

COLLABORATION AND CONFLICT IN SOFTWARE REVIEW MEETINGS

GIOVANA B. R. LINHARES

*Graduate Program in Informatics (PPGI), Federal University of Rio de Janeiro,
Rio de Janeiro, RJ 20010-974, Brazil
giovana_linhares@hotmail.com*

MARCOS R. S. BORGES

*Graduate Program in Informatics (PPGI), Federal University of Rio de Janeiro,
Rio de Janeiro, RJ 20010-974, Brazil
mborges@nce.ufrj.br*

PEDRO ANTUNES

*Faculty of Sciences, University of Lisbon, Campo Grande,
Lisbon, 1749-016, Portugal
paa@di.fc.ul.pt*

Received (Day Month Year)
Revised (Day Month Year)
Communicated by (xxxxxxx)

This paper discusses the collaboration-conflict process: a binomial process mixing collaboration and conflict. We applied the collaboration-conflict process to Software Review Meetings, commonly adopted to verify the functional specification of software. We developed a groupware tool demonstrating the dynamics of the collaboration-conflict process in review meetings. We also provide results from an experiment with the tool in a software engineering firm. The results show that the collaboration-conflict process promotes argumentation and generates better reviews.

Keywords: Review Meetings, Negotiation, Collaboration-conflict Process.

1. Introduction

Software Review Meetings (SRM) are recommended quality assurance activities in software engineering.¹ SRM involve designers, developers and testers in the verification of software at various points in the product development lifecycle. They allow determining if a product is being developed with quality and consistency with the specifications, i.e. it supplies the right solution to the requirements specified by the client.

In spite of common corporate goals, the participants in SRM often develop conflicting perspectives, interpretations and positions regarding the product quality. This type of conflict justifies the collaboration-conflict

process: a process integrating conflict *management* in collaboration. We thus have, on the one hand, the review activity that has to be fulfilled by a group of persons and, on the other hand, the collaboration-conflict process necessary to accomplish the review activity with success.

Groupware may simultaneously support SRM and collaboration-conflict processes. Unfortunately, resolving conflicts and getting to consensus is a complex problem. One major intricacy is dealing with the main assumptions behind conflict resolution: (1) the interlocutors have diverse profiles, interests, viewpoints, and strategies that should be respected and often promoted to reach high-quality results; (2) in this context, reaching consensus requires a collective cognitive effort to understand the different positions and negotiate acceptable solutions; and (3) the process should be simultaneously fast and thorough, two goals that are often difficult to reconcile.

Many groupware systems emphasize collaboration to the detriment of conflict management, for instance adopting a strict focus on participation and shared information. Such an approach may however fail, either because conflicts may remain dormant, just to arise later; or they may escalate to unacceptable levels, making it more difficult if not impossible to accomplish the corporate goals without explicit negotiation. It is therefore necessary to balance collaboration and conflict.

The problem discussed in this paper concerns the lack of collaboration-conflict balance observed in the current groupware tools.² Our research tries to supplant this lack of balance by integrating models of collaboration and conflict. This research guided the development of a groupware tool supporting SRM in the Functional Specification phase.

The adopted research approach is based on the Design Science paradigm.³ This problem-solving paradigm has its roots on engineering. It seeks to understand how technology may contribute to solve specific problems in particular domains. The Design Science paradigm emphasizes two main research goals: (1) establishing relevance through the identification of requirements and field-testing of concrete solutions, which in our case is accomplished by the FTR tool; and (2) establishing rigor by grounding the technology development in solid conceptual foundations, which in our case concerns the collaboration-conflict model.

The paper is organized in six sections. In Section 2 we discuss the research's theoretical foundations. In Section 3 we describe the collaboration-conflict process. Section 4 describes the developed prototype. Section 5 describes an experiment carried out with the

prototype. And finally Sections 6 and 7 present some points for discussion and the conclusions from this research.

2. Theoretical Foundations

2.1. Behavioral foundations of software review meetings

Many recent software development approaches emphasize participation and collaboration as critical to improve performance. Examples include agile⁴ and open source⁵ software development. SRM follow the same assumptions, relying on collaboration to improve the early detection and correction of defects in software development.⁶

SRM involve groups of experts, following formal procedures and designated roles, in the discovery of discrepancies between software specifications and other software documents, standards, and best practices.⁷ Johnson⁸ found out that these discrepancies can be one or two orders of magnitude less costly to remove when found in early development stages than after being released to customers; and also realized that SRM are effective in discovering certain soft, but nevertheless costly, defects such as logically correct but poorly structured code.

