Supporting Experimental Collaborative Systems Evaluation

Claudio Sapateiro[1], Pedro Antunes[1], José A. Pino[2]

[1] Faculty of Sciences, University of Lisbon, Portugal
[2] Department of Computer Science, Universidad de Chile, Chile

Abstract—**This paper proposes a framework and tool for conducting laboratory experiments with collaborative systems. The framework and tool seek to control a large number of experimental conditions, while obtaining fine-grained log data about the individual and collaborative activities. Furthermore, the proposed framework and tool allow reusing voice communication and questionnaire modules. The paper also reports a laboratory experiment where the framework and tool were used to analyze the impact of collaboration support on teams performing maintenance activities on a network infrastructure. The obtained results highlight the capacity of the framework and tool to facilitate collaborative systems evaluation.**

Index Terms—**collaborative systems, evaluation tool, laboratory experiments.**

I. Introduction

The evaluation of collaborative systems is difficult. It is frequently unclear what to evaluate, when and how to do it [1]. Trying to answer this last question, several techniques have been proposed.

One approach to collaborative systems evaluation is to use ethnography [2]. This technique has several advantages, such as doing the evaluation in the work place, in realistic conditions. The disadvantages are also large, such as the high cost it involves, the long period it involves and applicability only after the system is under operation.

Another approach is to evaluate by modeling the use of the system [2]. The advantages in this case are applicability at any time during the system development lifecycle and low cost. The problem, nevertheless, is that the evaluation is far from realistic. Thus, modeling seems to be at some opposite extreme from ethnography.

Somehow in the middle between the two extremes are experimental studies. They are not extremely long, expensive or user-demanding. They can be done for formative and confirmatory purposes, serving at the same time to explore design ideas and to validate theory. And they can also be done with some degree of realism.

Is it possible to effectively support this “jack of all trades” collaborative systems evaluation approach? Our proposal is to use instrumented microworlds [10, 11] for this purpose. The developed tool we present is sufficiently general to accommodate several collaborative scenarios. We illustrate its use with a case study in which we wanted to assess the improvement on team performance caused by the introduction of a collaborative system.

Section II below reviews related work on the problem context. Section III presents our evaluation framework. Section IV describes the experiment. Section V contains the discussion and Section VI provides the conclusions.

II. Related Work

Collaborative systems evaluation is a complex endeavour and it raises many methodological and practical concerns [1-3]. The research literature has been consistently reporting that between one third and one half of the developed systems end up not being evaluated [4, 5]. A recent study found out that, within those systems that were evaluated, only 10% were laboratory experiments [4].

Three main reasons have been put forward to explain this situation:

1) The experimental setting is very difficult to define and control, since one has to consider a large number of independent variables. Existing collaboration frameworks identify the most important variables associated with the group, task, process and technology [6-8]. However we should expect the emergence of new categories and variables in the future;
2) Laboratory experiments are focussed on closed and repeatable phenomena, while collaboration is inherently associated with the openness and uniqueness of the human behaviour;
3) Laboratory experiments are based on artificial settings, whereas real-world collaboration occurs in naturalistic settings.

Problem 1) is a very important one, since the quality of experimental data is highly dependent on the capability to control the experimental setting. This problem has been particularly acute is some specific areas of collaboration support, e.g. situation awareness [9]. Situation awareness depends on very fined-grained cognitive issues, such as attention, interruption, memory and information overload, which make laboratory experiments particularly challenging.

Cognitive engineers and human factors experts have been studying situation awareness using the microworld approach [10, 11]. Microworlds are task-oriented environments that allow studying human behaviour under simulated conditions within a laboratory setting. They simulate basic real-world characteristics, while omitting other aspects deemed secondary for the purposes of the research.

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

Antunes, P., Sapateiro, C., & Pino, J. (2011). Supporting Experimental Collaborative Systems Evaluation. Paper presented at the 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Lausanne, Switzerland.

Therefore, microworlds provide some degree of laboratory control while affording the complexity and dynamic nature of human behaviour when accomplishing a cognitive task [12, 13]. Being functional simulations, they allow collecting large sets of data typically necessary for hypothesis-testing. Moreover, they can also provide very cost-effective indications for concept validation early in the technology development lifecycle [14, 15].

