locations	Θ
13	Spatiality	Topological locations	O
14	Spatiality	Distances	O
15	Spatiality	Orientations	O
16	Spatiality	Focus/nimbus (center of activity)	O
17	Mobility	Wandering	Θ
18	Mobility	Visiting (e.g. others' offices)	Θ
19	Mobility	Travelling	⊗
20	Mobility	Fixed	⊗
21	Mobility	Mobile (human control)	⊗
22	Mobility	Autonomous (own control)	Θ
23	Mobility	Independent (from other devices)	Θ
24	Mobility	Embedded (in other devices)	⊗
25	Mobility	Pervasive (throughout the environment)	X
26	Physicality	Physical constraints	⊗
27	Physicality	Physical places	X
28	Physicality	Physical topology	⊗
29	Physicality	Phyisical attributes	⊗
30	Navigation	Viewports (over workspace)	⊗
31	Navigation	Links (to other places)	⊗
32	Navigation	Radar views (over workspace)	O
33	Navigation	Teleports (to others' foci)	⊗
34	Virtuality	Private objects	Θ
35	Virtuality	Group objects	Θ
36	Virtuality	Public objects	Θ
37	Virtuality	Data access privileges	Θ
38	Virtuality	Concurrency control	⊗
39	Virtuality	Floor control (channel's access)	⊗
40	Virtuality	Version control	Θ
41	Virtuality	Virtual constraints (e.g. editing)	⊗
42	Virtuality	Virtual places	⊗
43	Virtuality	Virtual topology (e.g. rooms)	⊗
44	Virtuality	Virtual attributes	⊗
45	Membership	Participants	Θ
46	Membership	Roles	Θ
47	Membership	Activities	Θ
48	Membership	Privileges	Θ
49	Membership	Group history	Θ
50	Attention	Eye-gaze orientation	⊗
51	Attention	Body orientation	⊗
52	Attention	Voice filtering (controlled by distance or orientation)	⊗
53	Attention	Porholes, peeholes (to others' working environment)	⊗
54	Task	Who	Θ
55	Task	What	Θ
56	Task	Where	Θ
57	Task	When	Θ
58	Task	How	Θ
59	Task	Task history	⊗
60	Interaction	Feedback (individual inputs)	O
61	Interaction	Feedthrough (group inputs)	O
62	Interaction	Backchannel (response tokens)	O
63	Interdependence	Individual activities	Θ
64	Interdependence	Paralell activities	Θ
65	Interdependence	Coordinated activities	O
66	Interdependence	Mutually adjusted activities	X
67	Interdependence	Loosely coupled	Θ
68	Interdependence	Tightly coupled	⊗
69	Internalization	Events	O
70	Internalization	Actions	Θ
71	Internalization	Resources	Θ
72	Internalization	Critical elements (e.g. ideas, problems)	⊗
73	Internalization	Meanings	⊗
74	Internalization	Future scenarios	⊗
75	Externaliz.	Individual cognition	⊗
76	Externaliz.	Distributed cognition	⊗
77	Externaliz.	Team cognition	⊗

Fig. 4. COIN awareness checklist (Θ=2; O=1; X=-1; ⊗=0)

The awareness checklist is used during FTR in the following way. The reviewers check the implementation against the 77 design elements. Positive relationships indicate the implementation contributes to realize the design element, while negative relationships indicate the implementation is detrimental to the respective design requirement.

Then the reviewers analyze the results in the awareness report. The positive and negative scores are discriminated according to the 14 design categories and 7 awareness categories. Overall scores for each awareness category are also shown. It should be noted the most positive outcome that may be achieved in one awareness category is having 100 positive and 0 negative scores, while the most negative outcome is having 0 positive and 100 negative scores.



Column #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Design elements / Awareness categories	Acessibility						Communication					Spatiality					Mobility					
	Same place	Different place	Any place	Co-located	Virtually co-located	Remote	Synchronous	Asynchronous	Network connectivity	Message delivery	Network management	Cartesian locations	Topological locations	Distances	Orientations	Focus/nimbus	Wandering	Visiting	Travelling	Fixed	Mobile	Autonomous
Positive																						
Time x Place	6	6	6	6		3	4	7	4	7	7		1	1								
Geographical space			2			1						14	7	7	7	7	2	1				
Physical space			1			1						1	1	1	1		5	5				5
Virtual space							1		1	2						1						
Social space							1	1		1	1					1	1	1				0
Workspace	1	1	1	1		1	1	2	0	1							1	0				
Situation		1	1			1	1	2	1	1			1				1					1
Negative																						
Time x Place					6																	
Geographical space																			3		2	
Physical space																			5	5	5	
Virtual space					1																	
Social space					1														1			
Workspace					1														0		0	
Situation					1																1	

Column #	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Design elements / Awareness categories				Physicality				Navigation				Virtuality										
	Independent	Embedded	Pervasive	Physical constraints	Physical places	Physical topology	Physical attributes	Viewports	Links	Radar views	Teleports	Private objects	Group objects	Public objects	Data access privileges	Concurrency control	Floor control	Version control	Virtual constraints	Virtual places	Virtual topology	Virtual attributes
Positive																						
Time x Place																						
Geographical space																						
Physical space	5																					
Virtual space										5		4	4	4	4			4				
Social space										1		0	1	1	0							
Workspace										1			1	1				0				
Situation										1			0	0								
Negative																						
Time x Place						1		1	1													
Geographical space									3													
Physical space		5	2	11	5	11	11															
Virtual space								11	11		11					4	4		4	4	4	4
Social space								1			1					0						
Workspace																	1		0	0	0	0
Situation		0		1	1			1			1					0						

Column #	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
	Membership					Attention				Task						Interact.			Interdependence						Internalization						Externaliz.		
	Participants	Roles	Activities	Privileges	Group history	Eye-gaze orientation	Body orientation	Voice filtering	Porholes, peeholes	Who	What	Where	When	How	Task history	Feedback	Feedthrough	Backchannel	Individual activities	Paralell activities	Coordinated activities	Mutually adjusted activities	Loosely coupled	Tightly coupled	Events	Actions	Resources	Critical elements	Meanings	Future scenarios	Individual cognition	Distributed cognition	Team cognition
Positive																																	
Time x Place										1		2	2			1	1		1	1	1		1		1	1							
Geographical space												2				2	2										2						
Physical space											1		1	1													1						
Virtual space																									1	1							
Social space	6	6	6	6	6					1	1					1	1	1	1	1	0		1		0	1							
Workspace	2	2	2	1						4	4	4	4	4		4	4	4	4	4	2		4		1	1	1						
Situation	2	2	2		1					2	2	2	2	2		1	1	1			1		1		2	4	4						
Negative																																	
Time x Place																						1		1							2		
Geographical space																																	
Physical space																																	
Virtual space						2																											
Social space						7	7	7	7													0		1				1	1			2	2
Workspace															4							2		4							2	1	1
Situation									1						1							1		2				4	4	4	9	9	9

Positive scores
70
58
30
32
50
64
39

Negative scores
14
8
59
60
43
18
50

Fig. 5. COIN awareness report

5 Examples of Use

This section briefly presents the inspection of two collaborative applications. The first application is MobileMap (Figure 6), which supports firefighters attending regular emergencies in urban areas. The second application is COIN (Figure 8), which supports construction inspectors reviewing physical infrastructures in construction sites.

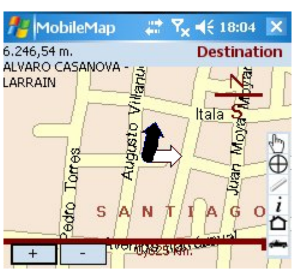


Fig. 6. MobileMap user interface

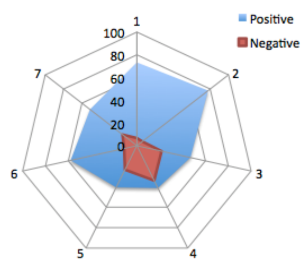


Fig. 7. MobileMap scores

Fire truck drivers use MobileMap to guide themselves to their destination. The user interface presents two arrows pointing from the current location (Figure 6): the white arrow indicates the direction in which the fire truck is moving; and the black one shows the direction in which the truck should move to get to the emergency place. This simple interface helps arriving faster to emergency sites (ref. omitted).

Two developers individually inspected MobileMaps using the awareness checklist. Figure 7 shows the obtained average scores. Analyzing these results, we may see that virtual space awareness (category 4) is the most problematic type of awareness. This should raise the developers' attention to understand if this type of awareness is required to guide the fire truck and realize how the application could better support the firemen.

Figure 8 shows the COIN user interface, which construction inspectors use to annotate digital maps related to construction projects. These annotations are done in the field and used in the office to schedule maintenance tasks to sub-contractors. Two developers also inspected COIN. Figure 9 shows the obtained results. COIN obtained low positive scores in physical and virtual space awareness (items 3 and 4). Situation awareness (item 7) also seems problematic because of the high negative scores.

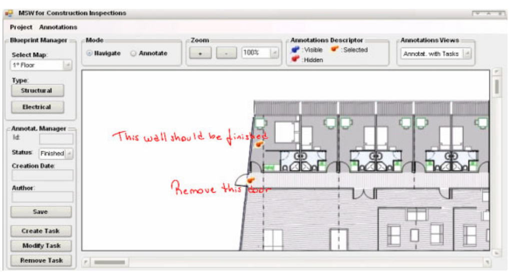


Fig. 8. COIN main interface

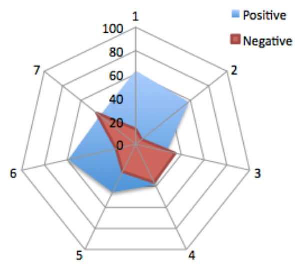


Fig. 9. COIN evaluation results

The situation with physical awareness in the two applications is particularly interesting to observe. In these applications, mobility and location awareness play an important role; however, it is not well supported. In the case of COIN, mobility support is quite appropriate but location support seems insufficient. On the contrary, in the case of MobileMap, location support seems to be appropriate. However, there is insufficient support to mobility. That is the main reason why the negative scores are high. Developers have here the chance to improve collaboration support by identifying the awareness categories and specific design elements requiring additional support. In that sense, the proposed checklist is an important instrument helping on the identification of deficiencies in collaborative applications.

6 Conclusions

Awareness is an important component of collaborative systems that helps users to conduct interaction processes. In this paper, we have studied the assessment of awareness support starting with the basic concepts of quality assurance of software systems.

We developed an awareness checklist helping developers inspect the quality of awareness support in collaborative applications. The checklist is founded on quality assurance principles and especially on the formal technical review technique. The checklist items were defined based on a comprehensive overview of awareness research that allowed us to identify 77 design elements contributing to seven different types of awareness. Of course, the developer is not forced to require all these design elements to be present in a certain system; the developer can use this checklist together with the possible mechanisms intended to provide awareness pondering the benefit of a certain awareness element with the estimated cost to the users in terms of information overload.

The correlations between design and awareness elements were defined according to theory and practice, incorporating the views of several experts in collaborative systems development. The awareness checklist allows obtaining a fast assessment of the quality of awareness support supplied by an application by simply inquiring about how effectively some key design elements have been supported. The awareness checklist serves to obtain positive and negative scores, both contributing to inform developers about which design areas require major interventions. The awareness checklist also serves to define quality metrics, control the development processes and benchmark various applications. The awareness checklist has already been used to inspect two collaborative applications. The obtained results indicate the checklist is adequate to formally review awareness support.

Acknowledgements. This paper was supported by the Portuguese Foundation for Science and Technology (PTDC/EIA/102875/2008) and Fondecyt (Chile) Grants Nº 11060467 and 1080352, and Proyecto Enlace VID 2010 (University of Chile), Grant ENL 10/10.

References

1. Antunes, P., Pino, J.: A Review of CRIWG Research. In: 16th CRIWG (2010) (in press)
2. Aurm, A., Petersson, H., Wohlin, C.: State-of-the-Art: Software Inspections after 25 Years. Software Testing Verification and Reliability 12, 133–154 (2002)
3. Baker, K., Greenberg, S., Gutwin, C.: Empirical Development of a Heuristic Evaluation Methodology for Shared Workspace Groupware. In: CSCW 2002, pp. 96–105 (2002)
4. Barney, M.: Motorola's Second Generation. Six Sigma Forum Magazine (May 2002)
5. Boaden, R.: What Is Total Quality Management... And Does It Matter? Total Quality Management 8, 153–171 (1997)
6. Brewer, I., MacEachren, A., Abdo, H., Gundrum, J., Otto, G.: Collaborative Geographic Visualization: Enabling Shared Understanding of Environmental Processes. In: Proc. of IEEE Symposium on Information Visualization, Washington, DC, p. 137 (2000)
7. Brewer, J., Dourish, P.: Storied Spaces: Cultural Accounts of Mobility, Technology, and Environmental Knowing. Int. Journal of Human-Computer Studies 66, 963–976 (2008)
8. Cecez-Kecmanovic, D.: A Sensemaking Theory of Knowledge in Organizations and Its Application. In: Davis, J., et al. (eds.) Knowledge Management. Organizational and Technological Dimensions. Springer, Heidelberg (2005)
9. Chan, L.: Quality Function Deployment: A Literature Review. European Journal of Operational Research 143, 463–497 (2002)

10. Cheverst, K., Davies, N., Mitchell, K., Friday, A., Efstratiou, C.: Developing a Context-Aware Electronic Tourist Guide: Some Issues and Experiences. In: Proc. SIGCHI Conf. on Human Factors in Comp. Systems, pp. 17–24. ACM Press, The Hague (2000)
11. Convertino, G., Ganoe, C., Schafer, W., Yost, B., Carroll, J.: A Multiple View Approach to Support Common Ground in Distributed and Synchronous Geo-Collaboration. In: 3rd Int. Conf. on Coordinated and Multiple Views in Exploratory Visualization, pp. 121–132 (2005)
12. Convertino, G., Neale, D., Hobby, L., Carroll, J., Rosson, M.: A Laboratory Method for Studying Activity Awareness. In: Proc. Third Nordic Conf. on Human-Computer interaction, pp. 313–322. ACM Press, Tampere (2004)
13. Daft, R., Lengel, R.: Organizational Information Requirements, Media Richness and Structural Design. Management Science 32 (1986)
14. Davis, G.: Anytime/Anyplace Computing and the Future of Knowledge Work. Communications of ACM 45, 67–73 (2002)
15. DeSanctis, G., Gallupe, R.: A Foundation for the Study of Group Decision Support Systems. Management Science 33, 589–609 (1987)
16. Dix, A., Rodden, T., Davies, N., Trevor, J., Friday, A., Palfreyman, K.: Exploiting Space and Location as a Design Framework for Interactive Mobile Systems. ACM Transactions on CHI 7 (2000)
17. Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. In: ACM Conf. on Computer-Supported Cooperative Work, pp. 107–114. ACM Press, New York (1992)
18. Dourish, P.: Re-Space-Ing Place: "Place" And "Space" Ten Years On. In: 20th Conf. on Computer Supported Cooperative Work, pp. 299–308. ACM Press, Alberta (2006)
19. Dourish, P.: Where the Action Is. The MIT Press, Cambridge (2001)
20. Endsley, M., Aircraft, N., Hawthorne, C.: Situation Awareness Global Assessment Technique (Sagat). In: Proc. IEEE National Aerospace and Electronics Conf., pp. 789–795 (1988)
21. Endsley, M., Garland, D.: Situation Awareness Analysis and Measurement. CRC Press, Boca Raton (2000)
22. Endsley, M., Jones, W.: A Model of Inter and Intra Team Situation Awareness: Implications for Design, Training and Measurement. New Trends in Cooperative Activities: Understanding System Dynamics in Complex Environments, pp. 46–67. Human Factors and Ergonomics Society, Santa Monica (2001)
23. Endsley, M.: Designing for Situation Awareness. Taylor & Francis, Abington (2003)
24. Endsley, M.: Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors 31, 32–64 (1995)
25. Fagan, M.: Design and Code Inspections to Reduce Errors in Program Development. IBM Systems Journal 15, 182–211 (1976)
26. Fisher, D., Dourish, P.: Social and Temporal Structures in Everyday Collaboration. In: Proc. SIGCHI Conf. on Human Factors in Comp. Syst., pp. 551–558. ACM Press, New York (2004)
27. Gaver, W.: Sound support for collaboration. In: Proc. of ECSCW 1991, pp. 293–308 (1991)
28. Greenberg, S., Boyle, M., Laberge, J.: PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal. Personal Technol. 3, 54–64 (1999)
29. Greenberg, S., Roseman, M.: Using a Room Metaphor to Ease Transitions in Groupware. In: Ackerman, M., et al. (eds.) Sharing Expertise. Beyond Knowledge Management, pp. 203–256. MIT Press, Cambridge (2003)

30. Grønbæk, K., Vestergaard, P., Ørbæk, P.: Towards Geo-Spatial Hypermedia: Concepts and Prototype Implementation. In: 13th ACM Conf. on Hypertext and Hypermedia, pp. 117–126 (2002)
31. Gutwin, C., Greenberg, S.: A Descriptive Framework of Workspace Awareness for Real-Time Groupware. Computer Supported Cooperative Work 11, 411–446 (2002)
32. Gutwin, C., Greenberg, S.: The Effects of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware. ACM T. on Comp.-Human Int. 6, 243–281 (1999)
33. Gutwin, C., Greenberg, S.: The Mechanics of Collaboration: Developing Low Cost Usability Eval. Methods for Shared Workspaces. In: WETICE 2000, pp. 98–103. IEEE Press, Los Alamitos (2000)
34. Gutwin, C., Roseman, M., Greenberg, S.: A usability study of awareness widgets in a shared workspace groupware system. In: Proc. of CSCW 1996, pp. 258–267 (1996)
35. Haag, S., Raja, M., Schkade, L.: Quality Function Deployment Usage in Software Development. Communications of the ACM 39, 41–49 (1996)
36. Hauser, J., Clausing, D.: The House of Quality. Harvard Business Review 66, 63–73 (1988)
37. Hazas, M., Scott, J., Krumm, J.: Location-Aware Computing Comes of Age. Computer 37, 95–97 (2004)
38. Herskovic, V., Pino, J., Ochoa, S., Antunes, P.: Evaluation Methods for Groupware Systems. In: Haake, J.M., Ochoa, S.F., Cechich, A. (eds.) CRIWG 2007. LNCS, vol. 4715, pp. 328–336. Springer, Heidelberg (2007)
39. Hinkley, C.: Defining the Best Quality-Control Systems by Design and Inspection. Clinical Chemistry 43, 873–879 (1997)
40. Holzinger, A.: Usability engineering methods for software developers. Communications of the ACM 48, 71–74 (2005)
41. Hornecker, E., Marshall, P., Dalton, N.S., Rogers, Y.: Collaboration and interference: awareness with mice or touch input. In: Proc. of CSCW 2008, pp. 167–176 (2008)
42. Howell, J.: Statistical Quality Control. Mathematics Magazine 25, 155–157 (1952)
43. Hoyle, D.: ISO9000 Quality Systems Handbook. Elsevier, Oxford (2009)
44. Jensen, E.: Sensemaking in Military Planning: A Methodological Study of Command Teams. Cognition, Technology & Work 11, 103–118 (2009)
45. Johansen, R., Sibbet, D., Benson, S., Martin, A., Mittman, R., Saffo, P.: Leading Business Teams. Addison-Wesley, Reading (1991)
46. Johnson, P.: Reegineering Inspection. Communications of ACM 41, 49–52 (1998)
47. Koch, M., Koch, J.: Application of Frameworks in Groupware—the Iris Group Editor Environment. ACM Computing Surveys (CSUR) 32 (2000)
48. Kock, N.: Media Richness or Media Naturalness? The Evolution of Our Biological Communication Apparatus and Its Influence on Our Behavior toward E-Communication Tools. IEEE Transactions on Professional Communications 48, 117–130 (2005)
49. Kristoffersen, S., Ljungberg, F.: Your Mobile Computer Is a Stationary Computer. In: CSCW 1998 Handheld CSCW Ws., Seattle, USA (1998)
50. Laitenberger, O., DeBaud, J.: An Encompassing Life-Cycle Centric Survey of Software Inspection. The Journal of Systems & Software 50, 5–31 (2000)
51. Liechti, O.: Supporting Social Awareness on the World Wide Web with the Handheld Cyberwindow. In: Ws. on Handheld CSCW at CSCW 1998, Seattle, USA (1998)
52. MacEachren, A., Brewer, I.: Developing a Conceptual Framework for Visually-Enabled Geocollaboration. Int. Journal of Geographical Information Science 18, 1–34 (2004)

53. MacMillan, J., Paley, M., Entin, E.: Questionnaires for Distributed Assessment of Team Mutual Awareness. In: Salas (ed.) Handbook of Human Factors and Ergonomic Methods, pp. 3–32. Taylor & Francis, Abington (2004)
54. Mills, H., Dyer, M., Linger, R.: Cleanroom Software Engineering. IEEE Software 4, 19–25 (1987)
55. Nacenta, M., Pinelle, D., Stuckel, D., Gutwin, C.: The Effects of Interaction Technique on Coordination in Tabletop Groupware. In: Graphics Interface 2007, pp. 191–198. ACM Press, New York (2007)
56. Pinelle, D., Gutwin, C.: Groupware Walkthrough: Adding Context to Groupware Usability Evaluation. In: Proc. SIGCHI Conf. on Human factors in computing systems, pp. 455–462. ACM Press, Minneapolis (2002)
57. Rodden, T., Blair, G.: CSCW and Distributed Systems: The Problem of Control. In: Proc. 2nd Conf. on European Conf. on Computer-Supported Coop. Work, pp. 49–64. Kluwer, Dordrecht (1991)
58. Rodden, T.: Populating the Application: A Model of Awareness for Cooperative Applications. In: ACM Conf. on Computer Supported Coop. Work, pp. 87–96. ACM Press, New York (1996)
59. Sacramento, V., Endler, M., Rubinsztejn, H., Lima, L.S., Goncalves, K., Nascimento, F., Bueno, G.: Moca: A Middleware for Developing Collaborative Applications for Mobile Users. IEEE Distributed Systems Online 5 (2004)
60. Snowdon, D., Munro, A. (eds.): Collaborative Virtual Environments: Digital Places and Spaces for Interaction. Springer, New York (2000)
61. Tang, J.: Findings from observational studies of collaborative work. International Journal of Man-Machine Studies 34, 143–160 (1991)
62. Vizcaíno, A., Martinez, M., Aranda, G., Piattini, M.: Evaluating Collaborative Applications from a Knowledge Management Approach. In: 14th WETICE 2005, pp. 221–225. IEEE Press, Los Alamitos (2005)
63. Weick, K.: Making Sense of the Organization. Blackwell, Oxford (2001)
64. Weick, K.: The Collapse of Sensemaking in Organizations: The Mann Gulch Disaster. Administrative Science Quarterly 38, 628–652 (1993)
65. Yanco, H., Drury, J.: "Where Am I?" Acquiring Situation Awareness Using a Remote Robot Platform. IEEE Conf. on Systems, Man and Cybernetics 3, 2835–2840 (2004)
66. Zhang, W., Hill, R.: A Template-Based and Pattern-Driven Approach to Situation Awareness and Assessment in Virtual Humans. In: 4th Int. Conf. on Autonomous Agents, pp. 116–123 (2000)

Supporting Asynchronous Workspace Awareness by Visualizing the Story Evolution in Collaborative Storytelling

Jana Schumann[1], Tanja Buttler[2], and Stephan Lukosch[2]

[1] University of Magdeburg, Faculty of Computer Science, Department of Simulation and Graphics, Universitaetsplatz 2, 39106 Magdeburg, Germany
jana.schumann@st.ovgu.de
[2] Delft University of Technology, Faculty of Technology Policy and Management, Section Systems Engineering, P.O. box 5015, 2600 GA Delft, The Netherlands
{t.buttler,s.g.lukosch}@tudelft.nl

Abstract. Workspace awareness support is mandatory for group support systems. In this paper, we present a novel approach to asynchronous awareness by means of traceability support. We integrate and evaluate our approach in the web portal of CASTing, a tool for audio-based collaborative storytelling. We describe the development of a prototype that visualizes how the collaborative story evolved over time. Our visualization helps group members assess who has modified the shared story, what exactly has been modified and when it has been modified. We evaluate different awareness factors in an experiment. The experiment proved that our visualization approach enables users to acquire workspace awareness by accessing information about previous work of other users.

Keywords: Information visualization, workspace awareness, traceability, collaborative storytelling

1 Introduction

The art of telling stories has a long and venerable history, dating back for centuries. Storytelling is a traditional way to share experience and is mostly oral [1]. Currently,there is an increased demand for audio books [2, 3] and podcasts [4, 5], which indicates arenaissance of listening. This renaissance of listening is also visible in organizations where stories are exchanged face-to-face or via the telephone in order to share knowledge.

Telling stories is not only a human way to share knowledge and experiences, but is also used as a method in different application areas under the designation *storytelling*. Collaborative storytelling aims at the development of a common understanding within a group through coordinated narrating activities, in order to make implicit knowledge explicit [6]. On this basis, audio-based collaborative storytelling uses the act of telling stories in groups in order to enable the exchange of experiences and knowledge within a group. Thereby, audio-based collaborative storytelling provides an alternative to mainly textual techniques, e.g. wikis [7].

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 218–232, 2010.

CASTing is a groupware system that supports collaborative audio-based storytelling [6]. CASTing supports users collaboratively creating non-linear stories. Compared, to a linear story with one story thread a non-linear story has several parallelthreads [8]. Non-linear stories are often visualized in a so-called story graph which displays the alternative story threads. In CASTing, a story graph consists of nodes containing audio material and linking edges. CASTing itself consists of two major components: the CASTing client and the CASTing web portal. The client application allows users to create a project team, add audio recordings, segment audio recordings, link audio recordings and select and publish a linear story. With the client users can retrieve the most current version of the story graph and synchronize their local changes. The CASTing web portalallows users to publish podcasts on the Web and discuss, comment, vote and reuse audio-based stories.

In CASTing, users thus mainly collaborate asynchronously in order to create an audio-based story, i.e. users can change the story graph while disconnected and synchronize their changes later on. The resulting asynchronous growth of shared data makes it difficult to trace the story graph evolution. To address this issue, it is necessary to offer asynchronous awareness support [9] and provide users with the necessary data to understand the recent activities in the CASTing system. By providing adequate awareness supports"an understanding of the activities of others, which provides a context for your own activity" [10] can be achieved.

Gutwin et al. [11] distinguish four types of awareness: group-structural awareness, social awareness, informal awareness and workspace awareness. In this paper, we focus on workspace awareness and how workspace awareness can be supported in the CASTing web portal. Workspace awareness is "the up-to-the-moment understanding of another person's interaction with the shared workspace". It is awareness of people and how they interact with the workspace, rather than awareness of the workspace itself [12].

We address the missing workspace awareness in the CASTing web portal with a novel approach on visualizing the story graph evolution which allows users to trace and understand how the current shared state has been achieved, i.e. the visualization displays who changed the story graph, what was changed and when it was changed.

In the following sections,wefirst determine the requirements for such workspace awareness support and the corresponding visualization. Then we discuss related work, and in detail present our solution as well as its integration in CASTing. We further present the setup and results of our evaluation experiment, before we conclude with a summary and outlook on future work directions.

2 Requirements Analysis

In this section, we determine the requirements for supporting awareness within the CASTing web portal. We describe a scenario to state the problem and the hypothesis for this work.

"A group of students at different universities in Germany and USAmade it their business to collect the differences between the educational establishments. For that they want to use CASTing as a tool for audio-based collaborative storytelling. The

group worked successfully for a few weeks on the project, but now they want to widen their circle, so they sent an invitation over the web portal to another person. That person decided to join the project and needs an overview of the events from the last few weeks. He wants to know who is working on the project and how the current story graph has developed over the time."

Whenever people work together in a shared environment (virtual or face-to-face) they need information about the activities and intentions of their co-workers. This information is important for a successful collaboration, especially in groupware systems [10].

Our scenario demands more advancedworkspace awareness. The new group member needs an overview about the recent activities. Within CASTing the story graph captures the current achievement of the collaborating team, as it shows the different alternatives for a story. Users add and remove nodes as well as edges from the story graph. To achieve awareness in relation to these activities, it is necessary to visualize how the story graph developed over time. We base our further requirements analysis on the following hypothesis:

Hypothesis: The visualization of the story graph evolution will enable group members to trace activities in the workspace.

We now analyze requirements for a suitable extension of CASTing based on the above hypothesis. We will evaluate the hypothesis in the evaluation section of our article.

In order to allow users to understand the story graph evolution and thereby provide workspace awareness, the visualization has to provide certain information. According to Gutwin and Greenberg [12], elements of workspace awareness can be divided into two parts: those related to the present and those related to the past. Our scenario focuses on elements of the past, which are action history, artifact history, event history and presence history. These elements of workspace awareness should answer the following questions: *How did an action happen? How did an artifact come to be in this state? When did that action happen? Who was here, and when? What has a person been doing?* When these questions are answered, users are able to identify what happened when, and who made each change. Thereby, users would not only understand how the story graph developed over time, but also who is responsible for the story graph evolution. Thus, the following requirements have to be met:

R1: The visualization of the story graph evolution has to show meta-information which allows users to assess who made any changes, and when those changes were made.

In order to display the meta-information, records have to be kept:

R2: CASTing has to record awareness information once actions take place.

Not every user might be interested in the complete evolution of the story graph. In some cases, users might only have missed a specific time period or might be interested in the changes by a specific group member. This leads to the following requirements:

R3: The visualization of the story graph evolution has to allow users to focus on a specific time period.

R4: The visualization of the story graph evolution has to allow users to focus on a specific group member.

In CASTing, users can select a single thread in the story graph and export this thread as a linear story. Therefore, stories are important artifacts and users might be interested in how this story was constructed:

R5: The visualization of the story graph evolution has to allow users to focus on the development of a linear story.

3 Related Work

In the previous section we identified the requirements for raising awareness regarding asynchronous growth of a story graph.

In this section we consider existing approaches and discuss whether these approaches are suitable to raise awareness regarding the asynchronous growth of the story graph.

Erickson and Laff [13] have added a timeline to the chat environment Babel, in order to better understand the history of a chat conversation. By design, a timeline allows one to focus on specific time periods (R3). In the timeline, each user is represented by a row. Each row displays the activities from all chats of that user, thereby focusing on that user (R4). Tool tips present additional information like the time of contribution and additional information about the user. The timeline enables users to discover when other users are interacting in the collaboration space and it allows users to adjust their working hours so that synchronous collaboration is possible when needed. However, a timeline does not allow users to focus on artifacts like a chat (5).

Virtual School is a collaboration space for student interaction. A user study has revealed that collaboration broke down several times due to lack of activity awareness [14]. One solution was to integrate a timeline into the students` workspace [15]. For each project, the timeline showed different documents. Changes to the documents were represented by the icons on the time axis. To access documents, users were forced to select them in the timeline instead of from a list of documents, making the timeline an integral part of daily work.

User studies [16] have shown that the timeline was of great value to people who were observing the group`s progress. E.g., when there were white areas on the timeline, teachers queried the students responsible for those documents about problems in their group process, and provided help. Here, the timeline allows teachers to focus on projects. However, the timeline does not allow them to focus on pupils across projects (R4).

In both cases, the timeline allows users to focus on two dimensions, the time period and one other dimension like actors or projects.

DreamObjects [17] is a platform for transparently managing shared data of synchronous groupware. It offers flexible and extensible solutions for data distribution, concurrency control, data persistency, latecomer support, and user interface notification. Within

DreamObjects, latecomers can choose between a direct state transfer and a replay of how the current state has been reached [18]. The replay mechanism addresses the collaboration awareness [19] and thus offers a possibility to target the evolution of the story graph through animation. However, in DreamObjects it is not possible to focus on actors (R4) or artifacts (R5).

Facebook offers an activity feed [20], to keep users appraise of the activities of their friends, thereby providing some meta-information (R1). This awareness mechanism allows users to follow actions, but does not offer help in focusing on a particular friend (R4) or activities in a time period (R3).

Apart from the discussed applications, there exist quite a few tools that support users in collaboratively creating stories. TellStory [21] is a collaborative application that supports groups in creating text-based stories. However, no awareness support regarding the growth of the story is provided.

StoryMapper[22] supports groups in telling a story modeled as a conceptual map. Each node in such a map represents an event consisting of facts, the time of the event, and the involved actors and links to related multimedia artifacts. The edges represent semantic relationships between the nodes. The border color of a node is used to indicate who created that node. StoryMapper provides some meta-information (R1) and utilizes color coding to show which user created a node, thereby partially addressing (R4). However, it does not have any mechanism that helps users to focus on time periods (R3) or parts of the story graph (R5).

PhotoStory[23] uses storytelling to increase the awareness in the group about its external presentation but also its social activities. For that purpose, the group can create stories that consist of a series of pictures with corresponding subtitles. PhotoStory uses BSCW [24] as a collaborative workspace. BSCW-specific data structures provide the basis for annotations of the pictures. Apart from story related attributes such as type of event or position of a picture in the drama arc, awareness related attributes like author and creation date are provided. While PhotoStory provides some meta-information about the images (R1), it does not support the users in understanding the evolution of the story by focusing on specific time periods (R3) or users (R4).

The internet-based storytelling application Voicethread[25]supports groups in creating sequences of images or video clips. Users can add textual, audio or video based comments to these multimedia artifacts and hence create a digital story. These comments can be associated with an author through a picture. The comments can be played back in the order of their creation. However, Voicethread does not employ filtering mechanisms for supporting users in understanding the evolution of the story (R3, R4).

The above discussion shows that there is currently no sufficient support for raising awareness of the asynchronous growth of a story graph.

4 Approach

In the CASTing web portal the user cannot keep track of the other users' activities which makes it difficult to stay aware of the changes in the workspace. In our requirements analysis, we identified requirements to improve the workspace awareness by visualizing the story graph evolution. In order to test our hypothesis, we developed and integrated a prototype in the CASTing web portal. The example scenario (Figure 1, Figure 2) was described in Section 2.

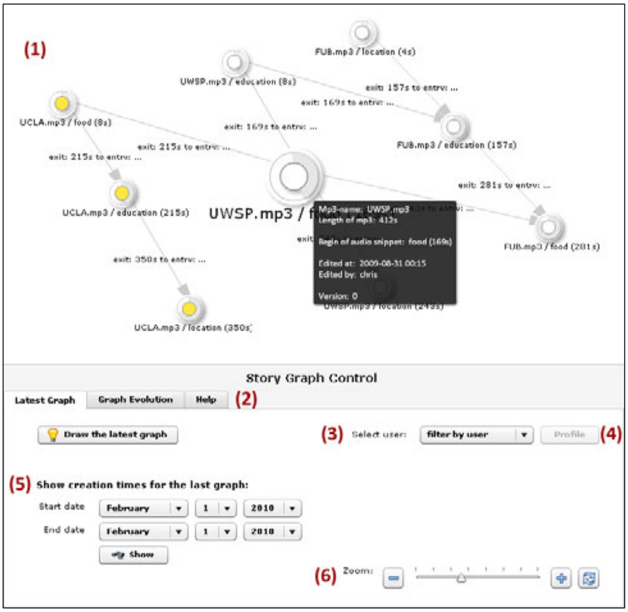


Fig. 1. Story Graph Visualization

The prototype is divided into two main parts (Figure 1). The upper part displays the actual story graph visualization (R1),and the lower part displays the control functions to filter the displayed information in the story graph (R3, R4).

The visualization itself(1) shows a story graph.We picked that kind of visualization, because the story graph is also used in the CASTing desktop application, which is needed by the user for creating the graph collaboratively and asychronously. In this way the users don't get confused. Tool tips provide additional information about the nodesand edges in the visualization (R1).

The story graph control (2) is divided into three parts: Latest Graph, Graph Evolution and Help. The Latest Graph control allows a user to explore who did what and when in the latest story graph, thereby addressing R4. The story graph can be filtered by a user (3), highlighting the edges and the outer rings of the nodes created by the selected user (R4). Thereby, this filter offers the ability to see which person created what artifact in the most current view. In addition,a profile of the selected user is provided (4).

The story graph can also be filtered by creation date of nodes and edges (R3) in the project (5). If a node or an edge was created in the selected time period, these nodes and edges are highlighted (R3). This function offers the ability to see when an artifact was created in the latest graph view. By using this function the question "*When did that action happen?*" is answered for the users.

The Latest Graph control also offers a zoom feature (6) for zooming in and out of the latest story graph. This was implemented to help users when the story graph is larger than the panel on which it is displayed. This allows for the user to take a look at the full graph or have a closer look at specific details. The zoom function is also available in the Graph Evolutioncontrol.

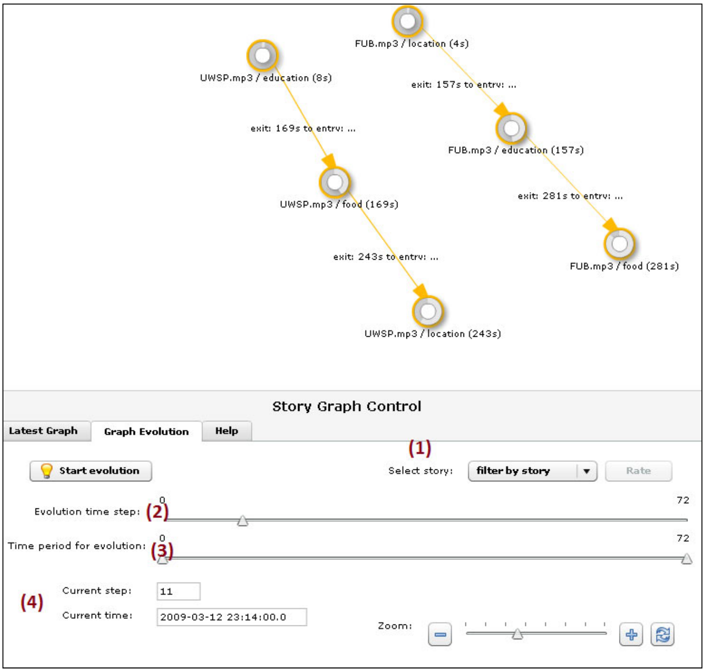


Fig. 2. Story Graph Evolution

The second part of the story graph control is the Graph Evolution (Figure 2). The Graph Evolution control provides several filter functions of all graph artifacts, which were created, modified or deleted over time, and thereby allowing users to understand the evolution of the story graph (R1, R5).

By selecting a story (1), the user filters the story graph evolution in regard to the created stories in the project. If a node or an edge is part of the selected story, then these nodes and edges are highlighted in the story graph. This filter offers the ability to see how the stories in the project were created. Additionally,a user can go directly to the rating of the selected story in the CASTing web portal.

The evolution function allows a user to see how the story graph developed over time (R5).Two sliders allow a user to replay the story evolution. The lower slider (3) allows a user to pre-select a specific time period and the upper slider (2) enables the user to slide through that selected time period. The current slide step and time are displayed in the lower part of the evolution function (4). By using this function the questions *"What has a personbeen doing?"*, *"When did that action happen?"* and *"How did an artifact come to be in this state?"* are answered.

The third and last part of the story graph control is the help section (Figure 3).It provides graphical descriptions of the meaning of the colors and displays of a node or an edge, therebyimproving the understanding of the story graph visualization.

The color coding is important for the users to recognize the state of the node and edge preattentively.The inner ring of a node (1) indicates its age, allowing a user to

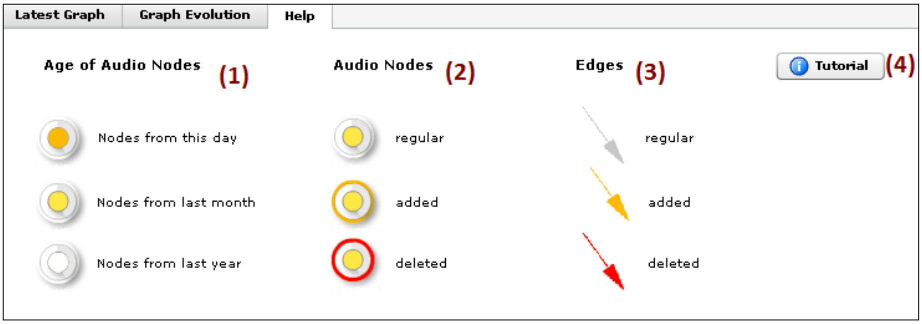


Fig. 3. Help Section

see the approximate age of a node at a glance. If the inner ring is colored orange it was created on that day. If it is yellow then it is at least one month old, and if it is white it is at least one year old. The outer ring of a node (2) is used to indicate the state of the node. If the outer ring of the node is highlighted in orange, the node was created at the chosen point in time. If it is red a user has deleted the node at the chosen point in time. The same color coding is used for edges (3). The help section is meant to be a legend of the story graph evolution, but it also offers a tutorial (4) for new users.

5 System Implementation

Two components were added to the existing CASTing system to implement the visualization of the story graph evolution; an activity log and a service which utilizes the activity log.

An activity log was implemented to record every change to the story graph (R2) in a database used in the StorytellingServer. Information about the kind of change, date and user were gathered. Additionally a service utilizing the activity log to provide information for the visualization and the filter mechanisms was implemented. Both the activity log and the service were integrated into the Storytelling Kernel.

The graph visualization itself and the control sections were provided as a rich internet application (RIA) based on BirdEye[26] and AdobeFlex [27]. BirdEye is a community project to advance the design and development of a comprehensive open source information visualization and visual analytics library for Adobe Flex. The library enables the creation of multi-dimensional data visualization interfaces for the analysis and presentation of information.

6 Evaluation

We conducted a lab experiment to test our hypothesis *"The visualization of the story graph evolution will enable group members to trace activities in the workspace"*.

With regard to this hypothesis (see Requirements Analysis), three questions (I – III) were explored in the experiment, an additional question (IV) relates to the handling of the visualization:

I. Does the visualization make it clear who changed what in the story graph? (R1, R4, R5)

This question supports the *"what"*-category and the *"who"*-category of workspace awareness and in this way advances the workspace awareness in group projects in the web portal.

II. Does the visualization make it clear when something was changed in the story graph? (R1, R3, R5)

This question supports the *"when"*-category and the *"what"*-category of workspace awareness and in this way advances the workspace awareness in group projects in the web portal.

III. Does the visualization make it clear who in general is working in the story graph? Are the users aware of other users in the project? (R4)

This question supports the *"who"*-category and the *"where"*-category of workspace awareness and in this way advances the workspace awareness in group projects in the web portal.

IV. Is the user interface easy to handle?

This question makes sure that the user interface is easy to handle for the project members. If it is easy then the users are able to get all the information they need. So the workspace awareness is advanced in an indirect way.

In the following section we describe the basic setup of the experiment, followed by a presentation of the results. Altogether ten voluntary participants (age 22 to 28; 7 male and 3 female) from four countries (1 from Canada, 5 from Germany, 3 from the Netherlands and the 1from the USA) participated in the experiment. All of them had no experience with the storytelling technique which goes with the basis of the experiment, the scenario "newcomer" mentioned in Section 2. The participants had different experience levels of accomplishing work with the computer. 5 of them had a background in computer science.

Four dummy persons were applied to a project. The employed graph consisted of nine nodes and ten edges and had seven different edit times of the nodes and edges. This equates to a small, but not too complicated storytelling project. That implicates that the study is limited to a small number of users and a small example project.

The execution of the experiment was divided into two phases. In the first phase the participants had to read basic information about storytelling and what they expect to see in the visualization prototype. Doing so should give them basic knowledge for completing the experiment. In the second phase the participants worked with the visualizationand filled out a digital questionnaire at the same time. They were able to complete the experiment independently from each other. It took on average 30 minutes to complete. The questionnaire consisted of two different sections of tasks and a total number of 14 tasks to the experiment. Every task had different answers that the participants were able to choose. The first 12 tasks are closed questions and the last two tasks used an ordinal rating scale to create a ranking. To answer the four questions mentioned above, the following 14 taskswere explored during the experiment:

1. Which persons work on the story graph?
2. What is Nadine's last name?
3. Which person is the most active in regards to the story graph?
4. Which person is the most inactive in regards to the story graph?

Table1. Overview of the relationship between questions, tasks and requirements

Questions	**Tasks**	**Requirements**
I. Does the visualization make it clear who changed what in the story graph?	5, 9, 10	**R1:** The visualization of the story graph evolution has to show meta-information which allows users to assess who made any changes, and when those changes were made. **R4:** The visualization of the story graph evolution has to allow users to focus on a specific group member.
II. Does the visualization make it clear when something was changed in the story graph?		**R1:** The visualization of the story graph evolution has to show meta-information which allows users to assess who made any changes, and when those changes were made. **R3:** The visualization of the story graph evolution has to allow users to focus on a specific time period. **R5:** The visualization of the story graph evolution has to allow users to focus on the development of a linear story.
III. Does the visualization make it clear who in general works in the story graph? Are the users aware of other users in the project?	2-8, 10-12	**R4:** The visualization of the story graph evolution has to allow users to focus on a specific group member.
IV. Is the user interface easy to handle?	13, 14	-

5. What did Jared do on January, 27th 2010?
6. Which stories were created in the story graph?
7. Which audio files were used in the project?
8. Which person was the last working on the project?
9. At which point in time was „UCLA.mp3 / location → UWSP.mp3 / location" deleted?
10. When was node (circle) „UWSP / education" created and by whom?
11. Which person would you contact, if you had questions regarding the UWSP Interview?
12. With which person would you cooperate, if you want to do something with the story "Food Comparison"?
13. The handling of the story graph visualization was easy to learn
14. I always knew where I am and what to do.

In Table 1 is shown an overview about the relationship between the overall questions, the tasks for the participants in the experiment and the requirements in Section 2.