In accordance with the International Software Testing Qualifications Board, the roles and responsibilities involved in SRM include:⁶

- Manager: Has responsibility for the final decisions;
- Moderator: Is responsible for the success of the review meeting. Leads the meeting and balances the discussions. Whenever necessary, also arbitrates conflicts;
- Authors: Submit software artifacts for review and explain and justify their decisions;
- Reviewers: Identify, analyze and question the defects found in the artifacts under review;
- Secretary: Documents what happens in the review meeting, registering the defects and final decisions.

D'Astous et al¹ conducted observational studies to identify and characterize the predominant configuration of exchanges associated with SRM: 1) solution-elaboration; 2) solution-evaluation; 3) solution-evaluation-elaboration; 4) proposition-opinion; and 5) opinion-arguments. This indicates that both conflict (negative evaluation) and collaboration (elaboration of an alternative solution) play an important role in SRM.

2.2. Collaboration-conflict model

Abstracting from the patterns observed by D'Astous et al,¹ we may define a *behavioral model* underlining two very distinctive behaviors:

- **Conflictive:** When reviewers assert negative evaluations of solutions proposed by the authors. When authors and evaluators provide negative opinions regarding the others' propositions.
- **Collaborative:** When reviewers seek to compensate negative evaluations by elaborating upon solutions or proposing new solutions. When authors and evaluators provide positive opinions regarding the others' propositions.

These behaviors define the *collaboration-conflict spectrum* of the exchanges between reviewers and authors in SRM. Of course the participants may continuously change from one behavior to another along the review process. Though what should be noted is that, depending on the contingencies of the specific situation, these behaviors may be equally supportive and harmful to the quality of the SRM outcomes.

For instance, excessive collaboration may lead to groupthink, which has been considered detrimental to the decision quality.^{9,10} Also, extremely conflictive behaviors may lead to unsuccessful review meetings. Interestingly, dealing with conflict has been considered a way to avoid groupthink¹¹ and collaboration is also a viable way to overcome conflict. Thus the two behaviors may actually be necessary to improve the SRM quality.

Our model is based on the assumption that (1) review meetings should not gravitate towards being strictly collaborative or strictly conflicting but instead should **reflect the whole spectrum of behaviors**. The model also considers that (2) both **collaboration and conflict should be stimulated** in particular circumstances, since they are necessary to counterbalance the negative effects of each other.

Fricker and Grünbacher¹² distinguish between single-party groups, which are highly cohesive and thus pursue the same goals, and multiple-party groups, which appear on different sides of the table. Multiple-party groups may be further classified as differentiated, homogeneous and collaborating. The former compete between each other, while homogeneous groups have the same aspirations but different opinions, and collaborating groups seek an agreement that may be beneficial to all group members. Thus in our behavioral model we should also consider that (3) **collaboration and conflict may emerge at different grades**, from

cohesive to differentiated, homogeneous and collaborating. Research in conflict resolution has found out that the adopted strategies depend on various factors such as personal style, gender, organizational influences and culture.¹³ This reinforces the argument that collaboration and conflict should coexist in SRM, and that no particular predisposition to benefit one over the other should be adopted.

2.3. Computational support to the collaboration-conflict model

Thomas¹⁴ considers that beyond behavioral predispositions, cultural factors and social pressures, the adoption of collaborative and conflictive behaviors may be influenced by rules, procedures and incentive structures. We ponder that, by controlling these elements, **technology may explicitly influence human behavior in SRM**. We may distinguish the following types of influence:

1. Using technology to manage the process;
2. Using technology to intervene in the process as a facilitator or mediator.
3. Using technology to develop incentive mechanisms that promote the process quality.

Exemplary of the first type, we find Online Dispute Resolution systems, which manage the definition of goals, preferences, offers and counteroffers, and settlements.¹⁵⁻¹⁷ In the second type we find intelligent mediation tools.¹⁸⁻²¹ They employ automatic or semi-automatic mechanisms to monitor activity, identify problems with participation, and to assist their resolution through human interventions and information management mechanisms.