Problem 2) has primarily been raised by *interpretivists* who emphasize questions of meaning over questions of cause and effect [16]. Interpretivists fundamentally seek to understand how people behave, interact and ultimately make decisions in a collaborative environment. However those who criticize the interpretivist approach also point out that although the questions of meaning are important when exploring scenarios, they are less relevant when designing and implementing systems. We should nevertheless realize that these two views are not in opposition, and both should be used in the technology development lifecycle to reinforce collaboration support.

Problem 3) questions one fundamental tenet of laboratory experiments: artificiality is necessary to control the independent variables and thus resolving problem 1) will actually raise problem 3), unless we are open to reduce the control over some variables, increasing the naturalness of the experimental setting. The microworlds approach, just described above, allows establishing these tradeoffs by retaining some of the real-world conditions in the artificial setting.

This is the main reason why microworlds have been adopted to study emergency management. Emergencies are particularly challenging because they require decision-making under time pressure, fluid conditions and scarcity of information. Microworlds may actually manipulate the task, the communication and the shared information to provoke these real-world constraints.

Considering the problems discussed above, the framework and tool proposed in this paper adopts the microworld approach to support the evaluation of collaborative systems. The purpose is to facilitate the control of the dependent variables affecting collaboration, while at the same time promoting some of the real-world conditions found in naturalistic settings. The proposed framework and tool may contribute to establish cause-effect relationships that may then be combined with more interpretative approaches to reinforce collaborative systems evaluation.

III. EVALUATION FRAMEWORK AND TOOL

A. Framework

The evaluation framework is built upon five main functional requirements:

Control the experimental conditions. This requirement is at the core of any laboratory approach. The main goal is controlling as many influencing conditions as possible. Currently, we have been controlling the conditions listed in Table 1.

Mediate interaction, communication and collaboration. The evaluation tool mediates all human-computer interaction, human-human communication, and all individual and collaborative actions necessary to accomplish a collaborative task. This control is accomplished through an experimental protocol, which specifies the roles, tasks, messages, actions and feedback that may be performed by the users.

Execute the experimental protocol. The evaluation tool runs the protocol in accordance with the specification, which typically includes a practice task (so the users get acquainted with the system) and one or more data collection tasks (in the case of repeated experiments). Often, during the experiment, it is necessary to suspend the task to inquire the participants about the task, the collaboration, situation awareness or other dependent variables. Thus the protocol also specifies when the task should be suspended and what questions should be given to the users.

Emulate group members. One problem with collaborative systems evaluation is that they require a large number of users, and users tend to rapidly become a scarce resource. However, because of the mediating role of the evaluation tool, we may consider emulating some group members. This requires having predefined user protocols.

Obtain experimental data. Naturally, the evaluation tool must obtain and preserve in context all the information regarding interaction, communication and collaboration. The granularity of experimental data often goes down to the keystroke level.

TABLE 1. CONTROLLED EXPERIMENTAL CONDITIONS

Condition	Observations
Team size	Number of users
Interruptions	The protocol defines the moments when the task is frozen and the questionnaires are delivered to the users
Training	The protocol specifies when the users are involved in training and data collection tasks
Task complexity	Task complexity is associated with the number of specified actions
Time to accomplish task	A time limit is specified in the protocol
Location of team members	The location and movement of users in the physical space is emulated
Team composition	Roles are specified and associated with specific users and actions
Team structure	Specific actions may be associated with privileged roles
Individual and group actions	The individual and group actions are controlled according with the team composition and structure
Application use	Controlled by the emulator according with the specified roles, actions and feedback

B. Tool

As illustrated in Figure 1, the developed tool adopts a client-server architecture. The server is responsible for executing the experimental protocol, a workflow that instantiates the practice and data collection tasks according with a predefined schedule. It also manages the client synchronization, triggering the questionnaires while freezing any other activities.