In the following section the questions and the results of the experiment are compared with each other.

I. Does the visualization make it clear who changed what in the story graph? (R1, R4, R5)

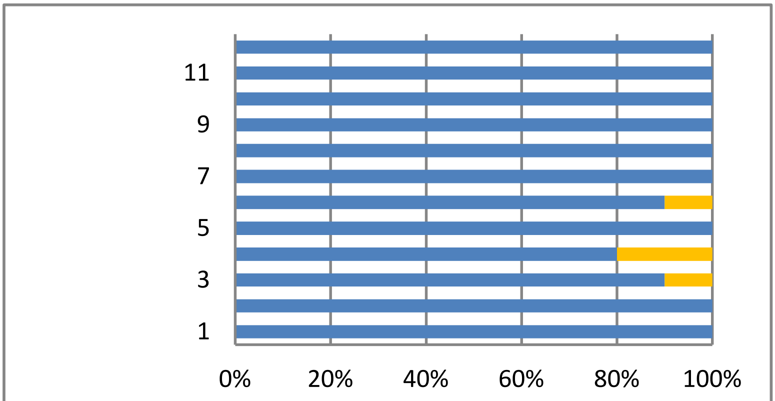
The first 12 tasks, besides task 1 and 9, support question I. All tasks were answered 100% correct, with the exception of task 3, 4 and 6 (Figure 4). Some participants had trouble answeringwhich project member is the most active or inactive related to the story graph. Another participant did not know which stories were created in the project. The issue here might lie in the way the questions were asked. It is possible that there were too many ways to interpret the question. However, most participants were able to solve the tasks. This implies that the visualization of the story graph evolution provides the needed functionality to emphasize the *"what"*-category and the *"who"*-category of workspace awareness and therefore supports our hypothesis.

II. Does the visualization make it clear when something was changed in the story graph? (R1, R3, R5)

The tasks 5, 9 and 10 were answered 100% right by the participants (Figure 4). This implies that the participants were aware when changes to the story graph were made (*"when"*-category of workspace awareness). This also supports our hypothesis.

III. Does the visualization make it clear who in general works in the story graph? Are the users aware of other users in the project? (R4)

Task 1 was answered 100% correct by the participants (Figure 4). This implies that the participants know which people worked on the story graph and in this way they are aware of the other project members. In this way the workspace awareness in group projects in the web portal is advanced. Additionally it empowers project members to be aware of other project members. This, like the two prior results, supports our hypothesis.



	1	2	3	4	5	6	7	8	9	10	11	12
right answers	10	10	9	8	10	9	10	10	10	10	10	10
wrong answers	0	0	1	2	0	1	0	0	0	0	0	0

Fig. 4. Answers of the participants to items 1-12

IV. Is the user interface easy to handle?

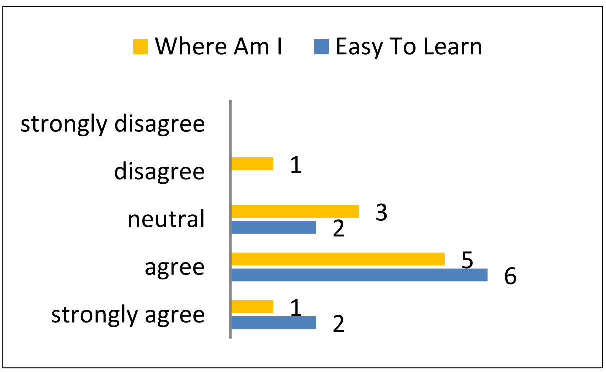


Fig. 5. Answers of the participants to items 13 and 14

The result of the last two tasks of the questionnaire is shown in Figure 5. The participants rated both tasks on average "agree", which means that the interface and the usability is good, but that there is still room for improvement. A possible explanation for the high variance of the rating is the usage of participants to such applications. The people who had no computer science background rated on average lower than the participants with higher computer science background.

In general it was shown that the visualization prototype provides the functionality to show the user all workspace related information he needs to get an overview about what happened in the past.

The application advances the workspace awareness within the CASTing web portal by showing who edited what artifact in the story graph and when it was edited. It was also shown that the visualization makes users aware of other users within the story graph and the project. This fact and the easy learning of the story graph handling enables users to track the activities of other users and empower users to question and understand the work results of other users.

The experiment has also shown that the visualization contains all the information that was only textually available before. Thus the main goal of the experiment has been achieved.

During the experiment limitations of the prototype were also discovered, which is the basis for improvements in the future. Users were not able to see the entire graph while looking at detailed information. An overview map could be used to tackle this issue. In addition, users were not able to access the audio files represented by the nodes.While the story graph visualization provided awareness regarding changes to the graph itself, awareness support for the evolution of the underlying content is still missing. This is crucial to understand how stories and the meaning they contain evolve over time. Further prospect improvements and extensions for the visualization prototype are discussed in the next section.

7 Conclusion/Future Work

Asynchronously growing story graphs make it difficult for users to trace the evolution of these graphs. Based on the concept of workspace awareness we identified requirements and presented a prototype visualizing the story graph evolution.
Furthermore, we evaluated different awareness factors in an experiment. The experiment showed that our visualization approach enables users to acquire workspace awareness by accessing information about previous work of other users and to follow the development process of the story graph.

The developed visualization could be used to explore its impact on the collaboration between users, e.g.: Does a user need less time to introduce themselves to the project? How big is the difference of workspace awareness in a storytelling web portal with and without the visualization of the story graph evolution? Does the visualization lead to a better coordination and communication between project members? Can a simplification of communication be achieved? How effective is that kind of visualization with much more displayed nodes and edges?

References

1. Totzke, R.: Erinnern – Erzaehlen – Wissen: Was haben (Erfahrungs-) Geschichten mit echtem Wissen zu tun? In: Reinmann, G. (ed.) Erfahrungswissen erzaehlbar machen. Pabst Science Publishers, Lengerich (2005)
2. Philips, D.: Talking books: The encounter of literature and technology in the audio book. Convergence 13(3), 293–306 (2007)

3. Friederichs, T., Hass, B.H.: Der Markt für Hörbücher. Eine Analyse klassischer und neuer Distributionsformen. MedienWirtschaft. Zeitschrift für Medienmanagement und Kommunikationsökonomie 3, 22–35 (2006)
4. Hein, G., Jakuska, R.: Podcast industry. iLabs – Center for Innovation Research. School of Management, University of Michigan-Dearborn (April 2007)
5. Martens, D., Amann, R.: Internetnutzung zwischen Pragmatismus und YouTube-Euphorie. Media Perspektiven 11, 538–551 (2007)
6. Lukosch, S., Klebl, M., Buttler, T.: Facilitating Audio-based Collaborative Storytelling for Informal Knowledge Management. In: Briggs, R.O., Antunes, P., de Vreede, G.-J., Read, A.S. (eds.) CRIWG 2008. LNCS, vol. 5411, pp. 289–304. Springer, Heidelberg (2008)
7. Wikipedia contributors: Wiki. Wikipedia, The Free Encyclopedia (February 2010), `http://en.wikipedia.org/w/index.php?title=Wiki&oldid=369908312`
8. Spaniol, M., Klamma, R., Sharda, N., Jarke, M.: Web-based learning with non-linearmultimedia stories. In: Liu, W., Li, Q., Lau, R. (eds.) ICWL 2006. LNCS, vol. 4181, pp. 249–263. Springer, Heidelberg (2006)
9. Pankoke-Babatz, U., Prinz, W., Schaefer, L.: Stories about Asynchronous Awareness. In: Darses, F., Dieng, R., Simone, C., Zacklad, M. (eds.) Cooperative Systems Design - Scenario-Based Design of Collaborative Systems, pp. 23–38. IOS Press, Amsterdam (2004)
10. Dourish, P., Belotti, V.: Awareness and Coordination in Shared Workspaces. In: Computer Supported Cooperative Work, Proceedings of the 1992 ACM conference on Computer-supported cooperative work, pp. 107–114. ACM Press, New York (1992)
11. Gutwin, C., Greenberg, S., Roseman, M.: Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. In: Proceedings of HCI on People and Computers XI, London, pp. 281–298 (1996)
12. Gutwin, C., Greenberg, S.: A framework of awareness for small groups in shared-workspace groupware. Technical Report 99-1, Department of Computer Science, University of Saskatchewan, Canada (1999)
13. Erickson, T., Halverson, C., Kellogg, W.A., Laff, M.: Social translucence: designing social infrastructures that make collective activity visible. Communications of the ACM 45(4), 40–44 (2002)
14. Carroll, J.M., Neale, D.C., Isenhour, P.L., Rosson, M.B., McCrickard, D.S.: Notification and awareness: synchronizing task-oriented collaborative activity. International Journal of Human-Computer Studies 58(5), 605–632 (2003)
15. Ganoe, C.H., Convertino, G., Carroll, J.M.: The BRIDGE awareness workspace: tools supporting activity awareness for collaborative project work. In: Proceedings of the third Nordic conference on Human-computer interaction, pp. 453–454. ACM Press, New York (2004)
16. Ganoe, C.H., Somervell, J.P., Neale, D.C., Isenhour, P.I., Carroll, J.M., Rosson, M.B., McCrickard, D.S.: Classroom BRIDGE: using collaborative public and desktop timelines to support activity awareness. In: Proceedings of the 16th annual ACM symposium on User interface software and technology, pp. 21–30. ACM Press, New York (2003)
17. Lukosch, S.: Flexible and transparent data sharing for synchronous groupware. International Journal of Computer Applications in Technology, Special Issue on Current Approaches for Groupware Design, Implementation and Evaluation 19, 215–230 (2004)
18. Lukosch, S.: Transparent Latecomer Support for Synchronous Groupware. In: Favela, J., Decouchant, D. (eds.) CRIWG 2003. LNCS, vol. 2806, pp. 26–41. Springer, Heidelberg (2003)

19. Lauwers, J.C., Lantz, K.A.: Collaboration awareness in support of collaboration transparency: requirements for the next generation of shared window systems. In: Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, pp. 303–311. ACM Press, New York (1990)
20. Facebook Developers: Activity Feed (March 2010), http://developers.facebook.com/docs/reference/plugins/activity
21. Perret, R., Borges, M.R., Santoro, F.M.: Applying group storytelling in knowledge management. In: de Vreede, G.-J., Guerrero, L.A., Marín Raventós, G. (eds.) CRIWG 2004. LNCS, vol. 3198, pp. 34–41. Springer, Heidelberg (2004)
22. Acosta, C., Collazos, C., Guerrero, L., Pino, J., Neyem, H., Motele, O.: StoryMapper: A Multimedia Tool to Externalize Knowledge. In: Proceedings of the XXIV Conference of the Chilean Computer Science Society, pp. 133–140. IEEE CS Press, Los Alamitos (2004)
23. Schäfer, L., Valle, C., Prinz, W.: Group storytelling for team awareness and entertainment. In: NordiCHI 2004: Proceedings of the third Nordic conference on Human-computer interaction, pp. 441–444. ACM Press, New York (2004)
24. Bentley, R., Horstmann, T., Sikkel, K., Trevor, J.: Supporting collaborativeinformation sharing with the World Wide Web: The BSCW Shared Workspace system. In: Proceedings of the 4th International World Wide Web Conference, pp. 63–74. O'Reilley& Associates, Boston (1995)
25. VoiceThread (March 2010), http://voicethread.com/
26. Birdeye – Information Visualization and Visual Analytics Library (October 2009), http://code.google.com/p/birdeye/
27. Adobe Flex (December 2009), http://www.adobe.com/de/products/flex/

Dealing with Device Collaboration Rules for the PCSCW Model

Kahina Hamadache and Luigi Lancieri

Orange Labs, 42 rue des Coutures, 14000 Caen, France
LIFL, University Lille 1, Lille, France
kahina.hamadache@orange-ftgroup.com
luigi.lancieri@univ-lille1.fr

Abstract. In this paper, we describe the design, the development and the use of devices collaboration rules for the PCSCW (Pervasive Computing Supported Collaborative Work) Model. These rules rely on the precise description of roles, tasks, actions, resources required by these actions and constraints associated to these resources to select the proper way to make devices cooperate with the final objective to facilitate the collaboration of humans. We suggest that by defining constraints on resources as triplets composed of a parameter, a value and an associated criticality it allows us to quantify, estimate, compare and then choose between several candidate rules. The finality given by these rules is a simple but efficient way to make devices choose automatically the most appropriate way to cooperate.

Keywords: Pervasive Computing, Collaborative Work, Constraints Modelling, PCSCW, Collaboration rules.

1 Introduction

The computer supported collaborative work (CSCW) domain is probably one of the most active research fields of recent years. Indeed, due to the facilitations brought by computers and smart devices it is almost impossible to find people working without them. For the past few years information technologies are evolving toward the multiplication of smart electronic devices such as smartphones, laptops, GPS and so on. Despite or maybe because of this proliferation the digital environment is a non-continuous space where miscellaneous devices can communicate, or not, with others. Thus, in order to make this space "continuous" the Pervasive Computing [7] is based on the communication between devices to smartly adapt their behaviour to the current context of users and offer them a seamless interaction with the digital world.

Given this aspect our work has rapidly focused on the way we could integrate the pervasive computing within CSCW. Such integration could bring various advantages: resource and time saving for companies, work simplification and task automation for workers. In a "green" consideration it could also help reducing work's energetic impact by accompanying users' in using lighter devices and services.

On the long road toward this accomplishment we have already sowed some seeds. Hence as we will describe in the next sections we have proposed the *PCSCW* model

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 233–248, 2010.

(see below) which is designed to improve the integration of collaborative work aspects with pervasive computing and make them benefit from each other, we propose a model that allows describing users with their roles, tasks, actions and resources required to perform them. Then by comparing required resources to available ones we can trigger device cooperation routines to facilitate the collaboration of users.

All devices don't natively support collaboration with others. In order to solve this kind of issue we argue that the definition of device collaboration rules could be of great help. These rules intend to define tasks that could be automatically performed by devices to collaborate in order to allow a user to do its own task. The main idea behind this is the following: a user needs two (or more) resources to complete an action related to a task; these resources are not available on a single device, but the combination of several of them can supply the resources. Thus, device collaboration rules define what actions can be performed by devices to collaborate, for finally providing required resources to the user. These collaborations can be of various kinds: network access sharing, heavy computing task delegation and notification of events, anything you can imagine to make several devices cooperate.

In this paper we will go down in the depths of these devices collaboration rules; we will present how these collaboration rules are designed and how we can use them.

This paper is organized as follows: we will first introduce the basic concepts of the PCSCW Model to be able in a second time to efficiently describe device collaboration rules we're using with. Third, to illustrate our work we propose a use case based on simple scenarios melting collaboration of users with cooperation of machines. The fourth section is dedicated to present a way to implement and use the PCSCW model and its collaboration rules. Next to the last we present a concise state of the art on pervasive computing modelling and reasoning. Finally we conclude by giving some perspectives of our future work.

2 PCSCW Model

As our research interest has focused on the integration of the pervasive computing aspect in the computer supported collaborative work, we have proposed in a previous work [1] an original model which aims at making smart devices cooperate seamlessly to improve and facilitate the collaboration of users. This model, named PCSCW for Pervasive Computing Supported Collaborative Work, relies on some simple but essential "sub-models":

- A *Task Model* composed of mainly two concepts:
  - *Task*: represents a meaningful process to be performed by one or more users to achieve a specific goal, for instance "creating a webpage", can be composed of a set of subtasks or actions;
  - *Action*: describes an atomic step of a task, it has no discriminatory meaning as it can't be understood outside of a task. To illustrate it we can consider the action "opening a web browser" that doesn't convey any specific meaning but can be integrated in tasks such as "searching the web" or "checking mails".

- A *Role Model* built above the task model, it extends it by providing one more concept and some refinements about tasks:
  - *Role*: it defines a role to be played by one or more users by wrapping tasks into subsets: mandatory, allowed and forbidden tasks;
  - Tasks can require one or more roles to be performed. Thus a single task may be shared among several roles and then becomes a "Collaborative Task".
- A *Resource Model* providing a common ground to represent:
  - *Required resources*: the set of resources an action requires to be performed. By describing these requirements in term of software, hardware, human and social resources at the action level we can efficiently describe resources required for a given task;
  - *Available resources*: the set of resources available in user's environment, it provides a structured representation of the context;
  - *Device*: the representation of a contextual device is merely a part of available resources but with a particular extent as it is considered as an active agent of the collaboration.

In addition to these sub-models the PCSCW Model includes *Device Collaboration Rules*. The main principle of these rules is the following: by comparison of resources required to perform a task or an action with available devices resources we can trigger specific interactions between devices to make them cooperate to finally provide all required resources to the user. Going a little further these rules can even perform whole actions or tasks and prevent users from doing repetitive and thoughtless ones.

All these features create a model allowing smart devices of users' context to automatically and seamlessly cooperate to facilitate, channel and enhance the collaboration of users.

3 Device Collaboration Rules

As it is our main focus for this article we will make an in-depth investigation of the devices collaboration rules, we will see how they are designed and how we can use them.

3.1 Rules

Making two or more devices collaborate doesn't only rely on resource matching; indeed you need to have defined a set of behaviours to trigger when user's context matches some rules requirements. To be coherent with its main principle, rule behaviour contains actions to be performed with the description of their associated resources. Indeed, each device collaboration rule is defined with the following syntax (1):

$$\text{IF } (context.resources \equiv rule.resources) \text{ THEN DO } rule.behavior \tag{1}$$

Obviously, several rules can have a similar or partially equivalent set of required resources, it implies that more than one rule can be matched by the current context and lead to some kind of conflict. Besides we need to express a specific need here: all device collaboration rules must have the same knowledge for their reasoning, it implies that context representation has to be "locked" during the reasoning process. Then, to be able to select the adequate rule to trigger we need a tool to evaluate their relative suitability. To fill this requirement and as we'll see in the next section we propose to define constraints on rules' resources.

3.2 Constraints

The design of Device Collaboration rules for the PCSCW already requires describing roles, tasks, actions and resources. These resources can be of various kinds mainly categorized along hardware and software ones. In order to complete this design we need to be able to express constraints over the required resources.

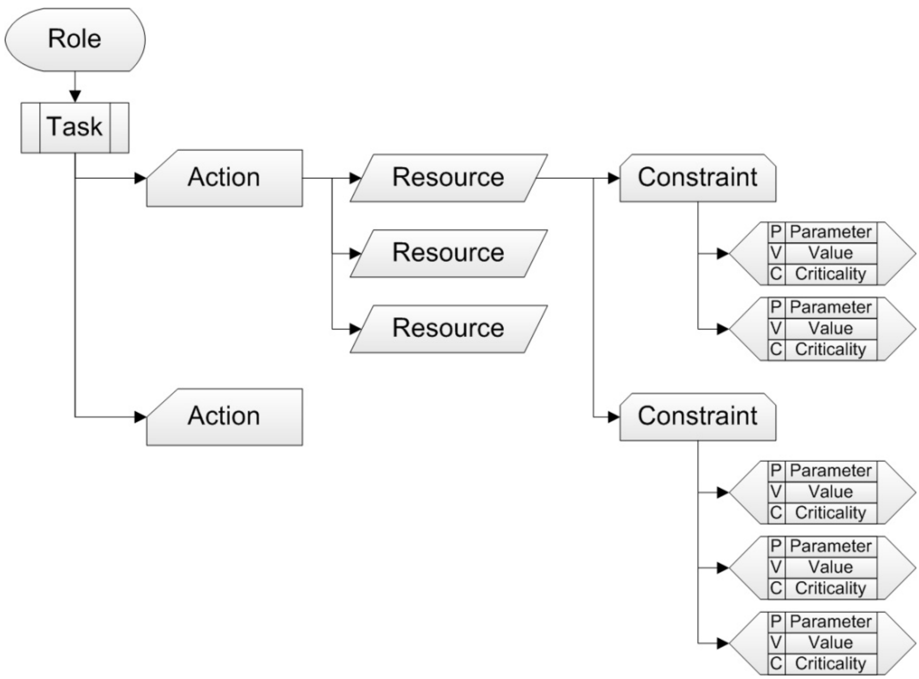


Fig. 1. Constraints for the PCSCW Model

Fig. 1 defines the addition of constraints on resources of the PCSCW Model. As it is depicted, a single resource may be related to several constraints, each of whom is described by a triplet {P, V, C} as: *P* a parameter which represents the precise point to be evaluated, *V* the expected (or required) value (or threshold) for this parameter and *C* the criticality of this parameter. This last component of a constraint has a specific impact as it is the one allowing a device to select the appropriate collaboration rule.

In addition to this triplet we propose to organize constraints in five main categories, facilitating and guiding rules designer in their work: *Availability, Cost, Privacy, Reliability and Security*.

Examples of parameters falling below these categories could be: CPU Load (Availability), connection price (Cost), data access (Privacy), website breakdown relative frequency (Reliability) and network secured protocol (Security). Given that values are related to parameters and are illustrated in the following use case, we won't give more examples of them.

While parameters and values are easily collectible from real use case, the criticality needs some more analysis and requires defining its own set of values. In this perspective we propose to use a really simple seven-level scale: *Optional*, *Very Low*, *Low*, *Average*, *High*, *Very High* and *Mandatory*.

Optional indicates the constraints doesn't need to be fulfilled but can provide a valuable benefit for the collaboration and can help choosing between two equivalent rules. On the contrary, the Mandatory level implies that if the constraint is not met the collaboration rule cannot be used in the current context.

3.3 Using Device Collaboration Rules

Until now we have described all required concepts to understand the PCSCW model. Let's have a look at the real use of all these descriptive levels and how the model helps at finding the right cooperative behaviour.

To formalize and facilitate the use of collaborations rules we have defined a six-step process describing how a specific rule can be triggered during the collaboration:

1. On context data update, an analysis of this update is automatically started;
2. If this analysis points out that some device collaboration rules may eventually improve the current collaboration by facilitating the accomplishment of an action we start the comparison between context information and rules activation requirements;
3. This comparison can end in three ways:
   a. No rule can effectively improve the collaboration in the current state of the context, we stop the process here;
   b. One rule can improve the collaboration, in this case we jump directly to step 6;
   c. Several rules can improve the collaboration, in this case we need to choose between them the most relevant and efficient, we go to step 4;
4. To choose between the selected rules we need to compare their suitability, their relative efficiency. To do it we confront action's required resources and their associated constraints with resources supplied and used in each rule' behaviour.
5. From this confrontation we bring out a score for each rule and all we have to do is to keep the rule with the higher score. In the case where several rules have the same score it means that none of them can be "automatically" preferred and the device has to take one of them arbitrary.
6. Run the chosen behaviour.

Hence to compare two selected rules we need to quantify each of them according to resources and constraints. In fact, the way we have defined constraints facilitates this comparison by relying on the quantification of criticality and the evaluation of the constraint fulfilment of each rule.

Table 1. Rule Comparison Matrix

Resource	Res_1		Res_2	…	Res_N			*Suitability*
Constraint	$C_{1,1}$	$C_{1,2}$	$C_{2,1}$	…	$C_{N,1}$	…	$C_{N,M}$	
Rule R_A	$V_{A,1,1}$	$V_{A,1,2}$	$V_{A,2,1}$	…	$V_{A,N,1}$	…	$V_{A,N,M}$	$S_A = \sum V_{A,I,J}$
Rule R_B	$V_{B,1,1}$	$V_{B,1,2}$	$V_{B,2,1}$	…	$V_{B,N,1}$	…	$V_{B,N,M}$	$S_B = \sum V_{B,I,J}$
…								…
Rule R_X	$V_{X,1,1}$	$V_{X,1,2}$	$V_{X,2,1}$	…	$V_{X,N,1}$	…	$V_{X,N,M}$	$S_x = \sum V_{X,I,J}$

Table 1 depicts the rule comparison process. On this matrix, each rule to be compared is evaluated along with its provided resources and their constraints. Thus, for each constraint of each resource we assign a value ($V_{r,i,j}$) which is limited by the criticality of the constraint evaluated ($V_{r,i,j} \leq Crit(C_{i,j})$). As for now we have decided to use a simple system: a value is comprised between 0 and 5, a *very low* criticality means 1, *low* means 2 and so on until *very high* which means 5. As we already evoked, an unsatisfied mandatory constraint eliminate the rule, while an optional one can only be used to decide between two equivalently suitable rules. Then if we consider a constraint with a *high* criticality the evaluation of this constraint can't be higher than 4. We know that the assignment of values can be sometimes problematic if the threshold value that was first defined in the constraint is not easily comparable. Thus, if we consider constraints such as *data encryption* it can be hard to give a value to a encryption method different from the one that was defined. Still, there are solutions to this kind of problem, for instance we can use predefined rankings.

Limiting rules' assigned values with the associated criticality and relying on constraints *Values* prevent from selecting a rule that does not satisfy most of the critical constraints but tremendously outperform a minor one. Thus, even if network bandwidth constraint has been defined with a *low* criticality and a value which has to be at least 0,5mbps, the rule allowing a ultra high speed connection faster than 100mpbs but poorly satisfying other constraints will certainly not be selected (except for the case where other rules are worse) as its connection can give it more than 2 "points".

Finally we obtain the suitability level of each rule by adding up all assigned values and we are able to compare rules and find the most appropriate by comparing their suitability.

4 Use Case

The description of the model and device collaboration rules' process is now complete. In order to illustrate their use and benefits we propose a use case based on a simple collaboration between 3 coworkers.

Leela, Amy and Philip are members of a team and have to collaborate on a new marketing campaign for the new product of their company. In this perspective they have to perform several tasks together. Let's suppose that they have to make a brainstorming session to design a new advertising board. Amy is working at their main office, but Leela and Philip are not physically present. Leela is working at her home while Philip is in mission in Kenya. In order to be able to work at the same time Amy has sent invitations to Leela and Philip for a virtual Brainstorming with a dedicated software at 3 PM (GMT). In a "device consideration" Amy is working on her usual workstation, Leela has its personal laptop, Philip on his side has a tablet-pc and a smartphone. At 3 Amy has started the server part of the application and has connected her station. At the same time Leela's laptop and Philip's tablet-pc need to connect to the Internet in order to be able to join the Brainstorming platform. To do it they rely on the PCSCW model that should allow their devices to make the right decision.

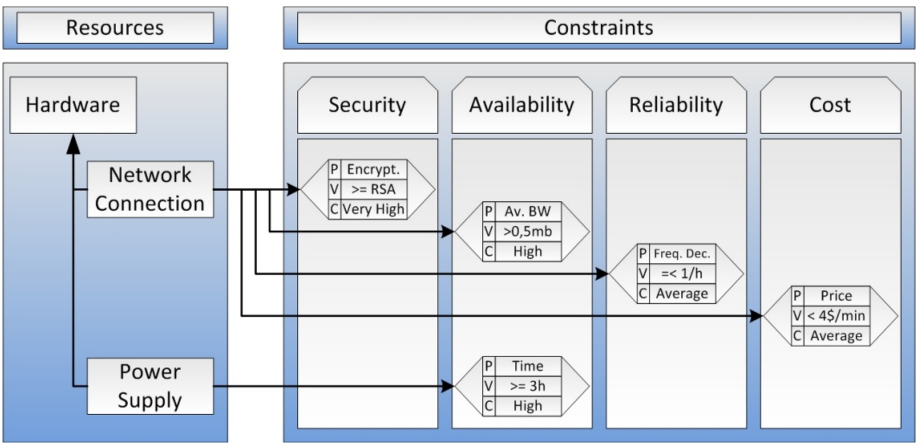


Fig. 2. Internet Connection Constraints

The task associated with the brainstorming activity described with the PCSCW implies several constraints on the resources used by the devices. **Fig. 2** defines and describes the set of constraints associated with the "Connect to Internet" action. On the left side we've got the set of resources required to perform the action and on the right their constraints. For this specific action there are two resources: a network connection and a power supply. For the network connection we've got:

- 1 security constraint: the encryption has to be at least RSA [4]; this constraint has a Very High criticality as the collaboration taking place is close to confidential;
- 1 availability constraint: the average provided bandwidth has to be at least 0,5mbps, this constraint has a High criticality as the application can work with less bandwidth but user's satisfaction and experience may be dramatically lowered by such limitation;

- 1 reliability constraint: the probability to experience network disconnections has to be less than 1 per hour. As this point doesn't completely stop the collaboration it has an Average criticality;
- 1 cost constraint: the price of the connection has to be less than four dollar a minute. As it doesn't obstruct the collaboration this constraint has an Average criticality.

As for the network connection we also have a constraint on the power supply resource:

- 1 availability constraint: the energy supplied has to be sufficient to maintain the connection for three hours in order to have enough time for the brainstorming session. This constraint has a High criticality.

Leela's laptop hasn't many choices and connects itself to the wifi access point of Leela's home's ADSL modem.

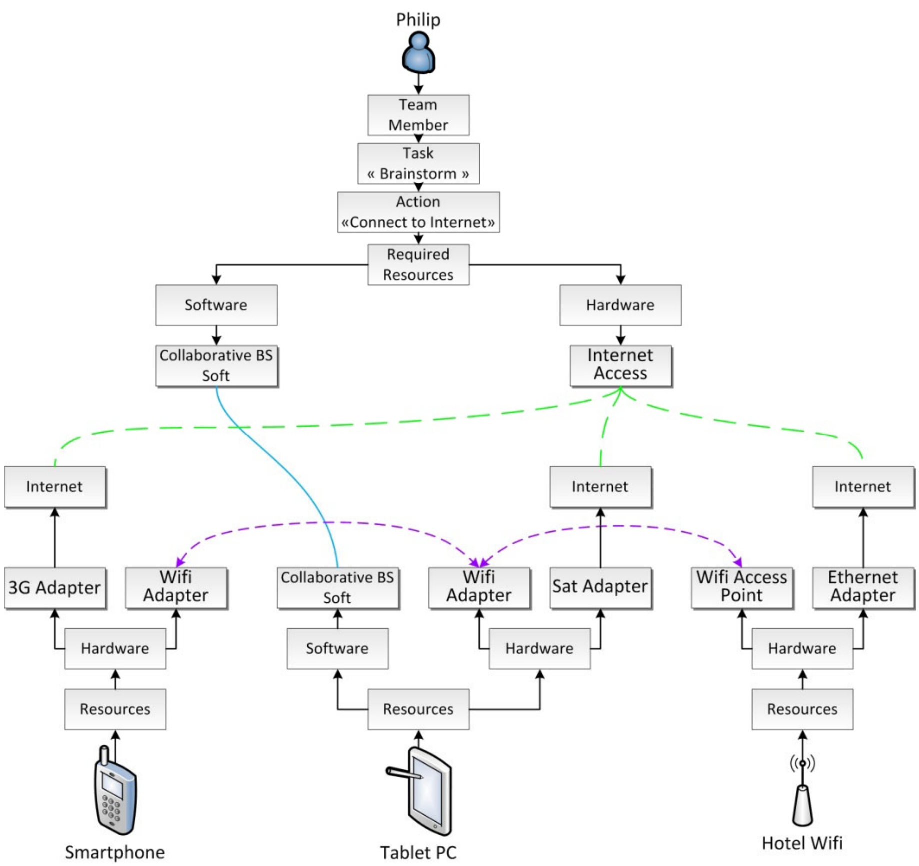


Fig. 3. Philip's Digital Environment

As depicted by Fig. 3 Philip's situation is totally different. In addition to the tablet-pc, the smartphone and the hotel wifi access point, we've got a description of resources required for the connection to Internet. It also depicts the three possible scenarios to establish the Internet connection:

- Direct connection of the tablet-pc through its satellite network adapter;
- Connection of the tablet with hotel's wifi access point;
- Connection of the tablet with Philip's smartphone with a connection bridge between cellphone's wifi and 3G networks to allow the tablet to access to Internet.

Each one of these possibilities has advantages and drawbacks:

- Direct connection with satellite network:
  - Advantages: highly secured, only rely on tablet's energy, relatively stable;
  - Drawbacks: slow connection (~0,2mbs) and costly, occasional disconnections;
- Connection to hotel's wifi:
  - Advantages: good bandwidth(~2mbps), free, low energy consuming;
  - Drawbacks: poorly secured (WPA), variable bandwidth, disconnections every fifteen minutes;
- Connection with smartphone:
  - Advantages: as we use ad-hoc wifi the security is up to the two devices and can be relatively good, the average bandwidth is fair (~1 mbps) and the connection is relatively stable;
  - Drawbacks: power supply is limited by smartphone's battery life which is limited to 2.8 hours due to the high energy consumption of the 3G and wifi adapters.

We consider that Philip's tablet has already acquired all these information; he must now find the best solution. This is simply realized by analyzing solutions constraints fulfillments such as displayed by Table 2.

Table 2. Comparison of Connections

	Network Connection				*Power Supply*	*"Score"*
	Security	*Availability*	*Reliability*	*Cost*	*Availability*	
Satellite	5	2	2	0	4	13
Hotel Wifi	1	4	0	3	4	12
Smartphone	4	4	3	2	2	15

In our case, even if the score are relatively close, the smartphone scenario offers more advantages than others as it combine a good security, availability and reliability for a limited cost. For the battery life, as it is not a mandatory constraint and given that the smartphone can maintain the connection during more than 90% of the desired time with the eventual possibility for Philip to simply plug its charger, it offers a good compromise.

Finally, after having evaluated the situation, Philip's tablet decides to establish a wifi ad-hoc connection with his smartphone and create a bridge between the 3G and wifi connections. Philip can now connect to the brainstorming platform and collaborate with Amy and Leela.

5 Developing Device Collaboration Rules

As we have described all concepts of the PCSCW model and have illustrated it through a use case, the last point we'd like to present here is a way to implement and use the PCSCW model and its collaboration rules.

In this perspective we needed a way to simply and efficiently represent all concepts of the model and reason on it. From our past experiences ([2], [3]) on representing context information and modeling reasoning rules we decided to use the combination of OWL [5] and SWRL [6].

The Web Ontology Language (OWL) is a language for defining and instantiating ontologies, and it can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. Web Ontology Language was adopted as the recommendation by W3C in 2004. OWL provides the required elements to represent and use complex information models. The language itself is an extension of the RDFS [9] language and provides additional features for a greater expressiveness. OWL can be used to define classes and properties and also provides constructs to create new class descriptions as logical combinations (intersections, unions, or complements) of other classes, define cardinality restrictions on properties and so on.

The Semantic Web Rule Language (SWRL) is based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML [11] sublanguages of the Rule Markup Language. It proposes the specification of rules in the form of an implication. The Atoms within the body and the head of the implication can be Class Predicates C(x) or Property Predicates P(x, y). Within the body or head, multiple atoms are treated as a conjunction. In order to use and manipulate the OWL and SWRL languages we rely on the Protégé [8] framework which is currently developed and maintained at the Stanford University. The reasoning itself is ensured through the Jess [10] rule engine which can be used within Protégé.

In a first time, let's see how the basic elements of the model are represented, then how collaboration rules are built on it.

5.1 Roles, Tasks, Actions, Resources and Constraints

The first step to use the basic elements of the PCSCW Model is to represent each of them and their relations. Fig. 4 provides a view of the implementation of the PCSCW Model that has been made with the OWL language.

On the top of the figure we can see the representation of an *Agent* as an OWL Class. As depicted an *agent* can be a *human* or a *device*. These *agents* can form groups by associating themselves to accomplish a specific *Task*.

Between the concepts of *Agent* (and *Groups*) and *tasks* we've got the *Role* concept. *Agents* and *Groups* are linked to *Roles* by the "*plays*" object property. Obviously *agents* and *groups* can be related with several *roles*.

Considering links between *roles* and *tasks* there are 3 different object properties involved: *mandatory*, *allowed* and *forbidden*.

The representation of *tasks* is relatively simple: a *task* may be composed of several *actions* or/and sub*tasks*.

Actions are mainly composed of requirements: they require *resources* and may also require the completion of some other *actions*. To organize them inside *tasks*, *actions* have a relative *order* as datatype property.

At this point we get back in the resource model by the description of *resources*. A *resource* can be required by an *action* or provided by an *agent*. A *resource* can also be *composed of* other *resources* and can naturally have some specific datatype properties. *Resources* can mainly be of three types: *human resources*, *device resources* and *other data* (such as current time, number of inhabitants in Springfield, all information that can't directly be related to an *agent*).

Finally, *resources* may have *constraints* that contain three datatype properties: *parameter, value* and *criticality.*

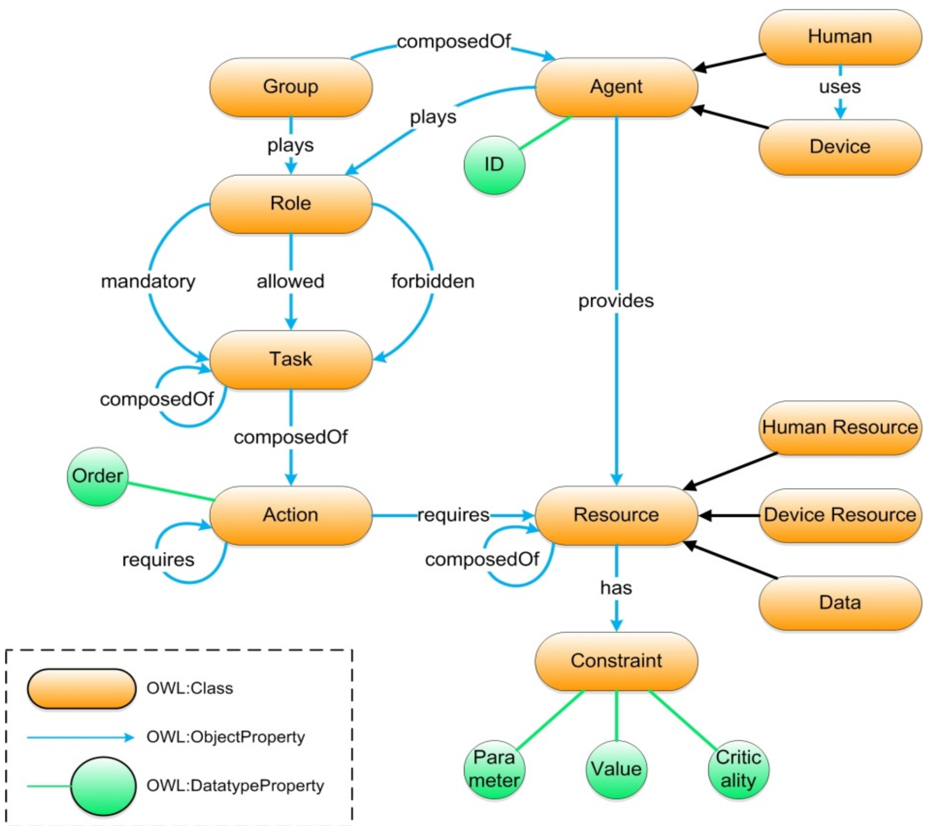


Fig. 4. OWL representation of the PCSCW Model

We have presented how the PCSCW model is mapped to an ontology with the OWL language. Hence the next step is to see how collaboration rules are represented.

5.2 Collaboration Rules

As it has been already evoked we have decided to use the SWRL language to represent device collaboration rules for its integration with OWL and its efficiency.

If we refer to (1) we know that devices collaboration rules are basically a comparison of required resources against available ones and a resulting behaviour. Then, as SWRL Rules are composed of an antecedent and a consequent, each of whom is containing a set of atoms the natural way to represent collaboration is to define required resources as the antecedent and resulting behaviour as the consequent.

$$\mathrm{RES}_{r1}(r_{r1}?) \wedge \ldots\ \mathrm{RES}_{rn}(r_{rn}?) \wedge \mathrm{ACT}_{r1}(a_{r1}?) \ldots \mathrm{ACT}_{ro}(a_{ro}?)$$

$$\Rightarrow \tag{2}$$

$$\mathrm{RES}_{i1}(r_{i1}?) \wedge \ldots\ \mathrm{RES}_{ip}(r_{ip}?) \wedge \mathrm{ACT}_{i1}(a_{i1}?) \ldots \mathrm{ACT}_{it}(a_{it}?)$$

Formula (2) summarizes the expression of collaboration in SWRL: the antecedent contains a set of resources and a set of actions representing the requirements while the consequent stores actions with their relative resources representing the behaviour of the rule.

During the definition of device collaboration rules we mentioned the fact that all device collaboration rules must have the same set of context information. To fulfil this requirement we rely on internal mechanisms of Protégé which are able to load knowledge inside a “Rule Engine Bridge” which has in charge to make the interface between the core of Protégé and the current rule engine. As we load context information only once for all rules (and each time we need to reason) it ensures that all rules use the same set of knowledge.

6 Related Works

We have presented our work on pervasive computing supported collaborative work. Obviously it hasn’t come out of nowhere and there are numerous researches that inspired us. In this section we present some of these works related to the context modelling and reasoning.

In [12] the authors present a rough set based methodology [16] to generate the appropriate minimal set of design rules for the Ubiquitous Smart Device (USD) design collaboration. They point out the fact that the USD design can be semantically represented in ontology, however, the computational complexity of semantic reasoning is a very sophisticated and time consuming task. They proposed in future work to compare the manually defined SWRL rules and inducted rules to validate their rough set framework.

In [13] the authors propose an ontology-based generic context management model; their model facilitates context reasoning by providing structure for contexts, rules and their semantics. Rules are derived or defined by users based on the requirements and policies of a specific application domain. Their underlying context model has been developed using the RDF and OWL languages. Obviously this model is meant to be extended and enriched by domain specific information according to its use and the requirements of the system. Reasoning rules are composed of two distinct sets: rules based on the ontology itself and user defined rules. Ontology based rules are those concerning the standard features of OWL such as "inverseProperty", transitivity of properties and so on. Users' defined can be of any type and express semantic implications in the ontology. This work is particularly interesting as it tries to give a generic model and a way to reason over it.

In a previous work [3] we have proposed a context modeling for communication services based on ontology. This ontology is enhanced into an active model by providing it a rule engine and a set of inference rules. We have used the SWRL rule language to reason on the context model. This mechanism consists in two specific phases. First, in the context model, we have defined a property named "hasAssociatedRule" which domain is Context_Behavior and range is swrl:imp. The class swrl:imp is the one that represents the SWRL Rules. In our ontology rules are directly associated to the classes by the owl restriction owl:hasValue. The second phase is the selection of the rules. This is quite simple since we have the set of rules associated with the classes of the behavior. This preceding work was far simpler than our current one but it gave us a good opportunity to evaluate the feasibility of such reasoning mechanisms and the efficiency of OWL and SWRL.

In [14] the authors propose Smart Device Collaboration for ubiquitous computing environment, which aims to establish the collaboration between portable devices and embedded computers, while realizing the basic function of portable devices and also applying the maximum advantages of embedded computers.

In [15] the authors propose an OWL context ontology (CANON) for modelling context in pervasive computing environment, and for supporting logic-based context reasoning. CANON provides an upper context ontology that captures general concepts about basic context, and also provides extensibility for adding domain-specific ontology in a hierarchical manner. They also studied and implemented the use of logic reasoning to check the consistency of context information, and to reason over low-level.

All these works are interesting, but we have to notice that we found extremely few researches dealing with the modelling of collaboration in a pervasive computing environment.

7 Discussion

In this work we have presented the real engine of our PCSCW Model. Indeed, the model itself is an essential part for our research toward the complete integration of Pervasive Computing within the computer supported collaborative work, it provides solid foundations to represent context information, should they be humans, devices, resources, roles, etc. Still, without reasoning it loses most of its interest. Thus, by completing it with device collaboration rules we dramatically increased its potential and usefulness. Indeed, more than an efficient structure, we've got a full process to decide how devices should cooperate to help users collaborate. Besides, this decision is achieved with no other features than the model, the rules and comparisons with current context information.

Even if devices collaboration rules are efficient, we already have some enhancement trails we'd like to explore. For instance, the description of constraints relies on the definition of criticality levels. We know the actual levels are sufficient for our needs, but we need to evaluate if more levels could bring a better accuracy for rules' selection.

In the perspective of providing a better adaptation to current context we also want to find out if there may be constraints on other elements of the PCSCW model. For this trail we'd like to consider if defining constraints directly on actions and tasks could give relevant information and rules selection refinements.

Until now we have set several theoretical bases, and we are currently focusing on the finalisation of the development of a simulator for pervasive computing collaborative work. This tool will assist us to validate and evaluate our model. Thus it should tell us if our model is actually working and if it really improves the collaboration of users. Besides, in a near future it could help us design collaboration rules by providing a simple way to evaluate and compare them.

Another issue considering the adaptation of devices behaviour to their current context is their possible engagement in another activity. Indeed if the evolution of user's context brings out the necessity to provide him specific resources for the accomplishment of his tasks, it may involve the modification of current devices behaviours and then create a conflict between two concurrent actions. We think this problem is naturally tackled by the PCSCW model as it is used to represent the current state of the context. In this perspective resources currently in use would not be available for the adaptation to context changes. A smarter way to manage potential conflicts it to allow the interruption of some part of current device behaviour in order to allow the accomplishment of a more important action. Such perspective seems not to be too complex as it could possibly be managed through reasoning rules.

A known challenge of our device collaboration rules is their elaboration. Indeed, even if the base principle is relatively simple and writing them with SWRL isn't really complex, the real difficulty is to design small, generic but still efficient rules. For this complex aspect of the life of our model we have planned to analyze real collaboration situations represented with the PCSCW Model and extract those where devices could potentially have cooperated. Then, we should be able to find the most relevant cooperation scenarios and their associated device collaboration rules.

On the long range such device collaboration rules could bring a new opportunity for coworkers, repetitive actions and even tasks could be automatically performed by a team of devices. In this perspective, if group of devices are able to automatically perform tasks, then they should be given a role. Going a little further with this idea we'll have to formalize how devices and group of devices can play some specific roles in the collaboration. Besides, we also have to consider that devices and humans can be mixed within a single role.

