Structuring Dimensions for Collaborative Systems Evaluation

PEDRO ANTUNES
University of Lisbon, Lisbon, Portugal
VALERIA HERSKOVIC
University of Chile, Santiago, Chile
SERGIO F. OCHOA
University of Chile, Santiago, Chile
AND
JOSE A. PINO
University of Chile, Santiago, Chile

Collaborative systems evaluation is always necessary to determine the impact a solution will have on the individuals, groups and the organization. Several methods to do this evaluation have been proposed. These methods comprise a variety of approaches with various goals. Thus, the need for a strategy to select the most appropriate method for a specific case is clear. This research work presents a detailed framework to evaluate collaborative systems according to given variables and performance levels. The proposal assumes that evaluation is an evolving process during the system lifecycle. Therefore, the framework is complemented with a collection of guidelines to evaluate collaborative systems according to product development status. Two examples illustrate the framework use.

Categories and Subject Descriptors: H.5.3 **[Information Interfaces and Presentation]**: Group and Organization Interfaces - *Evaluation/methodology* - *Theory and models;* K.6.1 **[Management of Computing and Information Systems]**: Project and People Management - *Life cycle* - *Systems development*.
General Terms: Measurement, Human-Factors, Management
Additional Key Words and Phrases: Collaborative Systems Evaluation, Human-Computer Interaction, Interaction Assessment, Evaluation Dimensions, Evaluation Guidelines

1. INTRODUCTION

The evaluation of collaborative systems is an important issue in the field of Computer Supported Cooperative Work (CSCW). Appropriate evaluation justifies investments, appraises stakeholders' satisfaction, or re-directs systems development to successful requirements matching. Several specific evaluation methods have been proposed [Herskovic et al. 2007] beyond those intended for Information Systems in general. However, many collaborative systems seem to be poorly evaluated. A study of 45 articles from 8 years of the CSCW conference revealed that almost one third of the presented collaborative systems were not evaluated in a formal way [Pinelle and Gutwin 2000]. Even when evaluations are done, many of them seem to be performed in an ad-hoc way, depending on the researchers' interests or the practical adequateness for a specific setting [Inkpen et al. 2004; Greenberg and Buxton 2008]. This shows a need for a strategy that helps choose suitable collaborative systems evaluation methods.

Authors' addresses: P. Antunes, Department of Informatics, University of Lisbon; email: paa@di.fc.ul.pt; V. Herskovic, Computer Science Department, University of Chile; email: vherskov@dcc.uchile.cl; S.F. Ochoa, Computer Science Department, University of Chile; email: sochoa@dcc.uchile.cl; J.A. Pino, Computer Science Department, University of Chile; email: jpino@dcc.uchile.cl.

Antunes, P., V. Herskovic, S. Ochoa and J. Pino (forthcoming) "Structuring Dimensions for Collaborative Systems Evaluation." ACM Computing Surveys, to appear. 5-year impact factor: 12.7.

This paper proposes a framework to evaluate a collaborative system under development or procurement, and also a set of guidelines to select the appropriate evaluation techniques. We understand evaluation as an evolving process that is in some way associated with the conception, design, construction and deployment activities of a system development. The guidelines also address the case of a collaborative system being purchased by an organization.

We consider two main structuring dimensions in order to frame the various contingencies of the evaluation process. One of these dimensions defines the set of relevant evaluation variables and the other one concerns the levels of human performance under evaluation. The considered evaluation variables are realism, generalization, precision, system detail, system scope and invested time. The adopted levels of human performance consider role-based, rule-based and knowledge-based tasks. The approach is generally applicable to all types of collaborative systems.

Section 2 analyzes the major problems associated with collaborative systems evaluation. Section 3 discusses the related work. In particular, it describes, categorizes and compares several well-known evaluation methods. Section 4 describes the proposed framework for evaluation. Section 5 presents the collection of guidelines for evaluation. Section 6 contains two case studies of collaborative systems evaluation. Finally, Section 7 presents the conclusions and further work.

2. STUDYING COLLABORATIVE SYSTEMS EVALUATION

2.1. Why is collaborative systems evaluation so difficult?

The success of a collaborative system depends on multiple factors, including the group characteristics and dynamics, the social and organizational context in which it is inserted, and the positive and negative effects of technology on the group's tasks and processes. Therefore, evaluation should attempt to measure several effects on multiple interdependent stakeholders and in various domains. What distinguishes collaborative systems from other information systems is indeed the need to evaluate its impact with an eclectic approach.

Ideally, a single collaborative systems evaluation method should cover the individual, group and organizational domains, assessing whether or not the system is successful at the combination of those realms. Unfortunately, no such single method is currently available, and may never be. The fundamental cause for it is related with the granularity and time scale of the information obtained at these three domains [Newell 1990]:

- The information pertaining to the individual is usually gathered at the cognitive level, focusing on events occurring on a time frame in the order of a few minutes or even seconds;
- Group information is gathered at the interaction/communication level, addressing activities occurring in the range of several minutes and hours;
- The information regarding organizational impact concerns much longer time frames, usually in the order of days, months and even years.

Moreover, the results of an evaluation should be weighted by the degree of certainty in them, which depends on the maturity of what is being evaluated. At the inception phase, the product to be evaluated may be just a concept or a collection of design ideas, so the

results have a high degree of uncertainty. When the development reaches full deployment, the product may then be tested in much more far-reaching and systematic ways, providing evaluators with an increased degree of certainty and relatively precise results.

The dependence between product development and evaluation is noticeable in the star model illustrated in Figure 1 [Hix and Hartson 1993]. Evaluation is a central aspect of a broad collection of activities aiming to develop a product, but it has to compete with the other activities for attention, relevance and critical resources such as people, time and money.

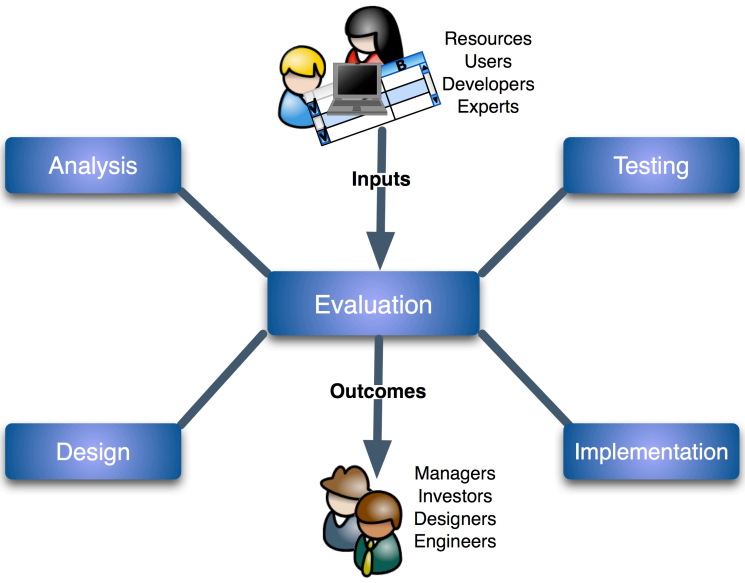

Figure 1. The dependence between product development and evaluation

2.2. Why to evaluate and how to evaluate?

McGrath [1984] characterized the purpose of conducting an evaluation as addressing three main goals: precision, generalizability and realism. The first goal concerns the precision of the data obtained by the instrument being used. This goal is inherently linked with the capability to control the dependent and independent variables, the subjects and the experiment. Laboratory experiments are usually selected to accomplish this high level of control.

Generalizability concerns the extent to which the obtained results may be applied to a population. High set goals on generalizability usually imply adopting large-scale inquiries and surveys, while low generalizability is obtained by interviewing a small audience.

Realism addresses how closely the obtained results represent real-world conditions, considering the work setting, the population of users, and the tasks, stimulus, time stress, absence of observers, etc. Laboratory experiments have been criticized for providing low realism, especially with collaborative systems, whereas field studies have been considered to score high on realism but low on precision.

Overall, the ideal evaluation should maximize the three goals, for instance using multiple evaluation methods and triangulating the obtained results. Nevertheless,

McGrath [1984] states this would result in a very costly and difficult to carry out evaluation, which ultimately may have to be considered utopian. McGrath then identified the major compromising strategies adopted to overcome the costs of an ideal evaluation:

- Field strategies – Set out to make direct observations of realistic work;
- Experimental strategies – Based on artificial experimental settings aiming to study specific activities with high precision;
- Respondent strategies – Obtaining evidence from sampling a large and representative population;
- Theoretical strategies – Using theory to identify the specific variables of interest.

2.3. What to evaluate?

Pinsonneault and Kraemer [1989] defined one of the pioneering collaborative systems evaluation frameworks addressing the practical aspects related with what exactly is the object under evaluation. The framework adopts an input-process-output view to conceptualize the relationship between the technology support and other factors related with the group, group behavior and work context:

- Contextual variables – The important factors in the group behavior. Contextual variables belong to five major categories: personal, situational, group structure, task characteristics and technology characteristics (e.g., anonymity and type of communication).
- Group process – The characteristics of the group interaction, including decisional characteristics, communicational characteristics and interpersonal characteristics.
- Outcomes – The outcomes of the group process affected by the technology support, including task-related outcomes and group-related outcomes.

This framework has been highly influential, especially because it created a common foundation for comparing multiple collaborative systems experiments. Also, the distinction between group process and outcomes highlights two quite different evaluation dimensions commonly found in the literature, the former usually addressing questions of meaning (e.g., ethnography [Hughes et al. 1994] and groupware walkthrough [Pinelle and Gutwin 2002]), and the latter addressing questions of cause and effect (e.g., value creation [Briggs et al. 2004]). Other collaborative systems evaluation frameworks, such as the ones proposed by Hollingshead and McGrath [1995] and Fjermestad and Hiltz [1999], are based on this framework.

Regarding more recent evaluation frameworks, Neale et al. [2004] proposed a simplified evaluation framework, basically consisting of two categories. One encompasses the contextual variables already mentioned. The other category concerns the level of work coupling attained by the work group, which combines technology characteristics with group process characteristics. Along with this proposition, Neale et al. [2004] also recommend blending the different types of evaluation. Araujo et al. [2002] also proposed a simplified framework based on four dimensions: group context (which seems consensual in every framework), system usability, level of collaboration (similar to the level of work coupling), and cultural impact. The cultural impact is seen as

influencing the other dimensions, thus introducing a feedback loop in the input-process-output view.

2.4. When to evaluate?

The timing of the evaluation is inherently associated with the development process. It is common to distinguish between the preliminary and final development stages [DeSanctis et al. 1994; Guy 2005]. The preliminary stage affords what has been designated formative evaluation [Scriven 1967], which mainly serves to provide feedback to the designers about the viability of design ideas, usability problems, perceived satisfaction with the technology, possible focal points for innovation and alternative solutions, and also feedback about the development process itself. The final stage, which sometimes is designated summative evaluation, provides complete and definitive information about the developed product and its impact on the users, the group and the organization.