The task descriptions consist of a set of records specifying the list of users, roles, actions, etc. These records are stored in the microworldSQL database. The server also populates the PFCSQL database with log data obtained when running the experiment.

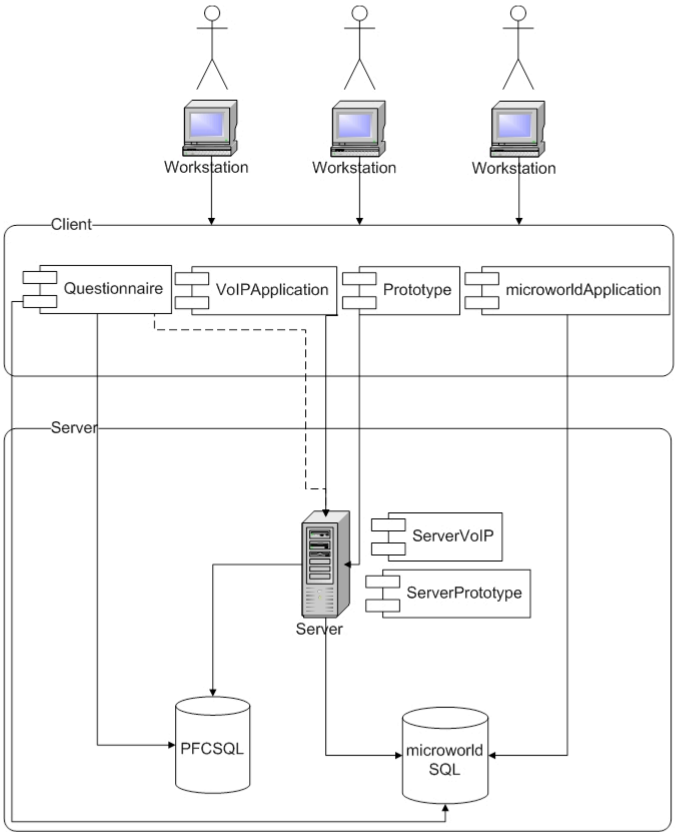

Figure 1. Architectural view of the evaluation tool

On the client side, we have four modules (standalone applications developed in C#) handling voice communication, questionnaires, application proxy and environment proxy.

Let us start with voice communication. In many collaborative settings, voice communication occurs in various ways such as face-to-face, through generic-purpose devices like telephones, and through audio channels embedded in collaboration technology. In order to control the experimental conditions, all of the existing communication channels should be levelled. By levelling the communication, one may effectively evaluate how the users communicate while avoiding channel-dependent influences such as gesturing, body language, tone, message delays, etc.

The voice communication module establishes links between selected user(s) using the Voice Over Internet Protocol. Users control the links by pressing a button. All voice messages are logged and time stamped in the server for later analysis. This is a generic module, since it does not have to be configured for the specific collaborative system under evaluation.

Originally, this module allowed users to define which type of message they would like to exchange by depressing different buttons. This would automatically code the voice exchanges. However, as reported later in the paper, we learned that users do not comply with the message types they select. Thus the message types have been removed and any necessary coding has to be done by hand at the analysis stage.

The application proxy must be tailored to the specific collaborative application being evaluated. It uses a set of user-interface tabs to invoke (or emulate) the real application functionality. The tabs provide typical user-interface elements such as labels, buttons and list options. The application proxy is based on the following relationships (specified in the microworldsSQL database): users-roles, roles-actions, actions-interface elements, actions-application feedback, and application feedback-interface elements.

The server logs and time-stamps all of the occurring actions, as well as the corresponding relationships. The number of tabs and user-interface elements that have to be created depends on the complexity of the collaborative application being evaluated.

Of course in most situations the users are not restricted to interact with the collaborative application. They may interact with other applications and physical devices. They may also move around the physical environment. The environment proxy is dedicated to emulate this functionality. Again, this requires developing a set of tabs and user-interface elements, as well as specifying the corresponding relationships.

Another important module implements the questionnaires that are presented to the users at designated times. The server triggers this module on some conditions: at particular time stamps, after some specific action, or after a certain number of actions has been performed by the group. This module is configurable to gather multiple open and closed questions. In Figure 1 we provide a conceptual view of the developed client modules. Real examples are given in Figures 4 and 5.