A very noticeable and original aspect of our work is the approach of the pervasive computing from the collaboration point of view. Indeed, even if there are numerous works related to the development of ontologies and systems to model and support the pervasive computing paradigm for users, there are very few works focusing on the computer supported collaborative work. Considering this lack we think our work around the PCSCW model can sets some bases to fully integrate the pervasive computing model within the collaboration of users.

One of the last points to evoke here is the evolution of rules. Indeed, the device collaboration rules used will doubtlessly need to evolve along with the life of the collaboration to adapt themselves to collaboration patterns and evolution of devices. In this perspective this evolution has to be at least partially dynamic; that is to say the system has to be able to evolve without the intervention of humans. This evolution may involve the modification of the ontology representing resources and roles and the modification of device collaboration rules. To allow this dynamicity we think the best way is the development of a knowledge deriving and learning mechanism. These mechanisms consist in the capacity of the system to infer and extrapolate context information and adapt device collaboration rules. Besides, the learning mechanism shall allow devices to use new collaboration rules extracted from extrapolation of already existing rules but also learn new collaboration rules from other devices. These learning mechanisms should allow the PCSCW model and its collaboration rules to evolve autonomously.

The PCSCW model and its associated collaboration rules form a coherent and consistent base allowing smart devices to continuously interact with each other and adapt their behaviour to users' context and collaboration. This base makes us foresee even more challenges toward the real ambient intelligence but strengthen us for our future work.

References

1. Hamadache, K., Lancieri, L.: Role-Based Collaboration Extended to Pervasive Computing. In: Proceedings of the International Conference on Intelligent Networking and Collaborative Systems, Barcelona, Spain, November 04-06 (2009)
2. Hamadache, K., Manson, P., Lancieri, L.: Pervasive services, brainstorming in situation of mobility. In: 3rd International Conference on Pervasive Computing and Applications, Alexandria, Egypt, pp. 709–714 (2008)
3. Hamadache, K., Bertin, M., Bouchacourt, A., Benyahia, I.: Context-aware communication services: an ontology based approach. In: 2nd International Conference on Digital Information Management (ICDIM 2007), Lyon, France (October 2007)
4. RSA patent: http://www.google.com/patents?q=4405829
5. OWL: http://www.w3.org/TR/owl-features/
6. SWRL: http://www.w3.org/Submission/SWRL/
7. Weiser, M.: The Computer of the 21st Century. Scientific American 265(3), 66–75 (1991)
8. Protégé: http://protege.stanford.edu/
9. RDFS: http://www.w3.org/TR/rdf-schema/
10. Jess: http://www.jessrules.com/jess/index.shtml
11. RuleML: http://www.dfki.uni-kl.de/ruleml/
12. Kyoung, K., Keunho, C., Ohbyung, K.: Rule Selection for Collaborative Ubiquitous Smart Device Development: Rough Set Based Approach. In: Sandnes, F.E., Zhang, Y., Rong, C., Yang, L.T., Ma, J. (eds.) UIC 2008. LNCS, vol. 5061, pp. 386–396. Springer, Heidelberg (2008)

13. Dejene, E., Vasile, S., Lionel, B.: An Ontology-Based Approach to Context Modeling and Reasoning in Pervasive Computing. In: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, pp. 14–19 (2007)
14. Naohiko, K., Kenta, M., Kazunori, T., Hideyuki, T.: Smart Device Collaboration for Ubiquitous Computing Environment. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, Springer, Heidelberg (2003)
15. Xiao, H.W., Da Qing, Z., Tao, G., Hung, K.P.: Ontology based context modelling and reasoning using OWL. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, PERCOMW 2004 (2004)
16. Pawlak, Z.: Rough sets- Theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht (1991)

Enabling Collaboration Transparency with Computational Reflection

Pedro Garcia Lopez, Enrique Fernández-Casado,
Carles Angles, and Maria Ferre

Universitat Rovira i Virgili
Department of Computer Science and Mathematics
Campus Sescelades 43007
Tarragona, Spain
{pedro.garcia,enrique.fernandez,carles.angles,maria.ferre}@urv.cat

Abstract. The conversion of legacy single-user applications to collaborative multi-user tools is a recurrent topic in groupware settings. Many works tried to achieve collaboration transparency: to enable collaborative features without modifying the source code of the single-user application. In this paper, we present a novel blackbox solution that achieves complete transparency by intercepting user interface libraries and input events. This is the first blackbox solution constructed on top of lightweight wrapper technologies (Aspect Oriented Programming) and unlike previous approaches it provides support to both AWT and Swing applications. Our solution solves four important problems: event broadcasting, management of external resources (random numbers), contextual information (telepointers) and transparent launching support. We validated our approach with several Swing-based and AWT-based tools demonstrating that our wrapper is generic and imposes very low overhead.

1 Introduction

Many researchers in CSCW (Computer-Supported Cooperative Work) have presented different solutions to convert single user applications to collaborative ones. The promise of collaboration transparency is to add collaborative interaction without modifying the source code of the original tools. This opens up novel unanticipated uses of existing applications like tele-assistance, monitoring, learning tools or joint work among others.

In general, application-sharing systems are normally classified in centralized (display broadcasting) or replicated architectures (event broadcasting). In the centralized architecture there is a single instance of the shared application, so it is simpler to enforce strict WYSIWIS (What you see is what i see) and sequential interaction. But in this model the central node must redistribute large amounts of graphics data so it can be inefficient in network usage [1].

On the other hand, in replicated architectures several instances of the shared application coordinate their state using event broadcasting. Although replicated architectures face more complex problems like consistency or management of

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 249–264, 2010.

external resources, they are normally preferred [2] for their versatility and support for relaxed WYSIWIS and concurrent interaction. For these reasons, we will focus in the replicated architectures based on event broadcasting.

We classify transparent collaboration in two categories: whitebox and blackbox models. None of them modify the code of the original application, but white box models require knowledge and access to this code whereas black box approaches are completely transparent. In the whitebox model, transparent adaptation techniques require a careful study and knowledge of the code to be wrapped or intercepted. In this line, some works used Aspect Oriented Programming (AOP) interception techniques to wrap specific parts of the application. As we can see, to create such specific interception points (pointcuts) the developer must understand the internals of the target code. On the other hand, blackbox approaches achieve complete transparency by intercepting or wrapping user interface libraries and input events [2], [3], [4].

In this paper, we present the first blackbox solution constructed on top of lightweight wrapper technologies (Aspect Oriented Programming). Our work is a clear evolution from Flexible JAMM [2] and we overcome many of their limitations thanks to the flexibility of AOP technologies. In this line, we can support both AWT and Swing applications, and our wrapping approach includes launching support for third-party applications. Our novel solution has an enormous potential since almost any Java application (Swing, AWT) could be used collaboratively without any modification.

In general, the major contribution of this paper is to outline the importance of computational reflection for both whitebox and blackbox collaboration transparency models. In [5], authors defined reflection as "the ability of a program to manipulate as data something representing the state of the program during its own execution". The authors show two major aspects of this manipulation: introspection (observing the state of the program) and intercession (modify its structure and behaviour). As computational reflection matures and it is incorporated in major languages and UI libraries, the transparent adaptation techniques are more powerful and easy to apply.

We have proven that Java has now suitable introspection (Event Queues, reflection packages) and interception mechanisms (AOP) to provide blackbox transparency. Instead of introducing collaboration-aware mechanisms at the operating/window system level [6], [7], we advocate for the incorporation of powerful and generic computational reflection techniques at the programming languages and interface libraries. In this article, we present a novel blackbox solution called AOPWrapper that benefits from event introspection and software wrapping (AOP).

2 Related Work

As stated in the introduction, we will focus in replicated architectures for achieving fast response, efficient network usage and flexible concurrent interaction. We will distinguish between blackbox and whitebox approaches:

Blackbox approaches try to obtain complete collaboration transparency by intercepting or replacing user interface components and input events. Most recent related work have been produced in the Java language thanks to the flexibility of the virtual machine. For example. [4] and [8] mainly aimed to intercept the event queue and to replace interface components for enabling group collaboration. They reported limitations in the event capture and injection capabilities that strongly complicated their transparent models. Fortunately, in these years the Java language have improved its reflective capabilities and those problems are solved now. For example, in [4] authors ended up recommending three reflective enhancements of the Java language (propagate all events, tag events with source information, and to allow cursor modifications). These three recommendations are now available in the Java language and they also solve the problems of event interception presented in [8].

Flexible JAMM [2] is a relevant work in blackbox approaches since it achieves complete transparency by replacing user interface components by collaborative ones. They also used proxies to manage the interaction with external resources like random generators or file access. They state four major requirements for their component replacement architecture: process migration, runtime component replacement, dynamic binding and user input events interception and replay. But in fact what they are demanding is computational reflection capabilities in the Java environment. Process migration or object serialization is necessary to restore the state in remote locations. But encoding the state of the program as data is in fact called reification in reflective terms. Runtime component replacement and dynamic binding are mechanisms to enable reflective intercession (to modify the structure and behavior of the program in runtime). And user event interception and replay is also directly related to reflective introspection (to observe and capture the program structure and behaviour at runtime).

The major problem of their intercessory approach (component replacement or code swapping) is the maintainability of the replaced components. If the original components change or add novel features, the replaced libraries must be updated. Other clear limitations of Flexible JAMM are the lack of support for AWT applications and the complexity for launching external applications. Flexible JAMM did not publish the source code to the public domain, so many of their techniques are hidden and they could not be adopted by the community. We will explore in this paper a different intercessory technique based on component wrapping. We have published our wrapper as open source so third-parties can extend and modify the tool.

Whitebox approaches require knowledge of the source code to create custom interceptors. Whitebox solutions are not as generic as blackbox ones, but they can use semantic information that is beyond the scope of the blackbox models. In some cases, a blackbox solution just will not work, since the collaboration must take place in specific locations or the collaboration needs semantic information of the underlying model. A typical reflective technique used in whitebox approaches is AOP. AOP permits to intercept and wrap code using modular units called aspects. The aspects are "weaved" with the original code creating transparent wrappers. This weaving proccess can occur in compile time, in load time or in runtime. Aspects aim to modularize cross-cutting concerns that are normally scattered in the code of the application. In this line AOP is usually exemplified with concerns like logging, security or persistence. In CSCW settings, the crosscutting concerns can be the collaborative modules like event broadcasting, floor control or late joining support.

A relevant work in applying AOP to collaborative work is [9]. They present several examples of transparent whitebox collaboration transparency with AOP using a replicated architecture. As a whitebox approach, the interception points in the code (Join Points) must be carefully located for every application. As authors state, this involves "code archaeology" where the main difficulty is to locate the right Join Points in the code. Another drawback of this approach is that Join Points may be brittle, since they rely on opportunistic calls that are vulnerable to change. In our work, we will evaluate the feasibility of AOP to provide generic blackbox transparency.

AOP has also been applied as a tool for achieving flexibility, tailorability and adaptivity in the design of CSCW systems. For example, authors in [10] present a framework where aspects contribute to the modularization of collaborative crosscutting concerns. This modularization in aspects is good for achieving flexibility and extensibility of the CSCW toolkit. Nevertheless, this is a collaboration-aware approach, so they are not focusing on providing collaboration transparency to existing applications.

Another important work in collaboration-aware designs is the seminal paper of Paul Dourish [11] entitled "Computational Reflection and CSCW Design". In this paper, Dourish outlines the importance of computational reflection techniques for achieving flexibility in the design of CSCW toolkits. He presented a CLOS (Common Lisp Object System) Meta-Object Protocol providing object locking for floor control. Thanks to this reflective approach, it is possible to define different locking actions (strong-lock-object, null-lock-object) in a flexible way. As we can see, Dourish already presented the importance of computational reflection as an important mechanism for constructing flexible CSCW shared tools. We focus on the importance of computational reflection for providing collaboration transparency for existing single-user tools.

Finally, a recent approach [12] presented a transparent adaptation technique based on operational transformation to ensure consistency among remote users actions. They converted two well-known tools (Word, PowerPoint) to collaborative ones (CoWord, CoPowerPoint) without modifying the original source code of these applications. Nevertheless, we consider this a whitebox approach since they rely on the Windows API of these tools to observe (introspection) and modify (intercession) their current behavior. Again, the whitebox approach required some "code archaeology" to study the data model of each application and introduce group collaboration features. In this case, the transparent adaptation technique is based on the reflective capabilities exposed by the specific application in the Windows API.

3 Computational Reflection and Collaboration Transparency in Java

In this section we will explain the major reflective capabilities of the Java language and how they can be used for achieving collaboration transparency. We have selected the Java language because it is a mainstream mature language with advanced reflective capabilities. Furthermore, our conclusions can be easily applied to similar programming environments under the Microsoft .Net Platform.

Both Java and .NET environments offer rich introspection functionalities in the form of APIs. In this line, the java.lang.reflect package or the System.Reflection namespace are heavily used by third party programming tools like debuggers or IDEs (Integrated Development Environments). Concerning User Interface (UI) introspection, Java and .NET platforms also offer powerful event handling mechanisms used in UI automation, testing and accessibility tools.

Concerning code interception, both Java and .NET environments provide sophisticated code manipulation tools (byte code transformations) and advanced code swapping mechanisms (Java hotswap class file replacement and custom classloaders). In Java, these class replacement mechanisms are part of the Java Platform Debugging Architecture (JPDA). JPDA is currently used by IDEs in code debugging and also in code profilers.

But neither Java nor .NET offer natively high level interception mechanisms like CLOS Meta Object Protocols or Smalltalk MetaClasses. The sole explicit interception mechanism in Java is the definition of Dynamic proxies. But dynamic proxies only offer very limited method interception that is not enough for transparent collaboration purposes. High level interception frameworks are however available as external software packages. In both Java and .NET exist AOP frameworks offering high level interception and wrapping techniques for developers.

We will evaluate now how computational reflection in Java can solve different collaboration transparency problems:

1. Event Broadcasting: local events must be captured, transmitted to remote collaborators and injected in the local queues. Java permits the creation of custom event queues that may replace the standard one. It is thus possible to capture all local user events (like Mouse and Key events) and inject them in a remote location. As we will explain in the following section, each event contains information about the component source that generated the event. When this event is to be injected in the remote site, it is possible to replace this source with the local one.
2. Contextual information: it must be possible to add contextual information like Telepointers or Radar Views in a transparent way. In Java, the GlassPane transparent component can be added to existing UI components. It is possible with GlassPane to add novel functionalities to the UI like remote Telepointers or contextual messages.
3. External Resources: another key problem for transparent collaboration is how to manage external resources used by the applications like random

generators, sockets or file access. The direct solution is to use software interception or wrapping to replace this calls with controlled proxies that show the same behavior in all remote locations. We will show in the next section how this can be solved with AOP techniques.
4. Launching Support: the transparent solution must ensure the launching of legacy single-user applications with minimal changes to convert them into collaborative ones. For example, code wrapping using aspect weaving can occur at compile time, load time or in runtime. The launching of the legacy tool with the collaborative wrapper can thus be achieved transparently with load time aspect weaving. Transparent launching can also be achieved with code replacement techniques available in the JPDA.
5. Late joining support: in order to support late joiners in a collaborative session, the current state must be transferred to the new collaborators. This can be solved with event replay in the entry node or transferring the state of objects. Java supports serialization and deserialization mechanisms that encode object state as data (byte arrays).
6. Floor Control: finally, there is a need for implicit or explicit mechanisms for arbitrating the access to shared resources. Normally, such floor control policies are related with the event broadcasting services. These policies can then decide with local events must be propagated and when, and which remote events must be injected and when.

As we can see, the problems presented for collaboration transparency can be solved with computational reflection techniques. There is however a question that still remains to be answered: code replacement or code wrapping ?. Up to now, blackbox transparent approaches [2] relied on UI component replacement, while many whitebox approaches [9] used software wrapping (AOP, application APIs). In this paper, we evaluate the feasibility of using AOP wrapping techniques to provide complete blackbox transparency to single-user applications. In the next section, we will present a generic wrapper using AOP and we will analyze its advantages and limitations.

4 AOPWrapper: A Generic Blackbox Wrapper for Java

In this section we present a lightweight wrapper that offers blackbox collaboration transparency to legacy Java applications. AOPWrapper addresses four out of the six problems presented in the previous section: event broadcasting, contextual information, external resources and launching support. The two remaining problems (late joining and floor control) will be undertaken in a future work due to lack of space.

The main idea of AOPWrapper is to allow each user running a Java-based application to transparently collaborate with other users by sharing all the local-generated events with other participants. Such sharing is performed via a distributed event queue in such a way that all participants will be consistently injecting into their applications all the events generated by all the participants.

Such consistency is reached by ignoring local-generated events and only injecting into the application's event queue those events which have been received from the distributed event queue. In this way, all users running a collaborative instance of the application will be executing the same sequence of events because they are only normally dispatching remote events.

As figure 1 shows, AOPWrapper includes four independent modules, each one responsible for one key collaborative feature. In addition to the wrapper, it is also necessary a distributed event queue in order to broadcast all events generated by the new multi-user applications. In our prototype we use as the distributed event queue a well-known publish/subscribe service (Apache ActiveMQ). We have chosen ActiveMQ for its high performance and strong event filtering capabilities. Nevertheless, other event broadcasting systems may be used such as group communication toolkits.

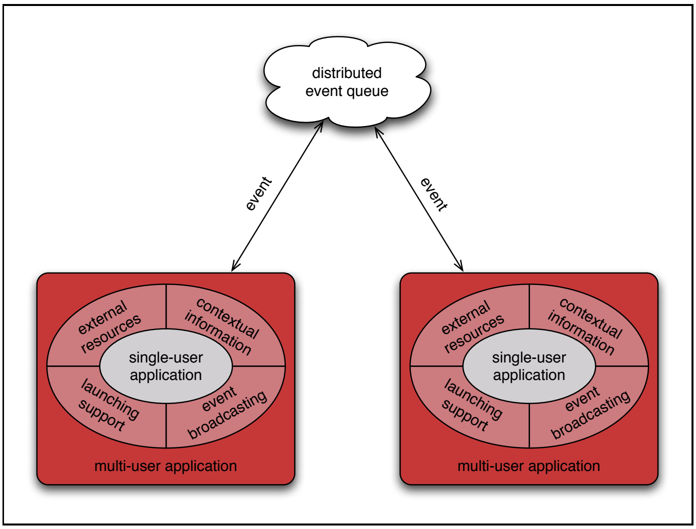


Fig. 1. Overview of the AOPWrapper's Architecture

We will describe how we solved each of the four problems using computational reflection techniques. We will also explain how we offer support for legacy AWT-based applications. Other blackbox approaches like Flexible JAMM were restricted to Swing-based applications. In our case, the flexibility of AOP permits inheritance rewirings that can solve the problem easily.

4.1 Event Broadcasting

In order to achieve an Event Broadcasting Collaboration System it is necessary to process all events generated by users and prepare them for sending via the Distributed Event Queue. In Java, it is possible to replace the standard Event Queue with a specific one to properly handle local and remote events. In this

```
Toolkit.getDefaultToolkit().getSystemEventQueue().push(
    new CustomEventQueue(this));
```

Fig. 2. Customizing Event Queue

line, we have implemented our own event queue by creating a class which inherits from EventQueue that we called CustomEventQueue. Using the method push(EventQueue eq) showed in Figure 2, we can replace the default event queue.

By overriding the method dispatchEvent of our CustomEventQueue we can manage all AWTEvents received by the System Event Queue in a non-intrusive way for the application which is running collaboratively.

Basically we process two kinds of events in our CustomEventQueue: Local events and Remote events. Local events are all AWTEvents generated by an user over an application. Remote events are those AWTEvents received via Broadcast from the Remote Queue (ActiveMQ). This queue is where all the applications on the same collaborative session send the local AWTEvents they generate.

1. Local Event processing: When a local event arrives to our custom event queue it is prepared for being published to the remote queue. With this objective we have created three new classes: CustomMouseEvent, CustomKeyEvent and CustomFocusEvents, which extend respectively from MouseEvent, KeyEvent and FocusEvent. Its purpose is to store extra information which is going to be needed by collaborating applications to dispatch remote events properly. Some of this extra information is: the application which generated the event (also known as 'event owner'), the identifier of the component over which the event has been triggered, and the screen resolution.
2. Remote Event processing: Remote events which are broadcasted by remote queue are received by a MessageListener. When a remote event arrives, it needs to be posted to the System event queue to have some effect on the application. To achieve this, we use the method postEvent which is provided by Toolkit. Once the remote events are posted, a previous processing is needed before our custom event queue dispatches them. This processing consists on adapting the incoming remote event to the environment changes present in the different collaborative applications. We have to change the event source with an instance of a component of the local application.

These methods can recover an instance of the right component from the index of the list with all-application-components. The method getAllComponents() in Figure 3 is responsible of this list generation.

This components list can be obtained transparently thanks to the method Window.getWindows(), which can return all windows used by an application. In order to get all windows of an application this process must be done recursively. We assume that doing the above process it will always return lists with the same order for any application on collaborative session. We can assume this because all collaborating applications are the same and their status is on the same point.

```
public static List<Component> getAllComponents() {
    components = new LinkedList<Component>();
    for (Window window : Window.getWindows()) {
        getComponents(window);
    }
    return components;
}

public static void getComponents(Component c) {
    components.add(c);
    if (c instanceof Container) {
        for (Component c1 : ((Container)c).getComponents())
            getComponents(c1);
    }
}
```

Fig. 3. Trasparent Component Obtaining Method

The incoming Remote Mouse events require extra processes: all processes tied to the telepointer drawing –this will be explained on Contextual Information section– and all operations for adapting mouse events to the new machine screen resolution. All mouse events have a field that indicates on which position of the screen the mouse event have been triggered; this position is measured in pixels. This magnitude is conflictive because each machine can be running collaborative applications with different screen resolutions. So, before posting any mouse event we need to convert its point position in accordance with the screen resolution of the local machine which is executing the application.

Finally, once the Local and Remote events processing it is done, the custom event queue has to dispatch certain types of events. Our dispatchEvent method ensures that only Remote Events are dispatched, the rest of them are discarded. We do this to avoid that local events affect the local collaborative application. To avoid consistency problems we only allow remote events to alter the collaborative applications. This is because our remote queue plays the role of a 'sequencer'. It receives all events of every application in a collaborative session and delivers them in the same order to all collaborating applications, insuring that event sequence will be the same in all machines.

4.2 Contextual Information

Another important issue if to add contextual information like Telepointers that improve the collaboration awareness in remote users. The telepointer is very important to be aware of what other users are doing in every moment of time. Since current Operating Systems and UI libraries do not explicitly support multiple cursors, we need to emulate them with our wrapper.

In our case, we can draw images representing contextual information like telepointers or contextual messages. To display this contextual layer on top of existing UI components, we used the Swing transparent GlassPane class that can be superposed to any swing component. In our case, we set a transparent GlassPane over all top-level swing containers (like jFrame and jDialog). Every

```
declare parents: (* && java.awt.Frame+ && !java.awt.Frame)
    extends javax.swing.JFrame;
```

Fig. 4. Component Replacement with AspectJ: Declare Parents Statement

time our wrapper receives a Remote Event from the Remote Queue we update telepointer positions and we force the glasspane to repaint, in order to simulate remote mouse pointers.

The main problem of our approach is that it is restricted to Swing applications. It is not possible to superpose a GlassPane to an AWT container like java.awt.Frame. This problem affects other transparent collaboration approaches like Flexible Jamm so they can only be applied to Swing-based applications. In our case, we provide a clean solution based on AOP interceptors. Using AOP intertype mechanisms it is possible to change the inheritance tree of classes. As we can see in figure 4, we can modify the superclass of all the classes which inherit from java.awt.Frame and change it to java.swing.JFrame. As we can see, we can transparently move an AWT application to Swing. This change allows the wrapper to superpose the transparent GlassPane and consequently to paint telepointers over AWT-based applications.

4.3 External Resources

The management of external resources like random numbers or files is another important problem that must be addressed. Again, we make use of AOP interceptors to wrap these calls and provide a controlled interaction between remote users. We consider here two external resources: random numbers and look and feel interface.

The random calls in applications make their progress differs from one execution to another. These calls make the application non-deterministic and may lead to consistency problems. We have defined an aspect with several pointcuts that intercept random generators. First line in Figure 5 defines the pointcut for the java Random constructor.

We intercept Random constructor calls and replace them with a new Random constructor with a prefixed seed. We want that all applications on a collaborative session create an instance of Random with the same seed. With this, we will ensure that all collaborating applications generate the same sequence of random numbers when nextInt, nextdoble, etc. methods are called. The pointcut can be seen in second line of Figure 5.

We also replace the method Math.Random for nextdouble(), following previously mentioned fixed seed approach. If no previous calls for random has been

```
pointcut randomCall() : call(java.util.Random.new());
pointcut mathRandomCall() : call(double java.lang.Math.random());
```

Fig. 5. Random's Pointcuts

made, it is necessary to create an instance of Random calling its constructor with a prefixed seed.

Look and Feel can also be a source of non-deterministic behaviour. That is because a Java program can have different appearance depending in which Operating System is running. For example Lobo web browser looks different in Windows 7 that in Windows XP. To avoid this problem, we intercept any change of Look and Feel. We force to set a Look and Feel which looks the same in any environment, in our case the Java Look and Feel (see Figure 6).

```
pointcut lookAndFeel() :
    call(void javax.swing.UIManager.setLookAndFeel(String))
        && !this(Interceptor);

after() : lookAndFeel() {
    try {
        UIManager.setLookAndFeel(
            UIManager.getCrossPlatformLookAndFeelClassName());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Fig. 6. Look and Feel Interception

4.4 Launching Support

Finally, our solution must provide a standard mechanism to launch legacy applications with our blackbox wrapper. To enable complete transparency we avoid any change in the original code and we do not require post compile operations. This can be done using AOP interception and load time or run time weaving. The idea is to modify the main method of the legacy application to include the required operations for running our wrapper. The pointcut in Figure 7 is responsible intercepting the main method.

Before the execution of main is when our collaborative wrapper initializes, 'before' statement in Figure 7. The CollaborativeWrapper performs two initial operations: it connects to the distributed event queue and it registers the CustomEventQueue in the UI toolkit. These initial operations ensure the correct event broadcasting capabilities explained in section 4.1.

After the main execution is when the graphic-related operations of our wrapper are done, 'after' statement in Figure 7. These operations consist in superposing glass-panels over all the windows that the legacy application has created. These glass-panels will allow us to paint the telepointer (image that represents the mouse of a remote user). This process must be done after the main execution, otherwise the application would not have created any Window to put GlassPanes on.

Finally, to launch the application including our wrapper, we use load-time weaving. With a simply modification in the command line we can launch any Java application in collaborative mode.

```
pointcut mainMethod() : execution(public static void main(String[]));

before() : mainMethod() {
    try{
        CollaborativeWrapper wrapper = new CollaborativeWrapper();
    }
    catch(Exception e){}
}

after() returning : mainMethod() {
    for (Window w : Window.getWindows()) {
        if (w instanceof JFrame) {
            CustomGlassPane glassPane = new CustomGlassPane();
            ((JFrame) w).setGlassPane(glassPane);
            glassPane.setVisible(true);
        } else if (w instanceof JDialog) {
            CustomGlassPane glassPane = new CustomGlassPane();
            ((JDialog) w).setGlassPane(glassPane);
            glassPane.setVisible(true);
        }
    }
}
```

Fig. 7. Interception of the Main Method