3. RELATED WORK

This section begins by describing the way we built a relevant corpus of papers to be analyzed, and also the literature review method used to classify the evaluation strategies. These retrieved strategies were split into two subsets: (1) evaluation methods that are presented in section 3.2, and (2) evaluation frameworks that are described in section 3.3.

3.1. Literature Review Methodology

We began our search for articles in the literature concerning collaborative systems evaluation by exploring various ways to get a large initial corpus of papers. The main technique to obtain papers was to search using pertinent search engines, e.g., Google Scholar and the ACM Digital Library, through combinations of keywords containing terms related with CSCW and evaluation (e.g. groupware evaluation, collaborative systems assessment, etc.). The proceedings from several relevant conferences and workshops in the area (e.g., CSCW, ECSCW, CHI, WETICE) were reviewed to find additional papers to add to the corpus. Then, we examined references of already found relevant papers, as well as searched through Google Scholar for papers citing those we had found. Each paper was carefully reviewed in order to determine if it had merit to be part of the set of pre-selected articles. The large set thus built was reduced by filtering out papers that did not present a distinctive evaluation proposal.

The initial analysis of the corpus of papers identified several types of proposals to evaluate collaborative systems. Some papers presented ad-hoc techniques or tools (e.g., questionnaires) defined specifically to evaluate a particular application. Such papers were not considered in our analysis because we were interested in finding evaluation methods with a clear and reusable evaluation strategy. Papers reporting just evaluation tools (i.e., single instruments intended to measure system variables) were also removed from the main corpus when they did not include an evaluation process. Once filtering out the tools and no-reusable evaluation proposals we analyzed the remaining contributions and realized those proposals could be classified in two categories: *evaluation methods* and *evaluation frameworks*.

On the one hand, we define evaluation methods as procedures used to apply evaluation tools with a specific goal. For example, the Perceived Value evaluation method [Antunes and Costa 2003] uses evaluation tools such as questionnaires and checklists with the goal of determining the organizational impact of meetingware. On the other hand, we define evaluation frameworks as macro-strategies used to organize the

evaluation process. Several evaluation methods and tools may be included in an evaluation framework.

After classifying the articles in these two categories, the analysis of the contributions was focused on the evaluation methods category. Each subset was then expanded to include seminal evaluation methods that have been adapted to the collaboration context.

The careful analysis of these selected papers led us to define a set of relevant questions that can be applied to each method in order to classify them more properly. These questions are the following ones: purpose of evaluation (why), evaluation tools being used (how), outcomes of the evaluation (what), and moment in which the evaluation is conducted (when). Section 3.2 presents this classification, which was complemented with a narrative summary of the procedures adopted by each method.

Moreover, we classified the evaluation methods by publication date, which served to build an understanding of their emergence and subsequent life. This classification allowed us to construct the timeline presented in Appendix A. The timeline analysis shows some identifiable patterns:

- The adaptation of single-user evaluation methods, developed in the Human-Computer Interaction field, to the specific context of collaborative systems. This has occurred, for instance, with walkthroughs (structured walkthroughs, cognitive walkthroughs, groupware walkthroughs), heuristic evaluation (heuristic evaluation, heuristic evaluation based on the mechanics of collaboration) and scenario-based evaluation.
- The assimilation of perspectives, methods and techniques from other fields beyond technology development. The clearest example is ethnography (observational studies, quick-and-dirty ethnography, workplace studies), but cognitive sciences also seem to have impact (KLM, cognitive walkthroughs, computational GOMS).
- The increasing complexity of the evaluation context. Most early methods (e.g., structured walkthroughs, KLM, discount methods) seem to focus on very specific variables measured under controlled conditions, while some of the latter methods seem to consider broader contextual issues (e.g., multi-faceted evaluation, perceived value, evaluating collaboration in co-located environments, lifecycle based approach).

Finally we also analyzed the proposals concerning evaluation frameworks. Section 3.3 presents the most representative ones.

3.2. Sample of evaluation methods

This section presents a sample of collaborative systems evaluation methods. Table 1 presents a summarized characterization of the selected evaluation methods, describing the purpose of the evaluation (why), the evaluation tools being used in each method (how), the outcomes of the evaluation (what), and the moment in which evaluation is conducted (when). Then, we present a brief description of the steps involved in each evaluation method.

Table 1. Characterization of evaluation methods

Method	Why	How	What	When
GHE	Precision	Software Analysis, checklist	Effectiveness, efficiency, satisfaction	Summative
GWA	Precision	Software Analysis	Effectiveness, efficiency, satisfaction	Formative
CUA	Precision	Software Analysis	Effectiveness, efficiency, satisfaction	Formative
GOT	Realism	Observation, checklist	Effectiveness, efficiency, satisfaction	Summative
HPM	Precision	Interaction Analysis	Group performance	Formative
QDE	Realism	Observation	Redesign	Summative
PAN	Generalizability	Formal analysis	Efficiency	Formative
PVA	Realism	Questionnaire, checklist	Organizational Impact	Formative
SBE	Realism/Precision	Interviews	Organizational Contributions	Formative
COS	Realism	Interviews, observation	Redesign	Formative
TTM	Generalizability	Interviews, observation	Predicted actual use	Formative
KMA	Generalizability	Software analysis, checklist	Knowledge circulation	Formative

Groupware Heuristic Evaluation (GHE). GHE [Baker et al. 2002] is based on eight groupware heuristics, which act as a checklist of characteristics a collaborative system should have. Evaluators who are experts in them examine the interface, recording each problem they encounter, the violated heuristic, a severity rating and optionally, a solution to the problem. The problems are then filtered, classified and consolidated into a list, which is used to improve the application.

Groupware Walkthrough (GWA) [Pinelle and Gutwin 2002]. A scenario is a description of an activity or set of tasks, which includes the users, their knowledge, the intended outcome, and circumstances surrounding it. Evaluators construct scenarios by observing users and identifying episodes of collaboration. Each evaluator, taking the role of all users or one in particular, walks through the tasks in a laboratory setting, recording each problem he encounters. A meeting is then conducted to analyze the results of the evaluation.

Collaboration Usability Analysis (CUA) [Pinelle et al. 2003]. Evaluators map collaborative actions to a set of collaboration mechanisms, or fine-grained representations of basic collaborative actions, which may be related with elements in the user interface. The resulting diagrams capture details about task components, a notion of the flow through them and the task distribution.

Groupware Observational User Testing (GOT). GOT [Gutwin and Greenberg 2000] involves evaluators observing how users perform particular tasks supported by a system in a laboratory setting. Evaluators either monitor users having problems with a task, or

ask users to think aloud about what they are doing to gain insight on their work. Evaluators should focus on collaboration and analyze users' work through predefined criteria, e.g., the mechanics of collaboration.

Human-Performance Models (HPM) [Antunes et al. 2006]. Evaluators first decompose the physical interface into several shared workspaces. Then, they define critical scenarios focused on the collaborative actions for the shared workspaces. Finally, evaluators compare group performance in the critical scenarios to predict execution times.

"Quick-and-dirty" Ethnography (QDE) [Hughes et al. 1994]. Evaluators do brief ethnographic workplace studies to provide a general sense of the setting for designers. QDE suggests the deficiencies of a system, supplying designers with the key issues that bear on acceptability and usability, thus allowing existing and future systems to be improved.

Performance Analysis (PAN) [Baeza-Yates and Pino 1997; Baeza-Yates and Pino 2006]. The application to be studied is modeled as a task to be performed by a number of people in a number of stages, and the concepts of result quality, time, and total amount of work done are defined. The evaluators must define a way to compute the quality (e.g., group recall in a collaborative retrieval task), and maximize the quality vs. work done either analytically or experimentally.

Perceived Value (PVA). PVA [Antunes and Costa 2003] begins by developers identifying relevant components for system evaluation. Then, users and developers negotiate the relevant system attributes to be evaluated by users. After the users have worked with the system, they fill out an evaluation map by noting whether the components support the attributes or not. Using these ratings, a metric representing the PV is calculated.

Scenario-Based Evaluation (SBE). SBE [Haynes et al. 2004] uses field evaluation. Evaluators perform semi-structured interviews with users to discover scenarios, or detailed descriptions of activities, and claims about them. Then, focus groups validate these findings. The frequency and percentage of positive claims help quantify the organizational contributions of the system, and the positive and negative claims about existing and envisioned features provide information to aid in redesign.

Cooperation Scenarios (COS) [Stiemerling and Cremers 1998]. Evaluators conduct field studies, semi-structured interviews, and workplace visits. They thus identify scenarios, cooperative behavior, users involved in it, their roles and the relevant context. For each role involved in the cooperative activity, evaluators analyze the new design to see how the task changes and who benefits from the new technology. Then, the prototype is presented as a scenario in a workshop with users to discover design flaws.

Knowledge Management Approach (KMA). Evaluation using KMA [Vizcaíno et al. 2005] measures whether the system helps users detect knowledge flows and disseminate, store and reuse knowledge. The knowledge circulation process is comprised of six phases (knowledge creation, accumulation, sharing, utilization, internalization), which are also the areas to be evaluated by this approach. The evaluation is performed by answering questions associated to each area.

Technology Transition Model (TTM). TTM [Briggs et al. 1998] predicts the actual system use as a function of the intent to use the system, the value that users attribute to it, how frequently it will be used and the perceived cost of transition. This model proposes that users weigh all factors affecting the perceived value of a system, producing an overall value corresponding to their perception of the usefulness of the system. Users' opinions

are obtained by interviews, archival analysis and observations. These opinions are the basis to predict actual use of the system. The collaborative application can thus be evaluated to increase the speed of its acceptance, while reducing the risk of technology transition.

3.3. Evaluation frameworks

This section presents existing macro-strategies to perform evaluation. Several frameworks adopt an input-process-output view [Pinsonneault and Kraemer 1989; Ross et al. 1995; Damianos et al. 1999; Araujo et al. 2002; Huang 2005], while others include evaluation in the software development cycle [Hix and Hartson 1993; Baecker et al. 1995; Veld et al. 2003; Huang 2005].

The star model [Hix and Hartson 1993] proposes evaluation as the central phase in the software development cycle. This means evaluation should be conducted after every development step. Baecker, Grudin et al. [1995] regard development as an iterative process of design, implementation and evolution, and apply appropriate evaluation methods after each development phase. The concept design is evaluated through interviews, the functional design through usability tests, the prototype through heuristics, the delivered system through usability tests, and finally the system evolution is evaluated through interviews and questionnaires.