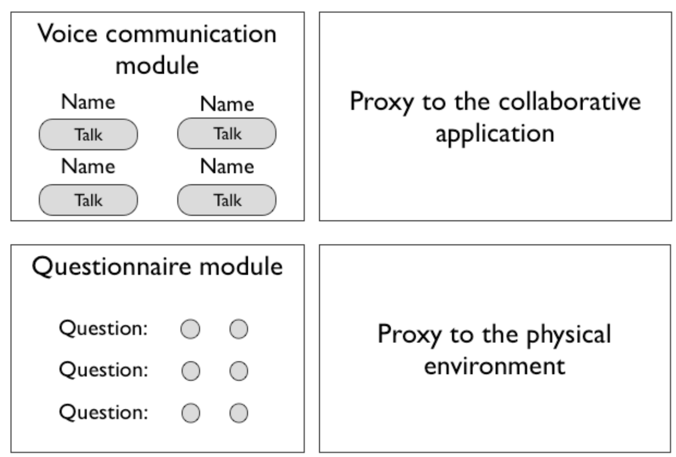

Figure 2. Client view of the evaluation tool

IV. EXPERIMENT

The experiments involved 17 teams of 3 participants each. To motivate participation and performance, a prize was drawn for the elements of the top three performing teams. The teams were randomly formed from a list of volunteers (undergraduate students in informatics).

The research goal was to assess the influence on team performance caused by a collaborative application providing data sharing and task coordination. Two main collaborative settings were evaluated: one setting based solely on verbal communication, and another combining verbal communication with collaboration support. Prior to the experimental sessions, the teams were briefed regarding the case they had to solve and their expected roles. The dependent variables under study are presented in Table 2.

TABLE 2. DEPENDENT VARIABLES

Studied Factor	Dependent Variables	Metric
Shared Awareness (ShA)	Scores from questionnaire	ShA = average score
Distributed Awareness (DA)	Scores from questionnaire # voice communications	DA1 = average score DA2 = average # voice communications
Performance (P)	Time to completion # activities	P = # activities / time to completion
Workload (W)	# activities	W = # activities
Effectiveness (E)	# activities	E = 1 / (1 - # unnecessary activities)

The overall hypothesis, in the null form, for the experiment was: *Teams operating with verbal communication and without application support have the same performance than teams operating with verbal communication and application support.*

A. Tasks

A total of four tasks were presented to the teams. Two of them were intended for practice purposes, while two others served for experimental data collection. The assigned tasks required resolving a failure in a network infrastructure. The defined network infrastructure is presented in Figure 3. The four tasks are depicted in Table 3.