```
java -javaagent:lib/aspectjweaver.jar -classpath
.;legacyapp.jar;aopwrapper.jar;aspectjrt.jar;activemq.jar legacyappMainClass
```

Fig. 8. Command-Line Launcher

5 Validation

To validate the usability of our approach, we have calculated the overhead imposed by our wrapper to both lightweight and heavy applications (Mahjongg and Lobo respectively), and using different workloads (intensive and non-intensive use). The execution environment consists of two computers with the following features: Pentium4 3Ghz with 1GB of RAM and Intel Core2 Quad 2.33Ghz with 3GB of RAM, both running Microsoft Windows XP Professional with Service Pack 3, respectively. The client computers broadcast events with an ActiveMQ server located in the same local area network.

Figure 9 shows the sequence followed by an event from its local generation and interception by the AOPWrapper, until it is finally dispatched by the system's event queue. It clearly shows that the execution flow of the single-user application has been modified, by adding new steps which add a speficic overhead.

$$overhead = overhead_{op} + overhead_{ip} + overhead_{c}\,. \tag{1}$$

– **Preparation Overhead (output)**, $overhead_{op}$ in (1), refers to the elapsed time from the interception of a locally generated event, until it is dispatched to the ActiveMQ server. It includes the time spent in the preparation of the event (the creation of the custom event and the addition of its meta-information).

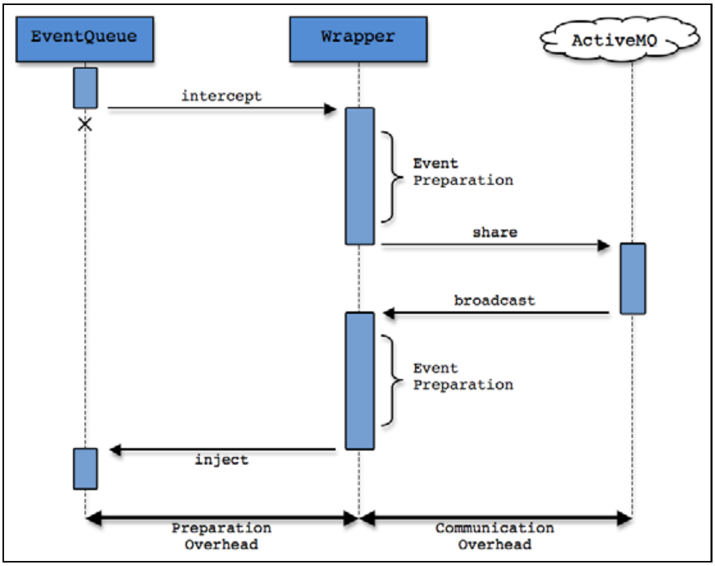


Fig. 9. Wrapper's Sequence Diagram

- **Preparation Overhead (input)**, $overhead_{ip}$ in (1), refers to the elapsed time since a custom event is received from the remote queue, until it is finally dispatched by the event queue. $overhead_{ip}$ has to be bigger than $overhead_{op}$ because the event has to perform the whole injection process again, while the interception (the cost of which is measured by $overhead_{op}$) takes place at the event queue, so the event has only to be prepared and sent.
- **Communication Overhead**, $overhead_c$ in (1), refers to the elapsed time since an event is sent to the remote queue, until it is received by the AOP-Wrapper. It depends on the context (broadband speed, remote queue implementation used) but in our case we assume a local area network and ActiveMQ server.

The results we have obtained show that the use of AOPWrapper only adds, independently of the application used and the workload applied, an average overhead of $35ms$, which is decomposed as follows: $14ms$ of preparation overhead (input), $0.1ms$ of preparation overhead (output) and $20.9ms$ of environment-dependant communication overhead. So we conclude that, in terms of user experience, AOP-Wrapper is a completely usable tool for collaborative synchronous applications.

5.1 Proof of the Prototype

We have validated our generic blackbox wrapper with some Java applications: three Swing-based (PDF Renderer, Lobo Web browser, Mahjongg game) and one AWT-based (Luke Lucene indexing tool). We have reached a level of transparency that allows the user to run a single-user application in a collaborative way by only adding our wrapper - and its dependencies - to the classpath of the single-user application. It is only necessary to slightly modify its command-line launcher (in order to enable weaving capabilities) and to fill the required fields (user's identifier, distributed event queue's ip and distributed event queue's topic) in a config.properties file.

Our lightweight approach only captures three types of input events (Mouse, Keyboard, Focus) and it is able to reproduce the interaction of users in a multi-user collaborative session. With the PDF renderer we were able to visualize a PDF file collaboratively with consistent event propagation (scrollbars, menus) and fast telepointer visualization. With the Lobo Web browser we also tested the feasibility of transparent collaborative browsing without modifications on the original tool. Finally, we also demonstrated the joint game play of the Mahongg game from remote locations.

We also tested an AWT-based application to prove our transparent solution. Obviously, without the rewiring of the inheritance tree from Frame to JFrame, the telepointer would not work. But in our case, the application was shared remotely without problems and the telepointer worked smoothly over all UI components.

We have shown how our generic wrapper can convert single user applications to collaborative ones. Our lightweight and transparent approach may have different uses like tele-assistance, learning purposes or cooperative interactions over existing single-user tools. We have solved this problem using advanced computational reflection techniques like event introspection and software wrapping (AOP).

Nevertheless, although blackbox wrapping is the more transparent collaboration approach, it certainly has some limitations that preclude its usage in all settings. Some applications only require collaborative functionalities at some specific execution point of their life-cycle, and leveraging the whole application may incur in security or functionality issues. For example, if the application already includes authentication and login in the launching process, our blackbox wrapper will not be a suitable solution. Furthermore, if semantic information is needed from the legacy application, whitebox models will be more appropriate.

6 Conclusions

In this paper we outlined the importance of computational reflection for achieving collaboration transparency. We classified collaboration transparency in two categories: blackbox and whitebox models. In blackbox models, complete transparency can be achieved with event capturing (introspection) and interface library wrapping or replacement (interception). In whitebox models adhoc interceptors must be carefully deployed in the original code to enable collaboration.

To the best of our knowledge, we are the first to present a transparent blackbox solution based on software wrapping (AOP). Previous solutions like Flexible JAMM provided backbox transparency replacing interface library components. Nevertheless component replacement is more intrusive and susceptible to changes in interface libraries. Our lightweight wrapping solution is less intrusive and can easily adapt to changes in the original libraries.

In this paper we solve four important problems of transparent collaboration: event broadcasting, management of external resources (Randoms), contextual information (telepointers) and launching support for legacy applications. We validated our prototype with existing Swing-based aplications like a Web browser (Lobo), a pdf viewer (PDF Renderer), a game (Mahjongg), and with an AWT-based toolbox for Lucene Indexes (Luke).

We have proven that advanced computational reflection tools like event introspection and software wrapping (AOP) can be used to provide blackbox collaboration transparency. We are now trying to solve the two remaining problems in blackbox collaboration transparency: late joining and floor control support. The generic wrapper can be obtained in http://ast-deim.urv.cat/wiki/AOPWrapper with examples and documentation.

It is true that our ideas cannot be applied to other environments not providing powerful introspection (EventQueues) and interception techniques (AOP). We could solve the problem in Java because it already provides such advanced capabilities. For this reason, we advocate for the future enhancement of computational reflection capabilities in programming languages, interface libraries, Web browsers and operating systems. Thanks to these reflection capabilities, we foresee potential applications in collaboration, logging, interface testing, security, fault tolerance and many others.

Acknowledgements

This work was partially funded by the spanish P2PGRID research project (TIN2007-68050-C03-03) from the 'Ministerio de Ciencia e Innovación' and by the Neurolearning project (TSI-020501-2008-154) from the 'Ministerio de Industria, Turismo y Comercio'.

References

1. Lantz, K.A.: An experiment in integrated multimedia conferencing. In: CSCW 1986: Proceedings of the 1986 ACM conference on Computer-supported cooperative work, pp. 267–275. ACM, New York (1986)
2. Begole, J., Rosson, M.B., Shaffer, C.A.: Flexible collaboration transparency: supporting worker independence in replicated application-sharing systems. ACM Trans. Comput. Hum. Interact. 6(2), 95–132 (1999)
3. Begole, J., Smith, R.B., Struble, C.A., Shaffer, C.A.: Resource sharing for replicated synchronous groupware. IEEE/ACM Trans. Netw. 9(6), 833–843 (2001)
4. Begole, J., Struble, C.A., Shaffer, C.A.: Leveraging java applets: Toward collaboration transparency in java. IEEE Internet Computing 1(2), 57–64 (1997)
5. Gabriel, R.P., White, J.L., Bobrow, D.G.: Clos: Integrating object-oriented and functional programming. Commun. ACM 34(9), 28–38 (1991)
6. Lauwers, J.C., Joseph, T.A., Lantz, K.A., Romanow, A.L.: Replicated architectures for shared window systems: a critique. SIGOIS Bull. 11(2-3), 249–260 (1990)

7. Lauwers, J.C., Lantz, K.A.: Collaboration awareness in support of collaboration transparency: requirements for the next generation of shared window systems. In: CHI 1990: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 303–311. ACM, New York (1990)
8. Li, W., Wang, W., Marsic, I.: Collaboration transparency in the disciple framework. In: GROUP, pp. 326–335. ACM, New York (1999)
9. Cheng, L.T., Patterson, J.F., Rohall, S.L., Hupfer, S., Ross, S.: Weaving a social fabric into existing software. In: Mezini, M., Tarr, P.L. (eds.) AOSD, pp. 147–158. ACM, New York (2005)
10. Pinto, M., Fuentes, L., Troya, J.M., Fayad, M.: Towards an aspect-oriented framework in the design of collaborative virtual environments. In: FTDCS, pp. 9–17. IEEE Computer Society, Los Alamitos (2001)
11. Dourish, P.: Computational reflection and cscw design. Technical report (1992)
12. Sun, C., Xia, S., Sun, D., Chen, D., Shen, H., Cai, W.: Transparent adaptation of single-user applications for multi-user real-time collaboration. ACM Trans. Comput. Hum. Interact. 13(4), 531–582 (2006)

Plasticity of Interaction Interfaces: The Study Case of a Collaborative Whiteboard

Gabriela Sánchez[1], Sonia Mendoza[1], Dominique Decouchant[2,3], Lizbeth Gallardo-López[4], and José Rodríguez[1]

[1] Departamento de Computación, CINVESTAV-IPN, D.F., Mexico
[2] Depto. de Tecnologías de la Información, UAM-Cuajimalpa, D.F., Mexico
[3] C.N.R.S. - Laboratoire LIG de Grenoble, France
[4] Departamento de Sistemas, UAM-Azcapotzalco, D.F., Mexico
gsanchez@computacion.cs.cinvestav.mx, smendoza@cs.cinvestav.mx, decouchant@correo.cua.uam.mx, glizbeth@correo.azc.uam.mx, rodriguez@cs.cinvestav.mx

Abstract. The development of plastic user interfaces constitutes a promising research topic. They are intentionally designed to automatically adapt themselves to changes of their context of use defined in terms of the *user* (e.g., identity and role), the *environment* (e.g., location and available information/tools) and the *platform*. Some single-user systems already integrate some plasticity capabilities, but this topic remains quasi-unexplored in CSCW. This work is centered on prototyping a plastic collaborative whiteboard that adapts itself: 1) to the *platform*, as it can be launched from heterogeneous computer devices and 2) to each *collaborator*, when he is working from several devices. This application can split its interface between the users' devices in order to facilitate the interaction. Thus, the distributed interface components work in the same way as if they were co-located within a unique device. At any time, group awareness is maintained among collaborators.

Keywords: plastic interfaces, context of use, redistribution-based plasticity, multi-computer collaborative environments.

1 Introduction

The increasing proliferation of heterogeneous computers combined with the unstoppable progress of communication networks allow to conceive the user [2] as "an entity that evolves within a multi-computer environment where he employs, in an opportunistic way, several interactive devices/systems in order to satisfy his information access needs anytime anywhere". However, an interactive system cannot display the same user interface on small, medium and big screens of such a multi-computer environment. Regarding usability, the automatic transposition of an interactive system from a PC to a smartphone results non-feasible, due to their so different displaying, processing, storage and communication capabilities.

At the user interface level, the most obvious solution to cope with this problem consists in reducing the size of the user interface components in order to

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 265–280, 2010.

display them in a unique view [6]. Unfortunately, this primitive solution might compromise the system usability because the handling of the included components might become complicated. Another solution consists in taking advantage of other modalities, besides the visual one, e.g., the incorporation of audio into the components to notify errors. Nevertheless, if the visual modality remains the only realistic option and it seems impossible to reduce the component size then the last solution, called plasticity, entails structuring the user interface into subsections through which the user can browse.

The user interface plasticity [5] allows interactive systems to manage variations based on: 1) the *user* (e.g., detecting when he is interacting with the system from one or more computers), 2) the *environment* (e.g., hiding some information when an unauthorized person is approaching to the screen), and 3) the hardware and software *platforms* (e.g., screen size and operating system capabilities), while preserving a set of quality criteria (e.g., usability and interaction continuity).

In the domain of single-user interactive systems, user interface plasticity has been mainly studied through the development of prototypes and the definition of concepts and models. However, this research topic remains quasi-unexplored in CSCW, despite the imminent need to provide the interaction interfaces of groupware applications with the adaptation capability to contextual changes. As is well known, designing interaction interfaces for groupware applications is more complex because of the need to add group awareness components to supply information that is naturally present during face-to-face interactions.

In this paper, we analyze and apply some adaptability principles to the development of a plastic collaborative whiteboard, specially designed and implemented to support nomadic work. Such an application is able: 1) to remodel its interaction interface in order to be launched from heterogeneous hardware/software platforms, and 2) to redistribute its interaction interface when it detects a user working from different hardware platforms. We select this groupware application because its functional core is relatively simple, whereas its user interface might be lightly complex. Thus, the collaborative whiteboard characteristics allow us to focalize on our goal: the study of interaction interface plasticity, without having to cope with secondary difficult problems.

This paper is organized as follows. After studying the plasticity problem space for single-user applications (Section 2), we present a detailed analysis of the most relevant plastic prototypes (Section 3). Then, we introduce a) the design and implementation of this plastic collaborative whiteboard (Section 4) and b) a scenario that highlights how our application facilitates both user-system and user-user interactions due to its plastic properties (Section 5). Finally, we conclude on the developed work and introduce some extensions.

2 Plasticity for Single-User Interactive Systems

Plasticity [2] is defined as the capability of interactive systems to adapt themselves to changes produced in their context of use, while preserving a set of predefined quality properties, e.g., usability. The **context of use** [3] involves three elements: 1) the *user*, which denotes the human being who is going to use the interactive system; 2) the *platform*, which refers to the available hardware and software of the user's computers to support user-system interaction; and 3) the *environment*, which concerns the physical and social conditions where interaction takes place. Plasticity can be achieved following two approaches [14]:

1. The **redistribution** approach consists in reorganizing the user interface (UI) on different platforms. Four types have been identified: 1) from a *centralized organization to another centralized one*, whose goal is to preserve the centralization state of the UI, e.g., migration from a PC to a PDA; 2) from a *centralized one to a distributed one*, which distributes the UI among several platforms; 3) from *a distributed one to a centralized one*, whose effect is to concentrate the UI into one platform; and 4) from a *distributed organization to another distributed one*, which modifies the distribution state of the UI.
2. The **remodeling** approach consists in reconfiguring the user interface (UI) by means of insertions, suppressions, substitutions and rearrangements of all or some UI components. Transformations apply to different abstraction levels: 1) *intra-modal*, when the source components are retargeted within the same modality, e.g., from graphical interaction to graphical one; 2) *inter-modal*, when the source components are retargeted into a different modality, e.g., from graphical interaction to hapic one; and 3) *multi-modal*, when remodeling uses a combination of intra- and inter-modal transformations.

Both plasticity approaches consider some factors that have a direct influence when adapting the user interface of single-user interactive systems [5]:

– The **adaptation granularity** denotes the smallest unit of the user interface (UI) that can be remodeled and redistributed. Three adaptation grains are identified: 1) *interactor*, which represents the UI smallest unit supporting a task, e.g., the "Save" button of an editor; 2) *workspace*, which refers to a space supporting the execution of a set of logically related tasks, e.g., a printing window; and 3) *total*, which implies that the whole UI is affected by the modifications. There is another grain, called *pixel*, according to which any UI component can be partitioned across multiple displays; however, this adaptation granularity only concerns the redistribution approach.
– The **state recovery granularity** represents the effort made by the user to carry on his activity after the user interface adaptation finished. Three recovery grains are considered: 1) *physical action*, which assumes the user does not lose any action; 2) *task*, which ensures that all finished tasks are persistently validated, except for the current interrupted task; and 3) *session*, which forces the user to restart, losing all the effects of his performed actions.
– The **user interface deployment** concerns the installation of the user interface (UI) in the host platform following: 1) a *static deployment*, which means that UI adaptation is performed when the system is launched and from then no more modifications are carried out; or 2) a *dynamic deployment*, which means that remodeling and redistribution are performed on the fly.

- The **coverage of technological spaces** denotes the infrastructure capability to support user interface (UI) plasticity across technological spaces (TS). Three coverage forms are possible: 1) *intra-TS*, which means the UI is implemented and adapted within a single TS, e.g., from Java to Java; 2) *inter-TS*, which supposes the UI is expressed in a single source TS and transformed into another target TS, e.g., from Java to Qt; and 3) *multi-TS*, which means the source or target UIs combine components of different TS.
- The **meta-user interface** (meta-UI) consists of a set of functions, whose goal is to evaluate and control the state of a plastic interactive system. Coutaz and Calvary [5] identifies three types of meta-UIs: 1) *meta-UI without negotiation*, which makes observable the adaptation process, but does not allow the user to take part; 2) *meta-UI with negotiation*, which is required when the system cannot decide between different adaptation forms, or when the user wants to control the process outcome; and 3) *plastic meta-UI*, which is able to instantiate the adequate meta-UI when the system is launched. The Meta-UI approach remains an open research problem [14].

3 Related Work

On the basis of the previously introduced factors, we analyze the most important plastic interactive systems. The majority of them are single-user systems, although others only provide a basic support for cooperative work. Few systems automatically remodel and redistribute their user interface, while others invite the user to participate to the adaptation process. After providing a brief description of each system, we present a comparative study of their plastic capabilities.

The **Sedan-Bouillon** Web site [1] promotes the tourist sites of Sedan and Bouillon cities; this system allows the user to take part of the redistribution of the site main page between a PC and a PDA. The **heating control** system [5] allows the user to consult and modify the temperature of the bedroom, bathroom and living room of his house; thus, the temperature can be controlled from a PDA, a PC, a mobile phone and even a watch. The **FlexClock** system [9] adapts its date and time components to the window size; this system provides seventeen predefined configurations of its components to carry out adaptation.

Unlike the previous single-user interactive systems, **Roomware** [11] supports working groups, whose members are co-located in a physical room; the main motivation of this system is to add computing and communication capabilities to real objects (e.g., walls, tables and chairs) in order to explore new interaction forms among collaborators. The **ConnecTables** system [13] facilitates the transition from individual work to cooperative one, by allowing collaborators to couple two personal tablets in order to dynamically create a shared workspace.

The first plastic capability, the **context of use**, refers to the user interface adaptation to three elements: the user, the platform and the environment. The *Sedan-Bouillon* Web site adapts to: 1) the “user” because it identifies him when he is working from different devices; and 2) the “platform” as it can be accessed from PC and PDA. The *heating control* system adapts to the “software and hardware platform" because it can be launched as Web and stand-alone applications, and it allows to consult the room temperature from heterogeneous devices (i.e., PC, PDA, mobile phone and watch). Likewise, *Roomware* is able to run on three special devices: 1) DynaWall, which is a large, touch-sensitive, information display and interaction device that is smoothly integrated in the wall; 2) InteracTable that provides a touch-sensitive plasma display into a tabletop; and 3) CommChair, which combines the mobility and comfort of armchairs with the functionality of a pen-based computer. A variation of platform adaptation is implemented by *ConnecTables* that allows to physically/logically couple two tablets to create a shared space. Finally, *FlexClock* can only be executed on PC but adapts its presentation to the window size. We can notice that none of the analyzed systems takes into account the environmental element.

There are four types of user interface (UI) **redistribution** that result from the 2-permutation with repetition allowed on a set of two possible transition states: centralization and distribution. The *Sedan-Bouillon* Web site supports all types of redistribution, e.g., full replication or partial distribution of the workspaces between different devices. *Roomware* supports transitions: 1) from a centralized organization to a distributed one, when its UI is shared out among the three smartboards of DynaWall; 2) the inverse of the previous transition, by reconcentrating its UI in an InteracTable or CommChair; and 3) from a centralized organization to another, when migrating from an InteracTable to a CommChair and vice-versa. *ConnecTables* only supports UI transitions from a distributed organization to a centralized one and vice-versa, when two tablets are respectively coupled and decoupled. Finally, *FlexClock* and the *heating control* only propose a centralized organization of their UI.

Remodeling consists in reconfiguring the UI components at the intra-, inter- or multi-modal abstraction levels. All the analyzed systems are intra-modal as their source components are retargeted within the same graphical modality.

The **adaptation granularity** defines the deep grain (i.e., pixel, interactor, workspace and total) in which the user interface (UI) can be transformed. *FlexClock* remodels its UI at the interactor grain as the date and hour are considered each one as a specific task, whose presentation (size, format and alignment) depends on the window size. The *heating control* system remodels its UI at the total and interactor grains; the first grain means that the PC and PDA user interfaces are graphical, whereas those of the mobile phone and watch are textual; the second grain means that the PC user interface is displayed on one view, whereas that of the PDA is structured into three views (one per room) through which the user navigates using tabs. The *Sedan-Bouillon* Web site remodels its UI at the workspace grain as the presentation (size, position and alignment) of the Web main page title, content and navigation bar is modified when this page is loaded from a PDA. *Roomware* uses the pixel grain when the UI is distributed on the three smartboards of DynaWall. Finally, *ConnecTables* also redistributes its UI at the pixel grain, because it allows the user to drag-and-drop an image from one tablet to another when they are in coupled mode.

The **state recovery granularity** concerns the preservation degree (i.e., physical action, task and session) of the user's current activity after the user interface (UI) adaptation. The *Sedan-Bouillon* Web site supports the task grain because only the user's current task is lost, e.g., if the UI redistribution occurs while filling a form, the user has to refill it. The remaining systems do not provide enough information to determine the lower state grain that can be recovered.

The **user interface deployment** can be static or dynamic. The *Sedan-Bouillon* Web site provides on the fly redistribution of its workspaces. *FlexClock* automatically remodels the window interactors (e.g., analogical clock enlargement and digital clock central alignment) when the user resizes the window. *ConnecTables* dynamically creates a shared workspace (vs. a personal one) when two users couple (vs. decouple) their tablets. On the contrary, the *heating control* system and *Roomware* only provide static deployment.

The coverage of **technological spaces** (TS) denotes the infrastructure capability (i.e., intra-TS, inter-TS and multi-TS) to support user interface plasticity across technological spaces. The *Sedan-Bouillon* Web site (HTML/PHP), the *heating control* (ART-Studio [2]) and *FlexClock* (QTk [9]) remain within the same technological space before and after the plastic adaptation (i.e., intra-TS). The other systems do not offer any information on the matter.

The Sedan-Bouillon Web site is the unique system that provides a **meta-user interface with negotiation**, because the user cooperates with the system for the redistribution of the UI workspaces (e.g., Web page title and navigation bar).

Currently, the adaptability of groupware applications is being analyzed as a side issue of the development of augmented reality techniques, which mainly rely on redistribution. The studied applications do not consider neither the *user* and *environment* elements of the context of use, nor most of the factors that affect the user interface. Therefore, we explore whether a plastic groupware application can be developed from the plasticity principles defined for single-user systems.

4 A Plastic Collaborative Whiteboard

Applying the plasticity approaches and factors of single-user interactive systems (cf. Section 2), we developed a plastic collaborative whiteboard. This application is able to remodel and redistributes the user interaction interface in response to changes occurred in the *platform* and *user* elements of the context of use. Firstly, we describe a MVC-based design of this groupware application. Then, we focalize on implementation issues related with the display space management of handheld devices. Finally, we present a scenario that highlights the benefits to provide groupware applications with plastic capabilities.

4.1 MVC-Based Design

The design of the plastic collaborative whiteboard is based on the architectural style Model-View-Controller (MVC) [8]. We prefer it to other styles (e.g., PAC* [10]) as, from our point of view, the MVC principles (several views for a model) match better with the plasticity principles (several user interfaces for an application). Thus, MVC simplifies the application structural representation before and after applying any plastic adaptation. MVC also facilitates software reutilization by modeling the application as independent interrelated components.

The basic MVC architecture consists of a *model*, which represents the application data; a *controller*, which interprets user input; and a *view*, which handles output. Like many MVC variants, our plastic collaborative whiteboard implements view-controller pairs as combined components. More particularly, as shown in Fig. 1, the MVC tree of our plastic collaborative whiteboard contains the root node, R, and three child nodes, H1, H2 and H3. At runtime, the R node view-controller is in charge of: 1) creating an instance of the application, 2) coordinating its children, and 3) communicating with other distributed instances of the application. The R node model stores information about these tasks (e.g., remote instance identifier or active children). The children of the R node are:

1. The H1 node is in charge of the collaborator authentication. Its view-controller receives the collaborator identification (e.g., name/password pair or photo) from a specific window. Then, the H1 view-controller calls the corresponding model functions to validate the collaborator identity. Finally, the H1 view-controller notifies its father of the validation result.
2. The H2 node administrates the collaborative whiteboard. Its view-controller receives the collaborator's input (e.g., clicks on the toolbar and drawing area), whereas its model maintains a log recording of each collaborator's actions (e.g., created figures/texts and their dimensions, coordinates, and used paintbrushes and colors). The H2 node is composed of three child nodes:
   (a) The H2.1 node manages the toolbar, which is composed of several figures, paintbrushes, and colors. The H2.1 view-controller calls the corresponding model functions in order to highlight the current tools (e.g., figure, paintbrush and color) chosen by the local collaborator.
   (b) The H2.2 node manages the drawing area. Its view-controller calls the corresponding H2.2 model functions that calculate the 2D dimensions and coordinates of each figure and text displayed on the screen. The H2.2 view-controller communicate with its remote pairs in order to provide and obtain the productions accomplished by respectively the local collaborator and the remote ones. The H2.2 model saves each figure and text properties (e.g., type, outline, color, size, position and creator).
   (c) The H2.3 node is in charge of the group awareness bar. Its view-controller manages the collaborators' status (e.g., present, absent and available) in the collaborative session. Moreover, the H2.3 view-controller coordinates with its remote pairs to organize each collaborator's name, photo and status in order of arrival. The H2.3 model stores relevant information about collaborators (e.g., identifier, status and incoming/leaving time).
3. The H3 node manages the **redistribution meta-user interface with negotiation** (cf. Section 2). Its view-controller is activated if: 1) the collaborator logs on to the plastic collaborative whiteboard from another computer device; or 2) the collaborator explicitly requests the redistribution meta-user

interface. The H3 model memorizes the redistribution configuration of the user interface components selected by the local collaborator.

As we saw in Section 2, the **adaptation granularity** of an application determines how deep its user interface is going to be metamorphosed. In the case of our plastic collaborative whiteboard, the adaptation granularity is the **workspace** because: a) it is a suitable unit (just like the interactor grain) when remodeling and redistributing the application user interface to computers that own a reduced screen; and 2) from the user's point of view, the interaction interface is easier to use if the metamorphosis concerns a set of logically connected tasks rather than some unrelated interactors or the whole interface.

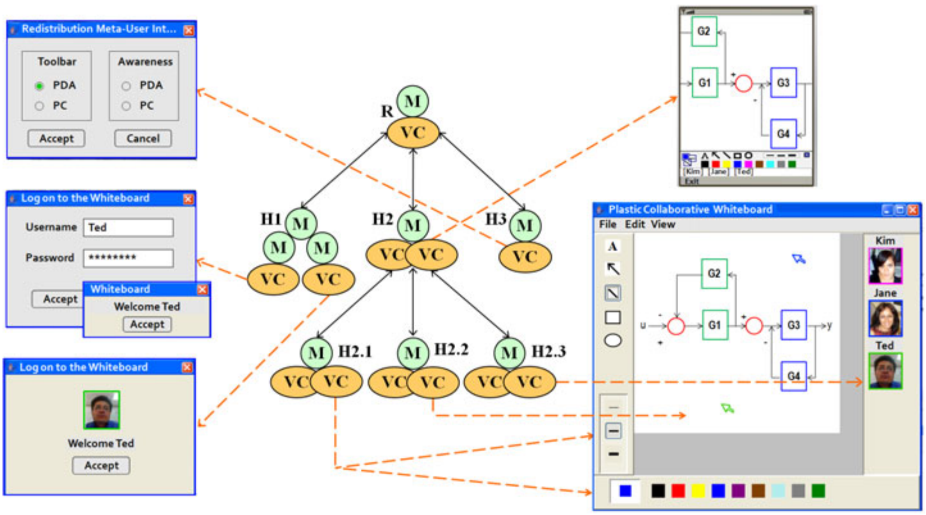


Fig. 1. The MVC-Based Architecture of the Plastic Collaborative Whiteboard

Regarding the H3 node, the plastic collaborative whiteboard supports the **user interface redistribution** categorized as *distributed organization to another distributed one* (cf. Section 2). The interaction interface state moves from: a) a *fully replicated state*, where all the workspaces (H1, H2 and H3 nodes) appear in the multiple devices used by the same user to log on to the working session to b) a *distributed state*, where the H2.1 and H2.3 nodes are hosted by one of the user's devices, according to his decision. Such a redistribution of the interaction interface aims to facilitate interaction (see Section 5).

The **context of use** (cf. Section 2) for the plastic collaborative whiteboard includes the *user* and *platform* elements, as it can adapt itself: 1) to the platform characteristics at starting time, and 2) to the collaborator identity when he is detected working from two computer devices. In the first case, the plastic collaborative whiteboard performs **inter-modal remodeling** (cf. Section 2) for the H1 node because, in computers equipped with a camera and the OpenCV (Open Source Computer Vision) library, the identification data only consists of the collaborator's picture that is automatically taken by OpenCV and processed by a face recognition system [7], which is in charge of identifying him; otherwise, the identification data only refers to the collaborator name and password. In the second case, when the collaborator is working from two computer devices, the plastic collaborative whiteboard performs **intra-modal remodeling** because it remains providing a graphical interaction support.

Once the interaction interface redistribution has taken place, the user's interrupted actions are fully restored: a **state recovery** support is provided at the *physical action* granularity (cf. Section 2) As we highlight in Section 5, the plastic collaborative whiteboard stores the user's actions performed on the H2.1 and H2.2 nodes. Remodeling and redistribution of the H2.1 and H2.3 nodes are performed on the fly, while the collaborative whiteboard is running. Thus, the **user interface deployment** is fully *dynamic*. As we discuss in the next section, the visible area (corresponding to the physic display of handheld devices) that is managed by the H2.2 node needs to be remodeled too.

4.2 Interaction Interface Implementation for Handheld Devices

Because application portability [4] is an important property of plastic interactive systems, we select the J2SE and JME to implement the collaborative whiteboard on heterogeneous platforms. Particularly, we use the NetBeans 6.8 IDE for: 1) PCs/Linux, 2) a SMARTBoard/MacOS, 3) a PDA HP iPAQ 6945/Windows Mobile 5.0, and 4) a smartphone HP iPAQ 610c/Windows Mobile 6.0. We also use the Kaffe JVM 1.1.9 for a PDA HP 3900/Linux Familiar. In addition, an access control mechanism based on computer vision was implemented using the OpenCV library and a face recognition system in C++ [7]. Thus, the implementation and adaptation of the plastic collaborative whiteboard encompass **multi-technological spaces** (as introduced in Section 2).

Most application developers usually design and implement interactive systems only for PC. However, when the device capabilities (e.g., screen size) are reduced, the management of some computer resources (e.g., display space) is especially difficult for groupware applications. Moreover, if collaborators are immersed in a multi-device environment, the implementation of the interaction interfaces becomes more difficult because the user's cognitive load should not be increased. Otherwise, the application usability might be put in jeopardy.

In order to satisfy this requirement, we implemented three prototypes of our plastic collaborative whiteboard for large and medium screens (e.g., SMARTBoard and PC) and five different prototypes for small screens (e.g., handheld), from which some were discarded. Thus, we conducted a basic usability study of these prototypes, which is based on questionnaires proposed to 25 master/PhD students of our institution during two working sessions (the first one lasted 30 minutes, whereas the other took 50 minutes). We firstly observed that, in the case of large and medium screens, the interviewees preferred the Microsoft Paint-like prototype because most of them already knew Microsoft Paint.

Selecting a prototype for small screens depends not only on the organization and appearance of the user interface components but also on their functionality to accomplish the planned tasks. Four of the five prototypes propose interaction interfaces including more than one window, whereas the interaction interfaces of the fifth prototype are made of one unique window. On this respect, we observed that most of the interviewees preferred this last one for two reasons: 1) they do not have to navigate through several windows; and 2) the number of pen clicks required to perform the planned tasks remains reduced.

Because the collaborative whiteboard implementation for big and medium screens is relatively trivial, in the next section, we describe the specific implementation for the smartphone iPAQ 610c and the PDA iPAQ 6945, as the aspects developed for these devices can be applied to any kind of handheld.

Collaborative Whiteboard Workspaces

The smartphone display surface, manipulable for programmers, is 240 width × 269 height px. If the OS menu bar located at the bottom is suppressed, the display surface height increases to 360 px, but the area occupied by this menu bar is not manipulable (see Fig. 2a). On the other hand, the PDA display surface is 240 width × 188 height px. At the opposite with the smartphone, the PDA area used by this OS menu bar can be configured. So, removing this menu bar, the display surface height rises to 214 px (see Fig. 2b). The upper part of this area is tactile, but the complement bottom one is not (i.e., writable but not readable). Thus, the tactile area may be increased to 240×195 px.

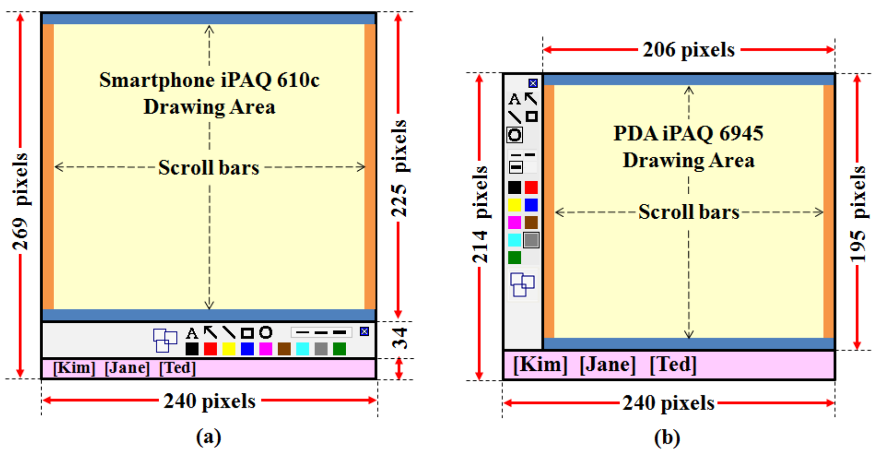


Fig. 2. Display Area Analysis for the Smartphone and PDA

The Fig. 2 also shows each workspace position within the display surface of the handheld devices. In both of them, the group awareness bar is shown in an horizontal way at the bottom of the display surface. Particularly, in the PDA, it occupies an area of 240×19 px in order to take advantage of the whole non-tactile area, as the group awareness bar does not need data input from the collaborator (see Fig. 2a). In the smartphone, the group awareness bar is reduced to 240×10 px to maximize the drawing area, while supporting homogeneous vertical scrolling jumps on it as we explain below (see Fig. 2b). The group awareness bar is always offered to the collaborator in order to provide him with up-to-date co-presence information.

The smartphone application toolbar is placed on top of the group awareness bar in order to reserve enough space for creating a quasi-squared rectangular drawing area, similar to the drawing area provided on computers with big or medium screen. Thus, the toolbar occupies an area of 240 width × 34 height px and is composed of two rows of interactors, e.g., figures, colors and paintbrushes (see Fig. 2a). By contrast, the PDA application toolbar is vertically placed on the left side of the display surface in order to define once more a quasi-squared drawing area. Thus, the toolbar uses an area of 34 width × 195 height px and contains two columns of interactors (see Fig. 2b). In all cases, the application toolbar can be temporarily hidden to make the visible drawing area larger.

Scrolling the Drawing Area

The drawing area comprises the surface unused by the previous workspaces, i.e., 240×225 px for the smartphone and 206×195 px for the PDA. However, we increase these dimensions in order to define the same drawing area for all device screens. Consequently, the drawing area requires: 1) both vertical and horizontal scrollbars to navigate across it, and 2) a buffer to manage the augmented non-visible information. It is important to mention that JME does not provide any primitives to create scroll bars. Thus, we provide four invisible scrollbars, one for each side of the drawing area: 1) two horizontal bars (at the top and at the bottom), that allow to move up and down; and 2) two vertical bars (at the left and at the right) that allow to move left and right.

To manage these scrollbars, a suited manipulation technique involves sliding the pen on the corresponding scrollbar (e.g., the horizontal top) towards the desired border (e.g., upwards) in order to move to such a direction. The drag and drop manipulation technique for traditional scrollbars is quite suitable for mouse computers, but when applying to pen computers, some users dislike the feeling of scratching the display surface with the pen tip [12].

The scrollbar implementation firstly entails verifying whether the handheld devices are able to acquire the coordinates when sliding the pen on the display surface. The PDA does not support it. So, scrolling only works when the pen taps on the area managed by each scrollbar. This limitation implies constraints for the design of the drawing area, which has to be reduced in order to implement the scrollbars. When the application toolbar is displayed on the PDA, the drawing area width is reduced from 206 to 180 px in order to reserve 13 px width for each of the right and left scrollbars. In the same way, the drawing area height is reduced from 195 to 175 px in order to reserve 10 px height for each of the top and bottom scrollbars. When the application toolbar is absent, the drawing area width changes from 240 to 225 px, whereas the left and right vertical scrollbars respectively measure 8 and 7 px width. This pixel width is sufficient enough to select and activate the scrollbar, while maximizing the drawing area and supporting homogeneous scrolling jumps.

As previously mentioned, the smartphone owns the capability to read coordinates. So, there is no need to reduce the drawing area. If the application toolbar is displayed (see Fig. 3a), the drawing area (240×225 px) is divided into 6 columns of 40 px each and 9 rows of 25 px each. Otherwise, the drawing area (240×250 px) is increased by 1 row, whereas the group awareness bar remodels itself by increasing its height from 10 to 19 px (like that of the PDA). On the other hand, if the application toolbar is displayed on the PDA (see Fig. 3b), the drawing area (180×175 px) is divided into 4 columns of 45 px each and 5 rows of 35 px each. Otherwise, the drawing area (225×175 px) is increased by 1 column.

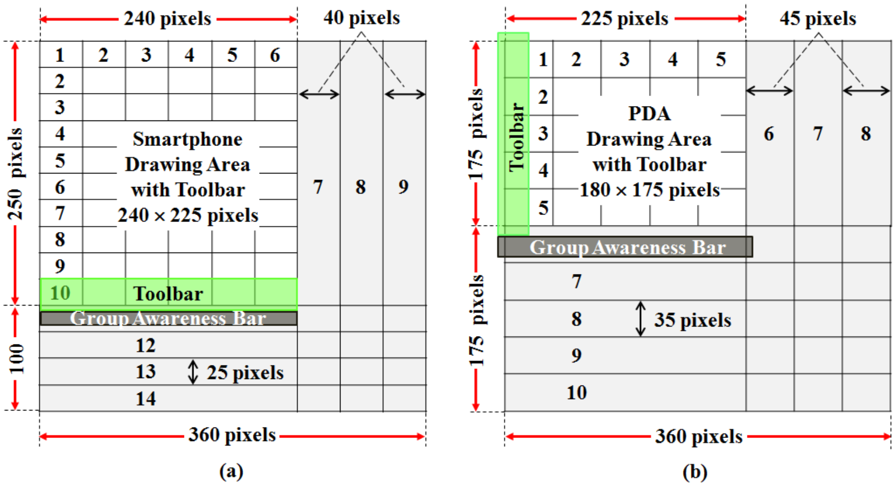


Fig. 3. Drawing Area Division for the Smartphone and PDA

Instead of scaling figures, we increase the drawing area of the handheld devices in order to make it the same as that for medium screen computers. The dimensions of the whole drawing area have been fixed to 360 width × 350 height px. Thus, the display surface of both smartphone and PDA has to be considered as a window the collaborator moves within the drawing area. To implement this window, the smartphone visible drawing area gains 3 columns and 4 rows (see gray area in Fig. 3a), whereas the PDA visible drawing area is increased by 3 columns and 5 rows (see gray area in Fig. 3b).

Thus, for instance, if the application toolbar is displayed on the smartphone, the collaborator has to slide the pen five times on the horizontal scrollbar located at the bottom of the drawing area, in order to see the content of the non-visible rows (the one hidden by the application toolbar plus the four augmented ones). Each time the collaborator slides the pen, the resulting vertical scrolling measures 25 px. However, when the toolbar is absent, the collaborator has to slide just two times (50 px per jump) because multiples of 25 px have been used to make scrolling easy for the collaborator.

5 Scenario of Use

To illustrate the plastic capabilities of the collaborative whiteboard, let us consider the following scenario: *Kim* logs on to the application from a camera-equipped PC/Linux connected to the wired network. The collaborative whiteboard first identifies *Kim*'s face [7] to authenticate her and then authorizes her to initiate a cooperative working session. The collaborative whiteboard displays its interaction interface within a unique view, which contains the three workspaces: 1) the application toolbar, 2) the drawing area, and 3) the group awareness bar. She recovers a document draft jointly initialized with her colleagues during a past working session. The group awareness bar indicates that *Kim* is currently the only collaborator present in the session.

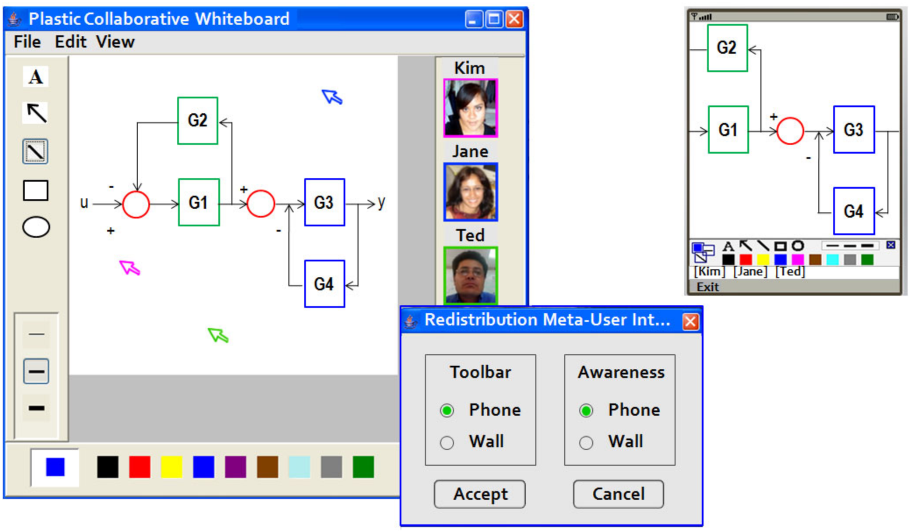


Fig. 4. The Plastic Collaborative Whiteboard on Ted's Wall Display and Smartphone

Few minutes later *Jane*, who is traveling by train, uses her PDA/Linux to log on to the collaborative whiteboard, which authenticates her by means of her collaborator name and password. After welcoming *Jane*, the collaborative whiteboard shows its interaction interface within a unique view that also contains the three workspaces. By means of them, *Jane* can perceive *Kim*'s presence and her document draft proposals. In a simultaneous way, *Kim*'s group awareness bar displays *Jane*'s photo, name and status.

Because *Jane* is using her PDA (see Fig. 2b), the group awareness bar is placed at the bottom of the view, where the names of the present collaborators are shown in order of arrival ("*Kim*" is before "*Jane*"). The application toolbar, situated above the group awareness bar, shows the tools (e.g., figure, paintbrush and color) selected by *Jane* just before logging out of the last session. At this point, the collaborative working session between *Kim* and *Jane* is established. Thus, when one of them draws on the whiteboard drawing area, the other can observe the effects of her actions in a quasi-synchronous way.

Some time later, *Ted* logs on to the collaborative whiteboard firstly from his wall-sized display/MacOS and then from his smartphone/Windows Mobile. The whiteboard instance running on the wall-sized display authenticates him via the face recognition system, whereas the whiteboard instance running on the smartphone identifies him via the well-known collaborator name/password technique. The group awareness bars of *Kim* and *Jane* show that *Ted* just logged on to the collaborative working session. In a symmetrical way, he perceives the presence of *Kim* and *Jane* (see Fig. 4). Then, *Ted* starts working with the same context (e.g., selections and tools) of the last session he left.

At the moment the collaborative whiteboard detects him interacting with two devices, it displays a redistribution meta-user interface (meta-UI) on the wall-sized display in order to invite him to participate in the plastic adaptation of his interaction interface (see Fig. 4). From this meta-UI, *Ted* selects the smartphone to host the group awareness bar and the toolbar, but moreover he decides to maintain the toolbar on the wall-sized display. As a result of this plastic adaptation, the smartphone hosts the group awareness bar and the toolbar, whereas the wall-sized display maintains the toolbar and the drawing area (see Fig. 5). Thus, the toolbar is displayed on both *Ted*'s interaction devices that allow him: 1) to produce in a more efficient way or 2) to eventually invite a colleague to take part of the collaborative document production.

We can observe that because the smartphone does not display the drawing area, the toolbar size has been increased allowing to offer more tools, whereas the group awareness bar can show each collaborator's name and photo (see Fig. 5).

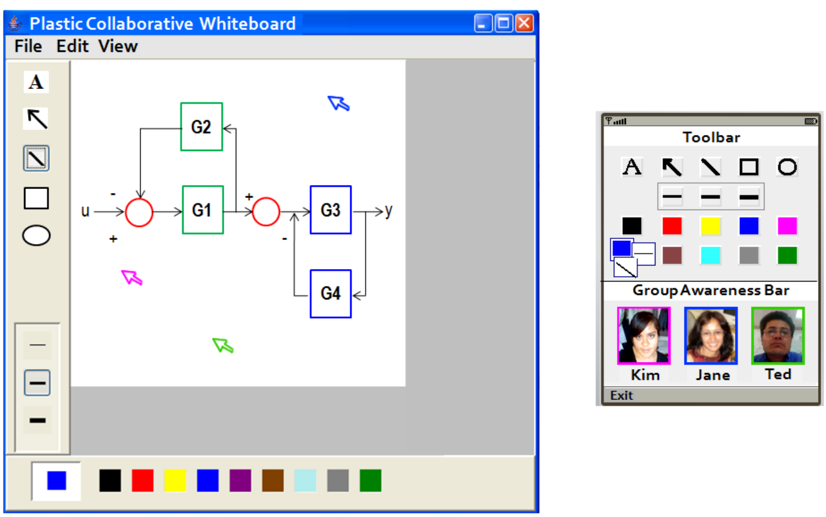


Fig. 5. The Plastic Collaborative Whiteboard After User Interface Redistribution

Putting the toolbar on a wall-sized display introduces several problems. For instance, in our scenario, *Ted* might not be able to reach the application toolbar at the top of the wall-sized display. By means of the multi-computer approach [12], he can use: 1) his smartphone, like a palette of oil-painting, to select a paintbrush type, a color or a figure and 2) his wall-sized display, like a canvas board, to draw. Like a traditional oil painter, *Ted* can tap on a color icon with his pen to change the pen color. As we can see, this multi-computer approach allows *Ted* to work with *Kim* and *Jane* in a remote way. Moreover, a *Ted*'s colleague might meet him in his office to participate to the collaborative session. In this case, both of them have a smartphone, but they are going to physically share the wall-sized display to produce the document draft.

6 Conclusion and Future Work

Nowadays, more and more colaborators are using different computer mobile devices in a dynamic way, e.g., laptops combined with a wall-sized computer as well as PDAs. Colaborators require functions to adapt and/or redistribute their interface of colaboration to be able a) to move and access the shared information from their heterogeneous mobile devices and b) to organize their work combining their devices to colaboratively produce in an efficient way.

The research topic of user interface plasticity is already being studied for single-user systems, where several concepts, prototypes and even reference models [2] have been proposed. However, in the domain of groupware applications, the adaptability of applications starts being studied as a side issue of the design and implementation of augmented reality techniques. Thus, these research efforts only focus on the platform, leaving aside the users and the environment.

Some of the factors influencing the user interface of single-user interactive systems (e.g., *state recovery* and *adaptation granularities*, *user interface deployment*, and *technological spaces*) can also be applied to groupware applications in order to provide them with the plasticity property. However, others factors (e.g., *context of use* and *redistribution meta-user interface*) need to be redefined/adapted to the particular requirements of groupware applications.

Particularly, the redefinition of the *context of use* must take into account: 1) a group of collaborators instead of a user only; and 2) their spacial interaction form (remote vs. face to face) in order to consequently adapt the user interface, e.g., suppression of the group awareness bar when some collaborators are co-located. However, the *redistribution meta-user interface* with plasticity appears of main concern for groupware applications because it has also to be adapted to the collaborative working context, e.g. some co-located colaborators are sharing a wall-sized computer and each one is interacting with a PDA: the meta-user interface should not show the drawing area as a redistributable UI component.

This work opens the research field of plastic user interfaces for collaborative work environments. From the results obtained from this research effort, we can logically imagine the possibility to define plastic generic concepts and mechanisms that can be adapted to the different kinds of groupware applications.

References

1. Balme, L., Demeure, A., Calvary, G., Coutaz, J.: Sedan-Bouillon: A Plastic Web Site. In: PSMD 2005, the INTERACT 2005 Workshop on Plastic Services for Mobile Devices, Rome, Italy (2005)
2. Calvary, G., Coutaz, J., Thevenin, D.: A Unifying Reference Framework for the Development of Plastic User Interfaces. In: Nigay, L., Little, M.R. (eds.) EHCI 2001. LNCS, vol. 2254, pp. 173–192. Springer, Heidelberg (2001)
3. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., Florins, M., Vanderdonckt, J.: Plasticity of User Interfaces: A Revised Reference Framework. In: TAMODIA 2002, First International Workshop on Task Models and Diagrams for User Interface Design, Bucharest, Romania, pp. 127–134. INFOREC Publishing House, Bucharest (2002)
4. Coulouris, G.F., Dollimore, J., Kindberg, T.: Distributed systems: concepts and design, 4th edn. Addison-Wesley, Essex (2005)
5. Coutaz, J., Calvary, G.: HCI and Software Engineering: Designing for User Interface Plasticity. In: The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications - Human Factors and Ergonomics Series, pp. 1107–1118. Taylor & Francis Press, New York (2008)
6. Crease, M.: A Toolkit of Resource-Sensitive, Multimodal Widgets. PhD Thesis, Department of Computer Science, University of Glasgow (2001)
7. García, K., Mendoza, S., Decouchant, D., Olague, G., Rodríguez, J.: Shared Resource Availability within Ubiquitous Collaboration Environments. In: Briggs, R.O., Antunes, P., de Vreede, G.-J., Read, A.S. (eds.) CRIWG 2008. LNCS, vol. 5411, pp. 25–40. Springer, Heidelberg (2008)
8. Giesecke, S.: Taxonomy of architectural style usage. In: PLoP 2006, the 2006 Conference on Pattern Languages of Programs, Portland, OR, USA, pp. 1–10. ACM Press, New York (2006)
9. Grolaux, D., Roy, P.V., Vanderdonckt, J.: FlexClock, a Plastic Clock Written in Oz with the QTk Toolkit. In: TAMODIA 2002, First International Workshop on Task Models and Diagrams for User Interface Design, Bucharest, Romania, pp. 135–142. INFOREC Publishing House, Bucharest (2002)
10. Kammer, P.J., Taylor, R.N.: An architectural style for supporting work practice: coping with the complex structure of coordination relationships. In: The 2005 International Symposium on Collaborative Technologies and Systems, St Louis MO, USA, pp. 218–227. IEEE Computer Society, Los Alamitos (2005)
11. Prante, T., Streitz, N.A., Tandler, P.: Roomware: Computers Disappear and Interaction Evolves. IEEE Computer 12(3), 47–54 (2004)
12. Rekimoto, J.: A Multiple Device Approach for Supporting Whiteboard-Based Interactions. In: CHI 1998, the 1998 Conference on Human Factors in Computing Systems, Los Angeles CA, USA, pp. 344–351. ACM Press, New York (1998)
13. Tandler, P., Prante, T., Müller, C., Streitz, N., Steinmetz, R.: Connectables: Dynamic Coupling of Displays for the Flexible Creation of Shared Workspaces. In: The 14th Annual ACM Symposium on User Interface Software and Technology, Orlando FL, USA, pp. 11–20. ACM Press, New York (2001)
14. Vanderdonckt, J., Calvary, G., Coutaz, J., Stanciulescu, A.: Multimodal User Interfaces: Models, Methods and Principles. In: Signals and Communication Technology, pp. 61–76. Springer, Berlin (2008)

Developing a Framework of Common Information Space (CIS): Grounded Theory Analysis of Airport CIS

Nallini Selvaraj and Bob Fields

Interaction Design Centre, School of Engineering and Information Sciences,
Middlesex University, London, UK
{N.Selvaraj,B.Fields}@mdx.ac.uk

Abstract. The notion of Common Information Space (CIS) has been proposed in the field of Computer Supported Cooperative Work (CSCW) as a conceptual framework for analyzing cooperative work processes. The area is still in its formative years and requires more research to contribute to its development. This paper presents findings from an investigation undertaken for such an endeavor. Three perceptions of CIS are presented, which are, CIS as a socio-technical arrangement, dynamic arrangement, and dependency management arrangement. These have been derived from review of existing research contributing to CIS notion development and Grounded Theory analysis of collaborative work process in air traffic control setting. The findings presented in this paper provide a comprehensive and consolidated view of the notion development. The paper contributes to the ongoing discussion of CIS notion development by making theoretical as well as methodological contribution.