Huang [2005] proposes a lifecycle strategy. An evaluation plan is defined before starting development considering five domains: context, content, process, stakeholders and success factors. The plan is improved at each cycle after analyzing the evaluation results. The E-MAGINE framework [Veld et al. 2003] has a similar structure: first, a meeting and an interview are done to establish the evaluation goals and group profile. This information guides the selection of evaluation methods and tools which will be used.

Damianos, Hirschman et al. [1999] present a framework based on Pinsonneault and Kraemer's proposal [1989]. The framework has four levels: requirement, capability, service and technology. Appropriate methods should be selected at each level to conduct the evaluation. At the requirement level, evaluation concerns the overall system quality. At the capability level, evaluation addresses the system capabilities. At the service level, evaluation is focused on performance and cost. Finally, the technology level concerns benchmarking technical issues.

The PETRA strategy combines the perspective of the evaluator and the perspective of the users, or participants [Ross et al. 1995]. In this way, it aims to achieve a balance between theoretical and practical methods.

The CSCW Lab proposes four dimensions to consider when evaluating collaborative systems: group context, usability, collaboration and cultural impact [Araujo et al. 2002]. Each dimension is a step of the evaluation process, which consists of characterizing the group and work context, measuring usability strengths and weaknesses and collaboration capabilities, and studying the impact of the application over time.

4. COLLABORATIVE SYSTEMS EVALUATION FRAMEWORK

4.1. Variables

Section 2.1 introduced the need to choose variables to assess a collaborative system under development. We should, then, characterize our framework according to a set of variables providing insights on the evaluation methods to be applied.

A starting point is McGrath's evaluation goals mentioned in Section 2.2. These goals are fundamental to layout the evaluation methodology. For our evaluation framework, it

seems thus appropriate to choose variables associated to these goals; if the evaluation methods change in succeeding evaluations, these variables will reflect the new evaluation methodology (Figure 2). Precision, generalization and realism are then our first three variables to describe the evaluation method. Precision focuses on the accuracy of the measuring tools, generalization concerns the extent (in term of population) to which the method must be applied, and realism refers to whether the evaluation will use real settings or not.

It is important to incorporate the level of system detail (depth) as one of the dimensions to characterize the evaluation activities. This dimension concerns the granularity of the evaluation. Evaluation methods with a high level of system detail (e.g. mouse movements of a user) will provide more specific and accurate information to improve the system under review.

Another dimension we would like to incorporate in the evaluation framework is the scope (breadth) of the system being evaluated. An evaluation having a large value for this variable would mean the system being evaluated has many functionalities and components being assessed. This variable complements the detail dimension. The breadth dimension can help identify the scope of a system that could be coped with a particular evaluation method. We note that while the first three variables in our framework consider theoretical issues, the system detail and scope concern the product development state.

Finally, an invested time variable describes the time used by the evaluators to carry out the work. This variable may not be completely independent from other variables, notably, detail and scope (since, e.g., a coarse-grain evaluation narrowing to a few functionalities will probably require little invested time). However, from a more practical than theoretical standpoint, it is an important variable to distinguish the efficient evaluation methods from those which are not. Therefore, invested time is included in the framework.

Other variables could be considered as candidate to be added to our framework; however after analyzing several of these variables, such as evaluation cost/effort, feedback richness or required expertise, we realized they could be inferred in some way by relating the results of the proposed dimensions. Moreover, the selected variables seem to be adequate to analyze evaluation methods, as shown in succeeding sections.

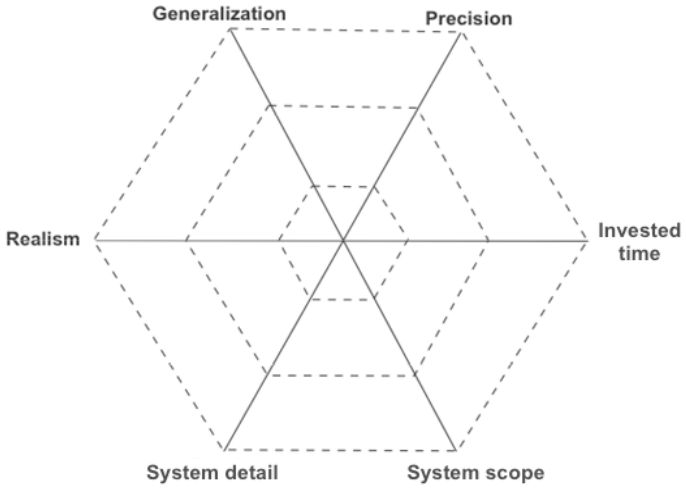

Figure 2. Variables adopted for the Evaluation Framework

Figure 2 shows a radar-graph representation of the evaluation variables. A specific method is represented by a dot in each of the axes (variables). Each axis has a scale from 0 (or minimum value) in the origin to a certain maximum value. These dots may be joined to show a certain evaluation shape. It may be noticed a numeric value for the area within a shape does not make much sense, since the scales are not the same for each variable. However, a "light" evaluation procedure will probably have low values for most or all variables, whereas a "heavy" one will probably score high in several evaluation variables.

4.2. Performance levels

Reason [2008] proposed a three-layered model of human performance in organizational contexts by extending a proposal by Rasmussen and Jensen [1974]. We will apply this model to the specific context of collaborative systems evaluation.

The model categorizes human performance according to two dimensions: situation and situation control. According to the situation dimension, the organizational activities may be classified as: (1) *routine*, when the activities are well known by the performers and accomplished in an almost unconscious way; (2) *planned*, when the activities have been previously analyzed by the organization and thus there are available plans and procedures to guide the performers accomplishing the intended goals; and (3) *novel*, when the way to achieve the intended goals is unknown to the organization and thus human performance must include problem analysis and decision-making activities. Shared workspaces, workflow systems and group support systems are good examples of collaborative systems technology supporting the routine, planned and novel dimensions.

The other dimension concerns the level of control the performers may exert while accomplishing the set goals. The control may be *mechanical*, when a human action is performed according to a predefined sequence imposed by the technology. The control may be *human*, when the technology does not impose any predefined action sequence. Finally the control may be *mixed*, when it opportunistically flows between the humans and the technology. These two dimensions serve to lay down the following performance levels (Figure 3):

- Role-based performance – Encompassing routine tasks performed with mechanical control at the individual level. Any group activity at this level is basically considered as a collection of independent activities.
- Rule-based performance – Concerning tasks accomplished with some latitude of decision from humans but within the constraints of a specific plan imposed by the technology. Unlike the previous level, the group activities are perceived as a collection of coordinated activities.
- Knowledge-based performance – Concerning interdependent tasks performed by humans in the scope of group and organizational goals.

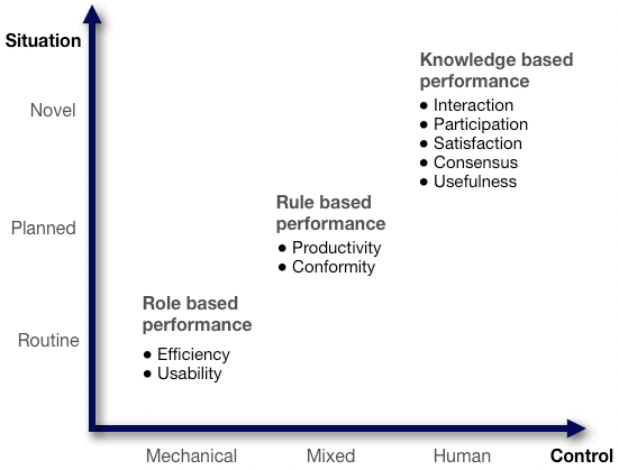

Figure 3. Performance levels, adapted from Reason [2008]

This model highlights the increasing sophistication of human activity, where simple (from the perspective of the organization) individual roles are complemented with more complex coordinated activities and supplemented by even more complex knowledge-based and information-rich activities. The group becomes more important than the individual. We will use this model to delineate three distinct collaborative systems evaluation scenarios.

4.3. Evaluation scenarios

Our evaluation scenarios follow the three-layer view previously mentioned:

Role-based scenario – The evaluation data is gathered at the individuals' cognitive level, focusing on events occurring during a time frame in the order of minutes or even seconds. The most adequate evaluation methods to employ in this scenario adopt laboratory settings and considerable instrumentation (e.g., key logging). To gather the data, the evaluators must accurately specify the roles and activities; and the subjects must exactly act according to the instructions and under strict mechanical control. In these circumstances the system detail is high (e.g., keystrokes and mouse movements) but the system scope is low (e.g., roles associated to some particular functions). This scenario also trades off realism towards higher precision and generalizability. The time invested in this type of evaluation tends to be low and mostly used in the preparation of the experiment. The various trade-offs associated to this evaluation scenario are illustrated in Figure 4.

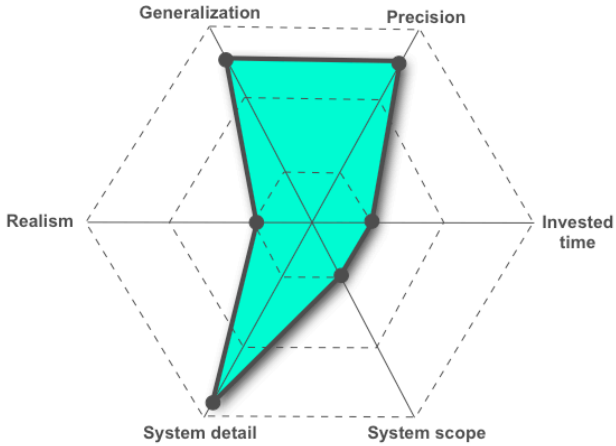

Figure 4. Role-based evaluation

Rule-based scenario – The evaluation data now concerns several subjects who must coordinate themselves to accomplish a set of tasks. The relevant events now occur over several minutes and hours, instead of minutes or less. The system details being considered have large granularity (e.g., exchanged messages instead of keystrokes). The system scope also increases to include more functions. The evaluation methods employed in this scenario may still adopt laboratory settings although using less instrumentation. This scenario also represents trading off realism in favor of precision and generalizability. As with the role-based scenario, the evaluators must plan the subjects' activities in advance; however, the subjects should be given more autonomy since control concerns the coordination level and not individual actions. The time invested in this type of evaluation is higher than the previous case, since the data gathering takes more time and the data analysis is less straightforward (e.g., requiring debriefing by the participants). The trade-offs associated with this evaluation scenario are illustrated in Figure 5.