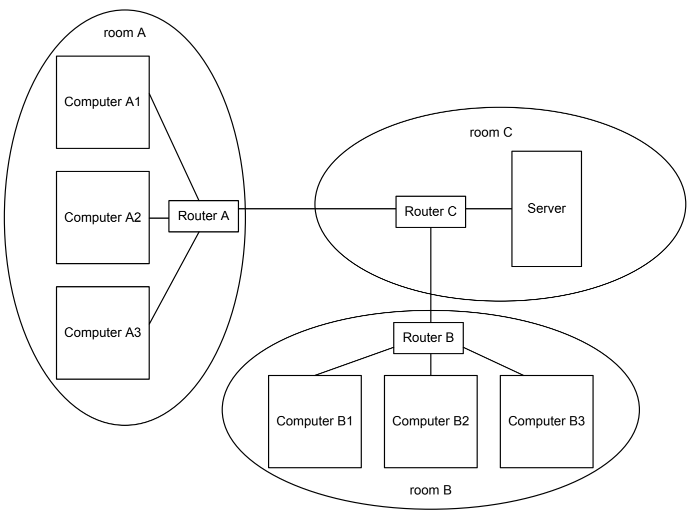

Figure 3. Simulated network infrastructure

TABLE 3. TASKS

Task #1	
Goal	Practice
Setting	w/o emulated application support
Incident	Lost of connectivity in computer A2
Problem	Router A malfunctioning
Solution	Router A firmware update
Task #2	
Goal	Data Collection
Setting	w/o emulated application support
Incident	Lost of connectivity in computer B1
Problem	Router C malfunctioning
Solution	Router C restart
Task #3	
Goal	Practice
Setting	w/ emulated application support
Incident	Lost of connectivity in computer B3
Problem	Router C malfunctioning
Solution	Router C firmware update
Task #4	
Goal	Data Collection
Setting	w/ emulated application support
Incident	Lost of connectivity in computer A3
Problem	Server malfunction
Solution	Server restart

B. Preparation

Table 4 shows the pool of individual actions that had to be emulated, while Table 5 presents the emulated collaborative actions.

TABLE 4. INDIVIDUAL ACTIVITIES

1. Check computer connectivity
2. Restart computer
3. Update/reinstall computer
4. Check router connectivity
5. Restart router
6. Update router firmware
7. Check server connectivity
8. Update server
9. Restart server
10. Move to room

TABLE 5. GROUP ACTIVITIES

1. Assign activity
2. Report device status
3. Request device status

The network devices (computers, routers and server) where considered to have just two states: Operating or Malfunctioning.

C. Questionnaires

At designated times, the task is suspended and the users are requested to answer a questionnaire regarding issues such as situation awareness and task comprehension. Relying on these answers, the scores for ShA and DA1 may be computed. Figure 4 shows two samples of these questionnaires.

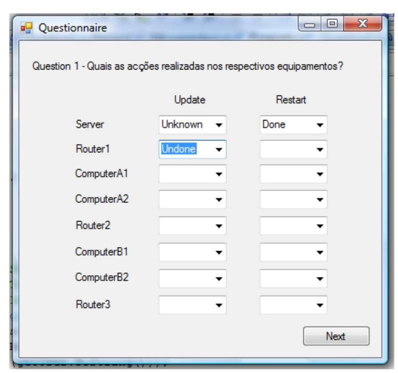

Figure 4. Questionnaires (sample)

D. Application proxy

The application proxy was created using three tabs: activity assignment, situation monitoring and activity reporting. Using the activity assignment tab, a team member may ask another one to verify the operating status of a device. The application indicates the team members' expertise and the device locations to facilitate task coordination. All of the assignments are listed in the situation monitoring tab. The activity reporting tab enables users to report on the outcome of their actions.

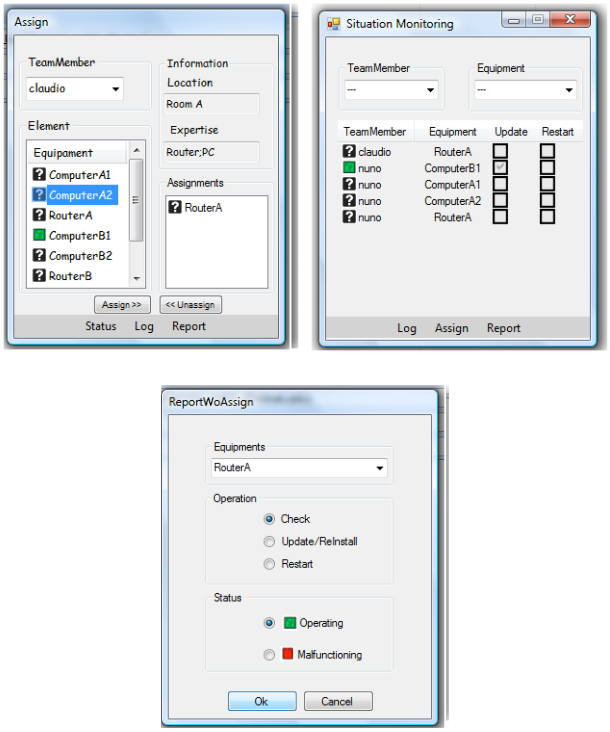

Figure 5. The activity assignment, situation monitoring and activity reporting tabs

E. Experimental design

The focus of the experiment was on the impact of the application on the teams' performance. The experimental design considered two scenarios. One scenario only allows voice communication. It uses the voice communication module to emulate the use of a telephone. The other experimental scenario, besides voice communication, supports task assignment and situation monitoring and reporting.