Keywords: Computer Supported Cooperative Work, Articulation Work, Common Information Space, Workplace Studies, Air Traffic Control.

1 Introduction

Modern work settings are collaborative ensembles that entail complex work processes and diverse social activities. Work is distributed among multiple personnel with dependencies between their undertakings. In order to manage the dependencies, personnel involved in the work process have to cooperate with each other by what is known in the field of Computer Supported Cooperative Work (CSCW) as "articulation work". Articulation work is the work undertaken to manage dependencies in the work process by *coordinating, scheduling, aligning, meshing*, etc. of distributed individual activities [1-3].

In the past few years research has been undertaken in the field of CSCW to provide support for articulation work through the construction of information spaces which are viewed as communication spaces or interaction spaces [4]. These spaces support collaborative work activities by facilitating communication through information exchange and information sharing. A number of terms have been formulated to represent them such as media spaces, shared workspace, shared information spaces, shared and common communication spaces, and more recently common information spaces [5]. These concepts are still evolving and are needed because of their significance in the discussion of features of cooperative work.

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 281–296, 2010.

One of the first discussion about the significant position of such information spaces in the articulation of cooperative work was under the label of "shared information space" by Bannon and Schmidt [5]. In a subsequent paper [4] they extended this concept under the label of "Common Information Space" (CIS). The difference between the two is in the perception towards the role of such information spaces in cooperative work process. In the case of shared information space, focus is on articulating cooperative work by using artefacts to mediate communication. However, in the case of common information space, the focus is not just on interaction through information sharing but also on establishing common understanding of the information held and propagated in such spaces. According to Bannon, one of the reasons for the shift in terminology is *to lessen the connotations associated with the word 'sharing' and to indicate the transient and instrumental aspects of people having information in 'common* [6].

Development of the notion of CIS is still in the formative years especially with respect to CIS for collaborative work across heterogeneous work communities. The focus of this research is to contribute to the notion development. In this paper, we first present a review of existing research contributing to CIS conceptualization through a simple framework. Our contribution to the development of the notion is then depicted by extending the framework through an empirical study conducted in the air traffic control work environment and Grounded Theory analysis of the collaborative work process of this setting.

2 Notion of Common Information Space (CIS)

There is a growing realisation lately that the complexities involved in a collaborative work ensemble such as dynamic interaction, distributed decision making, heterogeneous worker/group collaboration, etc. cannot be handled by just supporting information sharing or pooling information from multiple sources. Rather, there is also a need to incorporate an interpretive element to this process. Common Information Space (CIS) is a notion germinating in this evolution where the focus is on placing information in common as well as establishing common interpretation or at least "common enough interpretation" to achieve efficient task performance [4].

In the field of CSCW, CIS has been proposed as a concept for analysing cooperative work. Schmidt and Bannon introduced the concept of CIS to point out that information has to be "placed in common" explicitly involving creation in one context and usage in a different context by reformulating and re-contextualizing it to be relevant in latter [4]. Therefore, CIS does not represent just a repository of information to which people have common access but also how they incorporate it in daily usage and integrate it into the work practice.

In general, the notion of CIS focuses on the interrelationship between actors, artefacts, information, and cooperative work. Review of literature in this area reveals that researchers from various disciplines have discussed different aspects of CIS. Because the concept is still in its early stages of development there exists diverse perception towards the notion. The next section presents a framework constructed from a review of research leading to these varied conceptualizations of CIS. The framework is intended to help understand the concept development by synthesizing and organizing these diverse perceptions of CIS along two main attributes, which are, CIS as a socio-technical arrangement and CIS as a dynamic arrangement.

3 Framework of CIS Conception from Existing Research

Studies undertaken for developing the notion of CIS have focused on specialised cooperative work settings such as; air traffic control tower and software company [7], bank, football competition, and museum [6], hospital ward [8-10], airport [11], and oil and gas company [12]. While reviewing these studies two fundamental questions were addressed: What are the pertinent questions being addressed in the research, and How are the findings conceptualized? Some of the questions driving research in this area were found to be: How should CIS be conceived? What are the characteristics of CIS? and How can the notion of CIS be applied to the analysis and design of cooperative work arrangements?

Two main perceptions of CIS transpire from these studies and their findings. They are CIS as *a socio-technical arrangement* and as *a dynamic arrangement*. Conceptualizations from various research undertakings have been classified to formulate these two perceptions of CIS, as depicted in Fig.1. The three conceptualizations of 'Artefact as CIS', 'Workspace as CIS', and 'Achieved in Practice' contribute to the socio-technical arrangement perception and 'Malleable', 'Situated' and 'Temporal' contribute to the dynamic perception of CIS.

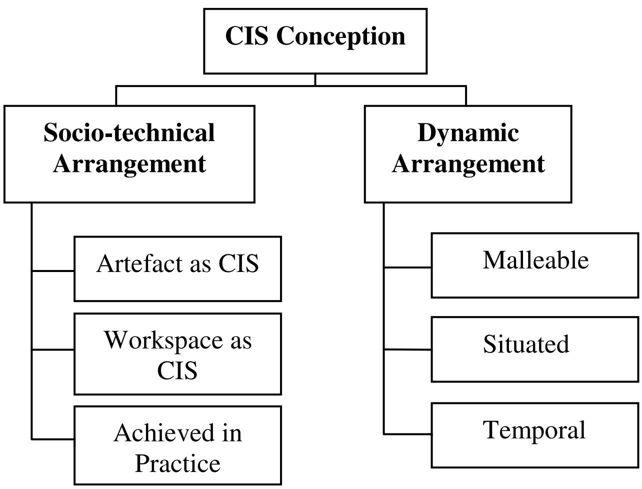


Fig. 1. Framework of CIS Conception

The framework is illustrated next through discussion of conceptualizations evoking the two perceptions of CIS. Table 1. presents conceptualization of CIS from various research undertakings informing the two perceptions of CIS depicted in the framework.

Table 1. Studies contributing to the perception of CIS as a socio-technical and dynamic arrangement

	Socio-Technical Arrangement			**Dynamic**		
Concepts / *Studies*	**Artefact as CIS**	**Workspace as CIS**	**Achieved in Practice**	**Malleable**	**Situated**	**Temporal**
Schmidt & Bannon '92[4]	X		X		X	
Clement & Wagner '95[13]			X	X		
Bannon & Bodker '97[7]		X	X	X	X	
Randall '00[14]	X		X			
Bertelsen & Bodker '01[15]		X	X		X	
Reddy et al. '01[8]	X		X	X		X
Bossen '02[9]		X				
Fields et al. '04[11]		X	X		X	
Rolland et al '06[12]	X	X	X	X	X	X
Munkvold et. al. '07[10]		X	X		X	X

3.1 Socio-technical Arrangement

The technological arrangement of the work setting along with the social practices of personnel functioning in the setting plays a significant role in the construction and maintenance of CIS. This section presents how different researchers have conceptualized such an arrangement of CIS.

Artifact as CIS. Schmidt and Bannon [4] conceptualize information artefacts as CIS by illustrating how these artefacts maintain a central archive of organizational information as well as disseminate information to cooperating actors. To function as a common information system artefacts need to be not only robust but also easily and quickly accessible to users from different background [14]. Reddy and her colleagues [8] explore how information propagated by a computerized patient record is incorporated into the work practices of a hospital intensive care unit where different representations of the same underlying information are provided to different work groups depending on their needs. However, additional work is required to synchronise interpretations. The role of common database as a CIS to share knowledge across different heterogeneous context has been explored by Rolland and other researchers [12] in a large oil and gas company. Although the database performs this function to some extent there are inherent limitations and barriers of such a system for achieving CIS across heterogeneous settings, for example, problems in interpretation due to lack of contextual knowledge and creation of new forms of fragmentation.

Workspace as CIS. The depiction of workspace as CIS varies with the work setting, i.e. when the collaborating actors are collocated and when they are distributed. Bannon and Bodker [7] present the workspace as CIS when actors are physically copresent. For example, the workspace of air traffic control room in an airport is a CIS constituted by the amalgamation of information artefacts, physical behaviour such as speaking aloud and gestures, visual observation, and openness of actions. Such a setting facilitates establishing common understanding of the field of work because of the physical co-presence of those working together. Rolland et. al. [12] provide a different take on CIS for collocated actors by presenting ‘collaboration rooms’ as a socio-technical arrangement where the arrangement of collaboration technologies in the room needs to be constantly re-negotiated to render a CIS unlike that of the airport control room which consists of stable arrangement of collaboration technologies.

In the case of distributed work setting, Bertelsen and Bodker [15] have challenged the notion that CIS is about access to everything everywhere by depicting the wastewater plant setting as a common artefact. They conceptualize the workspace as having several centres and peripheries and composed of overlapping regions, where establishing commonness of information is attributed to mobility within the workspace. Akin to this perception, Bossen [9] takes a broader perspective on CIS by portraying a hospital ward as massively distributed CIS and a common artefact like the wastewater plant. He has developed a framework of seven parameters to analyse the workspace as CIS. He attributes establishment of shared interpretations to not just the physical proximity of those involved but also to the number and means of communication available to people. In a similar setting, Munkvold et. al. [10] explore the infrastructural arrangement such as the electronic nursing module, inter and intra disciplinary discussions, conference room arrangement, and human mediators contributing to the establishment of CIS of a hospital ward. Taking the perception of overlapping regions in CIS further, Fields et. al. [11] depict the work environment of an airport as a constellation of overlapping CISs that are articulated through boundary objects.

Achieved in Practice. One perception common to all research undertakings in this area is the view that CIS is achieved in the daily practices of actors in the work process. Randall [14] does not consider technology to be the defining feature of CIS. Rather, it is the coalescence of pre-existing habits and practices of varying groups and individuals that establishes the commonness of information. CIS is jointly constructed and maintained by actors of the cooperating ensemble in a manner not necessarily constrained by prescribed procedures and conventions [4]. Besides sharing information additional work by actors such as incorporating contextual knowledge is required to establish common interpretation of shared information. Clement and Wagner [13] consider providing communication spaces to establish and maintain CIS. These electronic communication facilities allow actors to perform necessary negotiations by allowing them to rearrange the communication spaces according to changing needs. Bannon and Bodker [7] perceive CIS to be negotiated and established by actors involved where physical co-presence has an edge over spatially distributed actors and also mutual intelligibility of actions plays a significant role. They provide a different take by prescribing the use of human mediators to facilitate common interpretation of information by both producers and consumers of the information.

By shifting the focus from co-located control room like settings to cooperation in geographically dispersed settings of waste water treatment plant, Bertelsen and Bodker [15] present how CIS is established through the movement of people around the wastewater plant and through learning, participation, and experimentation. In a hospital setting even though people are not as dispersed as in the wastewater plant, Reddy et al. [8] illustrate that the benefits of collocation is lost due to the diverse work practices of different groups. In order to establish sufficiently common understanding of shared information to carry out individuals tasks, people have to discuss, exchange, and compare different representations of the same information. Negotiations of information interpretation are carried out informally during the course of work. It takes place during groups meetings by exchanging information about local work practices, thereby helping to gain better understanding of how changes made to information representation will affect other's work. Fields et. al. [11] emphasize the fact that commonness of information is achieved not by just having the information present and available but also in being able to build a 'common picture' by coordinating it with other elements in the setting. CIS is performed through the practices of those involved by switching between different alternatives and types of information representing the same phenomenon as well as by negotiating meanings held by the different representations [12].

The three conceptualizations presented above reveal how different researchers perceive the socio-technical arrangement of CIS. The perceptions vary depending on the work setting with varying focus on the technology, people, and work practice.

3.2 Dynamic Arrangement

Malleable. Researchers argue that mutable objects play a significant role in establishing CIS across heterogeneous context. For example, the technological arrangement of the collaboration room can be improvised according to the needs of the collocated and virtual participants involved in the discussion [12, 13]. Clement and Wagner [13] put forth the idea of integrating flexible regionalization into technical facilities by allowing actors to *erect, shift, blur, harden, dissolve, and strengthen boundaries of communication spaces*. Bannon and Bodker [7] have conceptualized CIS to be of open and malleable nature that allows translation and portability of information across boundaries where local contexts are re-established. In a similar light, Reddy et al. [8] have illustrated the importance of information malleability in a work setting by presenting how different representations of same underlying information help different work groups to coordinate their activities by de-contextualizing and re-contextualizing information as needed.

Situated. The notion of CIS is founded on the premise that emphasizes the importance of establishing common interpretation and not just sharing information to facilitate articulation work [4]. Interpretation of information however, takes place locally and on specific occasions of use. Bannon and Bodker [7] illustrate the situated nature of CIS by describing the degrees of openness and closure required with varying settings of CIS – i.e. when CIS is constituted for physically co-present actors or for those cooperating at "arms length". The situated nature of CIS is depicted for collocated settings by Rolland et. al. [12] who illustrate how by being present in the collaboration room during discussions provides additional context for interpreting information represented on various artefacts in the room. Someone walking into the room after the discussion ends might not be able to make complete sense of the representations. Emphasis is placed by Randall [14] on the need for understanding organizational context in which CIS has to operate because common information would be required by different actors with multiple work practices. In a geographically dispersed setting such as the wastewater plant, *overview, predictability and peripheral awareness are all related to how people move about in the plant, and not to a particular location*

[15]. Fields et. al. [11] place importance on contextualizing information to form common understanding through various means such as visual, verbal, and physical conduct, coordinating information from a number of sources and representations, and pre-existing common ground.

Temporal. Reddy et. al.[8] depict the temporal nature of CIS by presenting the *retrospective* and *prospective* attributes of a common information artefact and emphasize the importance of mediation between the two perspectives in order to render it into a CIS for different groups. Also, in case of CIS for heterogeneous groups, sharing and negotiation of common understanding is temporary and fluid where momentary understandings are achieved on specific occasions and is short-lived [12]. Rolland et. al illustrate this through the way a 'collaboration room' is a temporary arrangement that exists only for a short period of time as a CIS for the duration of discussions taking place in the setting. Munkvold et. al. [10] illustrate the temporal dimension of CIS through the temporality involved in the multiple trajectories of patients, doctors, nurses, and technologies. For example, the medical record evolves over time during a patient's illness trajectory that refers to past, present, and future. These are disconnected trajectories that briefly intersect where people from different work practices coordinate their activities.

From the above review of the work conducted in this area, we can infer that there exist quite varied and dispersed views on the characterization of CIS. The investigation being currently undertaken is an attempt to contribute to the development and clarification of the notion of CIS. This is done by studying how personnel from different work communities collaborate to manage various dependencies arising in the course of accomplishing tasks leading to a common goal. Also, from the review it was observed that most of the conceptualization of CIS was based on ethnographic studies. We feel that a more rigorous process of investigation is required to develop the notion of common information space, which is being addressed in this paper. In the next section we present the empirical investigation informing our contribution to CIS conception.

4 Data Collection and Analysis

The underlying principle of this research is that it is important to understand collaborative work process in its natural setting to inform the development of Common Information Space (CIS). In a collaborative work process there are many interacting elements. To explore such work processes the researcher has to obtain a practitioner's perspective of the system by situating oneself within everyday work activities. Various researchers [17, 4, 7, 16] have been advocating the importance of understanding phenomenon in a work process as it occurs in the real work setting in order to provide appropriate support for it. This research takes the qualitative approach because the study requires a methodological approach that would facilitate comprehending human behavior in a socio-technical context involving the three elements of human being(s), technical artifact(s) and context of use.

4.1 Study Site

The domain of interest for this research is the work process of Air Traffic Control (ATC) and in particular work taking place in an airport. The study has been conducted at a medium-sized single runway airport in the UK. The focus has been on collaboration between different work communities in and around the airport especially between personnel in the control tower, approach control, operations centre, and pilots. The focus was on these work communities because they have to collaborate with each other and share technological information systems to manage traffic movement in and around the airport. The control tower and operations centre setting was the direct field of this study while working of the other two work communities, pilots and approach control, was perceived from these two work settings.

4.2 Data Collection

Data for the research was collected through field studies. A series of studies have been undertaken at the airport over the last three years. Data was collected through ethnographic techniques of interviews, observation, field notes, collecting organizational and technical documents as well as literature on the field site. Formal and informal observational studies were undertaken in the study site. This involved taking notes of observed phenomenon and informal discussions with personnel about the observations made. Field notes contained information on environmental setting, behaviour of people, work practices, and questions arising from observations made. Data was also collected by conducting semi-structured interviews with personnel in the work communities of control tower and operations centre. Concurrent protocol was employed where participants were asked to talk through what they were doing while they were working. All the interviews and verbal protocols were recorded and later transcribed into text for analysis. Besides getting first hand data from the site, several secondary sources of data were obtained. This included organization and technical documents, studies conducted by others in the area of ATC, and literature on the field site.

4.3 Grounded Theory Analysis

This research employs the Glaserian [18] approach to Grounded Theory methodology application. Data is conceptualized through coding which is the foundation of Grounded Theory development. Glaser prescribes coding through the phases of: Substantive Coding, Theoretical Coding and Selective Coding [19] all of which is employed in this research. These phases are not entirely linear and work in conjunction with each other. The Grounded Theory process is both inductive and deductive. Inductive, as instead of starting with a hypothesis or theory, relevant theoretical concepts are allowed to emerge from the data during the coding and categorization process. *Deductive work in grounded theory is used to derive from initial codes as to where to go next in order to sample for more data to generate the theory* [19]. This is a cyclic process where the researcher goes back and forth between induction and deduction. A more detailed illustration of Grounded Theory methodology implementation in this research can be found in another paper [20] by the authors .

5 CIS Framework Extension

The notion of CIS focuses on the relationship between actors, artifacts, information, and cooperative work. The relationship between these elements have been portrayed in various research undertakings mainly by how information is represented and propagated through information artifacts and how it has been integrated into the daily work practices of personnel functioning in cooperative work settings, as depicted in the framework presented in section 3. We extend this framework through Grounded Theory analysis of the collaborative work process involved in the functioning of an airport setting.

In this research, we analyze the relationship between the four constituting elements of CIS in the course of managing interdependencies in the work process. There are two contributing factors for incorporating this perception in developing the notion of CIS. One is that the Grounded Theory analysis brings forth the centrality of the interdependency concept in the analysis of collaborative work. The other is that the notion of interdependence is at the core of cooperative work [4] and therefore it should form the crux of CIS which is aimed at supporting cooperative work.

The airport setting shares features of cooperative work settings investigated previously in studies contributing to CIS notion such as collocated actors, geographically distributed workspaces, arrangement of collaboration technologies, and the need for establishing and maintaining sufficiently common understanding of the field of work. The concepts generated during the Grounded Theory analysis are used here to extend the framework, which is presented in Fig. 2.

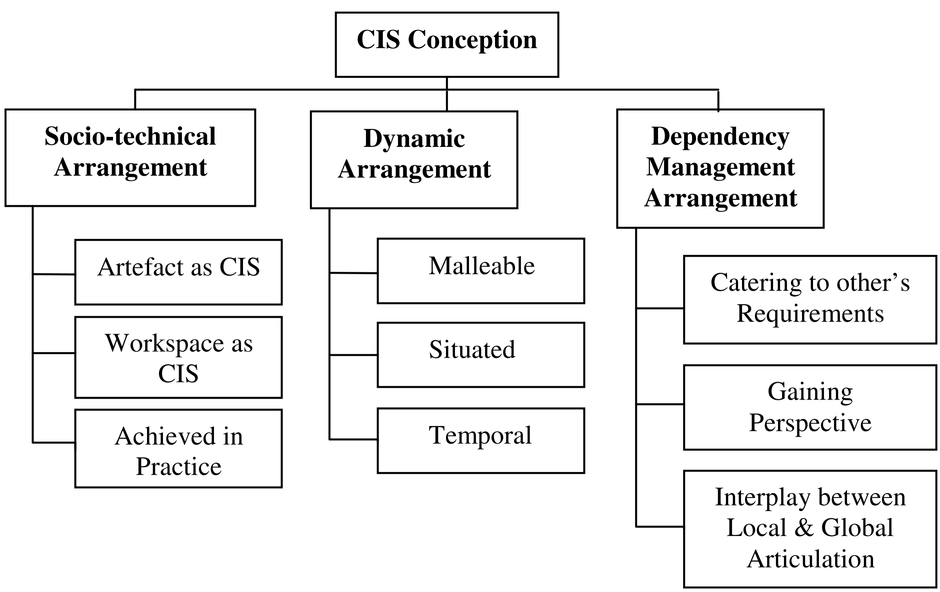


Fig. 2. Extended Framework of CIS Conception

In the following sections, two perceptions of CIS – as a socio-technical arrangement and dependency management arrangement - are described. The dynamic arrangement perspective is not described in detail here due to want of space. However, from the data analysis all the three concepts of malleability, situatedness, and temporality have been concurred. The descriptions are illustrated through interview transcripts from fieldwork data and the categories and properties generated during the Grounded Theory analysis.

5.1 Socio-technical Arrangement

Based on the Grounded Theory analysis, the workspace of work communities in the airport setting is perceived as CIS. It was found that the technological artifacts employed in these workspaces perform various mediation roles based on the practices by which information presented by them was put to use by those sharing it. Hence, the socio-technical arrangement entails the workspace and the practices by which CIS is established and maintained, which is illustrated in the ensuing sections.

Workspace as CIS. In the airport four work communities were studied, which are, the control tower, operations centre, approach control and pilots. The work communities are geographically distributed with the control tower, operations centre, and pilots located in the airport and the approach control outside the airport. The airport consists of multiple CISs where each work community's workspace setting is a CIS. This is because creating a CIS is not attributed just to the technology populating the setting but requires an amalgamation of various information resources of the workspace, procedural knowledge, responsibilities attributed to the roles of the personnel within the work community, benefits of collocation for observation, discussions, negotiations, and work practices established within the work community. The CIS of the airport then is a constellation of these overlapping CISs. This is because the work communities do not function independently. They are highly dependent on each other to manage the traffic movement in and around the airport. For example, the following transcript illustrates the dependency between the ground controller in the control tower and the departing aircraft pilot

"The first thing that you have to give is the Departure Route, which is his (aircraft pilot) clearance to move. No aircraft can go anywhere without a clearance. They need to know where to go basically and if you don't give them a point where to go and where to go from and where to go to and a route, they are in limbo. Basically that's what it is. You have to tell him (aircraft pilot) where to go. Otherwise he is going to come up to you and say 'what do I do? What stand am I? Which way do you want me to go?' So clearance is the main part of what we do when we are issuing instructions and this clearance is his permission to travel from here to his destination."

Here an overlap of information space is created between the CIS of the control tower and that of the aircraft pilot. Both the ground controller and the pilot need to establish common understanding of appropriate action in this context. By giving the pilot permission and direction to move around the airport the ground controller creates a brief overlap in the common information spaces until the aircraft has departed from the airport and control of the aircraft has been handed over to the approach control.

Achieved in Practice. The coding process revealed four main types of interdependencies between the different work communities: procedural dependency, information dependency, situation dependency and time dependency. This is elaborated in another paper by the authors [20]. The dependencies are managed through various social acts of coordination and cognitive acts of coordination. These acts of coordination are the categories in the grounded theory analysis, the properties of which reveal the practices by which CIS is established between work communities. The following table (Table 2) presents the categories and their properties. The codes within double quotes are in-vivo codes where the actual words of the interviewees are used to label the codes.

Table 2. Categories and Properties of Social and Cognitive Acts of Coordination

Category	Properties
Social Acts of Coordination	"Watching what's going on", Keeping People in the Loop, Requesting, Verbal Announcement, Exercising Authority, Helping Others, Sharing information, Sharing Responsibility, Act in Response, "Providing Required Information at right time", Verbally Concurring Course of Action
Cognitive Acts of Coordination	Expectation about Other's Behavior, Deciding Priority of Action, "Changing the plan quickly", "Making Informed Decisions", Planning

It is beyond the scope of this paper to illustrate and describe the properties of the categories. The following transcript provides an illustration of the two properties – Sharing Information and "Providing Information at the Right Time" - of the social acts of coordination category.

"When I give an aircraft pushback or annotate it with an active sign, the Assistant at the approach control will put the strip (flight progress strip) in front of the Coordinator. When it (aircraft) taxis out to the holding point, our Assistant in the control tower will then put a hold and again take-off on the screen (on her Departure Status Information screen)"

Information is shared by making changes to the common information system which allows it to act as a device for intermediating coordination of actions required for the task performance both within their respective work communities as well across communities. The information representation also depicts various aspects of work performance such as contextual information (status of aircraft departure), decisions made by the controller in the control tower (give permission for aircraft pushback), and task performance status (aircraft pushback, aircraft taxiing). By incorporating contextual aspects in information representation the system allows personnel in both work communities to gain common perspective of work being undertaken, thereby acting as a device for articulating interpretation. It also acts as a device for organizing coordination because when information is changed in the system by the controller in the controller tower it acts as an indicator or trigger for the Assistant in the approach control to take action.

5.2 Dependency Management Arrangement

Three main categories were identified in the grounded theory analysis that illustrates how CIS acts as a space for managing dependencies. Hence, the CIS of the airport is perceived as an arrangement for managing various dependencies arising in the work process. The three categories are:

Catering to other's Requirement. One of the main consequences of the two acts of coordination (presented in section 5.1.2) is catering to other requirements in terms of information, procedural compliance, or just helping each other out in performing tasks. An example illustration of this aspect of CIS is presented in the following transcript.

> *"……He (helicopter pilot) talks to me (Ground Controller)to start with to turn the engines on. I'll give him permission to lift, I don't clear him to take-off over there because you have to be very careful about that…because if something does go wrong. To be honest I can't give clearance to anything there. So I will just get him off the ground and transfer him to the tower and the tower once they know taxiway delta is clear will give him clearance to take off."*

This is an example of a situation where a police helicopter has to take-off from the airport. The police helicopters do not have to file a flight plan in advance. They can take-off whenever they are required to and as soon as possible. So when the pilot of the helicopter decides to leave the airport he calls the Ground Controller (GC) on his radio telephone frequency, and requests permission to start engine and move to taxiway. The GC will grant him permission to start his engine, lift and move to taxiway delta after ensuring that there are no movements on that area of the airport. Then he transfers control of helicopter to the Tower Controller (TC). After that, the TC will decide if he can grant the helicopter pilot permission to take-off from taxiway delta. This will depend on the traffic situation on the runway. As far as possible the TC will try and suspend traffic that might get in the way of the helicopter taking off. Also, under normal circumstances taxiway delta is under the control of the GC whereas in this situation the TC will take charge of movement on the taxiway. In this case the helicopter is not taking-off from the runway but from the taxiway. It is a crucial position to take-off from because there might be aircraft that have to move towards the runway from their stand in the Apron area or there might be those that are coming into the taxiway from the runway. Also, in the airport there are 'free range' vehicles that are allowed to move freely under the aerodrome authority's permission. The conversation taking place between the helicopter pilot and the controllers is broadcasted on the radio frequency which is available to these 'free rangers'. Once they know that the helicopter is planning to take-off from taxiway delta they are expected to keep away from that taxiway and the runway. If they need to go on or near the runway they have to get permission from the TC.

So if anything goes wrong with the helicopter taking off, according to the organizational norm the tower controller would be held responsible for the situation. Under normal circumstances however, the ground controller and tower controller coordinate their actions and make decisions about how and when it is appropriate to allow the helicopter to take-off. Therefore, the responsibility for resulting actions is now shared between the two controllers, at least under social conventions. This shared accountability now creates a context where those involved help each other by catering to other's requirements.

Gaining Perspective. Another aspect of CIS that enables it to act as an arrangement for managing dependencies is by facilitating those involved to gain common enough perspective on state of the work environment. Table 3 presents the properties of the category 'Gaining Perspective' generated in the Grounded Theory analysis.

Table 3. Categories and Properties of Category - 'Gaining Perspective'

Gaining Perspective	Synthesizing Multiple Information Sources, Mutual Intelligibility of Action, "Get tuned to each other", Common Practice, Identifying Information Availability, Notifying Information Availability, Anticipation, "play by the rules", "being proactive", Justifying One's Action, Updating, Verbally Concurring Course of Action, Determining Prospective Environmental Conditions, Determining Task Performance Status, Determining Temporality

The following interview transcript illustrates various properties of this category.

"Ground Controller - *(pointing outside to an aircraft in its stand) ....you see the guy (ground staff) walking over there now unplugging the leather flaps while he talks to the pilots and then we will be expecting him to taxi any minute now…any second now"*

This transcript was coded for the following properties "Get tuned to each other", "being proactive", Mutual Intelligibility of Action, Determining Task Performance Status, and Determining Temporality. The transcript illustrates the point that even though the two communities of control tower and aircraft pilots are geographically distributed they are still within visual range. The ground controller in the control tower is able to establish a sufficiently common understanding of events taking place in the work environment by proactively looking for information in the workspace to determine other's task performance status. He is able to infer the consequence of the ground personnel's actions in relation to the tasks performed by ground controller. He is able to do so because of Mutual Intelligibility of Action enabled by procedural knowledge.

Interplay between Local and Global Articulation. The overlapping CISs interweave local and global articulation required to collaborate across heterogeneous work communities. Local articulation is the work taking place within each work community to manage traffic movement and global articulation is the activities taking place between the dispersed work communities to manage interdependencies in the work process. The data analysis reveals that both local and global articulation needs to be addressed together. The perception of overlapping interdependent CISs in the airport addresses this local-global association. This is illustrated in the following scenario:

"We (Ground Controller (GC)) may have an aircraft that goes out to the hold and wants to get back to the stand, we may go to them and quickly and say 'can you go back to stand five'.....most of the time coordination with the Apron (in the operations centre) would be done through the Assistant"

In this case, a departing aircraft was waiting to take off near the runway but could not due to technical problems and wanted to go back to the stand in the parking lot. To perform this task, the ground controller in the control tower needs to articulate activities both locally within the control tower and globally across the work communities of pilots and operations centre. The actions taken to manage this situation are depicted below.

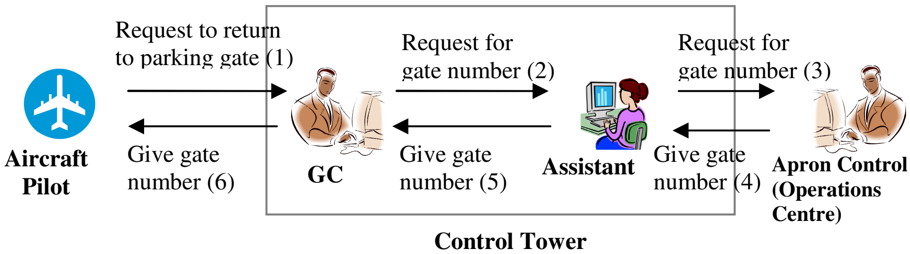


Fig. 3. Interplay between Local and Global Articulation Work

6 Conclusion

In this research we are endeavoring to contribute to the development of the notion of Common Information Space (CIS) through a Grounded Theory driven investigation. We have does this by placing the findings of the empirical investigation against conceptions from previous research. The notion of CIS is still in its formative stages and diverse conceptions of this notion have been developed by researchers. Considering these varied conceptualizations contributes to the notion development in various ways, such as

- It suggests that none of these present a coherent story
- Provides different starting points for analyzing the CIS notion
- Presents different insights into how CIS can be conceptualized

We have developed a framework to consolidate the different conceptions of CIS to provide an integrate representation of this notion. This helps to analyze conceptions in relation to others and provides a comprehensive insight into the development of this notion. We have extended this notion by incorporating another perception of CIS, which is CIS as Dependency Management Arrangement. This has been illustrated by describing how the overlapping interdependent CISs in an airport help cater to the requirements arising out the various dependencies in the work process, gain sufficiently common perspective of the work setting to perform interdependent tasks, and carry out the interplay between local and global articulation required to collaborate across work communities. Research informing the development of CIS notion hitherto has been undertaken primarily through ethnographic studies based on which inferences are drawn. In order to develop the notion of CIS there needs to be a systematic and rigorous methodology steering the theory development process. This is addressed in the investigation by employing Grounded Theory methodology to provide a systematic approach to conceptualizing CIS. This research makes theoretical as well as methodological contribution to the development of the notion of CIS.

Acknowledgements. We would like to thank NATS, U.K. for granting permission to conduct the field studies in the airport. We would also like to thank the reviewers of the conference for their valuable feedback and suggestions.

References

1. Gerson, E.M., Star, L.S.: Analyzing Due Process in the Workplace. ACM Transactions on Office Information Systems 4, 257–270 (1986)
2. Strauss, A.: The Articulation of Project Work: An Organizational Process. The Sociological Quarterly 29, 163–178 (1988)
3. Schmidt, K.: Cooperative Work and its Articulation: Requirements for Computer Support. Travail Humain 57(4), 345–366 (1993)
4. Schmidt, K., Bannon, L.: Taking CSCW Seriously: Supporting Articulation Work. Computer Supported Cooperative Work (CSCW): An International Journal 1, 7–40 (1992)
5. Bannon, L.J., Schmidt, K.: CSCW: Four Characters in Search of a Context. Studies in Computer Supported Cooperative Work: Theory, Practice and Design, pp. 3–16 (1991)
6. Bannon, L.J.: Understanding Common Information Spaces in CSCW. In: Proceedings of the Workshop on Common Information Spaces, Copenhagen (2000)
7. Bannon, L.J., Bødker, S.: Constructing Common Information Spaces. In: Proceedings of the Fifth European Conference on Computer-Supported Cooperative Work (ECSCW 1997), Lancaster, England, pp. 81–96. Kluwer Academic Publishers, Dordrecht (1997)
8. Reddy, M., Dourish, P., Pratt, W.: Coordinating Heterogeneous Work: Information and Representation in Medical Care. In: Proceedings of the Seventh European Conference on Computer Supported Cooperative Work (ECSCW 2001), Bonn, Germany, pp. 239–258. Kluwer Academic Publishers, Netherlands (2001)
9. Bossen, C.: The Parameters of Common Information Spaces: the Heterogeneity of Cooperative Work at a Hospital Ward. In: Proceedings of the Conference on Computer Supported Cooperative Work, pp. 176–185. ACM, New Orleans (2002)
10. Munkvold, G., Ellingsen, G.: Common Information Spaces along the illness trajectories of chronic patients. In: Proceedings of the Tenth European Conference on Computer Supported Cooperative Work (ECSCW 2007), Limerick, Ireland, pp. 291–310. Springer, London (2007)
11. Fields, B., Amaldi, P., Tassi, A.: Representing Collaborative Work: The Airport as Common Information Space. Cognition, Technology and Work 7(2), 119–133 (2005)
12. Rolland, H.K., Hepso, V., Monteiro, E.: Conceptualizing Common Information Spaces Across Heterogeneous Contexts: Mutable Mobiles and Side-effects of Integration. In: Proceedings of the Conference on Computer Supported Cooperative Work, Banff, Alberta, Canada, pp. 493–500. ACM Press, New York (2006)
13. Clement, A., Wagner, I.: Fragmentation Exchange: Disarticulation and the Need for Regionalized Communication Spaces. In: Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work (ECSCW 1995), Stockholm, Sweden, pp. 33–49. Kluwer Academic Publishers, Dordrecht (1995)

14. Randall, D.: What's Common about Common Information Spaces? In: Proceedings of the Workshop on Cooperative Organization of Common Information Spaces, Technical University of Denmark (2000)
15. Bertelsen, W.O., Bodker, S.: Cooperation in massively distributed information spaces. In: Proceedings of the Seventh European Conference on Computer Supported Cooperative Work (ECSCW 2001), Bonn, Germany, pp. 1–17. Kluwer Academic Publishers, Dordrecht (2001)
16. Crabtree, A.: Designing Collaborative Systems. In: A Practical Guide to Ethnography. Springer, London (2003)
17. Poole, M.S., DeSanctis, G.: Use of group decision support systems as an appropriation process. In: Proceedings of the 22nd Annual Hawaii International Conference on Systems Sciences, pp. 149–157. ACM, New York (1989)
18. Glaser, B.G., Strauss, A.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine, New York (1967)
19. Glaser, B.G.: Theoretical Sensitivity. In: Advances in the Methodology of Grounded Theory. The Sociology Press, Mill Valley (1978)
20. Selvaraj, N., Fields, B.: A Grounded Theory Approach towards Conceptualizing CIS for Heterogeneous Work Communities. In: Proceedings of the 23rd BCS Conference on Human Computer Interaction (HCI 2009), Cambridge, UK, pp. 471–479. British Computer Society, Swinton (2009)

The Mind's Eye on Personal Profiles: How to Inform Initial Trustworthiness Assessments in Virtual Project Teams

Ellen Rusman, Jan van Bruggen, Peter Sloep, Martin Valcke, and Rob Koper

[1] Open University of the Netherlands, CELSTEC, Valkenburgerweg 167,
6401 DL Heerlen, The Netherlands
[2] Ghent University, Department of Educational Studies, H. Dunantlaan 2,
B9000 Ghent, Belgium
{Ellen.Rusman,JanvanBruggen,Peter.Sloep,Rob.Koper}@ou.nl
Martin.Valcke@ugent.be

Abstract. Personal information is an important precursor for the trust formation process in virtual project teams. However, till today it has remained unclear what specific personal information most trustors prefer. Insight in their preferences as well as in their foundation could support the development of templates that provide communication support to virtual teams. In this paper, we describe and empirically test an approach that links trustors' common information preferences and a TrustWorthiness' ANtecedents (TWAN) framework. Thus, we provide 'the mind's eye' on interpreting and valuing information elements.

Keywords: trust, virtual teams, online identity.

1 Introduction

People form impressions of others every day, attributing properties to them they can never be certain the others do indeed possess. People make a 'best' guess based on signs and signals they perceive; this we call 'a first impression'. This first impression of others is the first seed of trust or distrust, and it colours perceptions of all subsequent behaviour [3,6,14]. In computer-mediated communicative (CMC) settings routes and available signs and signals to form an impression may be obstructed or different [4,5], but the impression-formation process remains just as important for human interaction [10,11,20]. Contrary to the initial belief that personal relationships would not be developed via CMC, since people would have less and not very useful information available with which to form an impression ('cues-filtered-out' perspective) [8], Walther [20,21] found that only the process of impression formation is delayed. He found that given enough time enough information about a person, personal as well as behavioural, is revealed and relationships grow as a result.

In face-to-face situations people use various routes to acquire information: via face-to-face interaction, via inferences based on social characteristics (e.g. communities the other takes part in) and via reputational information acquired via 'worth of mouth' [16]. In virtual project teams which use ICT (e.g., email, chat, videoconferencing) predominantly

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 297–304, 2010.

as their means of communication these routes are often not available or in different forms only. Team members of virtual project teams sporadically meet in person, they often do not have a prior history of working together and they may never meet in the future [9], so the routes of 'worth of mouth' and 'face-to-face' interaction are in many cases blocked. Furthermore, messages that are ICT-mediated do not convey the same type of signs and signals as they would in face-to-face settings. This type of teams are reported to have most problems with interpersonal trust formation, especially in the initial phases of a project [22,23].

In order to jump-start impression formation on trustworthiness in the first phases of a project one could offer team members information about their colleagues. This has been done for years by companies who organize special face-to-face team building activities, leaving the type of information exchanged up to spontaneous interaction. This approach has also infused online environments, of which evidence can be seen in the design and use of profile templates within social network sites or communities, such as Facebook or Elgg. The notion that a representation of people in online environments is beneficial for their collaboration is also supported from the perspective of research on presence [7].

Availability of information about virtual project team members can positively influence trustworthiness assessments [17], however it is not yet clear which specific information elements are considered most supportive for these assessments and why they are supportive. Information elements are small containers for data about a person. Examples are 'name', 'photo', 'hobbies', 'job title' and so on. The content of the information element 'name' could be 'Nathan'. First steps in the research on the significance of information elements have been made by Ten Kate (2009) and Berlanga et al. (in press), who in the context of social network sites (SNS) explored what elements displayed in profile templates contributed to a first impression of trustworthiness or were used to present oneself of perceive another [2,19]. Still, the information elements originally displayed in these templates were likely chosen by designers at the senders end of the computer-supported communication process and probably not specifically grounded in the cognitive processes at the receivers' end. Furthermore, the context of a social network site may differ from a virtual project team as their objectives are different. In addition, virtual teams have more mechanisms for social (institutional) control on the reliability of personal information displayed than SNS, as team members are embedded in existing organizations.

If one would know what type of information most people prefer in order to form an impression of their team members in a virtual project team, one could provide a pre-structured template for entering such information. Such predefined templates can have positive effects on the impression formation process as well on the collaboration process as a whole [15,17]. Unfortunately, the selection of the information elements to be displayed in these templates still is a 'best guess' rather than an informed decision.

When people form an initial impression of the trustworthiness of others, several factors interplay [18]. A trustor looks at the specific situation and the specific properties of a trustee, influenced by her mood as well as her trust disposition. While trying to gauge whether a trustee has characteristics which are desired in the specific context, a trustor collects information which can function as a cue and verifies it against the several antecedents of trustworthiness [18]. Although according to implicit personality theory people use different information elements as cues for specific properties of the other [1], we assume these elements overlap, that is, there are elements many people use. We furthermore expect that it is best to offer elements which are linked to different antecedents of trustworthiness and that elements revealing information on more than one antecedent are more worthwhile. In previous literature, many antecedents have been mentioned, but in recent literature three clusters of antecedents are discerned: ability, benevolence and integrity. This clustering and exclusion of previously mentioned antecedents was made mainly on analytical grounds, not so much on empirical grounds.

While reviewing literature, we have not excluded any antecedent found so far, but merely admitted them all to our list. This led to the following 'candidate' antecedents as the footing of a trustworthiness decision [18]:

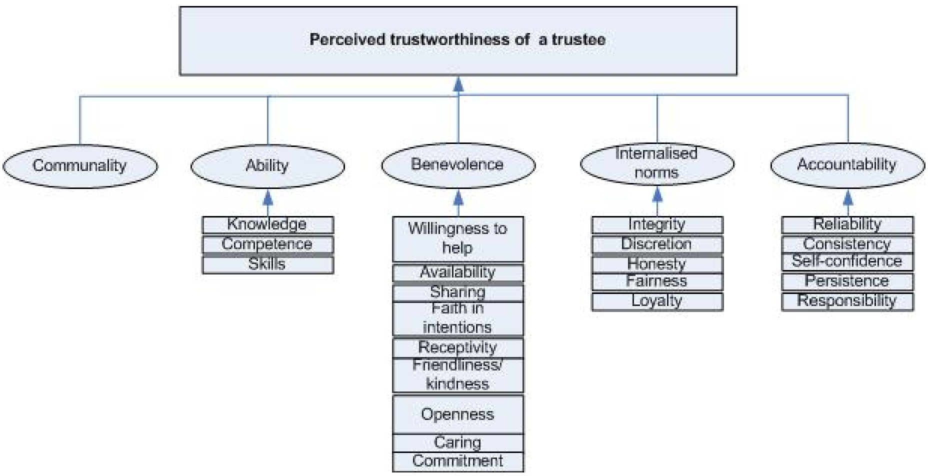


Fig. 1. The TrustWorthiness ANtecedent schema (TWAN)

This schema is currently tested in a large-scale empirical study among professionals, to determine whether all antecedents mentioned in literature are also relevant and add weight while assessing professional trustworthiness in practice. The question to be answered now is what information elements provide cues for reaching a trustworthiness decision and why these elements apparently matter most. Possibly, some information elements are preferred as they provide information for more than one antecedent. For example, one's education could address one's ability as well as one's consistency and responsibility. Also, from an economy principle people may prefer information elements which provide cues for multiple antecedents. Certain information elements will then have an increased 'weight' in a trustworthiness decision. However, people might also prefer information elements which provide 'unique' information related to a specific antecedent. These and similar considerations have led to the following research questions:

1. Do trustors in a virtual project team:
    a. use the antecedents in the TrustWorthinessANtecedent schema as a reference while selecting preferred information elements in order to decide on a trustee's trustworthiness?

   b. prefer information elements that provide cues for multiple antecedents within the TrustWorthiness ANtecedent schema?
   c. prefer information elements that provide unique cues for an antecedent within the TrustWorthiness ANtecedent schema?

2. Does trustors' total selection of information elements relate to multiple antecedents within the TrustWorthiness ANtecedent schema?

This paper describes the design as well as the first results of a study which aims to answer these questions.

2 Method

First, we determine what information elements trustors have in common when arriving at a trustworthiness decision. Subsequently, we test whether trustors' explanations of their preferences contain references to the TrustWorthiness Antecedent schema, thus testing whether the antecedents function as a reference framework for reaching their decision. In this paper we present the results of the first phase of this study and describe how we will proceed with the second phase.

2.1 Participants

Data were collected among bachelor level students, enrolled in the Educational Sciences programme at the Ghent University. A convenience sample existing of 226 students (mean age = 18,2 years, SD= 1,85) participating in a research course was obtained, 93% of which were female and 7% male. 99 % of the respondents had previous experience with collaboration in a face-to-face project team, either in a (part-time) job or during their study. 95 % had previous experience with collaboration in a virtual project team. 88% of the respondents had experience with online conversations with people they had never met before. The majority of online conversations took place via text-based media only, either via chat and/or e-mail (78%) or in combination with SMS (9%).

2.2 Instrument

The questionnaire contained open, as well as closed questions in the respondents' native tongue (Dutch). In this paper we restrict ourselves to the analysis of the open questions. In this part of the questionnaire participants were asked to select the 10 information elements they considered most important when forming a first impression of trustworthiness of a virtual project team member. They could select these information elements either from the pre-defined list they had just rated (closed questions) as well as from an initial open brainstorm of preferred information elements in a profile they did at the start of the questionnaire.

2.3 Procedure

Before they filled the questionnaire participants received a short presentation that clarified our definition of virtual project teams and showed examples of them.