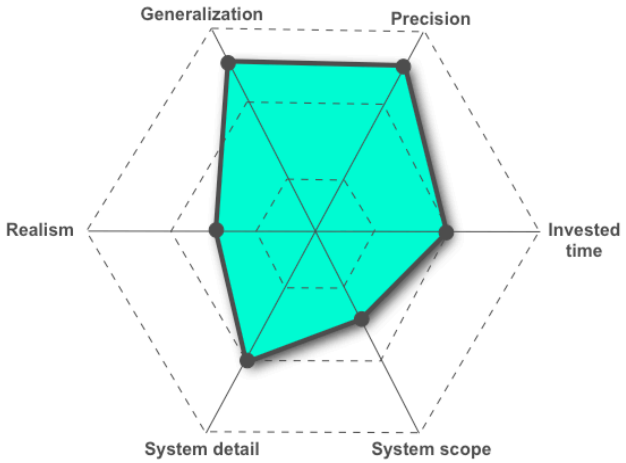

Figure 5. Rule-based scenario

Knowledge-based scenario – The evaluation is mostly focused on the organizational impact and thus concerns much longer time frames, usually on the order of days, months and even years, since the technology assimilation and the perception of value to the organization may take a long time to emerge and stabilize. The evaluation scenario is also considerably different when compared to the other scenarios, involving for instance knowledge management, creativity and decision-making abilities. Considering these main goals, it is understandable the system detail has coarse granularity, favoring broad issues such as perceived utility or value to business. The system scope may be wider for exactly the same reason. The evaluators may not specify the roles and activities beforehand in this case, since the subjects have significant latitude for decision, which leads to open situations beyond the control of the evaluators. Considering the focus on knowledge, the trade-off is usually to reduce the precision and generalizability in favor of realism. All these differences imply the laboratory setting is not the most appropriate for the knowledge-based scenario, in favor of more qualitative settings. Two examples of evaluation methods employed in this scenario are case studies and ethnographic studies. These techniques need significant time to gather the data in the field, and also time to transcribe, code and analyze the obtained data. The trade-offs associated to this evaluation scenario are illustrated in Figure 6.

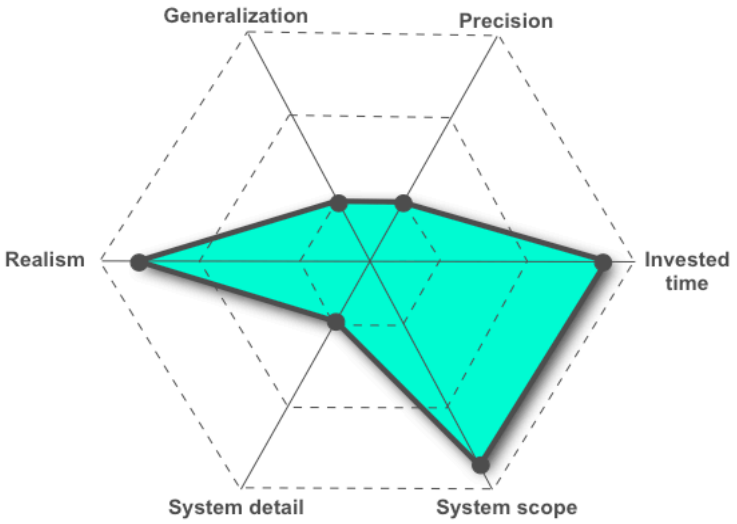

Figure 6. Knowledge-based scenario

4.4. Discussion

The analysis of the scenarios described above highlights interesting issues to ponder when taking into consideration a collaborative system evaluation. Regarding the ensemble of variables, the rule-based scenario seems to be the most balanced in the adopted trade-offs. On the contrary, the role-based and knowledge-based scenarios show a clear tendency for the extremes. The role-based scenario emphasizes detail, precision, generalization and time at the cost of scope and realism. On the opposite side, the knowledge-based scenario shows a clear emphasis in scope and realism at the expense of detail, precision, generalization and time. These differences highlight the so-called

instrumentalist and inter-subjectivist strategies, which have been quite influential in the CSCW field [Pidd 1996; Neale et al. 2004; Guy 2005]. The instrumentalist strategy is mostly focused on accumulating knowledge through experimentation, whereas the inter-subjectivist strategy is concerned with interpreting the influences of the technology on the individuals, groups and the organization.

The analysis of some individual variables may also give additional insights about the collaborative system evaluation. One such variable is invested time, which is distinct for the three discussed scenarios. From a very pragmatic perspective, the selection of the evaluation scenarios could be based on the time one is willing to invest on the evaluation process. Such considerations would lead to a preference for the rule-based and role-based scenarios and a devaluation of the knowledge-based scenarios. Nevertheless, this approach may not be feasible due to lack of system detail, e.g. whenever evaluating design ideas. This approach also has some negative implications, such as emphasizing details of little importance to the organization.

System detail and scope are also related with the strategy adopted to develop the system. For instance, a breadth-first strategy indicates a strong initial focus on broad functionality, which would mandate an evaluation starting with a knowledge-based scenario and later on continuing with role-based and rule-based scenarios. On the contrary, a depth-first strategy indicates a strong preference for fully developing a small functional set, which would mandate an evaluation starting with a role-based scenario and proceeding with rule-based and knowledge-based scenarios.

Figure 7 provides an overview of these evaluation issues. The two dotted lines show the limits suggested by the three evaluation scenarios. The arrows show the possible directions of the evaluation strategy and their basic assumptions.

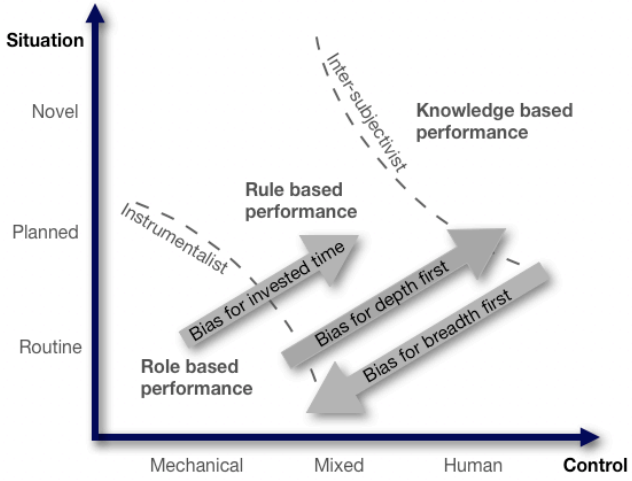

Figure 7. Evaluation lifecycle

The arrows in Figure 7 indicate possible evaluation processes adopted according to various biases. The arrow's starting point indicates which type of evaluation should be done first, while the end point suggests where to finish the evaluation process.

This graphical representation also affords equating the collaborative system evaluation on other dimensions. For instance, the specific control and situation characteristics of one particular system may determine the effort involved in evaluation. Consider a database under evaluation that only supports mechanical control. Then, we

may reckon the low dotted line shown on Figure 7 corresponds to the most adequate evaluation. An instrumentalist strategy should be adopted, assessing for instance the database usability. In the case of a workflow system, where control is mixed between the system and the users, we may consider the evaluation should be extended beyond the instrumentalist strategy, e.g. contemplating the conformity of the system with organizational procedures and rules.

5. EVALUATION GUIDELINES

This section presents a set of guidelines about the techniques and instruments used in collaborative systems evaluation. Figure 8 shows the evaluation methods which were presented in Section 3.2, organized considering the role, rule and knowledge-based categories.

1. The *knowledge-based evaluation* emphasizes variables pertaining more to the organization and group than to the individual performance. Examples of metrics which can be delivered by these methods include interaction, participation, satisfaction, consensus, usefulness, and cost reductions.
2. On the contrary, the *role-based evaluation* stresses the importance of the individual performance. Metrics that can be obtained using these methods are efficiency and usability.
3. The *rule-based evaluation* may be seen as being in the middle of the extremes. Some metrics may include the organizational goals, e.g. conformance to regulations, while others may concern group performance, such as productivity.

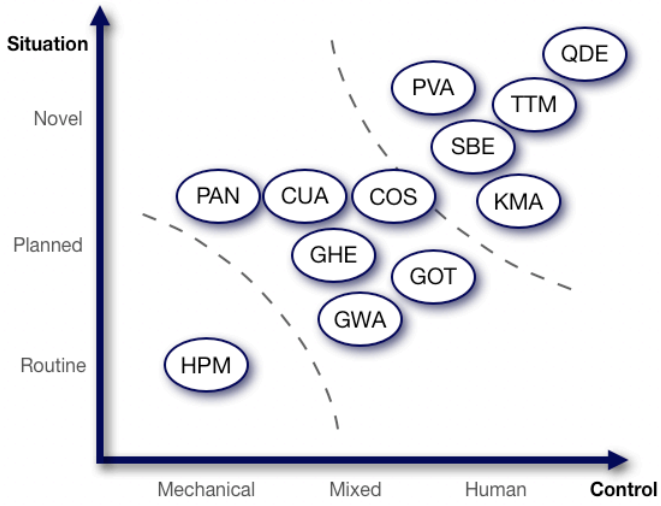

Figure 8. Classification of evaluation methods

Some methods may belong to one or two categories depending on the nature of the instruments involved in each method, e.g. *cooperation scenarios* (COS) are located in the area between rule and knowledge-based evaluation methods, because it has elements belonging to both categories. This classification allows evaluators to choose an appropriate method for their particular evaluation scenario.

We have also developed guidelines to select an evaluation method that depends on the development status of the product being assessed. We consider which of the following stages the product is in: *conception* (during analysis and design), *implementation* (during coding and software refinement), *production* (the product is already being used), *reengineering* (the product is being structurally redesigned), or *procurement* (the product is going to be acquired by the organization). Figure 9 presents a summary of these guidelines.

The rationale behind these recommendations is closely related to evaluation activities embedded in a typical software process. Validating the proposal of a collaborative system is mandatory during product conception or implementation phases. This validation typically involves a knowledge-based method intended to assess product usefulness for the organization. Further evaluation is usually justified if the results from this initial assessment are satisfactory, but the product requires some improvements. Following the same line of reasoning, rule-based methods should be applied before the role-based ones. If we want to evaluate already-implemented products (i.e. products in production, reengineering or procurement stage), the most suitable evaluation method will depend on what triggered the evaluation process; e.g. refinement, redesign or acquisition of a product. All guidelines and the rationale behind each are described below.