Three persons constituted the teams. This composition reflects the real-world situations that we found in two organizations we had previously studied.

The teams were randomly formed from a list of volunteers constituted by under-graduate students in informatics that had already completed a computer networking course. To engage the participants in the experiment, they were told that their performance would be analyzed and one additional credit could be added to the grade of one course they were still taking. Each participant signed a consent form allowing data collection and analysis, and committing not to discuss the experiment with their colleagues during the next days, while the experimental sessions were still being undertaken.

After completing the consent form, we briefed the participants about the task and the application they had to operate. The experimental sessions encompassed an initial ten-minutes period for reviewing and discussing the briefing.

Overall, the teams solved four exercises. The odd teams started with voice only and solved two exercises, the first for training purposes and the second for data collection. Then they solved two more exercises using the collaborative application, one for training purposes and the second one for data collection. Even teams performed the same tasks in the reverse order.

Each team member was randomly assigned a team number that was associated with a specific expertise profile and individual task capabilities:

- Team member #1 expertise profile allows Server, Router and Computer operations
- Team member #2 expertise profile allows Router and Computer operations
- Team member #3 expertise profile only allows Computer operations

All of the participants could also accomplish the following team activities:

- Assign: assign a task to a team member
- Report: report the outcome of an assignment
- Request: request information regarding a task

And finally, each one of the participants could communicate by voice with another participant. This could only be accomplished after pressing a button provided by the voice communication module.

After a number of actions performed on the application, the experiment was frozen and a questionnaire about the situation was delivered to the users. At the end of each session, a debriefing questionnaire composed by four questions was also delivered (measures for two of these questions are presented later):

- Problem identification: what was/were the device(s) causing the problem
- Solution identification: what were the main operations that resolved the problem
- Who operated a particular network device
- What operations were performed in a particular device

F. Pilot Tests

To fine-tune the experimental design, three pilot tests were conducted (Figure 6). These pilot tests lead to several important changes in the experimental design. First, we decided to adopt a flat team structure, since the teams operating under a team leader demonstrated a passive attitude. Another factor perceived in the pilot tests was that the initial voice communication module was not working as intended. Initially, this module supplied different buttons that should be pressed according with the type of voice communication. This intended to ease the post-hoc analysis. However, we observed that the participants were starting a call pressing any one of the buttons; and during the call different types of communication were actually taking place (e.g., a task assignment and a request). Thus the final experimental setting only used one button for initiating the voice communication.

Figure 6. Pilot tests

The pilot tests also allowed establishing two empirical conditions. One was the exercise duration, which was set at 15 minutes. The other one was the trigger for questionnaires, which would be after 10 actions on the environment proxy.

Another fine-tuning was related with the briefing sessions. A clarification had to be made insisting that the team members should avoid moving between different rooms to individually resolve the problem and alternatively communicate with the group to coordinate their activities. In the beginning, it was observed that the more proactive team members immediately started visiting all rooms to identify and solve the problem. This behaviour was possible since room changes were being emulated by clicking on a button. With this note of attention in the briefing sessions, the analysis of how the team members moved around the simulated environment could be more realistic.

Finally, the last adjustments regarded how to freeze the task to deliver the questionnaires. We determined that the freeze should finish only when all team members respond to the questionnaires, so that the answers consider a known and stable state.

G. Measures

1) Workload

Team workload was obtained by measuring the number of actions each team performed in the two settings that were defined. Table 6 shows the average sum of all actions in both settings (W - with application support; W/O - without application support). As can be observed, the workload was similar in both settings. The statistical analysis revealed no significant differences in the data sets[1].

TABLE 6. WORKLOAD MEASURES

	W	**W/O**
AVG	24,47	23,27
STDEV	5,26	6,78

2) Effectiveness

All teams under the two experimental settings finished the exercises. The effectiveness was evaluated considering the number of unnecessary actions:

- An update before a check operation
- An update before a restart operation
- A restart before a check operation

Regarding this measure, all teams were equally effective.