The presentation discussed the role of interpersonal trust for collaboration and the objectives of the questionnaire. At the start of the questionnaire, respondents were prompted by a scenario that described them as a member of a new European project, which required them to collaborate in a virtual project team. They were asked to imagine that they were part of this virtual team and told that they had to form a first impression of their team members' trustworthiness. They were told that they could determine what information they would want to have available from the profiles of their team members by selecting the information elements that they felt mattered most to their trustworthiness assessment. Respondents were told that the responses to this questionnaire would be kept anonymous and that it would take about 30 minutes to complete the open questions of the questionnaire: 10 minutes for the initial open brainstorm and 20 minutes for the selection of the 10 most important information elements, based on the results of the brainstorm as well as on the importance of information elements ranked previously in the closed questions of the list. They were asked to reflect on their answers and to select the 10 information elements they perceived as most important for the determination of trustworthiness within a virtual project team. They were also asked to explain what factual information about a person they derived from this information and how they interpreted this information in the process of determining the trustworthiness of a future team member. Respondents had no prior knowledge of the TWAN schema when they filled the questionnaire. Table 1 provides an example (translated) of the collected responses.

Table 1. Example response

Preferred information element	Facts which can be derived from this information element	Explanation of preference
Personal motivation for project	Reason for participation; expectation(s) of project	You get to know whether you are on the same wavelength. Do you have the same expectations?

2.4 Data Analysis

We focus on the analysis of the open questions of the questionnaire, as they provide an explanation for the preference of specific information elements for the design of a personal profile. We used a mixed quantitative and qualitative approach for this data analysis [12], to detect common preferences of information elements as well as their meaning for trustworthiness assessments of virtual team members. We here report the quantitative part and describe our approach for the qualitative part, not yet reporting the results of this part.

First, information elements selected as most important were categorized and counted and information elements mentioned most often by all respondents were listed according to their frequencies. All explanations referring to these information elements were listed as well, so that we could gauge both the shared importance of the information element across the respondents as well as the explanations of the advantage of using a particular information element for the assessment of trustworthiness. We here report on this first phase. For the qualitative part of the data analysis, we will use a coding approach for the analysis of explanations given in order to verify whether respondents used antecedents in the TWAN schema as a reference framework for selection of information elements. The different antecedents in TWAN are the coding categories (Figure 1), next to additional categories derived from theory on trust and trustworthiness, such as 'context' and 'trustors attitude' (comprises of trustors propensity and mood). Two raters will first individually analyse 10 % of the responses [13], to determine the similarity of their analyses. We use Cohen's Kappa as a measure of interrater-reliability, with a cutoff criterion of .8 [13]. The rest of the responses will be analysed by one rater.

3 Results

We received 2251 open entries from 226 respondents, of which 1882 entries were indeed rankings and 369 entries were missing data (16%). These entries were due to respondents which did not correctly follow the instruction and selected and described less than 10 information elements. The filled entries of these respondents were included for analysis. Table 2 shows the frequency distribution across the 15 most selected information elements.

Table 2. Frequencies of preferred information elements for trustworthiness assessment in VT's

Information element	Frequency
Personality traits/character	124
Work experience	118
Personal motivation for project	117
Education/studies/training/diplomas	94
Age/date of birth	87
Availability during project/agenda	82
Recommendations/references/reviews by third parties	74
Project work experience	67
Language/language proficiency/language skills	66
Photo (formal/informal)	65
Interests/hobbies	60
Family situation/marital status	54
Ideas in relation to project	49
Occupation/function/role/job	49
Nationality	47

In total, 106 different information elements were selected. 9 of them were not in the pre-defined list which respondents had available, e.g. stress immunity; computer skills and meeting skills, but resulted from the open brainstorm.

In the next phase of this study we will use the explanations participants' mentioned to explain their selection of information elements as input for a qualitative analysis.

In this phase we will test whether the reasons participants mention for their information preferences match with the antecedents within the TrustWorthinessANtecedent schema.

4 Discussion and Future Research

In this paper we described an approach to determine common preferences for information elements that are used to support trustworthiness assessments. We also describe an approach to the analysis of explanations virtual team members provide for these preferences. For the latter analysis, we will use a TrustWorthinessANtecedent (TWAN) schema.

Initial quantitative results show that often-used information elements such as 'name' hardly matter to trustworthiness assessments, as they were not commonly preferred. Such elements may just be an indicator of identity, merely used to distinguish people ('the flag on the ship'), but apparently they do not carry weight in a trustworthiness assessment. Results also show that each person in principle uses different information elements to assess trustworthiness of colleagues, but that there is quite some overlap in preferences. These common information preferences, here derived from receivers' instead of designers' perspective, can guide the design of profile templates aimed to accelerate the impression formation process on trustworthiness in the first phases of a project.

Further analysis of the obtained qualitative data is needed to provide more insight in the nature of peoples common information preferences. This additional analysis aims to determine whether the explanations people provide for their preferences are indeed related to the different antecedents of trustworthiness proposed within the TrustWorthinessANtecedent schema (TWAN).

Acknowledgments. We would like to thank the Ghent University students for their dedicated collaboration.

References

[1] Arnold, J., Cooper, C.L., Robertson, I.T.: Work psychology. In: Understanding human behaviour in the workplace, 3rd edn. Financial Times Professional Limited, Essex (1998)
[2] Berlanga, A.J., Bitter-Rijpkema, M.E., Brouns, F., Sloep, P.B., Fetter, S.: Personal profiles: Enhancing Social Interaction in Learning Networks. IJWBC (accepted)
[3] Cooper, A., Bott, M.W.J.: Influence of Expectancies and Experience on Impression Formation. JPI 4, 21–24 (1999)
[4] Donath, J.: Signals, cues and meaning (February 15, 2006), `http://smg.media.mit.edu/classes/IdentitySignals06/SignalingDraft.pdf` (Retrieved August 2006)
[5] Donath, J.: Signals in social supernets. JCMC, 13(1), article 12 (2007)
[6] Good, D.: Individuals, Interpersonal Relations, and Trust. In: Gambetta, D. (ed.) Trust: Making and breaking cooperative relations, pp. 31–48. Blackwell, Oxford (2000)
[7] Gutwin, C., Greenberg, S.: A descriptive framework of workspace awareness for real-time groupware. CSCW 11, 411–446 (2002)
[8] Hancock, J.T., Dunham, P.J.: Impression Formation in Computer-Mediated Communication Revisited. Communication research 28(3), 325–347 (2001)
[9] Jarvenpaa, S., Leidner, D.: Communication and trust in global virtual teams. JCMC 3(4) (1998)

[10] Lea, M., Spears, R.: Love at first byte? Building personal relationships over computer networks. In: Wood, J.T., Duck, S. (eds.) Understudied Relationships: Off the Beaten Track. Sage Publications, Inc., Newbury Park (1995)
[11] Liu, Y., Ginther, D.: Managing Impression Formation in Computer-Mediated Communication. Educause Quarterly 3, 50–54 (2001)
[12] Miles, M.B., Huberman, A.M.: Qualitative Data Analysis. In: An Expanded Sourcebook, vol. 2. Sage Publications, Inc., Thousand Oaks (1994)
[13] Neuendorf, K.A.: The Content Analysis Guidebook. Sage Publications, Inc., Thousand Oaks (2002)
[14] Petty, R.E., Cacioppo, J.T.: Communication and Persuasion: Central and Peripheral Routes to Attitude Change. Springer, New York (1986)
[15] Remidez, H., Stam, A., Laffey, J.: Web-Based Template-Driven Communication Support Systems: Using Shadow netWorkspace to Support Trust Development in Virtual Teams. IJeC 3(2), 65–83 (2007)
[16] Riegelsberger, J.: Trust in mediated interactions. University College London, London (2005)
[17] Rusman, E., van Bruggen, J., Cörvers, R., Sloep, P., Koper, R.: From pattern to practice: Evaluation of a design pattern fostering trust in virtual teams. Computers in Human Behaviour 25(5), 1010–1019 (2009)
[18] Rusman, E., Van Bruggen, J., Sloep, P., Koper, R.: Fostering trust in virtual project teams: towards a design framework grounded in a TrustWorthiness Antecedent (TWAN) schema. Manuscript submitted for publication (2010)
[19] Ten Kate, S.: Trustworthiness within Social Networking Sites: A study on the intersection of HCI and Sociology. Unpublished Business Studies Master. University of Amsterdam, Amsterdam (2009)
[20] Walther, J.B.: Relational Aspects of Computer-mediated Communication: Experimental Observations over Time. Organization Science 6(2), 186–203 (1995)
[21] Walther, J.B.: Computer-mediated communication: Impersonal, interpersonal, and hyperpersonal interaction. Communication research 23, 3–43 (1996)
[22] Zolin, R., Hinds, P., Fruchter, R., Levitt, R.E.: Interpersonal trust in cross-functional, geographically distributed work: A longitudinal study. Information and Organization 14, 1–26 (2004)
[23] Zolin, R., Hinds, P.J., Fruchter, R., Levitt, R.E.: Trust in Cross-functional, global teams. CIFE (2002)

Supporting Informal Interaction in a Hospital through Impromptu Social Networking

Alberto L. Morán[1], Juan Rodríguez-Covili[2], David Mejia[3,4], Jesus Favela[3], and Sergio Ochoa[2]

[1] Facultad de Ciencias, UABC, Ensenada, Mexico
alberto.moran@uabc.edu.mx
[2] Department of Computer Science, Universidad de Chile, Santiago, Chile
{jrodrigu,sochoa}@dcc.uchile.cl
[3] Department of Computer Science, CICESE, Ensenada, Mexico
favela@cicese.mx
[4] Centro de Ingeniería y Tecnología, UABC, Tijuana, Mexico
damejia@uabc.edu.mx

Abstract. Social networking systems allow users to keep in touch with relatives and friends in the absence of physical proximity. These tools are also increasingly supporting productive interactions in diverse working environments. In this paper, based on the understanding of informal communication in hospitals, we identify opportunities for the use of social networking software in support of hospital work. This has inspired the design of meetU, a tool aimed at supporting impromptu social networking through an ad-hoc communication infrastructure. The services offered by the system are illustrated through interaction scenarios, which were also used to evaluate the system with a group of medical interns.

Keywords: Mobile Workers Interaction, Social Networking Systems, Impromptu Social Networking, Hospital Work.

1 Introduction

Hospital settings offer a prolific area to study mobile collaborative work. Medical activities in hospitals are characterized by the need for coordination and collaboration among specialists with different areas of expertise, an intense information exchange, the integration of data from many devices or artifacts, and the mobility of hospital staff, patients, documents and equipment [24]. Additionally, a hospital can be considered as an information space that is "navigated" by the staff in order to get the information and resources required to accomplish work [3], since most hospital staff moves continuously to access people, knowledge, and resources [1].

Hospital workers are also in frequent need to locate colleagues. A physician might require the opinion of a specialist to confirm a diagnosis; a nurse might need to contact the doctor in charge of a patient who has been experimenting discomfort or pain; a resident physician might just need a couple of free hands to conduct an intervention. This type of interactions has been supported by casual, informal or lightweight communication mechanisms and social awareness [19, 31].

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 305–320, 2010.

By informal communication we mean those interactions that do not have a predefined schedule or place of encounter, that are spontaneous, brief, and where the topic of the conversation can change during the course of the interaction [19]. This type of communication has proved to be very important to accomplish work because it contributes to the social and production functions of the group [8, 16, 18]. Unfortunately, informal spontaneous communication comes at a cost: interruptions [7]. To date, several mechanisms have been designed to deal with interruptions, which mostly rely on the provision of social awareness [6, 7, 21].

By social awareness we refer to the basic knowledge that provides an overview of the social context where the situation arises: what is happening, who is around, whether others are relatively busy or can be engaged, who is talking to whom, who I can work or communicate with, and which other activities are occurring [4, 12, 31].

Based on these concepts, several models and systems have been proposed to provide support for informal communication and social awareness mostly based on the provision of artificial proximity in place of physical proximity. Examples include: VideoWindow and Cruiser [19], Piazza [15], Hubbub [14] and Aware Phone [2]. However, since hospital workers perform activities both while at a base location and while moving around the hospital [1], they require continuous (and interchangeable) support for both modes of work. For this reason, these tools are not adequate for such setting. Further, at-the-base and on-the-move work seems to change the rules and requirements for the provision of support for informal interaction in hospital work.

In order to understand how informal communication in settings such as hospitals, could be better supported, we conducted a field study in the area of internal medicine of a mid-size public hospital [1]. This study helped us identify design insights aimed at informing the design of tools for supporting co-located informal communication in a working environment with local mobility [22].

In this paper, we aim at investigating the use of social networking tools in support for hospital work. We conducted an additional study and analyzed the data of the former observational study to uncover opportunities for this technology and to understand how social networking systems can be designed and deployed to support informal interaction and work activities in this type of environment. The next section provides a brief introduction to social networking systems and some of its current uses in hospital work. In Section 3 we present the results of the workplace study from which design scenarios and insights are derived. Section 4 introduces the concept of Impromptu Social Networking (ISN) and describes the design of meetU a mobile tool developed on top of a mobile ad-hoc communications infrastructure. Section 5 presents the results of a preliminary evaluation of the concept. In Section 6 we present related work, and summarize and conclude in Section 7.

2 Social Networking Sites

Social Network Sites (SNSs) are attracting millions of users worldwide. Sites like Facebook, and MySpace, for example, have millions of registered users and are increasing their membership [17]. On these sites people present themselves via profile pages, establish ties with other users, communicate between them via text and voice messages, and interact with each other via optional leisure applications [24].

Concerning use, [20] found that SNSs are used for i) social searching –finding out information about people they have met offline-, and for ii) social browsing –using the SNS to develop new connections, perhaps with the aim of interacting with them offline. Further, [1] identifies three additional uses: i) “Looking at”, which intends to know more about people met offline (similar to social browsing [20]), ii) “Looking up”, which intends to search for people to meet offline (similar to social searching [20]), and iii) “Keeping up with People”, which intends to maintain weak connections with people met offline.

SNSs have also been classified as being Open or Work SNSs [10]. Open SNSs allow users from anywhere in the Internet, while Work SNSs refer to those that are used within an enterprise intranet by professional workers.

Professional workers use internal SNSs to build stronger bonds with their weak ties and to reach out to employees they do not know [10]. The main motivations for that use were i) *caring* - workers enjoyed connecting socially at work, both to those they work or may work with, and those they do not know; ii) *climbing* - they felt the use of the work SNS was or could specifically assist them in their personal career advancement; and iii) *campaigning* - they used the SNS to solicit support for their ideas and to drive traffic to their project web pages, which they saw as a means to move forward with their ideas.

Concerning the use of these tools in hospitals, some have started using and adopting them as communication tools for “hard to reach” groups in their area. For instance, NHS hospital trusts in the UK are using Twitter[1] as a service for the communication of press releases, awareness messages and updates, such as “cancelation of the services due to bad weather” [11]. They currently consider Twitter as “an auxiliary communication channel”.

Given the features of SNSs, and the evidence of its use as communication tools with users outside the hospital, we set to find how SNSs can be used to provide support to workers within the hospital. Although there could be more than one candidate community, we will concentrate on Medical Interns (MI), due to the specific characteristics of this community.

Medical interns are considered physicians-in-training; they provide the most hours of patient care in the unit and are in constant movement. In the area of internal medicine, where we did the observations, each intern is responsible for the care of five or six patients. One of their main responsibilities is to create clinical histories whenever a new patient arrives at the hospital. They are also responsible for providing care and following-up on patients during their stay in the hospital.

Further, medical interns are in the process of adapting to the hospital environment. For one full year they get their first experiences with the feelings associated to the loose of a patient, making a mistake, receiving a disciplinary action and the constant supervision from specialists, nurses and the director of the intern program. In addition, they spend most of their time in the hospital. They work daily shifts of 8 hours with a one-hour break, and every three days they have to work an extended 32-hour shift (medical ward). Under these conditions hospital work becomes a major part of their life, certainly more than what most professionals/students experience. Finally, most medical interns come from other cities to do their internship. Results of a survey

[1] Twitter: www.twitter.com

we applied to 36 medical interns reveal that when they arrive they barely know the city, and have no relatives or friends, other than the dozen or so fellow interns with whom they studied before initiating their internship. These conditions require medical interns to develop a strong sense of belonging to their community within the hospital.

Medical interns also experience an interesting duality as students and professionals, they are evaluated as part of their professional education, and their work is a crucial part of the hospital's operation. This duality combined with the confidence they gather throughout their internship, exposes them to situations where the support of the group or social network they belong to helps them sort out a diversity of issues that arise in their daily work.

3 Understanding the Work Social Networks of Medical Interns

To better understand the characteristics of the social networks that medical interns belong to, we performed a preliminary study at the internal medicine area of a local hospital. In this section we present the major findings of the study, as well as an ensemble of use scenarios.

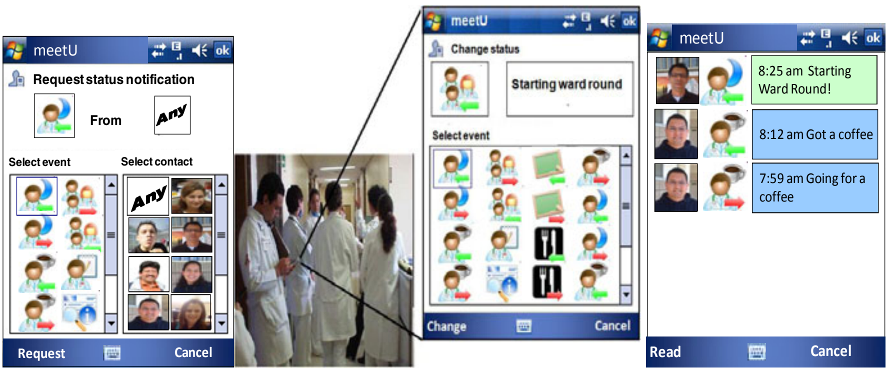
The study included two stages. In stage one we surveyed current and former medical interns from a local hospital. The participants in this stage were 36 medical interns (26 currently in their internship and 10 who just finished a few months ago). Their average age is 23 years old. The results show that all the medical interns know at least one social networking system and that 94% of them use at least one of them. Using a five-point Likert-scale, we identified that medical interns found social network systems very useful for social activities and just useful for productive activities. Most of them (89%) have been registered in one or more of these sites for at least one year. Moreover, all of them log into these sites at least once a month, and 17% of them do this every day. These results encouraged us to propose the use of SNSs to support hospital work and to deal with some breakdowns in informal communication in such settings.

Stage two consisted of a further analysis of observational data from a previous study, originally performed to understand informal interactions in hospital work [1]. Described briefly, the methodology consisted in collecting data through the non-participatory observations of ten medical workers including five physicians and five medical interns during two complete working shifts. In this second stage, we decided to place special emphasis on understanding how local mobility influences the interactions experienced by medical interns, and how their offline social networks affect those interactions while contributing to the completion and achievement of their work. For this, the second analysis aimed at having a good sample of interactions and capturing the communication instances in which those workers are involved, along with their social networks, as they conduct their work, including details with respect to the nature of the actions, artifacts used, contents of conversations and physical location of individuals, among others.

3.1 Application Scenarios for Social Networking in a Hospital

As a result of stage two of this study, we identified use scenarios that exemplify a distinct and typical use for a socially-aware support system. The scenarios were sketches of user activities that let us translate our findings into specific vignettes that captured facets of how socially-aware tools might fit into current work practices. The following scenarios illustrate the kind of social network support hospital work might benefit from and give a flavor of how this technology can enhance hospital practices.

Scenario 1. Juan, a medical intern, reviews the healthcare record of a patient just before the ward round usually begins. In order not to miss it, he indicates in his mobile phone that he wants to be notified when the status of any of his colleagues changes to "starting ward round" (see Fig. 1A). Ten minutes later the physician in charge initiates the ward round and Pedro (another MI) updates his status to indicate this by selecting the appropriate icon in the "set status" window (see Fig. 1B). At this moment, Juan receives the notification (see Fig. 1C) and he joins the ward round.



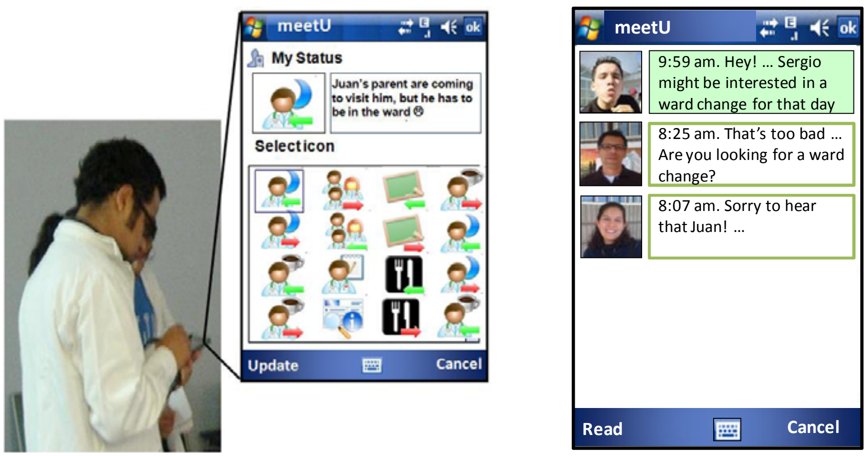
(A) (B) (C)

Fig. 1. (A) Requesting notification, (B) Update status, (C) On receiving notification

In this scenario, based on the actions of one of the medical interns participating in the ward round, valuable information is generated so that a notification regarding the actual event "Starting Ward round" could be generated and delivered to another medical intern. The additional value of this notification resides in that without this information, the medical intern could miss the start of the ward round and get a scolding or even a report from the physician in charge of the intern's program.

Furthermore, in this case the medical intern explicitly updates his status, but in some instances the system could potentially use contextual information to do it automatically. For example, an intern that stays in the cafeteria for more than five minutes at a certain time of the day (e.g. at Noon) could highly accurately be estimated to be having lunch. Previous work has established that activity estimation can be achieved with an accuracy of up to 90% when the location, time, companions and artifacts used by a medical worker are known [29], but even simple rules could be set to indentify some activities with a high degree of accuracy. To simplify the indication of a new state to the user, the interface shows a number of icons with common activities, such as *in ward round, out to lunch, writing medical notes*, as shown in Fig. 1B. When the user selects one of the icons, the corresponding text is displayed allowing the user to provide additional explanation.

Scenario 2. Juan is a MI and his parents will visit him this weekend. However, Juan remembers that he has a 32-hour shift this Saturday and is concerned that he won't be able to see them. Not knowing whom to ask for help; he enters into the Social Networking System on his mobile phone and sends the message: "Juan's parents are coming to visit him this weekend, but he has an extended 32-hour work shift this Saturday :-(" (see Fig. 2.A). Several of his fellow MI's receive the message and respond with empathy messages. Miguel, another MI, replies with: "Hey … Sergio might be interested in changing his shift to that day" (see Fig. 2.B), Miguel knows that Sergio (another MI) is free that weekend, and might be interested in switching shifts with Juan. Through Miguel's profile page, Juan accesses Sergio's profile page and contacts him through the SNS's chat to ask him if he could take his shift on Saturday.

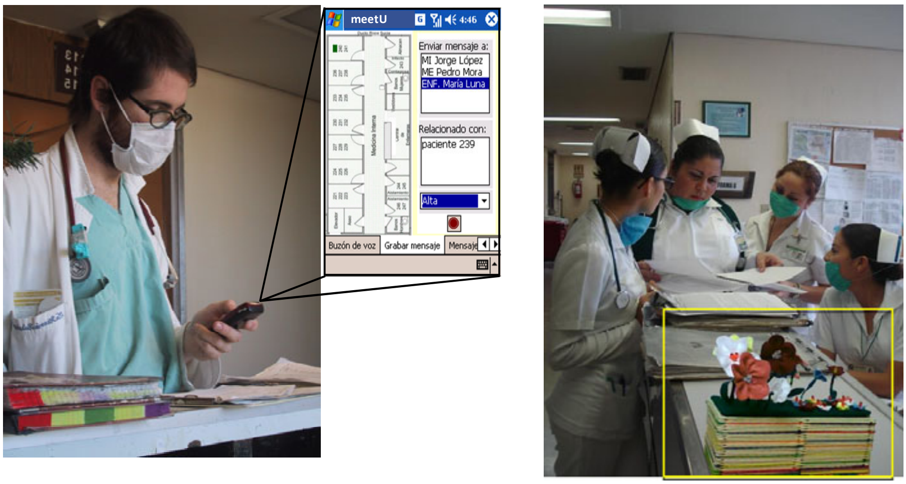


(A) (B)

Fig. 2. Updating the status with a custom message (A), and Receiving a reply from a friend with a suggested solution to the expressed problem (B)

In this scenario, updating the status of one member of the social network provides lightweight means of communicating the problem faced by the MI. This in turn generates an ensemble of answers, even including one with a possible unrequested solution to the MI's problem. Additionally, the seamless access to the profile of other MI's and shared communication channel (the chat tool) through the SNS are essential to solve the MI's problem.

Scenario 3. The physician is in his office reviewing the record of a patient when he decides to leave an indication to the nurse in charge of this patient during the next shift. He does not know who she is, so he pulls out his mobile phone and accesses the SNS (see Fig. 3.A). On the map of the ward shown, he clicks on the bed where the patient is and the names of the medical personnel in charge appear. The physician selects the name of the nurse and leaves a voice message for her. When the nurse is near the Nursing station she notices that she has a pending message through the FlowerBlink ambient display (see Fig. 3.B). The nurse accesses her SNS's message center and listens to the message. At that time, the physician receives a notification indicating that his message has been read.



(A) (B)

Fig. 3. Leaving a message to a nurse (A) and Notification delivery by means of the FlowerBlink ambient display

This scenario illustrates how subtle notification in the intensive working environment of the hospital could be achieved through an ambient display in the form of an artificial flower bowl that lights on when a person with a pending message is in its proximity [30]. Also, the scenario illustrates how the SNS and the hospital information systems could be integrated to achieve this kind of notifications.

3.2 Requirements of a Social Networking Infrastructure

As illustrated by the scenarios, hospital work has certain characteristics that shape the way in which hospital personal communicates and collaborates. Work is performed in the following context: (a) distributed personnel moving around work zones that are physically close (local mobility); (b) mobile workers performing tasks in which the current status of their activity is relevant to their colleagues; (c) people that want to share social information and collaborate on-demand with their colleagues; and (d) existence of no time-critical social activities, such as those illustrated in the scenarios, which although time-sensible, do not affect the patients' care if the MI's delay their execution a little.

In the mobile work scenarios we have, such as hospital healthcare or construction inspection activities [28], a social component may also help to improve collaborative work, providing an informal communication channel that allows workers to coordinate (e.g. changing turns, assigning work areas, using work resources, showing awareness about others), and generate informal conversations, messaging and discussions that can be also relevant for the main work purposes (patients' care). Also, in these settings it might be difficult to deploy networking infrastructure. In the case of the hospital, in addition to the considerable paperwork required to obtain the necessary permits to install networking equipment, there are problems related to deployment and maintenance costs, including the creation of a new work role, that of the network technician, which makes it less feasible.

Based on these requirements, in the next section we present our initial proposal to develop a social networking infrastructure to support mobile work.

4 meetU: A MANET-Based Social Networking System

Mobile Ad hoc Networks (MANETs) offer several advantages as a communication platform to support loosely coupled collaborative activities in mobile work scenarios [5, 28, 22]. In those contexts, MANETs can create a communication mesh to exchange data among participating devices, so that applications do not need to rely on a fixed communication infrastructure, such as a distributed network of access points, thus eliminating the need for an initial investment on the networking infrastructure, and on its later maintenance.

Previously, the authors have proposed an application level routing protocol based on IEEE 801.11 standards. The protocol, named High Level MANET Protocol (HLMP) [26], allows automatically setting and keeping a MANET. Its implementation can be used by groupware systems through an Application Programming Interface (API) that provides a set of services required by mobile collaborative environments deployed on top of a MANET, reducing the effort involved in the development of this type of applications [27]. The library interface keeps the abstraction of the communication processes by using a message exchange paradigm and an event delivery method. It also provides useful information about the network structure to the applications, including changes and current status. This information can be used to implement awareness mechanisms for mobile workers. Further details about the implementation of the library are provided elsewhere [27].

meetU is a mobile collaborative social system developed using the HLMP API. This system implements what we named an *Impromptu Social Network* (ISN) infrastructure that supports collaboration among mobile workers in a hospital. An ISN is an informal and distributed interaction space, which is used by mobile workers belonging to an organization to support their daily activities. The meetU social network system is available through the users' mobile devices and it provides not only user interaction services but also social and emotional awareness to mobile workers.

When using meetU at the hospital, workers are able to establish ties with their colleagues, coordinate their activities and collaborate in order to perform a particular task. Some of the advantages of implementing this application on a MANET are the following: (a) this communication system is independent of the hospital information and communication infrastructure, therefore formal authorization permissions to deploy it and use it is facilitated; (b) it is an automated and easy to deploy network, which eliminates initial deployment and maintenance costs; (c) users typically trust an informal interaction system that is not centralized, and thus not visible to managers; and (d) peer-to-peer communication allows each person to limit or allow the access of remote users to his/her social information.

Fig. 4 presents an example of meetU deployment, including two main communication zones within the hospital setting. The first case represents a “low density” zone (zone A), as there are no direct links between all users; therefore communication between some users (e.g. 1 and 3) transparently goes through an intermediate node (e.g. the device of user 2) to allow them to interact. The second case (i.e. zone B) is denser than the former due to the physical proximity of the users, which allows direct links among them all. Thus they can directly interact and exchange messages.

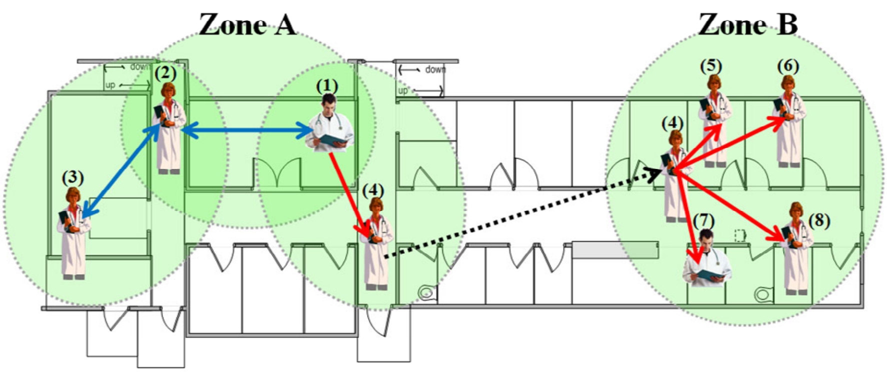


Fig. 4. Example of meetU usage at a hospital

Further, since both zones are disjoint (i.e. there are no links between them), if a user from zone A (e.g. user 1) wants to send a message to a colleague in zone B (e.g user 7), synchronous communication is not possible. For such cases, meetU provides a communication service named *gossip messaging*, which allows an intermediary mobile user to transparently transport and deliver an asynchronous message between two unreachable users [13]. Typically, these messages try to promote situations in which users could interact synchronously, or try to communicate status updates that could originate opportunistic encounters. For example, the messages could be: “please call me to my cell phone”, “I need to talk to you, I will be at neonatology at 10 am”, or status updates such as: “I’m going out for a coffee”, or “Starting ward round”.

In the situation depicted in Fig. 4, the medical intern (4) moves from zone A to zone B (denoted by the dashed arrow), hence meetU can automatically take advantage of her mobility and use her device as a message transport. For her it will be transparent that the system is using her device to support this interaction; therefore such activity is not disruptive for the users.

4.1 Architecture

Fig. 5 shows the meetU architecture, which includes the components that allow social interactions between users via text or voice messages, user awareness mechanisms and resource sharing. The application architecture is fully distributed; this means that the whole infrastructure is replicated in every mobile device that forms the ISN. The components hosted in a particular device can interact with components in other mobile devices through the HLMP Sub-protocols (i.e. using the message structure definition and the events triggered by the API) or the gossip mechanisms.

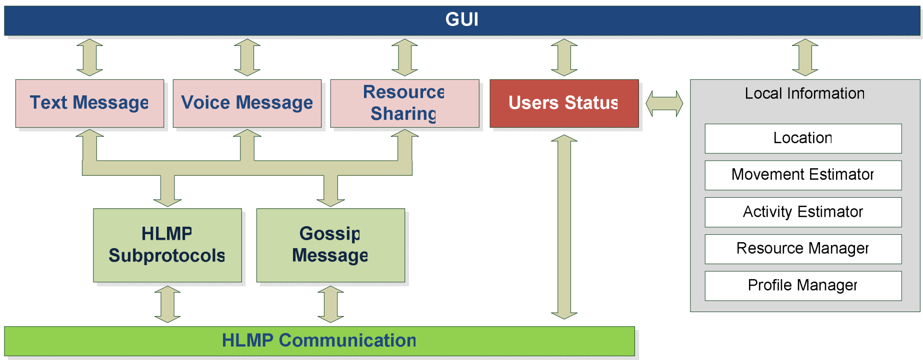


Fig. 5. Architecture of meetU

The profile of every user is updated through events of the MANET. Such events are exchanged among users status components hosted in each device, and indicate changes in the social network; e.g. a user joint or left the network, a user changed her status, etc. It is the responsibility of the HLMP Communication component to transport and deliver such messages. The data conforming the status of a mobile user includes: (1) user location inside the hospital (estimated by an autonomous software module), (2) user movement (manually configured or predicted based on the user's location history information), (3) user activity (estimated from information such as user location and time of day or explicitly provided by the user), (4) management of shared resources (where the users can create and manage shared files such as pictures, videos or documents), and (5) management of user profile (where the user can configure social information and its accessibility).

4.2 System Behavior

We now describe how the components of the architecture communicate to support the described functionality. A sequence diagram of the status notification procedure is shown in Fig. 6. Usually, a user requests a notification (as in scenario 1), in order to become aware of when certain status or condition is reported by a colleague. The petition is sent to the Users Status component. When updates on the information of another user match the requested status type, the component triggers back a notification to the GUI.

On the other hand, if a user changes his status in the meetU system using a new category, e.g. "Starting ward round", the new information is updated on his Users Status component and then settled in the "I'm alive" message of HLMP. This kind of message is constantly sent via multicast to the MANET by the communication engine,

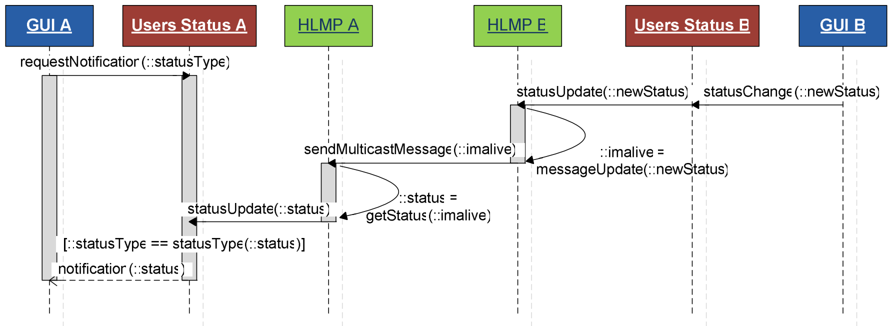


Fig. 6. Sequence diagram of meetU status notification

in order to propagate the device existence, user's network data, and profile information. The message is then received by the rest of the devices connected to the HLMP network, updating the colleague's profile with the attached status, and triggering a notification when it is required.

5 Evaluating the Perceived Usefulness of meetU

Aimed at evaluating the use of social networks and ubiquitous computing technology for supporting activities and communication of medical workers, we conducted a focus group session. This evaluation focused on one of the two main factors influencing technology adoption [9]: perceived usefulness. The focus group was conducted with three medical interns and was aimed at understanding how well social networks help to support their productive and social activities that require communication with other colleagues. The purpose of this session was to test and improve our design by focusing on the way in which medical interns communicate with others asynchronously, mainly due to unavailability of the collaborator or in absence of a communication channel (physical or artificial proximity).

5.1 Focus Group

The focus group was conducted in a room with a meeting table. We used a voice recorder to capture all events during the session. We invited three medical interns from a public local hospital in Ensenada, Mexico; who are currently performing their one year internship and consequently are very familiar with hospital work and its routines.

The session was conducted by one moderator who presented three scenarios using a personal computer. The session lasted approximately one hour.

The session started with an explanation of the "social networks" concept; after that, the moderator presented the scenarios to the medical interns that participated in the session; these scenarios illustrated some situations where the use of social networks could be helpful to them (medical interns). Each scenario was divided into two parts.

The first part showed a “problem” as it currently happens, while the second part showed our envisioned solution. After each scenario, the medical interns were asked to discuss how frequently these scenarios occurred in the hospital and the implications of these inconveniences in their work activities. The scenarios presented are those introduced in section 3.3 for which we foresaw that the application could be useful.

After each scenario, we conducted a session of questions and open discussion about the features, advantages and limitations of our proposal. At the end of the presentation of the scenarios, the medical interns were asked to complete a survey that helped us to gather quantitative information about the design of our proposal. We discuss the results of this evaluation next:

Scenario 1. Awareness of the start of the ward round. The three participants completely agreed that scenario 1 occurs at least once a day at the hospital and that it has a slight but clearly negative effect on their work. They all agreed that the solution presented in the scenario could help their current work and that it would be used in the hospital if available.

Additionally, they said that this solution (specifically the notification of status change) could be used in other scenarios, such as on the transfer of a patient from the emergency unit to bed in a ward, to notify when lab results are ready, and to notify that a medical worker is available to provide assistance.

Scenario 2. Request an exchange of medical ward. The three participants completely agreed that scenario 2 occurs one to three times a month in the hospital and that it has no significant effect on their work. They all agreed that the solution presented in the scenario could offer a slight benefit to their current work and that it could be used in the hospital if available.

The main concern with this scenario was privacy, since others could review the information. However, medical interns expressed that through the use of adequate mechanisms to deal with privacy, this service could be useful to them. Precisely one of the motivations for using a MANET-based infrastructure rather than a centralized one was to assure that the information won’t be stored, and thus accessible, in a central server.

Scenario 3. Leaving a message for a nurse. As for scenario 1, the three participants completely agreed that scenario 3 occurs at least once a week at the hospital and that it has a slight but clearly negative effect on their work. They all agreed that the solution presented in the scenario could help their current work and that it would be used in the hospital if available.

They mentioned that they currently need to ask other people for the person they are looking for or they ask a nurse to look for him, investing precious time in many occasions. They would find the tool useful since they often do not know who the nurse in charge of a shift is, and “it would make their relationship with nurses more cordial”. Also, they suggested including a photograph of the person because they do not always know her by name and after sending a message they might meet, and not be aware that it was her he sent the message to.

In summary, all three scenarios were considered realistic and the interns found the technology useful and easy to use given their previous experience with similar systems.

6 Related Work

Related work is revised under two different perspectives. A first perspective relates to the use of tools for informal communication in hospital work. One of the main challenges here is how to allow communication while workers are either at-the-desk or on-the-move. There are works that have mainly provided with the means to establish communication regardless of the location of the user within the premises. This has been achieved through the provision of artificial proximity tools using text, voice and hands-free communication on mobile devices; such is the case of the Vocera Communication Badge[2]. Our proposed ISN-based system, relies on the use of a Smartphone and of a MANET to support this kind of communication.

An additional challenge is how to identify a suitable moment to initiate a conversation with a colleague in the hospital. For this, there are works that have taken a context-aware approach to automatically acquire and provide information about the social context of a coworker; such is the case of the AWAREPhone [2]. In contrast, our proposal differs in that it takes advantage of the by-product information of the user's participation in the social network. This information is created by the ensemble of MIs through their explicit entries in the ISN, and thus it provides information that allows them not only to be aware of the social context but also to act upon it to provide an answer, solve a problem, and accomplish work at the same time.

A second perspective relates to the more general use of SNSs for work. In this case, the uses of the proposed Impromptu Social Network (ISN) differ from the social uses previously identified in [17] (i.e. looking at, looking up, and keeping up with people) and in [20] (i.e. social searching and social browsing), and from the work uses identified in [10] (i.e. caring, climbing and campaigning). ISN uses intend to provide support for, and are more related to, informal interaction during work processes of MIs rather than on maintaining or extending their social network, or on knowing more about co-workers, or spreading and championing the workers' ideas and projects. Possible explanations for this include that the ISN concept and our ISN-based tool have been conceived to support informal interaction and its functions at work [16, 19, 24], and that its initial implementation takes advantage of a MANET-based infrastructure, which is not provided nor managed by the organization, thus providing users with a sense of ownership and trust given its distributed and unmanaged nature. We are not aware of other similar uses of SNS in the context of hospital work.

7 Discussion and Conclusions

One of the main characteristics of social networks is that they provide social information regarding users that could be utilized by others to strength social relationships or to initiate collaboration. This seems to be one of the reasons for the growth of social networking systems (SNS) to support productive activities in collaborative working environments.

Based on the results of an observational study that we previously conducted in a hospital, we identified some situations where social networking services could be

[2] http://www.vocera.com

particularly useful, especially to support medical interns' activities, considering that most of them are already users of social networking software, and that being new to the environment, might not be aware of the resources at their disposal (including both colleagues and artifacts). Indeed, the medical interns that participated in the study have an offline informal social network that is used to support their work activities.

Thus, in this paper we introduce the concept of an Impromptu Social Network (ISN), an informal distributed ad-hoc interaction space, which is used by workers with local mobility to support their daily activities. Based on this concept we developed meetU, a tool that combines some of the main characteristics of social network systems and a set of pervasive services such as location estimation of users, awareness of use of artifacts, activity estimation and awareness of potential collaborators. meetU is implemented on top of the HLMP protocol, providing a lightweight communication infrastructure that can be easily deployed while offering privacy assurance to its users. The characteristics of this and similar working environments, including constant user mobility, the practical difficulties of deploying a fixed infrastructure, and the use of awareness of current activities to coordinate work, highlights the advantages of using an ad-hoc network to implement an ISN system such as meetU.

Potential users found the notification of status change to be useful for a variety of applications, especially if the system automatically determines the status of users; since hospital work is event-driven, the occurrence of some events could trigger the need to collaborate with others or to switch the task being performed.

Additionally, participants found it useful to have access to the network of contacts to facilitate tasks when users are not aware of their potential collaborators. This situation usually happens in the hospital due to the constant rotation of medical staff. Thus, the integration of the contact's network with the hospital information system could be particularly useful for working teams which composition is continuously changing (e.g. scheduled shift rotation).

Finally, concerning some of the limitations of the conducted evaluation, in this case we opted for a focus group scenario-based evaluation with three actual MIs acting as experts of the domain (informal communication in hospital work). This allowed us to validate the feasibility of our approach and to determine the MIs perception about the proposed tool. Future research should include further evaluation of the Impromptu Social Network concept and the proposed meetU tool with a larger number of hospital workers, as well as evaluating its actual use in the hospital.

Acknowledgments

This work was partially supported by LACCIR under grant No. R0308LAC005 and a travel grant provided by LACCIR to the second author, No. S1009LAC001.

References

1. Bardram, J., Bossen, C.: Moving to get aHead: Local Mobility and Collaborative Work. In: 5th European Conference on Computer Supported Cooperative Work (ECSCW), Copenhagen, Denmark, pp. 355–374 (2003)

2. Bardram, J., Hansen, T.: The AWARE architecture: supporting context-mediated social awareness in mobile cooperation. In: ACM Conference on Computer Supported Cooperative Work (CSCW 2004), Chicago, USA, pp. 192–201 (2004)
3. Bossen, C.: The parameters of common information spaces: The heterogeneity of cooperative work at a hospital ward. In: ACM conference on Computer Supported cooperative Work (CSCW 2002), New Orleans, USA, pp. 176–185 (2002)
4. Carroll, J., Neale, D., Isenhour, P., Rosson, M., McCrickard, D.: Notification and awareness: synchronizing task-oriented collaborative activity. International Journal of Human-Computer Studies 58(5), 605–632 (2003)
5. Corson, S., Macker, J.: Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. IETF, RFC 2501 (1999)
6. Cutrell, E., Czerwinski, M.H.: Notification, Disruption and Memory: Effects of Messaging Interruptions on Memory and Performance. In: Human-Computer Interaction (INTERACT), Tokyo, Japan, pp. 263–269 (2001)
7. Dabbish, L., Kraut, R.: Controlling Interruptions: Awareness Displays and Social Motivation for Coordination. In: ACM Conference on Computer Supported Cooperative Work (CSCW 2004), Chicago, USA, pp. 182–191 (2004)
8. Fish, R., Kraut, R., Root, R., Rice, R.: Video as a technology for informal communication. Communications of the ACM 36(1), 48–61 (1993)
9. Davis, F., Venkatesh, V.: A critical assessment of potential measurement biases in the technology acceptance model: three experiments. International Journal of Human-Computer Studies 45(1), 19–45 (1996)
10. DiMicco, J., Millen, D., Geyer, W., Dugan, C., Brownholtz, B., Muller, M.: Motivations for social networking at work. In: ACM Conference on Computer Supported Cooperative Work (CSCW 2008), San Diego, USA, pp. 711–720 (2008)
11. eHealthInsider, February 26 (2009)
12. Gutwin, C., Greenberg, S.: Workspace Awareness for Groupware. In: ACM Conference on Human Factors in Computing System (CHI), Vancouver, Canada, pp. 208–209 (1996)
13. Herskovic, V., Ochoa, S., Pino, J., Neyem, A.: Mobile Collaborative Systems: Behind the User Interface. Journal of Universal Computer Science (to appear, 2010)
14. Isaacs, E., Walendowski, A., Ranganthan, D.: Hubbub: a sound-enhanced mobile instant messenger that supports awareness and opportunistic interactions. In: ACM Conference on Human Factors in Computing Systems (CHI), Minneapolis, USA, pp. 179-186 (2002)
15. Isaacs, E., Tang, J., Morris, T.: Piazza: A Desktop Environment Supporting Impromptu and Planned Interactions. In: ACM Conference on Computer Supported Cooperative Work (CSCW 1996), Boston, USA, pp. 315–324 (1996)
16. Isaacs, E., Whittaker, S., Frohlich, D., O'Conaill, B.: Informal communication re-examined: New functions for video in supporting opportunistic encounters. In: Video-Mediated Communication. Lawrence Erlbaum Associates, New Jersey (1997)
17. Joinson, A.: 'Looking at', 'Looking up' or 'Keeping up with' people? Motives and Uses of Facebook. In: ACM Conference on Human Factors in Computing System (CHI), Florence, Italy, pp. 1027–1036 (2008)
18. Kraut, R., Streeter, L.: Co-ordination in software development. Communications of the ACM 38(3), 69–81 (1995)
19. Kraut, R., Fish, R., Root, R., Chalfonte, B.: Informal communication in organizations: form, function and technology. In: The Claremont Symposium on Applied Social Psychology, pp. 145–199. Sage Publications, Thousand Oaks (1990)

20. Lampe, C., Ellison, N., Steinfield, C.: A face(book) in the crowd: social Searching vs. social browsing. In: ACM Conference on Computer Supported Cooperative Work (CSCW 2006), Banff, Canada, pp. 167–170 (2006)
21. McFarlane, D.: Coordinating the Interruption of People in Human-Computer Interaction. In: Human-Computer Interaction (INTERACT), Edinburgh, Scotland, pp. 295–303 (1999)
22. Mejía, D., Favela, J., Morán, A.L.: Understanding and Supporting Lightweight Communication in Hospital Work. IEEE Trans. on Information Technology in Biomedicine 14(1), 140–146 (2010)
23. Monares, A., Ochoa, S., Pino, J., Herskovic, V., Rodriguez-Covili, J., Neyem, A.: Mobile Computing in Urban Emergency Situations: Improving the Support to Firefighters in the Field. Expert Systems with Applications (to appear, 2010)
24. Morán, E., Tentori, M., González, V., Martinez-Garcia, A., Favela, J.: Mobility in Hospital Work: Towards a Pervasive Computing Hospital Environment. International Journal of Electronic Healthcare 3(1), 72–89 (2005)
25. Ploderer, B., Howard, S., Thomas, P., Reitberger, W.: "Hey world, take a look at me!": Appreciating the Human Body on Social Network Sites. In: Oinas-Kukkonen, H., Hasle, P., Harjumaa, M., Segerståhl, K., Øhrstrøm, P. (eds.) PERSUASIVE 2008. LNCS, vol. 5033, pp. 245–248. Springer, Heidelberg (2008)
26. Rodriguez-Covili, J., Ochoa, S., Pino, J.: Enhancing Mobile Collaboration with HLMP. In: 14th Computer Supported Collaborative Work in Design (CSCWD), Shanghai, China, pp. 467–472 (2010)
27. Rodriguez-Covili, J., Ochoa, S., Pino, J., Messeguer, R., Medina, E., Royo, D.: HLMP API: A Software Library to Support the Development of Mobile Collaborative Applications. In: 14th Computer Supported Collaborative Work in Design (CSCWD), Shanghai, China, pp. 479–484 (2010)
28. Rodriguez-Covili, J., Ochoa, S.F., Pino, J.A., Herskovic, V., Favela, J., Mejía, D., Morán, A.L.: Towards a Reference Architecture for the Design of Mobile Shared Workspaces. Future Generation Computer Systems (to appear, 2010)
29. Sanchez, D., Tentori, M., Favela, J.: Activity Recognition for the Smart Hospital. IEEE Intelligent Systems 23(2), 50–57 (2008)
30. Segura, D., Favela, J., Tentori, M.: Sentient Displays in Support of Hospital Work. In: 3rd Symposium on Ubiquitous Computing and Ambient Intelligence (UCAmI), Salamanca, Spain, pp. 103–111 (2008)
31. Tollmar, K., Sandor, O., Schömer, A.: Supporting Social Awareness @ Work: Design and Experience. In: ACM Conference on Computer Supported Cooperative Work (CSCW 1996), Boston, USA, pp. 298–307 (1996)

Ambient Displays for Integrating Older Adults into Social Networking Sites

Raymundo Cornejo[1], Jesús Favela[1], and Mónica Tentori[2,3]

[1] Department of Computer Science, CICESE, México
{rcornejo,favela}@cicese.mx
[2] School of Computer Science, UABC, México
mtentori@uabc.mx
[3] Department of Informatics, UCI, USA
mtentori@uci.edu

Abstract. Social networking sites (SNS) help users sustain and strengthen ties with friends and relatives. However older adults who are less technically inclined individuals are often left aside these SNS felling as outsiders within their own family that uses this media to socialize. To assist these non-technical older adults we developed two ambient displays aimed at seamlessly integrate them into SNS. The first prototype, ePortrait, shows to older adults the photographs uploaded by their relatives in a SNS and the second prototype, eBowl, provides means to communicate their status through the manipulation of physical objects. The results of an evaluation shows that ambient displays are capable of monitoring older adults' context and provide them with continuous information about their social network in a subtle, peripheral and expressive manner. As a consequence, older adults use the information conveyed by such displays to enhance conversations with new topics helping them feel more integrated with their family.