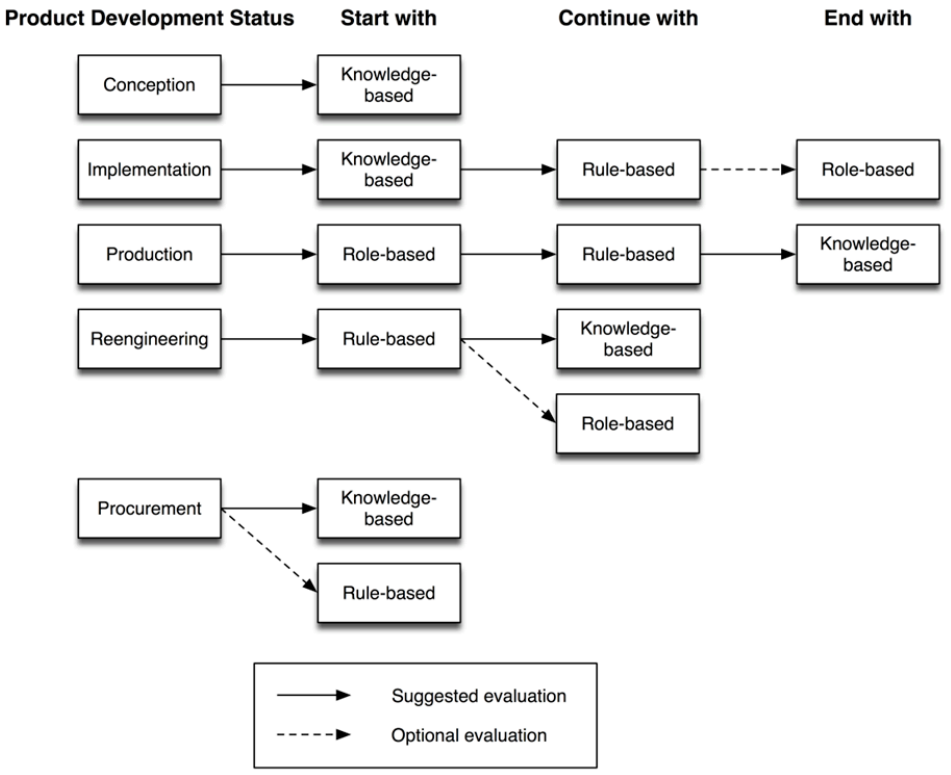

Figure 9. Summary of guidelines for selection of evaluation methods

1. If the product to be evaluated is in the *conception stage*, then the evaluation should be oriented towards obtaining coarse-grain information to help understand the role of the tool within the organization, the users' expectations and needs, the business case and the work context. This information, usually obtained from knowledge-based evaluation methods, may be very useful to specify or refine the user and software requirements, to establish the system scope, to identify product/business risks and to validate a product design. Performing this assessment, evaluators should adopt an inter-subjectivist view over the collected data, considering qualitative and interactive ways to obtain data, using various activities such as field studies, focus groups, and meetings. Rule-based and role-based methods do not provide a clear benefit in this stage because they require, at least, having a prototype of the system.

2. If the product is in the *implementation stage*, a knowledge-based evaluation method is recommended, because it will serve to understand if the product can address organizational goals. This evaluation also provides coarse-grain information concerning the issues/components requiring improvements. This type of evaluation is optional if the product was already evaluated with a knowledge-based method during the conception phase. However, the implemented product could differ from its design; therefore assuming the implemented system is still aligned with the organization needs and users' expectations could be a mistake.

   When the available time and budget allow additional evaluation actions, the process may be complemented with a rule-based evaluation method, since that will provide information necessary to adjust the product to the actual working scenario. For example, adjustments to concrete business processes may be identified this way. Optionally, a role-based evaluation method could also be used to fine-tune the product to the users. In case of rule/role-based assessments, the evaluation setting may be configured to assess the users' activities in a controlled or mixed environment, which may utilize laboratory settings. The evaluators may also adopt a more experimental view of the collected data.

3. An evaluation may also serve to determine the current impact a *system in production* has on its business operations. Therefore, the first recommended evaluation action considers diagnosing the current situation using precise information obtained from the actual production system. A role-based evaluation method may then be used to gather such information.

   As with the previous case, if the available time and budget allow additional evaluation actions, rule-based and knowledge-based methods could subsequently be applied. The aim could be identifying concrete performance issues and improving organizational behavior. Rule-based methods will provide performance diagnose information and the knowledge-based methods will contribute to identify the impact of the legacy system at an organizational level.

4. Many organizations often decide to *reengineer a legacy system*. The main purpose is to change the organizational behavior by extending the system support. The existing system may be used to guide this reengineering. In such a case it is recommended to start with a rule-based evaluation to avoid anchoring the evaluation on too fine-grained or too coarse-grained information. This type of evaluation helps identify particular improvement areas, which should be

addressed in the reengineering process. Nevertheless, a subsequent knowledge-based evaluation may determine the impact of the reengineered product on the organizational strategy. If the reengineering process involves significant changes to the systems' functions, user interfaces or interaction paradigms, a role-based evaluation may also be recommended. It allows focusing the evaluation on particular components and also getting fine-grain and accurate information to perform the reengineering.

5. Often an evaluation action occurs when *procuring a product*. In such cases, the evaluation should start with a knowledge-based method, in order to understand if the system functionality matches the organizational needs. Eventually, if the evaluators must also assess the system support of the organizational context and specific business processes, then the recommendation is to perform a rule-based evaluation. That process will allow identifying strengths and weaknesses of the product as support of particular activities in the organization.

Besides the generic recommendations mentioned above, the evaluators should also ponder the specific characteristics of the product under evaluation, namely the control and situation dimensions, which have impact on the evaluation scenario. The knowledge-based evaluation is naturally most adequate to products giving latitude of decision to the users and supporting interaction, collaboration and decision-making.

The evaluators should also ponder risk analysis. The risk adverse evaluator will set up a complete evaluation process, considering a combination of the three evaluation types, starting with the knowledge-based and finishing with the role-based scenarios. The risk taker evaluator will probably concentrate the evaluation only on the knowledge-based issues. The payoff of this high-risk approach is streamlining the evaluation efforts while focusing upon the issues that may have highest impact on the organization. The associated risk is the potential lack of quality of the outcomes.

6. THE COLLABORATIVE SYSTEM INCREMENTAL EVALUATION PROCESS

This section describes two case studies of collaborative systems evaluation. The first one involves the evaluation of a requirements inspection tool for a governmental agency [Antunes et al. 2006; Ferreira et al. 2009]. The second one shows the evaluation processes of a mobile shared workspace supporting construction inspection activities for a private construction company [Ochoa et al. 2008].

6.1. Evaluation of a collaborative software requirements inspection tool

Software requirements inspection is a well-known software engineering task. It engages a group of reviewers in the process of evaluating how well a software product under development accomplishes a set of previously established requirements. In a very simplified view, the tool under evaluation requests a group of software reviewers to synchronously complete a matrix with their perceived correlations between software requirements and specifications (from totally irrelevant to highly relevant). This matrix allows the reviewers to identify areas where software development has been underachieving and also to define priorities for further developing technical specifications.

This tool has been subject to two formal evaluation procedures, the first one being a knowledge-based evaluation and the second one being a role-based evaluation. The next

sections briefly describe the two procedures and the example ends with some comments about the overall evaluation process.

6.1.1. Knowledge-based evaluation

From a goal-oriented perspective, the major goal to achieve with this tool is a matrix of correlations expressing the reviewers' perspectives, expectations and worries about the software under development. The selection of correlations is necessarily a qualitative task, where the reviewers must agree upon the most appropriate link between "what" is being implemented and "how" the implementation corresponds to the reviewers' expectations. This task is naturally complex because there are several reviewers involved who may have different perspectives about the software application, interpretations of what is involved in application development, hidden agendas, etc. The tool supports the negotiation and reconciliation of these conflicting views.

Taking these problems into consideration, the initial evaluation step was focused on assessing the value brought by the tool to the evaluators, not only on assessing the software development but also on resolving their conflicting views in a productive and satisfactory way. This initial evaluation step was therefore focused on knowledge-based issues. The adopted evaluation method was based on Cooperation Scenarios (COS) [Stiemerling and Cremers 1998] using scenario-based workshops to elicit design flaws.

The evaluation procedure was set up as follows. The tool was evaluated in two pilot experiments involving two reviewers each. All of the reviewers were knowledgeable in software development, project management, requirements negotiation with outsourcing organizations and software analysis and design.

The pilot experiments were accomplished in the reviewers' workplace, which was a governmental agency responsible for the national pension system. The participants' task was to assess a project concerning the introduction of a new formula for computing pensions in the future. The specific goal set for the pilot experiments was to construct a matrix correlating a list of user requirements with a list of technical requirements, so that priorities could be set early in the project. The lists of user and technical requirements were specified at the beginning of the pilot experiments with help and approval from one of the most experienced participants. The evaluation itself was thus focused on negotiating and completing the correlations. The matrix under evaluation had 8x24 = 192 potential correlations to evaluate.

Each pilot experiment started with a brief tutorial about the tool, which took approximately 15 minutes. Then, a pair of reviewers used the tool until a consensus was obtained. During the experiment, whenever necessary, additional help about the tool was provided to the reviewers.

Afterwards, we asked the reviewers to complete a questionnaire with open questions about the tool's most positive and negative aspects, as well as closed questions concerning the tool's functionality and usability (Table 2). Regarding functional issues, the obtained results indicate the tool was convenient to use and accurate concerning the evaluators' view of the project. We also obtained positive indications about the consensus mechanism built into the tool, the reviewers' understanding of the overall positions from others, ease of finding agreements and simplicity revising their own opinions. Additionally, the reviewers agreed that the outcomes reflected their own opinions.

Concerning usability issues, the obtained results indicated that the participants could understand the working logic behind the tool and that they easily learned to deal with its functionality, as well as with the negotiation process. However, it was pointed out that

the tool was difficult to use by inexperienced users. Another drawback was related with bad performance, since the tool spent too much time synchronizing data. Several other minor functional and user interface details were also raised by the participants, e.g., the absence of graphical information and the difficulties obtaining a summary view of the negotiation.

Table 2. Results from the questionnaire

	Scores				
	1 (<)	2	3	4	5 (>)
Functionality					
Convenience (available functions and their appropriateness)				4	
Accuracy (reflecting the users opinions)			2	2	
Agreement (with the inspection method)				3	1
Usability					
Comprehension (understanding the tool)			2	1	1
Learning (how to use the tool)			2	1	1
Operability (effort controlling the inspection)			2	1	1

Another interesting outcome from this evaluation was the evidence of learning that the tool provided. Participants obtained new insights about negotiating software requirements. These two pilot experiments thus gave very rich indications about the value of this tool to the organization and to the group, as well as potential areas for improving the tool. The adopted evaluation approach also proved adequate to elicit knowledge-based design flaws and come up with design recommendations.

6.1.2. Role-based evaluation

The second formal evaluation procedure was aimed at evaluating in detail the user interaction with the tool. It was therefore a role-based evaluation.

The user interaction with the tool was centered on the notion of shared workspace. Shared workspaces are becoming ubiquitous, allowing users to share information and to organize activities in very flexible and dynamic ways, usually relying on a simple graphical metaphor. This evaluation procedure thus aimed at optimizing the shared workspace use, assuming such optimization would increase the evaluators' already positive opinion about the tool.

The adopted evaluation approach was to analytically devise different options for shared workspace use and predict their performance. The method applied well-known human information processing models to measure the shared workspace performance and to draw conclusions about the several design options. The adopted model was the Keystroke-Level Model (KLM) [Card et al. 1980]. KLM is relatively simple to use and has been successfully applied to evaluate single-user applications, although it had to be adapted to the collaborative systems context for this evaluation [Ferreira et al. 2009].