3) Completion Time

As can be seen in Table 7, the teams operating only with voice communication took slightly longer to complete the task. But the subsequent statistical analysis revealed no significant differences in the data sets[2].

TABLE 7. COMPLETION TIME

	W	**W/O**
AVG	6,67	8,16
STDEV	1,55	3,69

4) Task Perception

As already mentioned, the users completed a debriefing questionnaire after each exercise. The questionnaire had questions concerning the understanding of the problem and the solution. Problem understanding (PU) requires identifying the key devices causing the network connectivity problems. The solution understanding (SU) concerns the identification of the key operations necessary to re-establish the network connectivity. The answers to both questions were coded as: 0 – if the respondent has no idea about the problem/solution; 0,5 – if the respondent partially identified the problem/solution; and 1 – if the respondent completely identified the problem/solution. Three persons independently coded the responses. Whenever concordance was not found, the average score was used[3]. The team scores were then computed from the average scores of its members.

Table 8 summarizes the obtained results. The subsequent statistical analysis reveals significant differences in the solution identification[4].

TABLE 8. TASK PERCEPTION

	W		**W/O**	
	PU	**SU**	**PU**	**SU**
AVG	0.63	0.39	0.68	0.59
STDEV	0.24	0.21	0.23	0.20

V. DISCUSSION

The experiment discussed above highlights many of the advantages and drawbacks of the proposed evaluation framework and tool. On the positive side, we may account for the capacity to reutilize some instruments, such as the server responsible for the experimental protocol, and the voice communication and questionnaires modules.

[1] The Wilcoxon signed-rank test was used, since the data had not a normal distribution, giving $p = 0.62$, so the null hypothesis is accepted.

[2] The Wilcoxon signed-rank gave $p = 0.08$, so the null hypothesis is accepted, although being very close to being statistically significant.

[3] Only 5% of the scores had to be computed this way.

[4] The Wilcoxon signed-rank for PU gave $p = 0.35$, so the null hypothesis is accepted. In the SU condition, $p = 0.013$, so the null hypothesis is rejected.

We also emphasize the capacity to control a large set of experimental conditions, which have always been difficult to settle in collaborative settings. And we should not neglect that the proposed framework and tool not only automate the production of a large collection of experimental data but also ease the later analysis through a careful consideration for the relationships between teams, users, roles, actions and application feedback.

The experiment described in this paper also highlights the possible advantages of suspending a collaborative task to inquire the users about the task and the team situation. Suspending collaborative tasks may be difficult to achieve in naturalistic settings but is easy to orchestrate in a laboratory setting under controlled conditions. Furthermore, the users may be more promptly and effectively inquired than when using other approaches such as debriefings and post-hoc analysis. This is especially important when the data being collected concerns fine-grained cognitive phenomena, such as group attention, task awareness, mental load, memory, impact of interruptions, expectancies, etc.

Of course the associated costs include developing proxies for the collaborative application and environmental context under evaluation. Furthermore, the laboratory approach requires levelling the communication channels, such that, for instance, face-to-face and telephone talks must be substituted by a more artificial approach, where users have to explicitly request to communicate. When considered relevant to the evaluation, the interactions with the environment must also be levelled in the same way.

All in all, the proposed evaluation framework and tool are able to capture very insightful information about the group behaviour. The reported experiment points out towards the various types of measures that may be obtained, such as workload, task effectiveness, task completion time and perception, just to mention a few. As expected, many of these measures do not lead to statistically significant differences.

However, we have not yet been able to fully explore the possibilities of the proposed framework and tool. In particular, we have not yet experimented using mixed groups, where some group members may be real users while others may be artificially set up. This would further increase the value of our approach.

VI. Conclusions

We had initially presented the challenge of collaborative systems evaluation. It was also mentioned that the controlled experimental approach was perhaps the best tradeoff between realism and cost.

We noticed that not many evaluations are made with the experimental approach. Several reasons may explain that or perhaps a combination of all of them.