The WinWin negotiation model for requirements inspection²² has applied intelligent mediation to SRM, offering mechanisms to detect software defects like missing capabilities and hidden requirements, and promoting agreements using brainstorming, categorizing and polling tools.

The third type addresses the collaboration-conflict model in more subtle and diverse ways than the previous ones, using technology to influence the participants' behaviors but without explicit control. Within this category we may find several technology-designed mechanisms:

- Providing awareness of conflicts.^{23,24}
- Supporting conflict detection and traceability.²⁵
- Visualizing preferences and settlement spaces.^{26,27}

- Promoting knowledge exchange and alternative problem/solution representations.²⁸
- Promoting certain positive values such as anonymity, constructive criticism, participation and consensus.²⁹⁻³¹
- Detecting and discouraging certain malicious acts.³²

Some of these incentives have been used with success in crowdsourcing systems like the Wikipedia and open source software development.^{33,34} For instance, Wikipedia offers talk pages and controversial tags to facilitate conflict resolution.³³ Also in the software engineering field, Ramires et al³⁰ experimented several mechanisms to promote consensus in software requirements validation, supporting multiple individual preferences, consensus solutions, and also rating users according with their conflictive or collaborative behaviors.

3. The Collaboration-Conflict Process

In this section we elaborate the collaboration-conflict process, which provides a particular implementation of the model discussed in the previous section. This implementation is necessary to evaluate the model assumptions.

We conceptualize the collaboration-conflict process as a combination of three functions: (1) review, (2) negotiation, and (3) argumentation. Let us now elaborate these functions in detail. Words in bold call the attention to key concepts.

Review. The review meeting may be characterized according with the following phases:

- **Review statement:** What triggers the meeting, consisting of a list of **review items** such as specification documents and programming code.
- **Scores:** In this phase, the participants give scores to the review items. We currently support three scores: **accept**, **reject** and **accept with restrictions**. This phase may involve negotiation (described below).
- **Decision:** After assessing the various review items, a decision must be made about the review. This final phase involves analyzing the scores given to each review item, equating their impact on the overall review and determining if the review fails or succeeds.

Negotiation. The negotiation phase is prompted by conflicts. There is a **conflict** when two or more reviewers give different scores to a review item. A conflict should only be resolved through negotiation. Multiple

negotiations may occur during a review. A negotiation evolves according with the following steps:

- **Proposals:** In the first step, the different scores given to a review item are treated as negotiation proposals submitted to the participants.
- Search for consensus: The participants have to reach a **final score** for the item. This step may require argumentation (described below).
- Closure: A negotiation is closed when a final score is defined for a review item.

Argumentation. The search for a final score may require the confrontation of arguments. We adopted an argumentation model based on the Issue Based Information System³⁵ model, which has been used in software engineering to capture design rationale.³⁶⁻³⁸ The argumentation model defines the following elements:

- **Positions:** Several positions are expressed **in favor** or **against** a proposal. The positions are automatically inferred from the scores attributed by the participants to the review items (for instance, the participants that gave a reject score are against the participants that gave an accept score and vice versa).
- **Arguments:** Concise pieces of text giving strength to positions. In order to enforce argumentation, the participants are requested to complement the reject and “accept with restrictions” scores with arguments.

Notice that the review, negotiation and argumentation activities are entangled and concurrently executed. The data model of the collaboration-conflict process is organized around the various elements identified above: review statement, review item, score, final score, proposal, position, argument and decision. Figure 1 depicts this model.