Based on KLM, each user interaction may be converted into a sequence of mental and motor operators, whose individual execution times have been empirically established and validated by psychological experiments. This way we could find out which sequence of operators would minimize the execution time of a particular shared workspace implementation.

We modeled three low-level functions associated with the shared workspace usage: locating correlations, selecting correlations and negotiating correlation values. Several alternative designs for these functionalities were analytically evaluated. The adopted approach offered a common criterion, based on execution time, to compare the various

implementations and find out which implementation would offer the best performance. In Table 3 we show the obtained results, highlighting that Design A has better overall performance than Design B.

Table 3. Results of KLM evaluation of shared workspace

Design conditions	Design A	Design B
a) 3 users	5 s. MMPKKPKK	9.8 s.
a.1) no scroll (75% probability)		5 s. MMPKKPKK
a.2) scroll (25% probability)		11.3 s. MPKKMPKKPKKMMPKKPKK
b) 6 users	8.6 s. MPKPKMMPKKPKK	9.8 s.
b.1) no scroll (75% probability)		5 s. MMPKKPKK
b.2) scroll (25% probability)		11.3 s. MPKKMPKKPKKMMPKKPKK

(Operators: M-Mental; P-Pointer; K-Key)

6.1.3. Discussion

Overall, these evaluations allowed us to obtain several insights about the tool. The initial experiments were mostly focused on broader organizational and group issues, such as positive/negative effects, convenience and respect for the participants' opinions. Although the obtained results were characterized by low precision and generalizability, they were very insightful for further development and contributed to perceptions of the value attributed to the tool by the organization. The final experiments addressed fine-grained details about the tool usage and allowed us to experiment with alternative functionality and ultimately adopt the functionality that would offer the best performance. These latter results were characterized by high precision and generalizability, although they had low realism.

In both cases the time invested in the evaluation was low, due to different causes. In the first case it was low because we adopted a pilot study approach; in the latter case, it was low because we adopted an analytic approach. The system detail was quite different between the two evaluations. In the first case it was very low (positive/negative aspects), while in the second case it was very high (keystrokes). Conversely, in the first case the system scope was high (whole application) and in the second case was very low (few functions).

6.2. Evaluation of an application to support construction inspection activities

Construction projects typically involve a main contractor, which in turn outsources several parts of the whole project, e.g. electrical facilities, gas/water/communication networks, painting and architecture. The companies in charge of these activities usually work concurrently and they need to be coordinated because the work they are doing is highly interrelated. In fact, the project progress rate and the product quality increase when all these actors appropriately coordinate among themselves.

The main contractor is usually a manager responsible for the coordination process. The inspection activities play a key role in this process. The goal of these activities is to

diagnose the status of the construction project elements and to determine the need to approve, reject or modify the built elements based on the diagnosis. Each inspection is carried out by one or more inspectors using paper-based blueprints. These persons work alone (doing independent tasks) or by forming an inspection team (when their examinations are interrelated). The inspection process requires that these persons be on the move and record the contingency issues (problems identified by one inspector) related with particular components of the project.

Periodically, the main contractor informs the subcontracted companies about the list of contingency issues they have to address. The process required to deal with these issues may involve the work of more than one subcontractor, and of course at least one additional inspection.

In order to support the inspection activities and help coordinate the problem solving process, a mobile shared workspace named COIN (COnstruction INspector), was developed. This collaborative system manages construction projects composed of sets of digital blueprints, which are able to store annotations done with a stylus on a Tablet PC. The system also supports mobile collaboration among the users, and sharing data (file transfer and data synchronization) between two mobile computing devices.

Two types of evaluations were applied to this tool: knowledge and rule-based evaluations. The following two sections describe the evaluation processes; a third section presents a discussion of the obtained results.

6.2.1. Knowledge-based evaluation

During the first stage of the project, a Scenario-Based Evaluation (SBE) strategy [Haynes et al. 2004] was used to identify the scenarios and requirements involved in the construction inspection process. Two formal evaluations were done using this strategy; the first one during the software conception phase and the second one during the design phase. Each one involved two steps: (1) individual interviews to construction inspectors and (2) a focus group to validate the interview results.

Three experienced construction inspectors participated in the evaluation done during the conception phase. Each interview was about one hour long. The participants had to characterize the work scenarios to be supported, and also specify and prioritize the functionalities which are required to carry out the inspection process. The results showed consensus on the types and features of the scenarios to be supported by the tool. However, there was no consensus on the functionalities the tool should provide to the inspectors. After the interviews, the results were written and given back to the inspectors.

A week after that, a focus group was performed in order to try to get an agreed set of functionalities to implement in the software tool. The focus group was about three hours long and most of the participants changed their perception about which functionalities were most relevant to support the collaborative inspection process. A consensus was obtained after that session. The most important functionalities related with collaboration were the following: (1) transparent communication among inspectors, (2) selective visualization of digital annotations, (3) annotation filtering by several criteria, (4) unattended and on-demand annotations synchronization (between two inspectors), and (5) awareness of users' availability and location.

During the COIN design process, a preliminary prototype was used to validate the development team's proposals to deal with the requirements identified in the previous phase. Once again, a SBE strategy was used. Before the individual interviews, the inspectors received a training session lasting about 30 minutes. After that, each one explored the prototype features for about 45 minutes. Finally, a one hour interview was

done with each inspector. The main goal of the interview was to identify positive, negative and missing issues on the tool, and determine if the functionalities included in the prototype were enough to support a collaborative inspection process. The results showed a long list of specific and detailed comments with some kind of matching among the inspectors' opinions. Similar to the previous evaluation process, these issues were written and given back to the participants.

After a week we did the focus group session, where the COIN prototype was reviewed again and also the inspectors' comments were reviewed. The session main goal was to categorize the inspectors' comments in the following three categories: *critical* (it must be included in the tool), *recommended* (it is a good idea to include it) and *optional* (it could be included if there is enough time). The focus group took about three and a half hours, and identified 12 critical, 17 recommended and 8 optional issues. The developers were in the session (as observers) to get the requirements directly from the source.

The effort to carry out the second evaluation was at least double the first one. However, the result was highly accurate, detailed and valuable, which allowed us to adjust the proposed components in order to deal with the inspectors' comments. The development team members recognized these comments were a key piece to improve the matching between the prototype functionality and the inspectors' needs. However, it is important to acknowledge the opinions of three inspectors are not enough to determine the inspection requirements of a construction company. A larger number of participants implies not only more general and validated results, but also a larger evaluation effort.

6.2.2. Rule-based evaluation

Once the first version of COIN was delivered, an empirical evaluation experience was conducted at the Computer Science Department of the University of Chile using the tool. A variant of Cooperation Scenarios (COS) evaluation method [Stiemerling and Cremers 1998] was used in this case. The experience involved an area of 2000 m^2 approximately, deployed in two floors. These areas included mainly offices, meeting rooms, laboratories and public spaces. Forty labels simulating contingency issues were adhered to the physical infrastructure and electrical facilities.

Two civil engineers, who participated in the previous evaluation process, conducted the reviewing process. In turn they first used COIN running on a Tablet PC to carry out the inspection, and then they repeated the process using physical blueprints. In both cases an observer followed the activities of each inspector in order to verify the coherence between the inspectors' opinions and the empirical observation. In addition, these observers recorded the time involved in particular tasks of the inspection process.

The engineers agreed beforehand on a common strategy to conduct both inspections processes. The strategy consisted of performing two tasks in sequence: (1) to gather the contingency issues and (2) to determine the coherence between the inspectors' annotations.

During the first evaluation round, the inspectors identified the contingency issues and created the corresponding annotations using COIN. Subsequently, they met to review each annotation, and they decided the reviews were consistent. Afterwards, the labels simulating contingency issues were changed and relocated trying to reproduce the experimental conditions of the first experience. The inspection process was repeated, but now using blueprints.

Finally, the engineers were interviewed in order to assess their feeling about the use of the tool to support inspection processes. The observers provided information about the duration of several activities involved in the inspection process, such as the contingency

gathering and the integration of annotations. The idea was to establish a sound parameter to compare the whole process of inspection with/without COIN usage.

Table 4 shows the results of the inspections using the tool and blueprints respectively. These results indicate an improvement of the elapsed times when COIN is used. During the interview with the inspectors both indicated they preferred to use the collaborative application because: (1) digital maps are easier to use than paper-based blueprints, (2) writing annotations on the screen of a tablet PC is more comfortable than writing them on a blueprint placed on a wall, (3) the user mobility improves when COIN is used, and (4) reviewing annotations is faster when using the tool because both tablet PCs can be put together, and thus the distance between annotations to be compared is small (it eases the process).

Table 4. Results of the inspection process

Experience	Labels Found	Inspection - Elapsed Time	Annotations Review - Elapsed Time	Total Elapsed Time
With COIN	37	23 minutes	6 minutes	29 minutes
Without COIN	38	35 minutes	9 minutes	44 minutes

Although the use of COIN shows positive results, they do not represent a great improvement to the current inspection process. The most important advantages of using COIN are related with the coordination process. In that sense, Table 5 shows several interesting improvements in terms of coordination activities. For example, digital blueprints can be retrieved from the main contractor's server through a Web service, which is accessed via Ethernet or a cellular network, when COIN is used. This operation involved less than 2 minutes and avoided the trip to the main contractor's office required in the paper-based case.

Moreover, the process to integrate the annotations done by the inspectors involved less than a minute when the collaborative system was utilized. By contrast, the integration could have taken about one hour for paper-based inspection. The time to report the annotations to the main contractor is also considerably reduced with the system use.

Table 5. Results of coordination activities

Experience	Time for Retrieving Blueprints	Time to Integrate Annotations	Time for Reporting Annotations	Tasks Creation - Elapsed Time	Contingencies Report - Creation Time
With COIN	< 2 minutes	< 1 minutes	< 2 minutes	35 minutes	< 2 minutes
Without COIN	Go to the main contractor's office	1 hour (*)	Go to the main contractor's office	40 minutes (*)	1 - 2 hours (*)

(*): Estimations done by the inspectors.

The time spent to create the tasks related with the annotations is similar in both cases. However the creation of the contingencies report is considerably reduced when COIN is used.

This evaluation process gives us useful preliminary information to understand the possible impact of the tool in the construction inspection scenario. However, a large number of observations are required to get a more accurate diagnosis about the impact of the tool on a real construction company.

6.2.3. Discussion

In the first two evaluations (i.e. when SBE was used) just three inspectors were involved because of the effort required to carry out these evaluations. The evaluation effort (mainly time) in SBE grows considerably with each additional participant. Clearly this is a method which provides a high degree of realism when it is applied to a large number of participants. However it also requires long invested time. The reward for that work is an agreed set of specific and detailed (positive, negative and missing) issues, which must be considered during the development of a collaborative supporting tool. Counting on these issues is highly important to determine how well the product under development matches the users' needs.