Keywords: Ambient displays, social networking sites.

1 Introduction

Social ties and social integration often play a beneficial role for maintaining psychological well-being contributing to a better mental health of older adults. Strong social networks may enhance the quality of life of older adults [1] improving their health [2], reducing the chances for developing cognitive decline [3] and eventually preventing an earlier death [4]. Older adults with poor or weak social networks are 60% more likely to develop cognitive decline due to isolation or high stress levels caused by the loss of the spouse [5]. Also, socially active older adults are physically and mentally healthier in contrast with those who are isolated.

The above underlines the importance of friends and relatives and the impact of these interactions in older adults' life. However, the risk that a person becomes socially isolated increases with age [6] due to retirement, children living in different locations or because they suffer the loss of close friends. Social networking sites (SNS), such as Facebook or MySpace, have the potential of strengthening the social

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 321–336, 2010.

network of older adults. Social networking sites are a source of social capital, providing social or emotional support and access to information resources[7]. SNSs are attracting millions of users worldwide. Users of SNSs are predominantly in their twenties [8], but their use is gradually increasing in other age groups. In 2009 it was estimated that only 7.6% SNS users were more than 55 years old [9]. But the percentage of older adults online has increased over the past years, online older adults are getting Internet access for reasons unrelated to work and nearly 3 in 5 online seniors in the United States have said that the Internet has improved their connections with their family [10]. The top Internet activities among older adults are email, getting online hobby information, reading online news and searching for health and medical information.

However, online older adults are still a minority on SNSs compared to younger adults whose main SNSs activities are staying in touch with friends, planning activities or events and making new friends [11]. Thus, older adults might be missing opportunities for sharing and bonding with family members who are adopting these new interaction spaces. This low participation and engagement may be due to technology anxiety [12] (e.g. computers or cell phones) and because even older adults, who have been using computers at work, do not trust their own computer skills and consider computers as a working tool rather than a communication tool [13]. This might even make older adults feel like outsiders within their own family.

Although there are obviously many advantages to the availability of these services, they paradoxically, are limited in their potential for non-technical users to socialize through them. Moreover, these less technically inclined individual may even feel like outsiders in their off-line social networks or families, contributing to further cognitive decline and isolation. To fully support these populations, then, we require a new computing paradigm: one capable of seamless integration of both traditional and digital social communication tools.

Research in ambient information systems has included the development of ambient devices that provide relevant information to users through natural artifacts naturally interwoven in the environment. As stated by Mankoff et al.: "Ambient displays are aesthetically pleasing displays of information which sit on the periphery of users' attention. They generally support the monitor of information and have the ambitious goal of presenting information without distracting or burdening the user". Using these natural artifacts and interfaces is possible to design adaptable prototypes that provide older adults with a natural form of interaction, reducing their anxiety and increasing their self-efficacy.

Ambient displays are a subset of peripheral displays that continuously display information that can be monitored by the user without requiring her focused attention [14]. Commonly these displays present relevant information to users through natural artifacts being used in the environment like calendars [15], digital portraits [16-17] or decorative lamps [18]. For example, one of the first ambient displays that supports independent living by older adults, was the "digital family portrait" [17]. This design provide a qualitative sense of an older adult's daily activities, maintaining all the family members informed by capturing observations that would naturally occur if someone lives with the older adult. The use of ambient displays may help to support reciprocity in everyday asymmetric relationships helping older adults to socialize through SNSs being already used by younger generations.

In this paper, we propose two solutions for "unboxing" some of the services of SNSs providing older adults with a natural way to interact with their families SNSs. These unboxed services will create an ambient social network site composed with augmented everyday objects with digital capabilities acting as a natural interface while providing their users with continuous information about their social network in a subtle, peripheral and expressive manner. The development of such ambient social network sites open questions as to how they will enhance the use of SNS with ubiquitous services; answering questions such as "What are the main communication barriers between older adults and their relatives", "What are common conversation topics?" and "How meaningful SNS services can be integrated into the environment?"

To achieve this, we introduce the design of an ambient social network integrated by two ambient displays that incorporate social networks sites services for maintaining and strengthening family ties between older adults and their relatives. In section 2 we discuss how our work relates to previous work. In section 3 we present a qualitative study conducted to understand how older adults maintain their social ties. In section 4 we present the design and implementation of an ambient SNS. Section 5 presents the results of an evaluation of two ambient displays deployed in an older adult's home. Finally, in section 6 we present our conclusions and directions for future work.

2 Background and Related Work

There is considerable research on how technology can help older adults to live independently and provide awareness of their daily activities to their extended family. CareNet Display[16] and the Family Digital Portraits [17] are augmented digital frames designed to support older adults age in place and members of their care network be informed about sensitive data to facilitate day to day care. However, these systems don't provide noticeable ways of interaction to strengthen social and emotional ties between the relatives and the older adult.

Also research related to relationships and communication has explored the use of situated displays to enhance communication and emotional connectedness at a distance. Lumitouch [19], is a situated display focused on enhancing communication between loved ones through picture frames and light patterns. If a user picks up a picture frame, the displayed colors are transmitted to the corresponding Lumitouch. Likewise, the Relational Pillow [20] is an augmented pillow that provides a simple, intimate, and personable communication medium between loved ones through touch sensors and light patterns. Both projects implement a communication schema designed to support a one on one interaction.

Other researchers have enriched common home objects such as lamps or plants to induce a sense of presence. The 6th sense [21] is a light sculpture with multiple small lights each one representing a person close to the user. A similar project is the FamilyPlanter system [22], with infrared and ultrasonic sensors that detect a person's presence and sends this information to the sister plant in a remote location. The receiver plant obtains the motion information and gleams fiber optics to indicate the remote human presence and rotates to indicate the remote human motion.

Some researchers have focused on intergenerational interactions across homes. Virtual box [23] is designed to mediate intimacy between children and grandparents through a virtual hide and seek game. The virtual box carries various virtual items that can be hidden in the house, and while playing the game the grandparents and children exchange virtual objects such as messages, gifts or photos. A similar display to eBowl is GustBowl [24], a one to one interaction ambient display designed to share everyday experiences between mothers and sons. When the son's bowl detects the presences of an object, it records its movement and takes a snapshot of it, and then this information is sent to the parents' home. With this information the parents' bowl will wobble as recorded and display the snapshot of the object with the intention of transmitting the message of “arriving home”.

In efforts such as this, where the aim is to maintain a feeling of connectedness between generations, similar studies explored situated displays to keep older adults in touch with people they care about. For example the Epigraph [25] is a screen divided into a number of channels, one representing each family member. Channels can be updated via email, text or picture message. Nevertheless this prototype provides different ways of sending content to the older adult. This requires an extra effort from family members and has a fixed number of channels for a limited number of contacts. Similarly, the Sharing the Day's events [26] sends pictures, drawings or videos to a fixed screen in the older adult's home.

There is a considerable research in use of ambient technology to help older adults to live independently; also previous explorations have designed technology to provide a sense of connectivity between family members. The focus has been on providing a sense of awareness among family members by monitoring the older adult's activities or providing manners to share “digital snapshots” of the family members daily living. Our research, however, focuses on providing this asymmetrical awareness by integrating older adults to existing SNS with the use of ambient displays.

3 Understanding the Role of Communication and Interaction to Maintain Emotional Family Ties

For four months we studied how older adults maintain emotional ties with their relatives in three conditions: (a) older adults with cognitive decline that live in a private nursing home (b) older adults living with their relatives or independently and (c) older adults participating in a social network site. A total of 9 older adults, 32 of their relatives and the nursing manager participated in the study.

3.1 Methodology

Figure 1 shows the methodology of our qualitative study divided in three main phases. Our aim of this study was designed to capture four major themes: (1) how family ties are maintained, (2) which artifacts use during the social interactions, (3) how family interactions are promoted and (4) the main social barriers that difficult these family interactions.

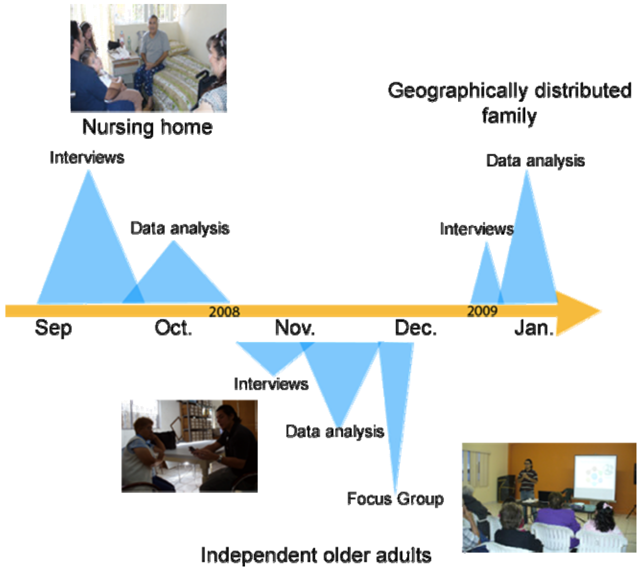


Fig. 1. Field trial methodology

Our first phase was conducted at a private nursing home that it is specialized in the care of older adults with Alzheimer or Dementia. Currently there are thirteen patients registered who are in different stages of Alzheimer ranging from initial to terminal stages. The medical practices are paper-based and video cameras are placed in public areas within the residence premises to allow relatives to watch their older adult via Internet. At the nursing home we conducted interviews with the geriatric animator (n=1), with older adults aged between 65 and 97 with or without mild cognitive decline (n=4, 1 female) and with the relatives visiting the residence (n=3, all males). We selected our participants based on their mild cognitive decline. However, all of them have also some physical illness or had sequels from a severe clinical episode such as a stroke.

During our second phase we studied older adults attending to a government social welfare facility where their main goal is to provide to older adults the tools they need to be economically self-sufficient guaranteeing their integrity and security. Based on this, the main requirement for older adults is to be mentally healthy, primarily because the occupational therapies require some skills, for example, painting or making imitation jewelry. We conducted interviews with the social coordinator (n=1) and with older adults living independently and assisting to occupational therapies (n=5, all females), aged 62 to 81.The selection criteria were based on them attending the therapies, at least twice a week.

Finally, our third phase, aimed at further explore communication issues between older adults and their family. We chose one family from the second group of our informants. We selected this family because their members are geographically distributed and most of them have been using Facebook for the last two years. The family includes an 89 years old adult (i.e. the grandmother) who lived with one daughter and 18 relatives of two generations, 8 males (i.e., 2 sons, 4 grandchildren and 2 grandchildren in-law), 10 females (i.e., 4 daughters, 1 daughter in-law, 5 granddaughters and 1 granddaughter in law), aged between 18-55. Family members are geographically distributed, in 6 cities within two countries. The grandmother is an active person who has consistent routines for each day of the week, everyday goes to a gym to walk on a treadmill for around 10 to 20 min. And despite of the fact that she takes English and multimedia classes, she is not a regular user of technology, for example, she seldom uses a computer and when she does, it is with the assistance of a relative, nor does she uses a cell phone. With this family we conducted interviews and a community attachment questionnaires with the grandmother (n=1) and with 2 relatives (n=2, females).

We later transcribed and analyzed these handmade detailed records using grounded theory a systematic research methodology for generating theory from data [27].

3.2 Results

We found that one of the biggest barriers in maintaining strong emotional ties is that older adults are often disassociated with events related to their relatives' everyday life. Interesting patterns of communication arise from the time participants spend with the older adult including their geographical proximity and their age. Also we can see how the use of SNS among our participants in the third phase has changed for incorporating strategies to keep in touch with their community using the SNS. Finally it was interesting to found that conversation topics varied from shallow interactions to emotional personal aspects when interacting face-to-face.

Barriers in Maintaining Emotional Ties. We found out that the major barrier for our participants to maintain emotional ties was geographical distance. Children who moved out of town for seeking new employment opportunities or for continuing their studies abroad had more difficulties to keep in touch with the older adult. As a consequence, older adults frequently felt disassociated with family events because they cannot communicate as often as they wanted due the high cost of social communication (telephone, cellular phone).

Particularly in the nursing home we found that severe cognitive decline impacts negatively in the frequency of family visits. Also, the lack of knowledge led to feelings of resentment towards the older adult during episodes of mental crisis.

It was interesting to observe how the grandmother felt disassociated with family events because family members were using a SNS or email to stay in touch with each other and she was unaware of the content of these interactions. As a consequence, some of their relatives act as proxy to the SNS by occasionally showing to the grandmother the type of information shared through such social media.

Communication Among Family Members. From the participants studied we identified three interesting cases that illustrate general patterns of communication between the grandmother and her relatives. These patterns of communication depend on the time participants spend with the older adult, their geographical proximity and their age. We illustrate through scenarios some of these communication patterns:

> *"Mary[1] is a teenager with an active social life that sporadically visits her grandma. As most teenagers, she is involved in several social activities including playing volleyball and partying with friends. She is also an active Facebook user, uploading on average 3 photos and making 4 comments, per day. She communicates with her grandma mostly by phone and they habitually talk about Mary's school, grandma's favorite television shows and English classes. Soon, Mary will move out of her hometown to attend college. She is a bit concerned on how she might be able to keep in touch with her grandmother when she leaves town."*

Despite Mary's interest in maintaining emotional ties with her grandma, grandma is quite unaware of Mary's everyday activities. In contrast, other members of the family have incorporated strategies to keep grandma up to date with important events in their lives. That is the case of Jimmy:

> *"Jimmy, her grandson, lives in Mexico and just had a son. He calls grandma twice a week and talks with her about his job, his son's activities and his family's well being. He also discusses with her the most important events that have happened in her life. In addition, as a lawyer, he is helping her with a legal problem. To complement their conversation he sends photos to his grandma through one of his aunts. Jimmy is a frequent Facebook user, uploading 2 photos and making 2 comments on average per day. He takes himself a daily picture and uploads it to the SNS."*

Even though Jimmy is not as active as Mary in the SNS, the information he uploads aims at keeping his family updated on his life. A third case is one of the older adult's daughters:

> *"Danna is grandma's oldest daughter, lives with her, and takes care of her well being. Danna uses Facebook once a month and shows grandma digital photos and emails sent to her by other family members."*

Becoming an Outsider for not Using Technology. The analysis of the family's communication patterns showed that the use of Facebook has been rather heterogeneous due participants' computer skills and computer access. The content of the information uploaded by the participants into Facebook has changed over time. At the beginning participants uploaded information aimed at building a persona in the SNS. Before Facebook, this type of information wasn't being shared and learning about it raised an interest in relatives in preserving the community at the SNS. Therefore the type of information being shared in Facebook became more personal and highly relevant to the family. It is interesting to note that despite of grandma not participating in Facebook there is an interest of both, relatives and her, to become part of it.

> Grandmother: *"Well, I tell them (her family members) to show me the photographs, otherwise they may forget…"*

As consequence she misses a lot of information and sometimes she is not able to follow a conversation in a family reunion when family members start talking about

[1] All names have been changed to protect the privacy of our informants.

"Facebook events", such as a comment in a photo. What is clear is that this family shares a lot of information through the SNS and as a result, the grandmother feels disconnected with such community and even feels like an outsider in her own family.

Information Shared. The type of information shared among members of this family is a spectrum from merely shallow interactions, such as jokes, to such information shared with an emotional component, such as problems or even feelings.

It was interesting to found that the type of information shared when participants interact face-to-face was very shallow. Participants who live nearby interact quite often; therefore, there is no need for them to exchange information about their well being, their location or activities. Instead the family used the time they spend together to have a playful and relax time as gateway to escape from their problems and their everyday routines. In contrast, the type of information shared in Facebook is very explicit and frequently includes information about feelings and moods. In addition, this information shared by participants in Facebook includes a lot of detail about their location, activity, status and relevant pats or upcoming events. For instance, one might know specifics about one's schedule of work/school and social activities. Work details generally include knowing the days and times that one is working/studying, rather than knowledge of work appointments and meetings. For instance, once Mary changed her status to "boring at […] class". As a result, six of her relatives make fun of her telling that she must pay attention to her class instead of playing around with Facebook. Then, Mary explained to them that she had just finished the class' exercise so she had time to play a bit.

4 An Ambient Display to Connect Older Adults with Their Family SNS

The study findings motivated the development of an ambient social network that seamless integrates older adults to SNS. The ambient SNS is made of two ambient displays, the ePortrait and the eBowl. The aim of the ePortrait is to provide awareness to less technically inclined individual of their on-line social networks. The eBowl provides a natural interface to interact directly into the SNS by sharing shallow information as usually happens when relatives interact face to face. Following a user-centered approach we developed design requirements, indentified limitations and scenarios that were later used to iteratively develop the ePortrait and the eBowl.

4.1 Implementation of the ePortrait

Figure 2 shows the general architecture of ePortrait. The system collects new photographs published in Facebook, edits them and publishes them in a RSS file read by the digital frame. The main module retrieves, through Facebook API calls, the list of friends associated to the older adult's Facebook account and retrieves the last ten photos uploaded by each one of them. Once the photographs are downloaded, the Editing module adds at the bottom of each photograph a caption, written by the family member, and a reference number at the top right corner. The photo's context is enriched with these references, helping the older adult to associate each photo with his/her relatives.

Fig. 2. The ePortrait's architecture

The RSS feed contains the last four days of photographs uploaded, giving the older adult the opportunity to enjoy them for a reasonable period of time. The digital frame is configured to read the RSS feed each time it is turned on and each photograph is displayed for one minute. Figure 3 shows an example of a photograph displayed in the ePortrait.

Fig. 3. Photograph showed in the ePortrait

4.2 Integrating Feedback into the ePortrait: The eBowl

The objective of the eBowl is to integrate feedback to the ePortrait to fulfill two main purposes: (1) share a joke and indicate how funny the joke seems to the user and (2) monitoring the older adult's home presence.

We enhanced the ePortrait with an ambient display to provide a bidirectional interaction between the older adult and her family social network avoiding the high cognitive load that a traditional interface of the SNS presents and activity cues for inferring the older adult's availability.

The eBowl consists of a bowl decorated with an array of leds and a set of USB building blocks denominated Phidgets [28]. The array of leds that emit derided light patterns, RFID tags inside the vine balls, a RFID phidget to monitor the RFID tags, a PhidgetLED to control the array of LEDs and a netbook which controls both phidget boards (see Figure 4). The ePortrait retrieves a daily joke from a repository of jokes that is maintained in a server and creates a photograph that contains it, which is added at the end of the RSS file. This content can be easily monitored by the older adult and promotes family interaction through the eBowl.

Our design allows the older adult to share the current joke displayed in the digital frame by placing a unique "sharing" vine ball inside the eBowl. The ball contains a RFID tag which, when identified, triggers the service to share the current joke with the family, via Facebook. Every family member can then read and rate the joke. At a given time, the service collects the rates of the current joke and displays a light pattern on the older adult's eBowl as an indication of the relatives' laughs.

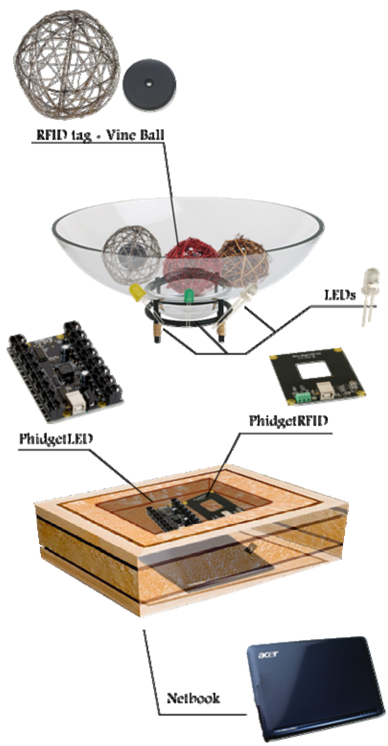


Fig. 4. The eBowl sentient display

We added an extra feature to the eBowl to notify relatives the whereabouts of the older adult. When the older adult places her home keys inside the eBowl, her status in the SNS changes to "at home", providing an activity cue to the relatives (see Figure 5). Similarly, when she takes out the keys from the eBowl, members of the social network will be notified that she is going out. This gives relatives the opportunity of knowing when it might be a good time to call.

Fig. 5. Activity cue using eBowl

5 Evaluation

For a period of 21 weeks, we conducted a field trial of our prototype system with one family. From the older adults who participated in the original qualitative study, we selected Meg, an 88 year-old grandmother whose relatives are geographically distributed, many of whom have been using Facebook at least two years. A total of 18 relatives participated in the use of both ambient displays and had on average a 19 moths (s.d. 6.4) experience using SNS services such as sharing, tagging and commenting photographs.

A total of eight participants were selected, four grandchildren (n=4, two females) and four daughters. Two grandchildren and two daughters who participated in the study lived abroad and one daughter lived with the older adult and was the primary caregiver. These 8 relatives are a representative sample of the family presenting a diversity of characteristics including geographic proximity, frequency of contact and relationship with the elderly. We conducted 6 interviews with the older adult and 8 interviews with relatives, 2 by phone, 4 by IM and 2 face to face, during the period of deployment.

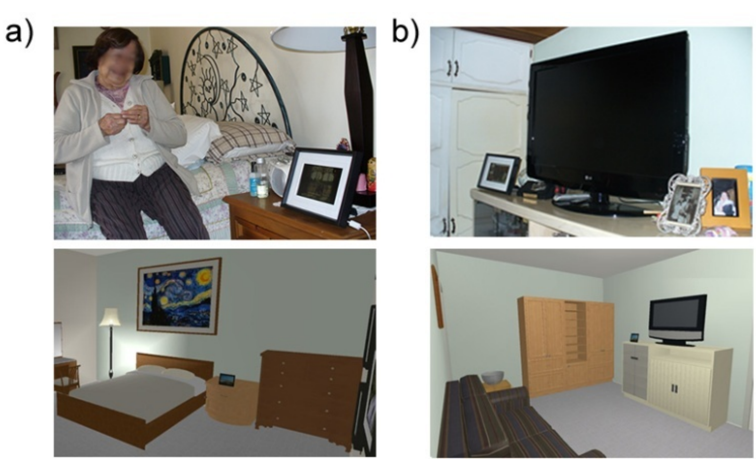


Fig. 6. a) The ePortrait in the bedroom b) The ePortrait and eBowl in T.V. room

The evaluation was conducted incrementally, introducing each ambient display at a different moment so the older adult and relatives became familiar with the designs and to evaluate the impact of each characteristic of both prototypes. During the first 9 weeks, the ePortrait was deployed in the bedroom of the grandmother (see Figure 6a) because grandmother Meg identified this location as a private space and where she spends most of their time. The next 12 weeks, the ePortrait was placed in the TV room (see Figure 6b) to explore a more periphery used of the display. And the eBowl was deployed in the TV room during the last week of evaluation.

5.1 Results

In this section we present the impact of both prototypes on certain aspects of grandmother Meg's life. These findings also reflect the impact of using SNS's outside the desktop environment and helping older adults maintain asymmetrical relationships with their relatives. Meg rapidly adopted both ambient displays and described them as pleasant and attractive.

> *"very pleasant, not difficult to use, I just need to turn it on/off"*

Also she stated that both displays were very useful, the ePortrait was a way to keep up with family events and eBowl as home awareness device for her relatives:

> *"very useful [ePortrait] to see if there is something new, to be informed…".*
>
> *"I love It [eBowl], because they know if I am home or not and if they call me they know I am going to answer"*

Pathways and Routines. When the ePortrait was deployed in Meg's bedroom we found out that she included to her daily routine the activity of watching the photographs every morning for around half an hour. We did not expect this type of behavior since older adults are very careful on managing their time and have specific communication routines [25]; nevertheless it was important to observe how the grandmother became interested in observing the activities of her relatives that she was unaware of. During one interview we asked why she spent so much time every day viewing the pictures. She stated that the ePortrait was a way for her to keep up with family events.

When the ePortrait was moved to the TV room we observed that Meg rather used it as a peripheral display providing her continuous awareness of their relatives' life. In contrast with the use of the ePortrait in the bedroom, in the TV room she just glance at the ePortrait when watching TV or passing nearby. Whenever she found something of interest she approached the digital frame to read the caption and/or look at the photograph closer and with more detail. Although in this setting she did not change her routine she did change her patterns of movement, recognizing the TV room as the center of his paths crossing by as much as possible to glance new information at the display.

Meg's relatives included new activities when using Facebook to their usual routines or persuaded them to more frequently use Facebook. For example, one granddaughter expressed that after the deployment of the ePortrait she started to upload photographs and tagging people to them.

> *"I upload more photographs and tag them more often. Before my grandma had the ePortrait I did not tag her in any photograph"* (granddaughter Mary)

Finally Danna who lives with the older adult now joins the SNS less often since she can see the photographs with the ePortrait rather than with the computer. She prefers to know what is new in the family through the digital frame rather than using the computer.

> *"I only use it once a month; it is easier just to glance at the digital frame".* (daughter Danna)

Empathy and Conversation Enhancer. We found that the ePortrait helped Meg develop empathy to her relatives, feeling part of their daily life experiences. Now, she is more aware of where they go, who their friends are, and what is their social environment. Despite the fact that the use of SNS involves a general concern about privacy issues, relatives did not show a concern related to sharing their daily life events with grandma Meg.

> *"I am not worried, because there aren't pictures that grandma should not see" (granddaughter Mary)*

Grandma Meg used the ePortrait as a virtual entity of the relatives in the SNS to induce a sense of presence. She described feeling emotionally connected with her loved ones even if they could not visit or call her. Also the grandmother realized how the photographs trigger and enrich their conversations:

> *"I keep seeing them… as when they come to visit me […] and I realized that we have more topics to talk about".*

Similarly, the eBowl help grandmother Meg to have new topic conversation that she always wanted, jokes. Besides the use of sharing jokes through eBowl, grandmother Meg uses the daily joke to enrich face to face conversations:

> *"I told a joke at my grandsons' home and they laugh a lot"*

Instead of inducing the sense of the physical presence in the older adult's home, the ePortrait induces a sense of presence by using the already existing virtual entity of each relative on the SNS, helping to establish an emotional connection through the photographs. During the interviews the older adult express that even though some of their relatives can visit her that often she felt more close to them almost like she is seeing them every day.

> *"I feel [my relatives] closer; if they can't come I still see them and keep remembering them. I am up to date on what they are doing; where they go… it is similar as if they came to visit me. If they cannot come for some reason, I do not feel sad because they are still thinking about me"*

Activity on the Family Social Network. Relatives expressed enthusiasm with the display. For example, one subject created his account, at the beginning of the project, just for the purpose of sharing photographs with the grandmother. Privacy issues was not a concern with relatives, they did no express any concern about the content being shared with grandmother Meg, mainly because they have an open relation with her grandmother. Table 1 shows the number of photographs uploaded before and after the use of ePortrait.

Table 1. Number of photographs uploaded

		Prior to ePortrait		With ePortrait	
Relationship	n	Total photographs	Average	Total photographs	Average
Grandsons	6	598	99.66	678	113
Granddaughters	7	555	79.23	765	109.29
Sons	2	31	15.5	0	0
Daughters	3	545	136..25	451	112.75

Relatives' activity in the SNS was rather heterogeneous—largely due to similarly heterogeneous computer usage patterns, but overall, they described being pleased with the participation of their grandmother in the SNS. As a result, they incorporated several strategies to get her involved. For instance, more photographs with family content were uploaded to Facebook with the aim of increasing the probability of sharing them with their grandmother. Another strategy was to tag the grandmother when photographs so she could see them more quickly. Consequently, Meg began using digital information as an information source when socializing with their loved ones. For instance, the older adult explained:

> *"I told them last night a joke about a photo. I like it because now I know about what can I talk to them".*

Also we observed a change in the content of the photographs and an increment in some activities in the family social network. After ePortrait was introduced relatives started to upload photographs with more family content:

> *"I try to upload more photographs or comments related to what I am doing"* (grandson Jimmy)

Relatives' activity in the SNS was rather heterogeneous—largely due to similarly heterogeneous computer usage patterns, but overall, they described being pleased with the participation of their grandmother in the SNS. As a result, they incorporated.

6 Conclusion and Future Work

Ambient social networks offer a new type of interaction, helping older adults to connect with SNS while avoiding the high cognitive load present in the standard interfaces of these systems. The ePortrait is an affective digital frame that allows an older adult to become part of a social virtual network and keep in touch with her relatives' life. The eBowl provides means to communicate their status through the manipulation of physical objects. The results of an evaluation highlighted the importance of providing feedback so that a bidirectional sense of presence in the social network can be established. This research will contribute to new opportunities of use of the online communities by giving new ways of interaction to the elderly.

We plan on investigating the advantages and problems of using ambient information systems to integrate the older adults to these online communities. In this longer evaluation a couple of ambient displays will be deployed with at least 3 families which not necessarily will have an expertise in using social networks. This will allow us to understand how the family dynamics might change over the use of these ambient devices or how privacy or cost issues might arise. Also with a longer evaluation we expect to gather interesting data to provide feedback into the design of these systems and might help prevent isolation, disassociation and enhance their quality of life.

References

1. Giles, L.C., Glonek, G.F.V., Luszcz, M.A., Adrews, G.R.: Effect of social networks on 10 year survival in very old Australians: The Australian longitudinal study of aging. J. Epidemiol. Community Health 59, 574–579 (2005)
2. Belvis, A.G.d., Avolio, M., Sicuro, L., Rosano, A., Latini, E., Damiani, G., Ricciardi, W.: Social relationships and HRQL: A cross-sectional survey among older Italian adults. BMC Public Health 8 (2008)
3. Fratiglioni, L., Wang, H.-X., Ericsson, K., Maytan, M., Winblad, B.: Influence of social network on occurrence of dementia: a community-based longitudinal study. The Lancet 355, 1315–1319 (2000)
4. House, J.S., Landis, K.R., Umberson, D.: Social relationships and health. Science (N.Y.) 241, 540–545 (1988)
5. Seeman, T.E.: Social ties and health: the benefits of social integration. Ann. Epidemiol. 6, 442–451 (1996)
6. Iliffe, S., Kharicha, K., Harari, D., Swift, C., Gillmann, G., Stuck, A.E.: Health risk appraisal in older people 2: the implications for clinicians and commissioners of social isolation risk in older people. Br. J. Gen. Pract. 57, 277–282 (2007)
7. Joinson, A.N.: Looking at, looking up or keeping up with people?: motives and use of facebook. In: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pp. 1027–1036. ACM, New York (2008)
8. DiMicco, J., Millen, D.R., Geyer, W., Dugan, C., Brownholtz, B., Muller, M.: Motivations for social networking at work. In: Proceedings of the 2008 ACM conference on Computer supported cooperative work, pp. 711–720. ACM, San Diego (2008)
9. comScore, `http://www.comscore.com/press/release.asp?press=1019`
10. Fox, S.: Wired Seniors. Pew Foundation, Washington (2001)
11. Lenhart, A.: Adults and Social Network Websites Pew & Internet (2009)
12. Czaja, S.J., Charness, N., Fisk, A., Hertzog, C., Nair, S., Rogers, W., Sharit, J.: Factors predicting the use of technology: findings from the Center for Research and Education on Aging and Technology Enhancement (CREATE). Psychology & Aging 21, 333–352 (2006)
13. Lehtinen, V., Näsänen, J., Sarvas, R.: A little silly and empty-headed: older adults' understandings of social networking sites. In: Proceedings of the 2009 British Computer Society Conference on Human-Computer Interaction, pp. 45–54. British Computer Society (2009), 1671017

14. Mankoff, J., Dey, A.K.: From conception to design: a practical guide to designing and evaluating ambient displays. In: O'Hara, K.P.M., Churchill, E., Russell, D. (eds.) Public and Situated Displays: Social and Interactional Aspects of Shared Display Technologies, pp. 210–230. Kluwer Academic Publishers, Dordrecht (2003)
15. Plaisant, C., Clamage, A., Hutchinson, H.B., Bederson, B.B., Druin, A.: Shared family calendars: Promoting symmetry and accessibility. ACM Trans. Comput.-Hum. Interact. 13, 313–346 (2006)
16. Consolvo, S., Roessler, P., Shelton, B.: The CareNet Display: Lessons Learned from an In Home Evaluation of an Ambient Display, pp. 1–17 (2004)
17. Mynatt, E.D., Rowan, J., Craighill, S., Jacobs, A.: Digital family portraits: supporting peace of mind for extended family members. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 333–340. ACM, New York (2001)
18. Hindus, D., Mainwaring, S.D., Leduc, N., Hagström, A.E., Bayley, O.: Casablanca: designing social communication devices for the home. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, New York (2001)
19. Chang, A., Resner, B., Koerner, B., Wang, X., Ishii, H.: LumiTouch: An Emotional Communication Device. In: Proc. SIG-CHI on Human factors in computing systems, pp. 313–314. ACM, New York (2001)
20. Fluid Interaction Group, http://fluid.media.mit.edu/people/sajid/past/pillowtalk.html
21. Tollmar, K., Persson, J.: Understanding remote presence. In: Proceedings of the second Nordic conference on Human-computer interaction, pp. 41–50. ACM, New York (2002) 572027
22. Itoh, Y., Miyajima, A., Watanabe, T.: 'TSUNAGARI' communication: fostering a feeling of connection between family members. In: CHI 2002 extended abstracts on Human factors in computing systems, pp. 810–811. ACM, Minneapolis (2002)
23. Davis, H., Skov, M.B., Stougaard, M., Vetere, F.: Virtual box: supporting mediated family intimacy through virtual and physical play. In: Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces, pp. 151–159. ACM, Adelaide (2007)
24. Keller, I., Hoog, W.v.d., Stappers, P.J.: Gust of Me: Reconnecting Mother and Son. IEEE Pervasive Computing 3, 22–28 (2004)
25. Lindley, S.E., Harper, R., Sellen, A.: Desiring to be in touch in a changing communications landscape: attitudes of older adults. In: Proceedings of the 27th international conference on Human factors in computing systems, pp. 1693–1702. ACM, New York (2009)
26. Evjemo, B., Svendsen, G.B., Rinde, E., Johnsen, J.-A.K.: Supporting the distributed family: the need for a conversational context. In: Proceedings of the third Nordic conference on Human-computer interaction, pp. 309–312. ACM, New York (2004)
27. Strauss, A., Corbin, J.M.: Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. SAGE Publications, Thousand Oaks (2007)
28. Greenberg, S., Fitchett, C.: Phidgets: easy development of physical interfaces through physical widgets. In: Proceedings of the 14th annual ACM symposium on User interface software and technology, pp. 209–218. ACM, New York (2001), 502388

A Simple and Portable Command Post to Coordinate Search and Rescue Activities in Disaster Relief Efforts

Rodrigo Bartels[1], Valeria Herskovic[2], Alvaro Monares[2], Sergio F. Ochoa[2], José A. Pino[2], and Marcos R.S. Borges[3]

[1] Computer Science and Informatics School, University of Costa Rica, Costa Rica
rodrigo.bartels@ecci.ucr.ac.cr
[2] Computer Science Department, Universidad de Chile, Chile
{vherskov,amonares,sochoa,jpino}@dcc.uchile.cl
[3] Department of Computer Science, Univ. Federal do Rio de Janeiro, Brazil
mborges@nce.ufrj.br

Abstract. The lack of communication channels and support information can make the critical search and rescue of survivors after a disaster an ineffective process resulting in losses. This work addresses this problem by proposing a simple and portable command post application based on mobile devices and ad-hoc networks. The application provides support to communication and collaboration aiming to speed up the SAR process. The system architecture and some implementation issues are also presented.

Keywords: Portable command post, search and rescue, coordination activities, information support, mobile workers support, low-cost system.

1 Introduction

When large-scale disasters affect urban areas, large response efforts must be done in order to try to minimize the effect of the extreme event on the population. Earthquakes, tsunamis, volcano eruptions and floods are some of these extreme events. A large number of organizations participate in the relief effort: police, firefighters, health organizations, government agencies, ONGs, and even citizens, who may look for and try to rescue other people.

In these situations, it is common to hear about improvisation, delayed assistance, lack of information to make decisions, and lack of communication and coordination among the first responders conducting the response process [13]. The empirical evidence shows there is no country prepared enough to deal with large-scale disasters. This situation is worse in countries with limited resources (such as personnel, equipment, and transportation systems), where first responders need to conduct fast and effective response processes. Recent earthquakes in Haiti and Chile unveiled the challenges to be addressed by assistance organizations when the need for response largely overcomes the capability of the task force.

A very urgent and critical activity to be conducted after a natural catastrophe is the search and rescue of survivors. Typically, firefighters play a key role in this activity, because they are trained to be the first-line task force during emergencies. In the case

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 337–344, 2010.

of Chile and several other countries, firefighters are basically volunteers. Furthermore, their organizations typically have little state-funded support. Thus, any solution they apply or use in the field must involve a low cost for the organization.

Firefighters have a clear protocol that allows them not only to work autonomously, but also to cooperate with other units in order to reach a common goal. Moreover, the search and rescue (SAR) protocol allows firemen to trigger relief efforts within a short time period of 48-72 hours. However several studies indicate the first 24-48 hours are the most critical ones to rescue survivors [2; 4].

This article proposes a simple and portable command post to improve the effectiveness and speed of the search and rescue process. This command post can be implemented through a lightweight and autonomous collaborative system including pre-loaded information to support decision-making.

Section 2 describes the main issues involved in the search and rescue international protocol. Section 3 explains the design aspects involved in the proposed portable command post. Section 4 presents and discusses the related work. Section 5 presents the conclusions and future work.

2 Search and Rescue International Protocol

The first preventive task that has to be done before a large-scale emergency is the definition of a protocol to guide rescue efforts. The Incident Command System (ICS) establishes a well-known and validated protocol for addressing emergencies of various sizes, within a flexible and scalable organization [9]. This response protocol establishes the first team arriving to the emergency site is in charge of the response process. The most experienced firefighter in the field becomes the incident commander (IC). The incident commander works in the command post, which is located in a safe area; usually outside of the emergency site. This person is responsible for organizing, coordinating and controlling the relief efforts. The IC also acts as a hub that gathers and distributes information from/to the rescuers.

The IC task force includes personnel in charge of the operations, logistics and planning area (Fig. 1). The operations team leader (OTL) is in charge of requesting and coordinating actions in the field. This person divides the team in small units and assigns them a unique work area and a specific goal. Each response unit has a leader, who is in charge of coordinating the rescuers and reporting information to the OTL. Similarly, the OTL reports to the incident commander.

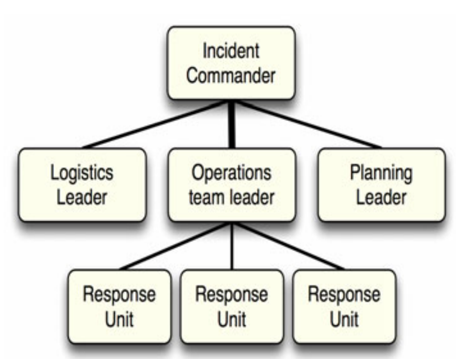


Fig. 1. Incident command system hierarchy

While the work done by response teams is important, the work performed by the IC and the OTL plays a key role in the search and rescue process. Most decisions made in the field are supported by information provided by these persons.

Today many IC and OTL continue using whiteboards, paper and pencils to perform their job. Most of the information these persons exchange among them and with the response unit leaders is based on paper and radio messages. The main reasons to use these tools are two: (1) they have proven to be useful in traditional emergency management, and (2) the alternative solutions available in the market are too expensive for volunteer firefighting organizations.

This paper proposes to replace this traditional command post by a portable one able to run on a laptop or tablet PC. This command post is easy to transport and deploy. Its cost is similar to the cost of the device where it is running. The collaborative system implementing the command post functionality is used by the IC, OTL and response unit leaders. Thus the information exchanged among these people may become fast and accurate, making the SAR process more efficient and less error-prone. The proposed portable command post also improves the way in which the information is presented in order to improve the support for decision-making.

3 Portable Command Post

This proposal is based on the authors' previous work [8], which currently helps firefighters to manage daily emergencies in urban areas, e.g. fires and car crashes. This legacy system implements a fully distributed peer-to-peer architecture that allows collaboration between firemen using two versions of the same system: a lightweight version (that runs on PDAs and smartphones) and a full functional version (running on laptops, desktops and tablet PCs). This application can be used standalone, but in such case the system features are reduced to a Geographical Information System that embeds location resource functionalities (through GPS) and support to do annotations.

The system also allows users to mark interest points and areas on the digital map (e.g. the emergency site, nearest hydrants and hospital), and also request information about the current response process conducted at a certain emergency (e.g. emergency type and location, involved trucks, and fire companies attending the event).

In case of large-scale disasters affecting urban areas, the probability to find the physical infrastructure just as it was before the extreme event is very low. In these situations, there is a need to count on geographical information of the affected area that allows rescuers to perform their work and collaborate with other teams. Having on-time and accurate information can help improve the effectiveness and speed of the SAR process; and therefore it may help reduce the number of victims.

3.1 Supporting Fieldwork with Mobile Technology

In order to help improve information availability, delivery and exchange in the field, this article proposes to adapt the legacy system to create a portable command post (PCP) that allows incident commanders and operations team leaders to make on-time and effective decisions during SAR processes. The collaborative system implementing the PCP inherits the fully distributed peer-to-peer architecture and also the autonomy provided by the legacy system.

The PCP also implements two versions of the system: a lightweight version (that runs on smartphones) and a full functional version (that runs on laptops/tablet PCs). Although the nodes running full PCP versions act as regular nodes of a mobile ad-hoc network, they embed particular functionality to support the work of the IC and OTL. The lightweight PCP supports the work of the response unit leaders, whom use the application to display information and mark interest points/areas on the digital map.

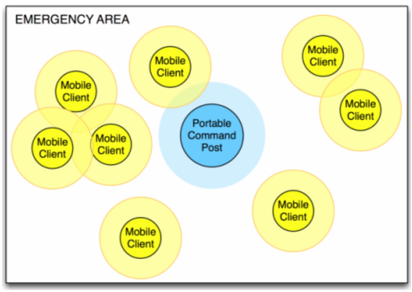


Fig. 2. PCP deployment in the field

Periodically, they have to synchronize their work, either to inform the command post of progress made, or to transmit information to other teams beginning their work shifts. Fig. 2 displays a possible distribution of the PCP and mobile users in the emergency site. Each device has an area of connectivity with other devices; intersection of these areas mean the devices can connect to each other.

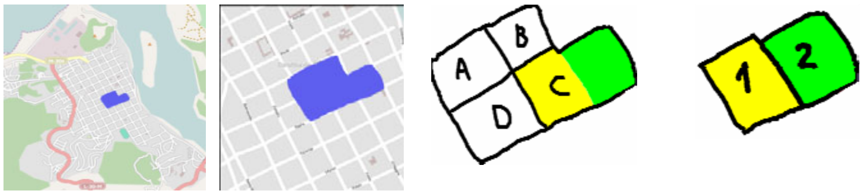
Since the needs of these actors are different, they require different devices for accessing data, e.g. laptops, smartphones. The ideal situation would be to have a truck or building dedicated to communications support. The operations team leader can input hand-written information and make decisions using a tablet PC, as he is more likely to be on the move. Unit leaders can access relevant information (e.g. the marked building plan of the area assigned to his unit) through mobile devices that support high mobility, such as smartphones.

This ideal situation may be unlikely, due to insufficient resources or the impossibility to reach an isolated location by land. In that case, the portable command post may consist of laptops, and the rest of the users may interact using smartphones. The portable command post is suitable to support work in these three scenarios, and it is therefore adaptable to diverse conditions. The PCP can provide several services that are not available in a whiteboard or paper-based information, e.g. timely access to information, and to exchange information with other interested parties.

3.2 Portable Command Post Services

The PCP system is based on maps, since they provide firemen with orientation in an unknown area. The map can help them find access to water sources, safe places in case of sudden threats, and location of resources. This base map may be a road map, satellite, hybrid map, or all three depending on the availability of data for the affected area. However, since the geography of a place may change after an emergency, the tool also provides firefighters with services to pinpoint the emergency area over the base map, as shown in Fig. 3a and Fig. 3b, as well as mark other points of interest (e.g., field hospitals to treat the wounded).

Firemen usually need a higher level of detail than the one provided by the georeferenced maps (e.g. distribution of apartments on a building floor). In these cases, hand-drawn maps can be created like the ones in Fig. 3c and Fig. 3d. The maps are organized in a hierarchical (zoom in – zoom out) collection, in which firefighters can explore an emergency area at various levels of detail. Relevant information added to the lower levels should also be visible at higher levels in aggregate form. For example, the incident commander has marked in green those areas where work is finished,



(a): General map (b): Detailed Area (c): Subdivided area (d): Subdivision of area C

Fig. 3. Example of the hierarchical display of maps within PCP

and in yellow areas with ongoing work (Fig. 3d). This information is also visible in the aggregate view presented in Fig. 3c.

Since SAR protocols may temporarily change because of unexpected circumstances, we propose to add semantic meaning to a few fixed options in the PCP system (e.g., status of an area, or number of victims found) to permit users to input any other type of information they consider is important for the situation (e.g., stable walls, or evacuation routes). Each map should allow users to enter information on it. Some information will be used in the aggregate view that is displayed to the IC.

Typically, the first operation assigned to rescuers is a reconnaissance stage. At this stage, some groups will be in the field identifying the type of area, e.g. residential, commercial, and industrial. The operations team leader will determine those areas that should be searched and those that should not. Then, the OTL will assign the task force to the area, splitting them in groups and assigning a sub-area to each group. Thus, each sub-area could be in one of three stages: under review, already reviewed, or to be reviewed. As a way to communicate the status of each area in the aggregated view, we propose a color for each possible status: yellow for under review, green for already reviewed, red for an area with a review still pending. Areas not needing review will be marked black.

3.3 Portable Command Post Architecture

Fig. 4 shows the architecture of the application, which is divided into three layers: interactive tools (UI), views handling and data management. From a UI perspective many of the provided features depend on the underlying device. For example, a Tablet PC would allow all the proposed features through a stylus, while a regular notebook might provide them with a mouse.

Services embedded in the second layer manage two information sources: regular geo-referenced maps and sketched maps. The former ones come from GIS and spatial databases, and they are stored as vectorial data. The latter ones are drawn by the user and they are completely self-contained.

Data management layer embeds services that allow the application to work in two modes: as a standalone or as a collaborative application.

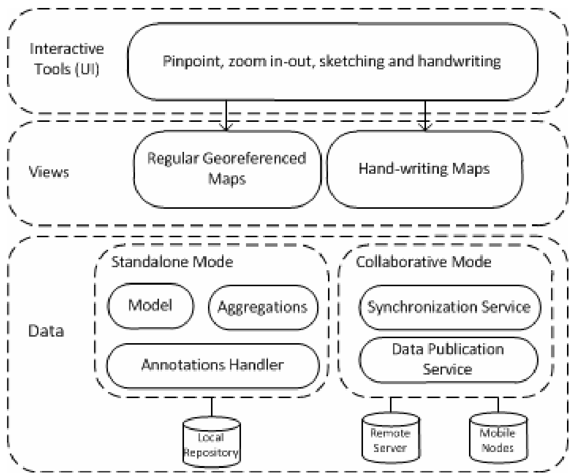


Fig. 4. PCP Software Architecture

In the first case the PCP works autonomously and uses local data and services to provide its functionality. The data repository can be a relational, XML or spatial database. When the PCP works in collaborative mode, it requires counting on a communication channel and computing devices to form a mobile ad hoc network. In such case, the application can work similar to a traditional collaborative peer-to-peer system, where the nodes can exchange and synchronize information among them, and also provide awareness of user's availability and activities. This operation mode can use several data sources; typically local and remote repositories. In both operation modes PCP provides services for handling basic annotations like search area status and also for other semantic annotations like the ones proposed by the INSARAG standard. PCP also allows handle the abstraction level of these annotations.