This paper has presented an evaluation framework and tool for the experimental approach to evaluation. It may help to ease further application of the technique by various means. First, by providing a set of clear requirements for a suitable kind of experiments applicable to collaborative systems. This can be useful by itself to developers of such systems.

The second contribution of the paper is the tool. Ours or similar tools can implement monitoring, communication and data collection in a rigorously conducted experiment.

Finally, the tool may exploit simulated users, something that we will evaluate in future research. This feature would be particularly useful when conducting large-scale collaborative experiments. Of course, the quality of the obtained results will depend on the validity of the users's simulation.

References

[1] P. Antunes, V. Herskovic, S. Ochoa, and J. Pino, "Structuring Dimensions for Collaborative Systems Evaluation," *ACM Computing Surveys,* vol. to appear, forthcoming.

[2] V. Herskovic, J. Pino, S. Ochoa, and P. Antunes, "Evaluation Methods for Groupware Systems," in *Groupware: Design, Implementation, and Use. 13th International Workshop, CRIWG 2007, Bariloche, Argentina, September 2007 Proceedings*. vol. 4715, J. Haake, S. Ochoa, and A. Cechich, Eds. Heidelberg: Springer-Verlag, 2007, pp. 328-336.

[3] D. Neale, J. Carroll, and M. Rosson, "Evaluating computer-supported cooperative work: models and frameworks," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, Chicago, Illinois, 2004, pp. 112-121.

[4] P. Antunes and J. Pino, "A Review of CRIWG Research," in *Groupware: Design, Implementation, and Use. 16th CRIWG Conference on Collaboration and Technology. Maastricht, The Netherlands*. vol. 6257 Heidelberg: Springer-Verlag, 2010, pp. 1-15.

[5] D. Pinelle and C. Gutwin, "A review of groupware evaluations," in *Proceedings of 9th IEEE WETICE Infrastructure for Collaborative Enterprises*, 2000.

[6] J. Fjermestad and S. Hiltz, "An assessment of group support systems experimental research: Methodology and results," *Journal of Management Information Systems,* vol. 15, pp. 7-149, 1999.

[7] J. McGrath, *Groups: Interaction and performance*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[8] A. Pinsonneault and K. Kraemer, "The impact of technological support on groups: An assessment of the empirical research," *Decision Support Systems,* vol. 5, pp. 197-216, 1989.

[9] C. Sapateiro and P. Antunes "An Emergency Response Model Toward Situational Awareness Improvement," in *International Conference on Information Systems for Crisis Response and Management*, Göteborg, Sweden, 2009.

[10] B. Brehmer and D. Dorner, "Experiments with computer-simulated microworlds: Escaping both the narrow straits of the laboratory and the deep blue sea of the field study," *Computers in Human Behavior,* vol. 9, pp. 171-184, 1993.

[11] W. Gray, "Simulated task environments: The Role of high-fidelity simulations, scaled worlds, synthetic environments, and laboratory tasks in basic and applied cognitive research," *Cognitive Science Quarterly,* pp. 205-207, 2002.

[12] J. Funke, "Dynamic systems as tools fo analyzing judgment " *Thinking and Reasoning,* vol. 7, pp. 69-89, 2001.

[13] M. Jobidon, R. Breton, R. Rousseau, and S. Tremblay, "Team response to workload transition: The role of team structure," in *Cognition: Beyond the brain: Embodied, situated and distributed congnition* Montréal, Canada, 2006, pp. 22-32.

[14] B. Jhoansson, J. Trnka, and R. Granlund, "The effect of geographical information systems on a collaborative C2 task," in *4th International Conference on Information Systems for Crisis Response and Management*, Delft, The Netherlands, 2007.

[15] J. Schraagen and J. van den Ven, "Improving decision making in crisis response through critical thinking support," *Journal of Cognitive Engineering and Decision Making,* vol. 2, pp. 311-327, 2008.

[16] R. Davison, "An Action Research Perspective of Group Support Systems: How to Improve Meetings in Hong Kong," Department of Information Systems, City University of Hong Kong, 1998.