The second evaluation process (i.e. when COS was used) provides an interesting strategy to obtain a diagnosis of the tool usefulness, and its impact on the process in which it is utilized. The feedback is detailed and precise; however it requires a large number of participants to generalize the results. This implies an increase in the evaluation effort. Such effort could be reduced using agents running in the background and recording the time involved in the several tasks. Thus, the observer would no longer be required.

Finally, the main limitation of the COS evaluation method could be the realism level of the obtained results. If the testing scenario (laboratory) is similar to the real scenario, then the results will be representative. Otherwise, the evaluation effort could be meaningless. If COS is going to be used for an evaluation process, then it is important to consider the cost of having a testing scenario similar to the real one.

7. CONCLUSIONS AND FUTURE WORK

The second section of this paper starts with a tough question: why is collaborative systems evaluation so difficult? As we have thoroughly discussed, there is no single culprit. Indeed, the difficulties are practical (e.g., dealing with many subjects and groups), theoretical (addressing different cognitive levels, specifying satisfying criteria) and methodological (e.g., dependence of the evaluation on the development process).

The various evaluation methods reviewed in this paper and the timeline showing their emergence corroborate the complexities. Many of these methods are not competing for the same goal, but instead they complement the whole framework necessary to evaluate collaborative systems.

Of course, then, the task of the evaluator is to define the necessary trade-offs and select a set of satisfactory evaluation methods. This paper tries to ease this task.

To accomplish this goal, we started by identifying the set of variables which may be necessary to build a comprehensive evaluation framework. Such a framework must deliver a balanced albeit concise combination of variables addressing the practical, theoretical and methodological issues that make collaborative systems evaluation so difficult. We defined six variables: generalization, precision, realism, system detail, system scope and invested time.

The generalization, precision and realism variables fundamentally concern theoretical issues regarding how satisfying the evaluation results may be to the evaluator. The system detail and system scope concern methodological issues associated to the product development strategy. The invested time variable concerns a very practical issue, which is assessing the amount of time available to the evaluator in order to conduct the evaluation.

Yet these six variables still constitute a quite complex evaluation framework. We must ease the evaluator's decision-making task. Thus, we have also considered three performance levels: role-based, rule-based and knowledge-based performance. These levels of performance lay out the relative importance attributed to each one of the six variables previously described. For instance, the role-based level assigns high importance to the generalization, precision and system detail variables and low importance to realism, invested time and system scope.

Overall, the performance levels define three distinct evaluation scenarios aiming to reduce the number of choices considered by the evaluator without significantly compromising the comprehensiveness of the evaluation process. Given the evaluation scenarios, we then proceeded with the discussion of which evaluation lifecycle, i.e. combination of scenarios, could be adopted by the evaluator. The discussion is essentially based on two criteria: bias for invested time and product development criteria. Considering the bias for invested time, the issue is to recommend the evaluation lifecycle and corresponding scenarios that are cost-effective with respect to the time spent doing the evaluation. On the other hand, the product development criterion is concerned with aligning the evaluation with the development cycle, which may have adopted a depth first or breadth first approach.

Thus this approach leads the evaluator towards a fairly straightforward decision-making process considering the product being developed, the development lifecycle and the time available to evaluate the product.

Finally we also relate the existing evaluation methods to the evaluation scenarios mentioned above, thus easing the definition of the concrete evaluation plan. The paper also describes two case studies illustrating the use of the evaluation framework and showing how the three evaluation scenarios complement each other towards assessing prototypes at various levels of granularity.

The main contributions of this paper are twofold. The most important one is offering decision-making support to evaluators wishing to disentangle the inherent complexity of collaborative systems evaluation. The proposed approach covers the whole endeavor ranging from the selection of evaluation variables, definition of satisfying criteria and adoption of an evaluation lifecycle. The second contribution is laying out a foundation for classifying evaluation methods. The evaluation methods seem to emerge in a very ad-hoc way and cover quite distinct goals regarding why, how, what and when to evaluate. This situation makes it difficult to classify them in a comprehensive way. We have proposed a classification highlighting their major distinctions. We hope this classification will be helpful to future research and practice in the CSCW area.

ACKNOWLEDGEMENTS

This paper was partially supported by the Portuguese Foundation for Science and Technology (PTDC/EIA/102875/2008), Conicyt PhD scholarship, Fondecyt (Chile) Grants Nº 11060467 and 1080352, and LACCIR Project No. R0308LAC004.

REFERENCES

ANTUNES, P. and COSTA, C. Perceived value: A low-cost approach to evaluate meetingware. Proceedings of CRIWG'03, Autrans, France, Lecture Notes in Computer Science 2806 (2003), 109-125.

ANTUNES, P., FERREIRA, A. and PINO, J. Analyzing Shared Workspace Design with Human-Performance Models. Proceedings of CRIWG'06, Medina del Campo, Spain, Lecture Notes in Computer Science 4154 (2006), 62-77.

ANTUNES, P., RAMIRES, J. and RESPÍCIO, A. Addressing the conflicting dimension of groupware: A case study in software requirements validation. Computing and Informatics 25, (2006), 523-546.

ARAUJO, R., SANTORO, F. and BORGES, M. The CSCW lab for groupware evaluation. Proceedings of CRIWG'02, La Serena, Chile, Lecture Notes in Computer Science 2440 (2002), 222-231.

BAECKER, R. M., GRUDIN, J., BUXTON, W. and GREENBERG, S., Eds. Human-computer interaction: toward the year 2000. San Francisco, CA, Morgan Kaufmann (1995).
BAEZA-YATES, R. and PINO, J. A first step to formally evaluate collaborative work. Proceedings of the ACM Int. Conference on Supporting GroupWork (GROUP '97), Phoenix, AZ (1997), 55-60.
BAEZA-YATES, R. and PINO, J. Towards Formal Evaluation of Collaborative Work and Its Application to Information Retrieval. Information Research 11, 4 (2006), paper 271.
BAKER, K., GREENBERG, S. and GUTWIN, C. Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration. Proceedings of the 8th IFIP international Conference on Engineering For Human-Computer interaction, Lecture Notes In Computer Science 2254 (2001), 123-140.
BAKER, K., GREENBERG, S. and GUTWIN, C. Empirical development of a heuristic evaluation methodology for shared workspace groupware. Proceedings of the 2002 ACM conference on Computer Supported Cooperative Work, New Orleans (2002), 96-105.
BIAS, R. Interface-Walkthroughs: efficient collaborative testing. IEEE Software 8, 5 (1991), 94-95.
BIAS, R. The pluralistic usability walkthrough: coordinated empathies. Usability inspection Methods. J. Nielsen and R. Mack. New York, John Wiley & Sons (1994), 63-76.
BRIGGS, R., ADKINS, M., MITTLEMAN, D., KRUSE, J., MILLER, S. and NUNAMAKER, J. A technology transition model derived from field investigation of GSS use aboard the U.S.S. CORONADO. Journal of Management Information Systems 15, 3 (1998), 151-195.
BRIGGS, R., QURESHI, S. and REINIG, B. Satisfaction Attainment Theory as a Model for Value Creation. The Thirty Seventh Annual Hawaii International Conference on Systems Sciences, IEEE Computer Society Press 1 (2004), 10013.
CARD, S., MORAN, T. and NEWELL, A. The keystroke-level model for user performance time with interactive systems. Communications of the ACM 23, 7 (1980), 396-410.
CARROLL, J. Making use: Scenario-based design of human-computer interactions. Cambridge, Massachusetts, The MIT Press (2000).
COCKTON, G. and WOOLRYCH, A. Sale must end: should discount methods be cleared off HCI's shelves? Interactions 9, 5 (2002), 13-18.
CONVERTINO, G., NEALE, D., HOBBY, L., CARROLL, J. and ROSSON, M. A laboratory method for studying activity awareness. Proceedings of the Third Nordic Conference on Human-Computer interaction, Tampere, Finland (2004), 313-322.
DAMIANOS, L., HIRSCHMAN, L., KOZIEROK, R., KURTZ, J., GREENBERG, A., WALLS, K., LASKOWSKI, S. and SCHOLTZ, J. Evaluation for collaborative systems. ACM Comput. Surv. 31, 2es (1999), 15.
DESANCTIS, G., SNYDER, J. and POOLE, M. The meaning of the interface: A functional and holistic evaluation of a meeting software system. Decision Support Systems 11, (1994), 319-335.
EREBACK, A. and HÖÖK, K. Using cognitive walkthrough for evaluating a CSCW application. Conference Companion on Human Factors in Computing Systems, Boston, Massachusetts (1994), 91-92.
FERREIRA, A., ANTUNES, P. and PINO, J. Evaluating Shared Workspace Performance using Human Information Processing Models. Information Research 14, 1 (2009), paper 388.
FJERMESTAD, J. and HILTZ, S. An assessment of group support systems experimental research: Methodology and results. Journal of Management Information Systems 15, 3 (1999), 7-149.
GREENBERG, S. and BUXTON, B. Usability evaluation considered harmful (some of the time). Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, Florence, Italy (2008), 111-120.
GUTWIN, C. and GREENBERG, S. The effects of workspace awareness support on the usability of real-time distributed groupware. ACM Transactions on Computer-Human Interaction 6, 3 (1999), 243-281.
GUTWIN, C. and GREENBERG, S. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. WETICE '00 (2000), 98–103.
GUTWIN, C., ROSEMAN, M. and GREENBERG, S. A usability study of awareness widgets in a shared workspace groupware system. Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work, Boston, Massachusetts (1996), 258-267.
GUY, E. "...real, concrete facts about what works...": integrating evaluation and design through patterns. Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work, Sanibel Island, Florida (2005), 99-108.
HAYNES, S., PURAO, S. and SKATTEBO, A. Situating evaluation in scenarios of use. Proceedings of the 2004 ACM conference on Computer supported cooperative work, Chicago, Illinois (2004), 92-101.
HERSKOVIC, V., PINO, J. A., OCHOA, S. F. and ANTUNES, P. Evaluation methods for groupware systems. Proceedings of CRIWG'07, Bariloche, Argentina, Lecture Notes in Computer Science 4715 (2007), 328-336.
HIX, D. and HARTSON, H. R. Developing user interfaces: ensuring usability through product and process. New York, NY, John Wiley & Sons, Inc. (1993).

HUANG, J. P. H. A conceptual framework for understanding collaborative systems evaluation. Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, Linköping, Sweden (2005), 215-220.

HUGHES, J., KING, V., RODDEN, T. and ANDERSEN, H. Moving out from the control room: ethnography in system design. Proceedings of the 1994 ACM conference on Computer supported cooperative work, Chapel Hill, North Carolina (1994a), 429-439.