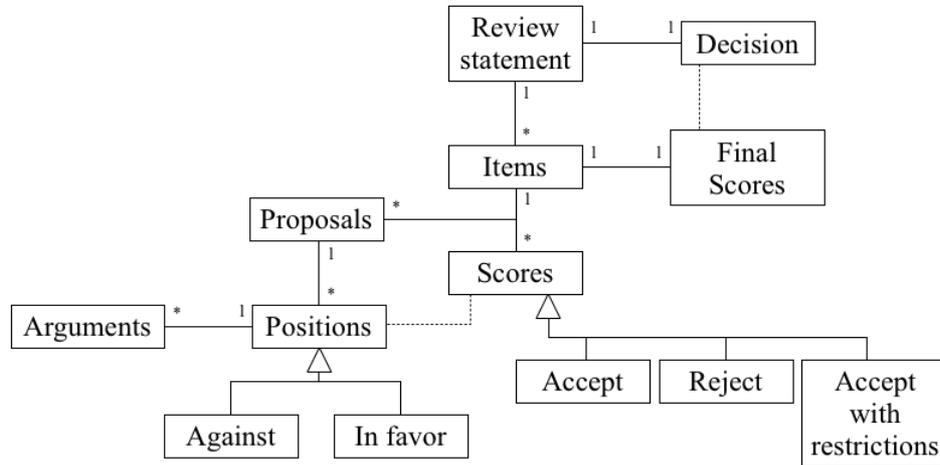


Fig. 1. Data model of the collaboration-conflict process

We also observe that the goal of the collaboration-conflict process is not necessarily to obtain consensual scores for every review item. Several rules may be defined regarding what results should be drawn from the individual scores. The following rules may be considered: majority voting, where the final score is determined by the majority of the participants; consensus voting, i.e. there is only a result if it corresponds to the same score selected by all participants; and manager decision, where the manager decides the final score based on the participants' scores.

After obtaining the final scores, the whole review statement should be subject to a final decision. Again, several rules may be adopted to reach the final decision. We adopted the following types of decisions in our implementation: (1) full acceptance, when all participants accept; (2) general reject, if there is at least one reject; (3) postpone, if there is more than a predefined number of accepts with restrictions; and (4) general acceptance otherwise.

3.1. Factors affecting the process

Every collaboration-conflict process, although structured according with the phases previously described, has its own dynamics and depends on a set of factors that interact between themselves, interfering with the process outcomes. We highlight the following contextual factors:

- Level of conflict - As the level of conflict increases, so does the cognitive effort necessary to negotiate and argue. At the limit, a destructive level of conflict will lead to a failed process. The number of

suggested proposals, positions and arguments may serve to measure the level of conflict.

- Number of participants - A large number of participants may also turn it more difficult to negotiate.
- Status differences - Status differences address the dependence relationships between leaders and subordinates. Groups having status differences may be negatively affected by the dependence on people with more power.³⁹ The balance between the participants' proposals, positions and arguments may serve to measure the effects of status differences.
- Problem involvement - A low involvement with what is under discussion may turn it more difficult to participate in the process. The number of suggested proposals, positions and arguments may serve to measure the problem involvement.
- Group expertise - The lack of expertise about the problem under discussion may also affect the process outcomes. This factor may be measured by assessing the quality of the presented arguments.

3.2. Quality criteria for assessing the process

It is fundamental to define quality criteria for assessing the collaboration-conflict process. However, the selection of criteria is quite challenging. Let us consider, for example, a situation where a decision is immediately reached after a small number of proposals; and contrast it with another situation in which, after a long argumentation, several proposals were discussed.

We may assume the first case has low quality while the second case has high quality. This assumption may however be misleading. For instance, it is possible that the first case has low complexity and relevance, and the adopted decision is not only adequate but also efficient. On the contrary, the second case may correspond to a situation where conflicts may have lead to a suboptimal decision, having the additional cost of spending too much time to finish the process.

When considering negotiation processes, quality has been fundamentally associated with efficiency. For instance, the distance between the agreed solution and the best possible solution that could be obtained by continuing the process, designated value-left-on-the-table, is commonly used to evaluate the quality of negotiation processes.⁴⁰ This approach is however more adequate to bargaining than to collaboration-

conflict, since the former is influenced by the zero-sum game while the later is more influenced by “satisfying” trade-offs.⁴¹

When considering collaboration processes, quality tends to be measured according with a diverse set of variables categorized as efficiency, effectiveness, satisfaction, and consensus.⁴² This suggests the quality of collaboration-conflict processes should be measured according with a combination of criteria, for which we suggest:

- Efficiency - Time to complete the task.
- Flexibility - Number of positions changes to converge with the majority.
- Contribution - Number of arguments produced by the participants.

4. FTR Tool

This section describes the tool we developed to support the process described in Section 3. We first describe the specific requirements of the groupware tool and associate those requirements with the particular characteristics of the collaboration-conflict process. We then describe the tool’s architecture and interface.

4.1. *Addressing the collaboration-conflict model*

One fundamental characteristic of the FTR Tool is making the collaboration-conflict process explicit to the participants. It is not enough to manage messages exchange according with the typical tags like topics, contents, authors, etc. Specific tags are necessary to position messages within the collaboration-conflict spectrum.