4 Related Work

The development of Information Technology Systems (ITS) to support the emergency response teamwork has many works published in the literature. For a long time, crisis management systems were based in the military model of Command and Control (C&C) [3]. For many authors, the need to change the theories of emergency management creating new paradigms is imperative to improve the flexibility of the C&C structures [1, 10]. Their aim is to make them more efficient, multi-disciplinary and multi-institutional, increasing the collaboration between C&C and the field responders and allowing sharing planning and resources to stabilize the crisis [12].

The decision process in C&C rooms has been covered in the literature and several studies indicate its complexity [1; 3]. The same situation is true for the first responder process in the field, which is very dependent on its contextual knowledge of the event as well as the responder's experience. These issues are also addressed in the literature [7; 11]. Recent disasters such as 9/11, the Asian tsunami, the Katrina hurricane and the Haiti earthquake demonstrated these limitations [5].

Similar applications have been developed after several large-scale disasters. Jiang et al. [6] developed a similar incident commander supporting tool, but contextualized in a developed country (USA). They support functionality for map sketching and resource management. However, since their work was done during 2003, the available mobile technology was not as widespread as nowadays, which limits the proposed system. Their work is also focused in daily firefighter emergencies instead of large-scale disasters. Wu et al. [15] present a system prototype (CIVIL) developed to support map-based decision-making. Their main goal is to allow a panel of three experts (*Public Worker, Environmental expert, and Mass Care expert)* to develop a plan during an emergency situation. Therefore, the application supports the exchange of information between the different roles and was not designed for chaotic and unplanned task forces. Finally, Wagner et al. [13] developed a system called COORDINATORs; it basically provides "…*decision support for first response teams and the incident commander by reasoning about mission structures, resource limitations, time considerations, and interactions between the missions of different teams*". However, the focus of this application is the use of multi-agent systems to actually perform these tasks. This approach proposes the automation of the planning and evaluation phases. It differs from the approach presented here which is to facilitate search and rescue activities through mobile technological support.

5 Conclusions and Future Work

This paper proposes a portable command post (PCP) to support the work of rescuers in large-scale emergencies. It was designed considering the SAR activities performed mainly by firefighters, whom usually are in charge of doing and coordinate this process. The proposal was based on the existing incident command system protocol, and provides several advantages over what is in use today. First, the proposed tool is low-cost and can be adapted to different scenarios depending on existing resources. Second, the tool can be used for various types of emergencies, since it supports work that is not routine or planned. Third, this system is superior to whiteboard and paper-based systems in that it may help firefighters quickly find relevant information as well as exchange it, and it may even provide mechanisms to share information with outside organizations, such as government agencies and the press. Fourth, it is easy to transport and deploy the PCP, providing firemen in the field with the autonomy (in term of supporting information and services) required for this type of activity. Finally, the PCP also allows firefighters to review the emergency and analyze it by providing technological support. This capability could help firefighters to review their protocols and learn for future emergencies.

The next steps in this initiative include the integration and tuning of the PCP components in order to have a complete and stable version of the product. Then, the PCP will be used (as a concept proof) to support SAR activities during the search and rescue national training program, which is performed by Chilean firefighters.

Acknowledgements

This work was partially supported by Fondecyt (Chile), grants Nº 11060467 and 1080352, and LACCIR grants Nº R0308LAC004 and Nº S1009LAC004.

References

[1] Beroggi, G.E.G.: Preface: A new paradigm in emergency management. Int. J. Emergency Management 1(1), 1–2 (2001)

[2] Coburn, A., Pomonis, A., Sakai, S., Spence, R.: Assessing human casualties caused by building collapse in earthquakes. In: Int. Conf. on the Impact of Natural Disasters, USA (1991)

[3] Dynes, R.: Community Emergency Planning: False Assumptions and Inappropriate Analogies. Int. J. of Mass Emergencies and Disasters 12(2), 141–158 (1994)

[4] Fiedrich, F., Gehbauer, F., Rickers, U.: Optimized resource allocation for emergency response after earthquake disasters. Safety Science 35, 41–57 (2000)

[5] Government Accountability Office: First Responders: Much Work Remains To Improve Communications Interoperability. Report to Congressional Requesters (2007)

[6] Jiang, X., Hong, J.I., Takayama, L.A., Landay, J.A.: Ubiquitous Computing for Firefighters: Field Studies and Prototypes of Large Displays for Incident Command. Computer Human Interaction 6(1), 679–686 (2004)

[7] Landgren, J.: Shared use of information technology in emergency response work: Results from a field experiment. In: Proc. of 2nd ISCRAM Conf., pp. 35–41 (2005)

[8] Monares, A., Ochoa, S.F., Pino, J.A., Herskovic, V., Rodriguez-Covili, J., Neyem, A.: Mobile Computing in Urban Emergency Situations: Improving the Support to Firefighters in the Field. Accepted for Expert Systems with Applications (2010)

[9] National Fire Protection Association: NFPA 1670: Standard on Operations and Training for Technical Search and Rescue Incidents (2009)

[10] Rubin, C.B.: What Hazards and Disasters are Likely in the 21st Century - or Sooner? Nat. Haz. Res. Work. Paper #99, Nat. Haz. Res. & Appl. Inf. Center, Univ. of Colorado (1998)

[11] Smalley, J.: Cognitive Factors in the analysis, design and assessment of command and control system. In: Handbook of cognitive task design, pp. 223–253. L. Erlbaum Assoc., Mahwah (2003)

[12] Trnka, J., Le Duc, M., Sivertun, A.: Inter-Organizational Issues in ICT, GIS and GSD – Mapping Swedish Emergency Management at the local and regional level. In: Proc. of 2nd ISCRAM Conf., pp. 75–82 (2005)

[13] Wagner, T., Phelps, J., Guralnik, V., VanRiper, R.: COORDINATORS: Coordination Managers for First Responders. In: Proc. of the 3rd Int. Joint Conference on Autonomous Agents and Multi-agent Systems, vol. 3, p. 1147. IEEE Press, Los Alamitos (2004)

[14] Webb, G.R.: Role Improvising During Crisis Situations. Int. Journal of Emergency Management 2(1-2), 47–61 (2004)

[15] Wu, A., Zhang, X., Convertino, G., Carroll, J.: CIVIL: Support Geo-collaboration with Information Visualization. In: Proc. of the ACM 2009 International Conference on Supporting Group Work, pp. 273–276. ACM Press, New York (2009)

Digital Workbook: A Mobile Learning Environment to Support Collaborative Examinations

Iván D. Claros, César[1] A. Collazos[1], Luis A. Guerrero[2], and Sergio F. Ochoa[2]

[1] IDIS Research Group, Universidad del Cauca, Colombia
{iclaros,ccollazo}@unicauca.edu.co
[2] Computer Science Department, Universidad de Chile, Chile
{luguerre,sochoa}@dcc.uchile.cl

Abstract. It is already known that some collaborative assessment processes can provide students not only the possibility to show how well they have understood knowledge content, but also the opportunity to enhance and increase such knowledge. Such learning activities also allow instructors to diagnose the quality of the instructional process. However there is evidence that this type of activities involve an important effort. This article presents a mobile learning environment named digital workbook, which was designed to support students and instructors when applying a Collaborative Examination Technique. The tool was evaluated in a Computer Science undergraduate course, and the obtained results are highly encouraging.

Keywords: collaborative assessment process, mobile computing.

1 Introduction

From a collaboration point of view, sharing information is one of the most important activities in teaching-learning scenarios. Most of the time students share information without using exchange structures or adequate supporting tools. In that sense, collaborative learning techniques provide strategies to support the student learning process by doing some activities more than simply watching and listening. However, collaborative learning activities are still difficult to design and apply inside the classroom; especially if those activities include new technological elements. In that way, we are interested in the development of technological solutions to be integrated in an educative setting supporting social interactions among different group members. The solutions will allow students to achieve the pedagogical goals in an effective manner, by acquiring abilities like problem solving, critical thinking, meta-cognitive thinking (learning to learn), and information retention, as it was proposed by Johnson & Johnson [1]. Some years ago the authors proposed a Collaborative Evaluation Technique (CET) [2], which involves three stages composing a collaborative examination process: a *pre-test*, a *test* and a *post-test*. The pre-test main goal is to help students to understand and assimilate, in an early stage, the knowledge that is going to be considered in the test phase. The test phase involves the individual resolution of an exam, similar to a traditional examination process, which is face-to-face and hand-written. Finally, the post-test intends to reach two main goals: (1) that students find the right

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 345–352, 2010.

answers to the test questions/items through a discussion among them, and (2) that students identify the right points and the mistakes in their answers. At the end, in an individual manner, each student grades his/her own exam and give it back to the instructor. For the correction of their own exams, the students use the solution outline constructed by the group during the discussion session. Finally, students grade their own exam as a whole, in accordance with the individual grades assigned to each answer of the test. The students' answers are then examined and graded by the instructor or the teaching assistant. Students that find the right answers during the post-test receive a bonus in their exam final score.

On this context, the authors propose to use a collaborative software tool to support some activities involved in the CET. It will help to reduce the effort involved in performing the CET process. Such tool has been named the *Digital Workbook*. This solution keeps the traditional paper notebook metaphor in order to allow students to express their ideas and annotations in a natural way. The digital workbook also offers to the users several interaction mechanisms among group members, which can be used to support collaborative activities. In that way, students can build in a collaborative manner, their responses to the problems proposed in the pre-test and post-test stages by using services of the digital workbook, e.g. the creation of work sheets they can share, annotate and publish. Mechanisms such as annotations or publications convey communication elements in synchronous and asynchronous way.

A particular module of this tool, which was named CETProfessor, was implemented to support the instructor during the activity preparation and monitoring processes. This module includes several services, e.g.: (1) a list of questions that can be reused during tests, (2) services to configure access grants for the accounts of users participating in the activity, and (3) various communication services that are useful to publish the questions and responses involved in the activity. Some of these services include: a sociogram, participation records, and registered users list and on-line.

The rest of the paper is organized as follows: next section presents background information related to the development of the proposed collaborative tool. In section 3 describes the architecture of the digital workbook and its main components. Section 4 presents the experimentation process performed to evaluate this tool, and also the preliminary results. Finally, section 5 presents the conclusions and further work.

2 Background

It is important to mention that CET already counts on a collaborative system that automates several of its activities. The tool, named MOCET [6] runs on tablet PCs and PDAs, and utilizes a stylus to keep the metaphor of the paper notebook. This system allows each student to share information just with the instructor. For example, a student can retrieve or submit his/her exam using this service. Data persistency and synchronization are managed by a platform named SOMU (Service-Oriented Mobile Units) [3].

Shared objects, e.g. images, text documents or exams, can be accessed on-demand by members of a mobile work session. These persons communicate among them using a mobile ad-hoc network [7], which forms a distributed system. During pre-test, each student can share his/her resources with other students; however during the test and post-test students can interact just with the instructor. In every stage these persons can do annotations on their private or public resources. If the stage allows them, two or more students can exchange or combine their annotations in order to get a consolidated view of their comments. MOCET separates the object from the annotations done over them. It simplifies the implementation of several services, such as the annotation exchange and synchronization, and also the replication of shared objects.

This tool uses an application programming interface provided by Microsoft One-Note [1], to access services for hand-writing on tablet PCs and handheld devices. Although these services are useful, every device running MOCET must include the product OneNote as a supporting service. It limits the possibility to reuse the tool in more than one scenario. In addition, MOCET have shown some usability limitations mainly in the module that allow instructor to prepare and monitor a CET activity. These reasons motivated to the authors to develop a completely new system that solve the already mentioned MOCET limitations. The result was the Digital Workbook.

3 Digital Workbook

The digital workbook was also designed to be used on a Tablet PC; however it does not require any plug-in or software product to provide the services to students and instructors. The tool implements a public and a private space to manage the resources in each computing device. It also implements two communication services: *ICET* and *IPageShare*. ICET represents the design contract between CETProfesor (module used by the instructor) and digital workbook (module used by the students) in order to obtain information about proposed questions and group information. IPageShare allows sharing pages between digital workbooks.

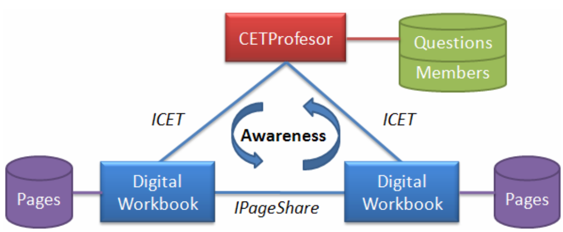


Fig. 1. General Diagram of the architecture in the communication level

Such service is executed in an autonomous way, and allows students to perform distributed presentation or share information (e.g. pages of the workbook) as part of a private session. Fig. 1 depicts a general diagram that allows understand how the workbook works.

The module CETProfesor manages a database of questions and users. This information and also the exams can be shared with the students using the ICET communication service. Each digital workbook manages its own pages (or working sheets) that could be published or shared according to the participants needs.

3.1 Digital User Interface

Figure 2 presents a screenshot of the digital workbook main user interface. The interface involves five components or interface areas:

(A) A list of questions given by CETProfessor. Each question has its own responses, describing the communication process.

(B) A list of users that are part of the work group. It uses green color for the connected users and white for the disconnected ones.

(C) A working area. There it is possible to add lines and some graphical components like geometric figures, images and text. Notes or annotations are a special type of components allowing students to make some comments about the content of certain task. Based on these annotations a sociogram is displayed about the interaction of the students.

(D) A menu bar with different components could be integrated and also the communication services that allow sharing page, working sheets or students' options that can be sent to the instructor.

(E) An information line that allows accessing to the awareness management tool, which was proposed in [5]. In order to obtain awareness independence, the digital workbook and CETProfessor modules implement their own awareness mechanisms. This aspect is going to be evaluated in future experimentations.

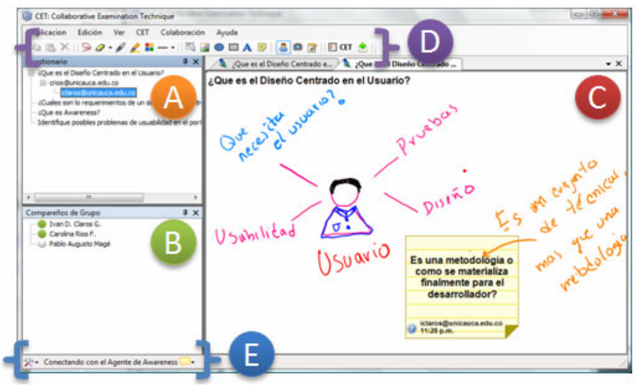


Fig. 2. General View of the Digital Workbook (in Spanish)

3.2 CETProfessor

This module is used just by instructors and it allows: 1) managing the list of questions and users; 2) executing communication services that allow users access shared information, and publish students' contributions through pages or working sheets; and 3) facilitating tutor mechanisms to monitor a CET activity.

Fig.3 presents a screenshot of the CETProfessor user interface. In N) we can observe an example of how different users have made some comments to the working sheets of the teammates. In B) it is possible to observe the list of users registered per activity, their connection states and the different responses published in every working sheet. If the user clicks a question in L), the module displays the responses. In M) we can observe the buttons to access the edition window, the questions and the commands to launch and stop the ICET service.

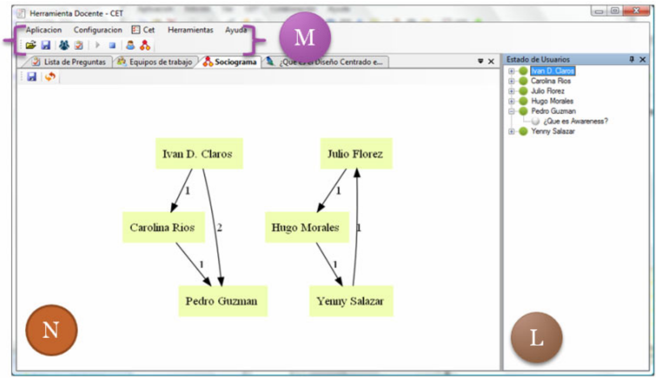


Fig. 3. General view of CETProfessor

The CETProfessor module does not include functionality to edit the responses in the students working sheets. If the instructor wants intervene, s/he must send a message to the students. In future versions we hope to include mechanisms that allow tutor to intervene in a direct manner over the responses working sheets. Next we describe the experimentation process performed to evaluate this tool.

4 Experimentation

During the experimentation phase we worked with students of a Human-Computer Interaction course in the Computer Science Program at University A (Colombia). The topic evaluated was *user interfaces in collaborative systems*; particularly: awareness mechanisms in CSCW environments, usability and tangible interfaces. A set of questions were formulated to support the learning of these topics. We worked with 21 students divided into working groups composed of three randomly selected students. Due to lack of Tablet PCs only two of these groups worked with the digital workbook (ULD), and the other ones (UT) without digital workbooks. In this experiment three tabletPC were used, the ULD groups had to interchange the tabletPCs during the activities. In one of these tabletPC, both Digital Workbook and CETProfesor were executed, in the others, only the Digital Workbook was executed.

4.1 Methodology

The process began with an introductory talk about the topic, where concepts like collaborative work and collaborative learning were explained. The working groups were organized and every student was responsible to acquire the knowledge related to a certain topic (awareness mechanisms in CSCW environments, usability and tangible interfaces). Each student studied in an individual manner the assigned topic and s/he proposed a brief summary that is going to be worked in the activity. Next section explains how do the different activities that conform CET were performed during the experimentation process.

4.2 The Process Step-by-Step

During the pre-test stage the following activities were performed:

Definition of a coordinator (5 min): Each group selected a coordinator, who has the responsibility to promote and solve discussions, and also to coordinate the activity in a way that permit to achieve the group goal in an efficient manner.

Resolution of questions in an individual way (15 min): Based on the preliminary research, every student gave an answer to questions of every topic. UT groups did the work using paper sheets, while ULD groups used the Digital Workbook.

Experts Meeting (15 min): Students from different working groups, that have the same question to answer, were organized by pairs in order to discuss every question and response (Fig. 4). Changes or updates to their responses are saved as a new version of the answer.

Group meeting (15 min): Once the experts meeting has finished, working groups meet and discuss the responses to the assigned questions. Any change is saved as another version of the response. All documents are sent to the instructor; the ULD groups do not need to do that, because the tool provides such service automatically.

Fig. 4. Expert meeting (ULD left, UT center, working groups right)

During the test stage we applied a traditional test, where one of pre-test question was included. We did not use any technological application to support this stage. During the post-test the working groups were organized in order to analyze their responses, including some annotations and notes if they needed. The post-test considered an additional bonus in the score that was added to the average between pre-test and test. The activity finished when students graded the exam as a whole.

After that, students participated in a discussion about the topics worked and they talked about the experience of this collaborative practice. At the end, they did an exercise that consisted in an essay about the formulation of awareness mechanisms in tangible interfaces, in order analyze if students really understood the topics. Although this phase is not considered as a part of CET, we think adequate to perform this activity to analyze if there was an appropriation of the concepts worked in the CET activity. Finally, we did an interview to the students and they also fill a survey about the experience.

4.3 Obtained Results

Table 1 depicts the results obtained during the experience. Groups A, B, C, D, and E were working without digital workbook. However groups F and G were working with the digital workbook. At the end of each phase the students presented a document with responses to the proposed questions. According to these results we can observe all groups improved their scores after pre-test. This could imply the CET is a mechanism that permits to support teaching-learning processes in an independent way, if it is mediated by a computer. Although all groups improved their results comparing grades from pre-test to test, those who were working with Tablet PC got a better performance (their grades improved from 3,5 to 4,37 and from 3,8 to 4,4 into a scale from 1 to 5). More important is the level of improvement obtained by the groups using the digital workbook, which was between 0,6 and 0,9 points.

Table 1. Experimental Results

	CET				Final Document	Final Grade
	Pre-test	Test	Bond Post-test	Final CET		
A – UT	4.00	4.10	+0.10	4.20	4.1	4.15
B – UT	4.27	4.43	+0.05	4.48	4.4	4.44
C – UT	4.23	4.23	+0.10	4.33	4.6	4.46
D – UT	3.90	4.07	+0.05	4.12	4.3	4.21
E – UT	4.3	4.33	+0.15	4.48	4.5	4.49
F – ULD	3.50	4.37	+0.10	4.47	4.5	4.48
G – ULD	3.80	4.40	+0.10	4.50	4.6	4.55

It is clear that in CSCL scenarios technology is not the only variable that could influence an improvement of the students' performance. It is necessary to structure the collaborative activities convey a real collaboration. In our experience CET has shown to be a good technique that helps improve teaching-learning process and the use of supporting technology helps to perform the activity in a better way. The results we obtained were based on quantitative date of the results of the test, for future experimentations we are going to include qualitative aspects like some survey about the experience of the technique and software tool. During these experimentations we have not included the comments of the teacher about the experience. We hope to include experience of the teachers using the Model and software tool we have developed for further work.

5 Conclusions and Further Work

This article presented a mobile collaborative application, named digital workbook, which supports activities involved in the collaborative evaluation technique (CET) [2]. The metaphor embedded in the digital workbook seems to be appropriate to perform the CET activities. This result can be observed through positive impact the tool has had not only to support the experience but also the learning goals. However there is a need to establish design guidelines to mitigate the usability problems that still have the technologies for hand-writing on Tablet PCs.

The digital workbook can also be used to support other collaborative learning activities, because it is possible to use its services in an autonomous way and independently of any other component. The monitoring capability of the tool can also be used to support other collaboration activities. Although the obtained results are still preliminary, the use of the tool has shown to be consistently attractive and useful for students and instructors. The next steps in this work involve more experimentation instances that allow understand the real impact of this proposal including the evaluation of the CETProfessor in order to determine how awareness aspects we have includes could support instructor in the teaching-learning process.

Acknowledgements

This work has been supported by the LACCIR Grant R0308LAC001, and Proyecto Enlace VID 2010 (University of Chile), Grant ENL 10/10.

References

1. Johnson, D., Johnson, R.: Circles of Learning. Association for Supervision and Curriculum Development, Washington (1984)
2. Ochoa, S., Guerrero, L., Pino, J., Collazos, C., Fuller, D.: Improving Learning by Collaborative Testing. Student-Centered Learning Journal 1(3), 123–135 (2003)
3. Neyem, A., Ochoa, S.F., Pino, J.: Integrating Service-Oriented Mobile Units to Support Collaboration in Ad-hoc Scenarios. J. of Univ. Comp. Science 14(1), 88–122 (2008)
4. OneNote Developers Center, http://msdn.microsoft.com/en-us/office/aa905452.aspx (last access: March 2010)
5. Claros, I., Collazos, C., Lara, A., Ochoa, S., Guerrero, L.: Arquitectura Multi-agente para la Gestión de Información de Awareness en Entornos CSCL. In: XXI Encuentro Chileno de Computación (ECC 2009), Santiago, Chile, pp. 240–246 (November 2009)
6. Ochoa, S., Collazos, C., Bravo, G., Neyem, A., Ormeño, E., Guerrero, L.: Una técnica de evaluación colaborativa soportada por computador para escenarios de educación superior. In: IX Congreso Internacional Interacción, pp. 71–80. Albacete, España (June 2008)
7. Tschudin, C., Lundgren, H., Nordström, E.: Embedding MANETs in the Real World. In: Conti, M., Giordano, S., Gregori, E., Olariu, S. (eds.) PWC 2003. LNCS, vol. 2775, pp. 578–589. Springer, Heidelberg (2003)

Evaluating a Prototype for Geo-referenced Collaborative Psychotherapy with Mobile Devices

Luís Carriço[1], Gustavo Zurita[2], Marco de Sá[1], Nelson Baloian[3], João Faria[4], and Isabel Sá[4]

[1] Department of Informatics, University of Lisbon, Portugal
{lmc,marcosa}@di.fc.ul.pt
[2] Department of Management Control and Information Systems, Universidad de Chile, Chile
gzurita@fen.uchile.cl
[3] Department of Computer Science, Universidad de Chile
nbaloian@gmail.com
[4] Faculty of Psychology, University of Lisbon, Portugal
jnunofaria@gmail.com, mirealsa@fp.ul.pt

Abstract. Social competency training, as part of psychotherapy, for children and teenagers, requires them to engage on outdoor activities in which they have to complete tasks such as talking to someone or visiting a specific place. Currently, the inability for therapists to monitor their patients, to promote collaborative efforts and to reinforce positive attitudes is a major issue that affects both the therapy process and its results. In this paper we present an evaluation experience of a mobile prototype for a geo-referenced collaborative system that supports in-situ group therapy. The system aims to provide means for therapists to monitor their patients, their locations and achievements and includes communication mechanisms that facilitate cooperation between patients and therapist. We describe the concept behind the project, our initial low-fidelity prototypes and the experiments that were undertaken to validate them. Initial results are discussed and future work is defined.

Keywords: Prototyping, Collaborative Psychotherapy, Geo-referenced systems.

1 Introduction

Social competencies and skills training (SCST) is a type of psychotherapy procedure in which patients are generally required to perform a set of activities individually (e.g., ask for a coffee at a bar; compliment someone on the street) or as a group (e.g., talk to each other; work as a team) [1, 2], working both with their peers and parents [3]. This kind of therapy is rather frequent and applied to people who feel uncomfortable when faced with traditionally common social situations, in some cases, to the point of impairing their daily lives [4].This type of issue affects all ages, genders and cultures, being, however, most noticeable, but also more likely to be easily treatable, at younger ages. More severe cases, such as Autism and Asperger's Syndrome, are also common examples of disorders with similar symptoms [1].

G. Kolfschoten, T. Herrmann, and S. Lukosch (Eds.): CRIWG 2010, LNCS 6257, pp. 353–362, 2010.

Currently, this process is generally synchronous. During a consultation, therapists prescribe a set of field tasks for their patients, that are perform at home (with their parents) or at school (with their peers). Tasks, although accomplished individually, are preferably undertaken within the context of a group, so that motivation is kept even when the therapist's presence is inaccessible. SCST tasks are usually defined as a set of consecutive activities that should be accomplished by following a predetermined route. At each specific place a task or task repetition is proposed. Every time the patient is able to complete it, he/she annotates his/her thoughts and registers a set of values for feelings such as shyness, stress, anxiety, and difficulty.

These annotations are afterwards shared with their therapist during a following consultation, in which the experience is discussed and feelings are explained. Paper questionnaires and annotations are used to collect this information while on the field and to support the exchange of information with the therapist. Sometimes these consultations are held in group, with other patients having similar behaviors, discussing the achievements of each other, congratulating them on the successful accomplishments and encouraging them to surpass the felt difficulties.

Given the used medium, this process is affected by several drawbacks. Firstly, as there is only intermittent therapist supervision, patients are frequently detached from the therapy process and provide, more often than desired, fake comments and experiences. Additionally, cooperation between the various actors during the execution of the tasks is hindered by the usual lack of communication and social skills: motivation is extremely low while away from the therapist's office, whenever parental supervision is unavailable and, especially, when cooperation between patients is not possible or sometimes avoided.

In this paper we present an evaluation experience that aimed at assessing the feasibility, adequateness and acceptance of a geo-referenced collaborative mobile tool to support SCST. We discuss the design process and present the initial prototypes that emerged, also stressing the validation process that was conducted and the initial results that were encountered. Future directions and improvements are also detailed.

2 Related Work

Recent studies have shown that the use of technology, applied to psychotherapy, can improve clinicians' work [5, 6]. In particular, mobile device usage has been gaining popularity and several tools have emerged recently [7, 8]. Most of them use handheld devices such as PDAs or Tablet-PCs for self-control or relaxation procedures [9].Work reporting on mobile devices used to support healthcare range from theoretical approaches, establishing design patterns based on the needs of the person being treated (mental health [10]), to the development and testing of prototypes of monitoring systems. Among the various applications developed we can find system to: monitor the heath condition of people with chronic asthma [11, 12]; monitor and recommend behaviors during physical activity [13]; monitor and guide people with mental disorders [14]; and aid on the recovery of addiction problems [14]. Other works use sensors to collect data about the physical and mental condition of people with severe chronic diseases like Alzheimer [15] and bipolar mental disorders [16]. The prototype described in [17] addresses the treatment of depression though a cognitive behavior therapy approach based on a proposed ontology. In recurs to a set of tools supporting communication through chats, audio conversations and messaging services that allows a group of distributed located specialists to make a collaborative diagnostic.

Still, cooperation and support for group activities is such settings is still very scarce. In other domains, there are some systems addressing similar issues usually in a generic manner. Related projects have addressed the need to support communication between several hand-held and even non-mobile devices [18, 19]. Managing information that is shared between several devices is the usual goal of a system supporting cooperative work in a spatially distributed environment. However, the specificity of the group therapy scenario requires a special care. In such domains, significant emphasis should be devoted to the easiness of use, the adequacy to the clinical procedures, and the specificity of the used artifacts [20]. In addition, and most importantly, for SCST, it is essential to allow therapists to maintain awareness of their patients' locations and activities, not only in order to supervise the process but also to be able to intervene whenever necessary.

3 Background and Design Process

This project emerged as the conjunction of two different systems with distinct purposes. The first one aims at supporting geo-collaborative work and knowledge creation, supporting geo-referenced annotations (in various formats) that can be monitored and shared in real time through mobile devices [21]. The goals of the second project are to support cognitive behavioral therapy practices by providing means for therapists and patients to complete their activities (e.g., thought and feeling registration) ubiquitously, also using mobile devices [22].

During the course of some trials with the psychotherapy tool in group settings [20], in which several therapists were involved, their need for means to manage out-of-the-office sessions and monitor their patients while completing activities outdoors was noticed and frequently mentioned. Accordingly, the idea to combine these existing systems, which allow therapists to share a set of data collection artifacts with their patients, into a system that could provide awareness on the location of various users and communicate with them in real time, was quickly and enthusiastically accepted by all the involved therapists.

3.1 System's Goals

Based on the needs that SCST therapists claimed, the system aims at addressing several issues and enhancing the process by: (1) providing an integrated collaborative platform for both therapists and patients; (2) supporting real time, monitoring of patients while accomplishing their tasks; (3) offering communication mechanisms between all participants; (4) including multimodal experience records and (5) promoting collaborative motivation between patients and therapist.

This can be achieved by merging a set of geo-referenced collaborative features, inherited from the first project with enhanced therapy artifacts that support in-situ multimodal data collection and reviewing mechanisms, offered by the second. Overall, this results in a system that, on the one hand, allows users (patients) to access and create data (e.g., video/audio/text annotations) that is geo-referenced, using a digital map, directly on their mobile devices (e.g., smartphone, PDA) and; on the other hand, a power-user (the therapist) to manage all the information that is generated, access all the users' locations, manage, and distribute it through the various participants.

3.2 System's Features

Geo-Referenced Knowledge Creation. On a general perspective, the geo-referenced knowledge creation system provides mechanisms for users to create geo-referenced information, by selecting a location on a map and creating structured annotations, either textual or sketch-based [21]. The system also includes an underlying communication mechanism that creates an ad-hoc network between the various mobile devices. With this mechanism, generated information can be distributed between several users, who can browse through the existing information by its location, directly on a map.

Multi-modal Psychotherapy Data Collection. The psychotherapy framework includes a set of functionalities that allow therapists to create patient-specific artifacts that support data collection using various modalities (e.g., voice, video, images, text) [22]. Each artifact can be used on a mobile device. Results, together with usage logs, are locally stored in XML files and can be synchronized to other devices.

3.3 Design Approach

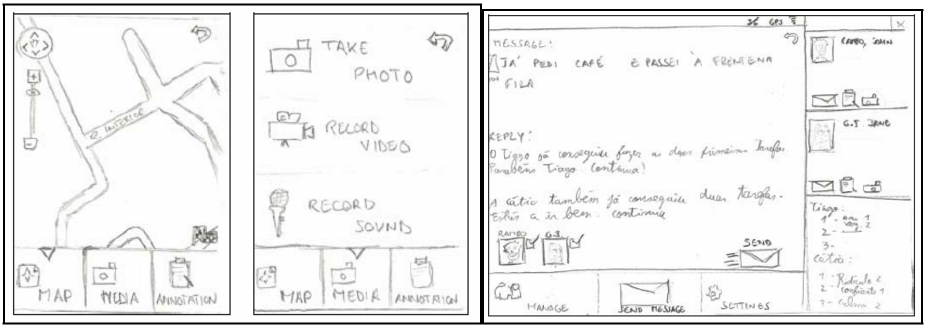
The design of the system is following a User Centred approach [23], in which the various stages have been closely accompanied by end-user and their input has been provided since the beginning. As a bootstrap, several meetings were held, with therapists and members from the two projects, spawning the system that is being designed. Requirements were set based on the existing features, the therapists' requests and description of patients' needs. Some scenarios were also defined and simulations took place to identify additional issues and requirements for the envisioned activities.

These requirements were thoroughly discussed with practicing therapists and psychotherapy researchers and, once consensus was achieved, the design concept evolved into the prototyping stage.

3.4 Prototypes

To test the concept, two different low-fidelity prototypes were created. One for the therapist application, designed for a Tablet-PC and one for the patients' tool, directed to smart-phones. The creation process of these prototypes followed an advanced methodology, specifically conceived to support mobile interaction design [23].

Figure 1a shows two sketches for the patient application. On the left, the main screen depicts the main user interface, in which patients can keep track of their tasks and location of their route, as defined by the therapist. Below, there's a quick navigation menu that allows patients to quickly select a media capturing option or create an annotation. On the right, the second screen displays the various media options. Patients can quickly take a photo, record a video or record an audio annotation or sound and immediately send it to the therapist. If allowed by the therapist, the other patients on that session will receive the message as well. Broadcasting should always be confirmed by the therapist depending on his/her assessment on whether it will contribute or not to the encouragement of the other patients.



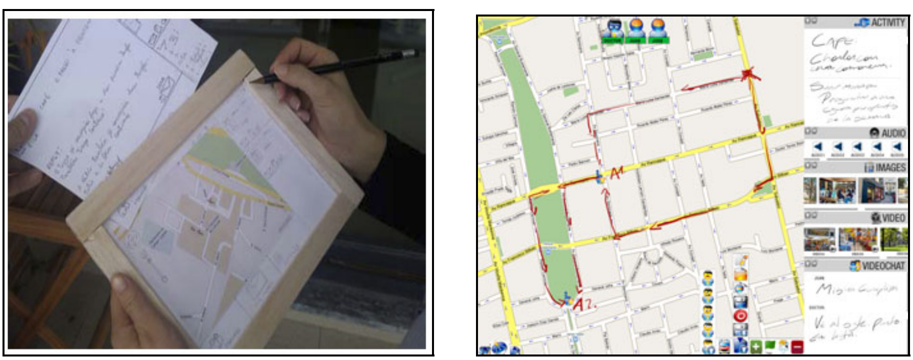
a) Patients' PDA tool

b) Therapist's Tablet-PC tool

Fig. 1. Low-fidelity prototypes

Figures 1b shows the therapist's counterpart. On the main canvas, the figure presents the messaging textbox, in which messages can be read, composed, and recipients can be added or removed by (de)selecting their photos (left-bottom corner above the menu). The lower menu allows the therapist to manage the patients, to send global messages, directly to all the patients, and defined the general settings of the session. On the right side of the canvas, individual information on each participant is always available, including completed tasks, feelings and their intensity, and current progress. Additionally, shortcuts for each patient's data (e.g., annotations, photos) are available, together with individual messaging options.

In figure 2a it is visible the main canvas when the manage patients' option is selected. Here, the therapist is able to monitor patient's locations and movement as well as set new challenges and tasks that are automatically updated on the patient's route. Figure 2b depicts the therapist user interface's main screen, displaying a map in which therapists can manage the various locations of the pre-defined challenges, as well as their patients' positions and movement.



a) Low-fi in out-of-the-lab experiment

b) Screenshot of the developed application

Fig. 2. Therapist's Tablet-PC tool

4 Experiment

To validate the two prototypes, the therapy tool and the patient tool, an initial Wizard-of-Oz experiment was conducted. The low-fidelity prototypes were used by one therapist and two simulated patients, at a university campus. Simulated patients were young students, one male and one female, selected randomly at the university campus. None had undergone therapy before. The therapist had extensive experience in SCST and was not involved in any of the preceding projects.

The sketches were used with rigid frames, emulating real devices and offering a good support for annotations and writing on the sketches if needed. Pencils were offered for the participants to annotate (whenever they wanted to create textual annotations) and mark any required information directly on the sketches.

One Wizard was assigned for each participant, shadowing the user and acting both as the application and as the communication mechanism. To support communication between the various Wizards, cell-phones were utilized and calls were made sharing each participant's location. The therapist's Wizard would then mark on the application's map the current location of each simulated patient. Annotations were exchanged through SMS and MMS if including images, videos or sound.

5 Results and Discussion

Results from the experience were very positive. Regarding the feasibility and the effectiveness of the entire system, the envisioned features were highly appreciated by the therapist. The ability to easily access patients' location, including the time spent at specific points and tasks were considered extremely useful. Besides allowing the therapist to control the patient's route, for safety reasons (given that most patients are children), it also allowed the therapist to control whether the time spent at one spot was sufficient to complete a task or even just to control if the patient was actually following the route. According to the participating therapist, this provided the opportunity to gather real, effective, and reliable information and provided much less space for error. Additionally, this also endows therapists with the ability to send motivating messages at crucial moments (just before the patient reaches the location for one given task), which is a decisive moment in which patients usually disengage from the process and feel more stressed. The communication options were also very much appreciated by the therapists, especially the possibility of sharing or concealing achievements from one patient with the others. Nevertheless, some suggestions and improvements also emerged. In particular, the inclusion of an S.O.S. button on the patient's tool and the possibility of allowing patients to freely exchange messages between each other were also requested.

Participants that acted as patients also provided valuable input. Both participants felt that the tool was very easy to use and allowed them to quickly understand their tasks. Following the pre-defined routes on the mobile device, and gathering data in different formats was also especially pleasing. Once again, some suggestions were made. One participant suggested the inclusion of a list of feelings that could be selected and quantified, rather than typing in the feeling. In addition, the second participant mentioned that if tasks could be subdivided, they would see more tasks completed, increasing motivation, and would help them while setting goals and priorities. The therapist concurred with this suggestion.

During the experience, it was also noticeable that voice annotations and pictures were frequently used and clearly preferred to textual annotations/descriptions of achievements, as it allowed them to quickly provide, and receive, feedback from the therapist, increasing confidence throughout the process. The use of mobile devices was also deemed very adequate as it allowed them to pass unnoticed while collecting data and sharing information with the therapist. The therapist also agreed and considered the entire infrastructure very appropriate for these situations, as most patients already carry their own devices.

Overall, the system was deemed very helpful and easy to use. From the therapist's perspective, it would be highly beneficial to apply on children and teenagers SCST and, with the addition of communication between patients, would allow groups to greatly increase cooperation and, consequentially, motivation, engagement and results during therapy.

6 The Application

Based on the results of the experiment we completed and improved the design of the application which is currently under development, using Tablet-Pc's as platform for the implementation. The whole system consists of two modules: one for the patient and the other for the therapist. The current therapist's module interface is shown in the figure 2b. Although the therapist's and the patient's interfaces look very similar their functionality differs from each other. This therapist's application module runs in two different modes: one allows the therapist to define tasks the patients have to follow and the other allows her to monitor the patients while they are performing the task. A task definition may contain handwritten text and sketches (for the instructions), photographs, videos and sound that might clarify the task's nature. Each task is geo-referenced, which means that is "anchored" to a certain geographic place where it should be carried on by the patient using the map displayed in the workspace of the therapist's module interface.

Additional to tasks, a therapist can define routes that the patient may have to follow, also using the map on the workspace. Various, already defined tasks may be assigned to a route and a certain order in which the tasks have to be performed can be defined. In the Figure 2b shows the interface during the definition of a route. As seen, routes are defined by free handwriting the route over the map. Tasks are anchored to geographical places by marking the point on the map with a double click and then entering a label for that task also by free handwriting. In the figure we can see two Tasks geo-located and labeled as A1 and A2.

At the right-hand side we see the elements of the task currently being created or edited. On the top we see the area where the activity is described, also using free handwriting, below that area is the one for the audio files, followed by the area for the image and the video files. The area on the bottom of the right border is used to exchange hand written messages with a patient and is not associated to a certain task. This area is activated when the system is being used in the mode for monitoring the patients. In this mode the therapist can see which patients are currently using the system for performing tasks. An icon identifying the patient appears in the upper area of the workspace (see figure 2b the three icons on the top). By selecting one of them the therapist can see where the patient is located, which task is performing now and can exchange messages with him.

The patient's module allows him to explore a task definition and to follow it. While following the task the patient can see the task elements in the same way as they are displayed on the therapist's interface. Additionally, the patient can input freehand annotations, voice messages, images taken with the camera of the mobile device. These elements will help the patient and the therapist to analyze the performance together, reflecting about the things that were good done and which could be improved. The patient has also the possibility of expressing his current mood by choosing an icon from a menu. If the patient chooses to use this icon, it will be displayed beside the patient's icon on the interfaces of the therapist.

Currently all the necessary communication between patients and therapist is realized with a client-server architecture. The patients' mobile devices access the server using a wide area wireless network. We are evaluating the convenience of using also Bluetooth communication protocol for situations where a wireless wide area network is not available.

7 Conclusions and Future Work

In this paper we presented the validation study for the concept of a geo-referenced collaborative in-situ group therapy system. The design aims at overcoming current limitations of traditional social competencies and skills training, a process that is frequently required for children and young teenagers, suffering from extreme shyness, Asperger's Syndrome, Autism, and related disorders.

Our design concept is based on two existing systems and merges functionalities from both in order to allow therapists to manage patients that circulate through different locations but collaborate in order to achieve specific goals and motivate each other. To test its feasibility and the overall design, a set of low-prototypes were created and a Wizard-of-Oz experience took place. A practicing therapist and simulated patients participated on a set of sessions that provided insight on the various facets and dimensions that compose the system. Overall, results were significantly positive as the participating therapist greatly appreciated the concept and strongly encouraged its continuation into a future development stage. In addition, the remaining participants, acting as patients, were also very pleased with the experience and clearly stated that they would use such a system for a wide variety of purposes.

Based on these results, further experiences, with a larger group of participants and additional therapists, using new and improved prototypes, will be conducted and new features will be tested. As the process evolves, a working software prototype is already being developed and will be tested in real world scenarios, with children currently undergoing therapy for social competencies training.

Acknowledgements

This work was funded by FCT, through project PTDC/EIA-EIA/103676/2008 (InSIThe) and the Multiannual Funding Programme, by Fondecyt, through project 1085010, and by Latin American Collaborative Research Federation (LACCIR), through project R1209LAC002.

References

1. White, S.W., Keonig, K., Scahill, L.: Social skills development in children with autism spectrum disorders: a review of the intervention research. J. Autism Dev. Disorders 37(10), 1858–1868 (2007)
2. Chan, D., Ramey, S.L., Ramey, C.T., Schmitt, N.: Modeling intraindividual changes in children's social skills at home and at school: a multivariate latent growth approach to understanding between-settings differences in children's social skill development. Multivariate Behavioral Research 35(3), 365–396 (2000)
3. Bass, J.D., Mulick, J.A.: Social play skill enhancement of children with autism using peers and siblings as therapists. Psychology in the Schools 44(7), 727–735 (2007)
4. Marchant, M.R., Solano, B.R., Fisher, A.K., Caldarella, P., Young, K.R., Renshaw, T.L.: Modifying socially withdrawn behavior: a playground intervention for students with internalizing behaviors. Psychology in the Schools 44(8), 779–794 (2007)
5. Garrard, C.S.: Human-computer interactions: can computers improve the way doctors work? Schweitz Med. Wochenschr 130(42), 1557–1563 (2000)
6. Proudfoot, J.G.: Computer-based treatment for anxiety and depression: is it feasible? Is it effective? Neuroscience & Biobehavioral Reviews 28(3), 353–363 (2004)
7. Grasso, M.A.: Clinical Applications of Hand Held Computing. In: 17th IEEE Symposium on Computer Based Medical Systems, pp. 141–146. IEEE Press, Los Alamitos (2004)
8. Norton, M., Wonderlich, S.A., Myers, T., Mitchell, J.E., Crosby, R.D.: The use of palmtop computers in the treatment of bulimia nervosa. European Eating Disorders Review 11(3), 231–242 (2003)
9. Przeworski, A., Newman, M.G.: Palmtop computer-assisted group therapy for social phobia. Journal of Clinical Psychology 60(2), 179–188 (2004)
10. Doherty, G., Coyle, D., Matthews, M.: Design and evaluation guidelines for mental health technologies. Interacting with Computers 22(4), 243–252 (2010)
11. Anhøj, J., Møldrup, C.: Feasibility of Collecting Diary Data From Asthma Patients Through Mobile Phones and SMS (Short Message Service): Response Rate Analysis and Focus Group Evaluation From a Pilot Study. Journal of Medical Internet Research 6(4) (2004)
12. Holtz, B., Whitten, P.: Managing asthma with mobile phones: a feasibility study. Telemedicine Journal and e-Health 15(9), 907–909 (2009)
13. Maitland, J., Sherwood, S., Barkhuus, L., Anderson, I., Hall, M., Brown, B., Chalmers, M., Muller, H.: Increasing the Awareness of Daily Activity Levels with Pervasive Computing. In: 1st International Conference on Pervasive Computing Technologies for Healthcare (2006)
14. Campbell, S.W., Kelley, M.J.: Mobile phone use among Alcoholics Anonymous members: new sites for recovery. New Media & Society 10(6), 915–933 (2008)
15. Hayes, T.L., Pavel, M., Larimer, N., Tsay, I.A., Nutt, J., Adami, A.G.: Distributed Healthcare: Simultaneous Assessment of Multiple Individuals. IEEE Pervasive Computing 6(1), 36–43 (2007)
16. Tacconi, D., Mayora, O., Lukowicz, P., Arnrich, B., Tröster, G., Haring, C.: On the Feasibility of Using Activity Recognition and Context Aware Interaction to Support Early Diagnosis of Bipolar Disorder. In: Ubiwell 2007: Ubicomp Workshop Proceedings (2007)
17. Hu, B., Hu, B., Wan, J., Dennis, M., Chen, H.-H., Li, L., Zhou, Q.: Ontology-based ubiquitous monitoring and treatment against depression. Wireless Comms. and Mobile Computing (2009)

18. Geyer, W., Richter, H., Fuchs, L., Frauenhofer, T., Daijavad, S., Poltrock, S.: A team collaboration space supporting capture and access of virtual meetings. In: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, pp. 188–196. ACM Press, New York (2001)
19. Myers, B.A.: The pebbles project: using PCs and hand-held computers together. In: CHI 2000 extended abstracts on Human factors in computing systems. ACM Press, New York (2000)
20. Carriço, L., de Sá, M., Duarte, L., Carvalho, J.: Managing Group Therapy through Multiple Devices. In: Jacko, J.A. (ed.) HCI 2007. LNCS, vol. 4553, pp. 427–436. Springer, Heidelberg (2007)
21. Zurita, G., Baloian, N., Peña, G.: A Mobile Collaborative Visual System to Support Knowledge Creation. In: ICKM 2009 (2009)
22. de Sá, M., Carriço, L., Antunes, P.: Ubiquitous Psychotherapy. IEEE Pervasive Computing 6(1), 20–27 (2007)
23. de Sá, M., Carriço, L.: Lessons from early stages design of mobile applications. In: 10th international conference on Human computer interaction with mobile devices and services, MobileHCI 2008, pp. 127–136. ACM Press, New York (2008)

Author Index