HUGHES, J., SHARROCK, W., RODDEN, T., O'BRIEN, J., ROUNCEFIELD, M. and CALVEY, D. (1994b). Field Studies and CSCW. Lancaster, UK, Lancaster University.

HUMPHRIES, W., NEALE, D., MCCRICKARD, D. and CARROLL, J. Laboratory Simulation Methods for Studying Complex Collaborative Tasks. Proceedings of the Human Factors and Ergonomics Society 48th Annual Meeting (2004), 2451-2455.

INKPEN, K., MANDRYK, R., DIMICCO, J. and SCOTT, S. Methodology for Evaluating Collaboration Behaviour in Co-Located Environments. CSCW 2004 Workshop, Chicago, IL (2004).

KIERAS, D. and SANTORO, T. Computational GOMS modeling of a complex team task: Lessons learned. Proceedings of the SIGCHI conference on Human factors in computing systems, Vienna, Austria (2004), 97-104.

MCGRATH, J. Groups: Interaction and performance. Englewood Cliffs, NJ, Prentice-Hall (1984).

NEALE, D. and CARROLL, J. Multi-faceted evaluation for complex, distributed activities. Proceedings of the 1999 Conference on Computer Support For Collaborative Learning Palo Alto, CA (1999), 425-433.

NEALE, D., CARROLL, J. and ROSSON, M. Evaluating computer-supported cooperative work: models and frameworks. Proceedings of the 2004 ACM conference on Computer supported cooperative work, Chicago, Illinois (2004), 112-121.

NEWELL, A. Unified Theories of Cognition. Cambridge, Massachussetts, Harvard University Press (1990).

NIELSEN, J. Usability engineering at a discount. Designing and Using Human-Computer Interfaces and Knowledge Based Systems. G. Salvendy and M. Smith. Amsterdam, Elsevier Science Publishers (1989), 394-401.

NIELSEN, J. Usability inspection methods. Conference on Human Factors in Computing Systems, Boston, Massachusetts (1994), 413-414.

NIELSEN, J. and MOLICH, R. Heuristic evaluation of user interfaces. Proceedings of ACM CHI '90 Conference, Seattle, WA (1990), 249-256.

OCHOA, S., PINO, J., BRAVO, G., DUJOVNE, N. and NEYEM, A. Mobile Shared Workspaces to support construction inspection activities. Collaborative Decision Making: Perspectives and Challenges. P. Zarate, J. Belaud, G. Camileri and F. Ravat. Amsterdam, IOS Press (2008), 211-220.

PIDD, M. Tools for Thinking. Chichester, J. Wiley & Sons (1996).

PINELLE, D. and GUTWIN, C. A review of groupware evaluations. Proceedings of 9th IEEE WETICE Infrastructure for Collaborative Enterprises (2000), 86-91.

PINELLE, D. and GUTWIN, C. Groupware walkthrough: adding context to groupware usability evaluation. Proceedings of the SIGCHI conference on Human factors in computing systems, Minneapolis, Minnesota, USA (2002), 455-462.

PINELLE, D. and GUTWIN, C. Evaluating teamwork support in tabletop groupware applications using collaboration usability analysis. Personal and Ubiquitous Computing 12, 3 (2008), 237-254.

PINELLE, D., GUTWIN, C. and GREENBERG, S. Task Analysis for Groupware Usability Evaluation: Modeling Shared-Workspace Tasks with the Mechanics of Collaboration. ACM Transactions on Computer-Human Interaction 10, 4 (2003), 281-311.

PINSONNEAULT, A. and KRAEMER, K. The impact of technological support on groups: An assessment of the empirical research. Decision Support Systems 5, 3 (1989), 197-216.

PLOWMAN, L., ROGERS, Y. and RAMAGE, M. What are workplace studies for?. ECSCW'95: Fourth European Conference on Computer-Supported Cooperative Work, Stockholm, Sweden (1995), 309–324.

POLSON, P. G., LEWIS, C., RIEMAN, J. and WHARTON, C. Cognitive walkthroughs: A method for theory-based evaluation of user interfaces. International Journal of Man-Machine Studies 36, 5 (1992), 741-773.

RASMUSSEN, J. and JENSEN, A. Mental procedures in real-life tasks : a case-study of electronic trouble shooting. Ergonomics 17, (1974), 293-307.

REASON, J. The Human Contribution: Unsafe Acts, Accidents and Heroic Recoveries. Surrey, England, Ashgate (2008).

ROSS, S., RAMAGE, M. and ROGERS, Y. PETRA: participatory evaluation through redesign and analysis. Interacting with Computers 7, 4 (1995), 335-360.

ROWLEY, D. and RHOADES, D. The cognitive jogthrough: a fast-paced user interface evaluation procedure. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Monterey, California (1992), 389-395

RUHLEDER, K. and JORDAN, B. Video-Based Interaction Analysis (VBIA) in Distributed Settings: A Tool for Analyzing Multiple-Site, Technology-Supported Interactions. PDC 98 Proceedings of the Participatory Design conference, Seattle WA (1998), 195-196.

SCRIVEN, M. The Methodology of Evaluation. Perspectives of Curriculum Evaluation. R. Tyler, R. Gagne and M. Scriven. Chicago, Rand McNally (1967), 39-83.
SONNENWALD, D., MAGLAUGHLIN, K. and WHITTON, M. Using Innovation Diffusion Theory to Guide Collaboration Technology Evaluation: Work in Progress. Proceedings IEEE International Workshop on Enabling Technologies (2001), 114-119.
STEVES, M., MORSE, E., GUTWIN, C. and GREENBERG, S. A Comparison of Usage Evaluation and Inspection Methods for Assessing Groupware Usability. Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, Boulder, CO, USA (2001), 125-134.
STIEMERLING, O. and CREMERS, A. The use of cooperation scenarios in the design and evaluation of a CSCW system. IEEE Transactions on Software Engineering 24, 12 (1998), 1171-1181.
SUCHMAN, L. Plans and Situated Actions: The problem of human-machine communication. Cambridge, UK, Cambridge University Press (1987).
TANG, J. Findings from observational studies of collaborative work. International Journal of Man-Machine Studies 34, 2 (1991), 143-160.
TWIDALE, M., RANDALL, D. and BENTLEY, R. Situated evaluation for cooperative systems. Proceedings of the 1994 ACM conference on Computer supported cooperative work, Chapel Hill, North Carolina (1994), 441-452.
URQUIJO, S., SCRIVENER, S. and PALMEN, H. The Use of Breakdown Analysis in Synchronous CSCW System Design. Proceedings of the Third European Conference on Computer Supported Cooperative Work - ECSCW 93, Milano, Italy (1993), 289-302.
VAN DER VEER, G. Task Based GroupWare Design: Putting theory into practice Proceedings of the 2000 Symposium on Designing Interactive Systems, New York (2000), 326-337.
VAN DER VEER, G., LENTING, B. and BERGEVOET, B. Gta: Groupware task analysis - modeling complexity. Acta Psychologica 91, 3 (1996), 297–322.
VELD, M. A. A. H. I. T., ANDRIESSEN, J. H. E. and VERBURG, R. M. E-MAGINE: The Development of an Evaluation Method to Assess Groupware Applications. Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (2003), 153-158.
VIZCAÍNO, A., MARTINEZ, M., ARANDA, G. and PIATTINI, M. Evaluating collaborative applications from a knowledge management approach. 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05), Linköping, Sweden (2005), 221–225.
WHARTON, C., RIEMAN, J., LEWIS, C. and POLSON, P. The Cognitive Walkthrough Method: A Practictioner's Guide. Usability Inspection Methods. J. Nielsen and R. Mack. New York, John Wiley & Sons (1994), 105-140.
YOURDON, E. Structured Walkthroughs. New York, Yourdon Inc (1978).

Appendix A. Timeline of evaluation methods

1978	[Yourdon 1978]	Structured walkthroughs
1980	[Card et al. 1980]	Keystroke-Level Model (KLM)
1987	[Suchman 1987]	Ethnomethodological studies
1989	[Nielsen 1989]	Discount usability engineering
1990	[Nielsen and Molich 1990]	Heuristic evaluation
	[Wharton et al. 1994]	Cognitive walkthroughs
1991	[Tang 1991]	Observational studies
	[Bias 1991]	Interface walkthroughs
1992	[Polson et al. 1992]	Cognitive walkthroughs
	[Rowley and Rhoades 1992]	Cognitive jogthrough
1993	[Urquijo et al. 1993]	Breakdown analysis
1994	[Twidale et al. 1994]	Situated evaluation
	[Nielsen 1994]	Usability inspection
	[Nielsen 1994]	Heuristic evaluation
	[Ereback and Höök 1994]	Cognitive walkthrough
	[Bias 1994]	Pluralistic usability walkthrough
	[Hughes et al. 1994a]	Quick-and-dirty ethnography
	[Hughes et al. 1994b]	Evaluative ethnography
1995	[Plowman et al. 1995]	Workplace studies

1996	[Gutwin et al. 1996]	Usability studies
	[Van Der Veer et al. 1996]	Groupware task analysis
1997	[Baeza-Yates and Pino 1997]	Formal evaluation of collaborative work
1998	[Stiemerling and Cremers 1998]	Cooperation scenarios
	[Ruhleder and Jordan 1998]	Video-based interaction analysis
	[Briggs et al. 1998]	Technology Transition Model
1999	[Neale and Carroll 1999]	Multi-faceted evaluation for complex, distributed activities
	[Gutwin and Greenberg 1999]	Evaluation of workspace awareness
2000	[Gutwin and Greenberg 2000]	Mechanics of collaboration
	[Carroll 2000]	Scenario-based design
	[Van Der Veer 2000]	Task-based groupware design
2001	[Steves et al. 2001]	Usage evaluation
	[Baker et al. 2001]	Heuristic evaluation based on the mechanics of collaboration
	[Sonnenwald et al. 2001]	Innovation diffusion theory
2002	[Baker et al. 2002]	Groupware heuristic evaluation
	[Cockton and Woolrych 2002]	Discount methods
	[Pinelle and Gutwin 2002]	Groupware walkthrough
2003	[Pinelle et al. 2003]	Collaboration usability analysis
	[Antunes and Costa 2003]	Perceived value
2004	[Haynes et al. 2004]	Scenario-based evaluation
	[Convertino et al. 2004]	Activity awareness
	[Humphries et al. 2004]	Laboratory simulation methods
	[Inkpen et al. 2004]	Evaluating collaboration in co-located environments
	[Kieras and Santoro 2004]	Computational GOMS
	[Briggs et al. 2004]	Satisfaction Attainment Theory
2005	[Vizcaíno et al. 2005]	Knowledge management approach
2006	[Baeza-Yates and Pino 2006]	Performance analysis
	[Antunes et al. 2006]	Human performance models
2008	[Pinelle and Gutwin 2008]	Tabletop collaboration usability analysis