To illustrate the problem, consider that messages exchange is supported through a typical e-mail tool. The tool preserves the exchanged messages in their temporal order, but the collaborative and conflicting behaviors are not easy to discriminate and follow. This is particularly true with asynchronous interaction.⁴⁴ As participants tend to mix several types of contributions into a single message, it is not easy for a remote participant to keep track of the interventions according with the collaboration-conflict continuum, which means the participants have to overcome this ambiguity by constantly assessing and reassessing the messages’ contents.

To reduce these problems, the FTR Tool adopts the argumentation model described in Figure 1. This model assures that exchanges messages may be tracked according with relevant criteria like positions in favor or against and arguments.

It is also important to give the moderator an overall view of the participants' contributions according with the collaboration-conflict continuum. The FTR Tool addresses this issue with a *participameter*.⁴¹ Table 1 summarizes the participameter information collected by the FTR Tool and delivered to the moderator.

Table 1. Individual assessment information

Assessment	
Contributing Positions	Number of proposals from a participant in relation with the total number of proposals.
Contributing Arguments	Percentage of arguments from the participant in relation with the total number of registered arguments.
Punctuality	Average of time to complete the task, as a percentage of time assigned to the task.
Relevance of Arguments	Number of arguments from a participant that contributed to the final score in relation with the total number of arguments.
Flexibility to converge	Number of score changes to converge with the majority, in relation with the total number of score changes to converge with the majority.

4.2. FTR implementation

Any FTR requires several pre-arrangements from the moderator. The FTR Tool supports some of these activities. It allows importing the review documents into the system. It also allows presenting the reviewers' initial proposals and comments, and selecting for discussion (with control by the moderator) the artifacts that seem more conflicting.

Another important supported function is allowing the moderator to check for duplicates and equivocal statements. Using the FTR Tool, the moderator may turn doubts, problems, comments, alternatives, and solution into validated proposals for assessment by the reviewers in the next phase. We note however that this preliminary phase is not the main focus of our research. We actually concentrated our research on the support to the second phase: the collaboration-conflict process.

The second phase starts when the moderator sends the first validated proposals to the reviewers. New proposals may be delivered during the review if necessary. To ensure confidentiality, the proposals are dissociated from the original authors.

During the second phase, the reviewers register their scores. Each reviewer may associate a score to a proposal, reflecting his/her judgment about the proposal (0 - not an error/accept; 1 - light error/ accept with restrictions; 2 - serious error/reject). In case the chosen score is 1 or 2, the reviewer is requested to complement the score with arguments, consisting of small text sentences. All arguments should be linked to a Functional Specification Document. Examples include: "the item cannot be related

with the Functional Specification and should be removed”; “the item does not comply with the specification of function X”; or “the item fails to implement requirement Y”.

The positions in favor and against each proposal are automatically calculated. The divergences are shown to the reviewers without exposing the identities of the opponents. After evaluating the arguments associated with one proposal, a reviewer may change his/her own position or add additional arguments. The changes in positions update the associated arguments. This procedure may be repeated until closing a proposal with the final score. Updates to positions and arguments are visible to all reviewers.

When there are no positions against a proposal, it is immediately “closed” and the final score is known. In order to cover all proposals assigned to a FTR session, a “closed” proposal cannot be reopened in the same session. Also for efficiency reasons, the proposals are controlled by a timeout mechanism. The moderator is responsible for setting the time limits and closing the proposals when the time limits are reached. The reviewers are notified before the proposal time is out. After all proposals are closed the process advances to the decision phase.

The decision phase will determine the output of the FTR. As previously mentioned, a consensual score may not be achieved for every proposal. In order to close the review, the moderator may adopt three different strategies: majority voting, deciding another negotiation round, and assigning his/her decision. The moderator selects one of these rules before starting the session to guarantee the transparency principle.

4.3. *Additional implementation details*

The FTR tool was built using the Microsoft .Net framework and C# language. Being a Web application, it can be used at any time and place. The adopted database manager was SQL Server. To illustrate the prototype, we present some screen dumps.

Figure 2 shows the beginning of the FTR session. The proposals selected by the moderator are displayed at the left. The participants enter their positions on the right.

Enter your positions

Meeting
EF_29520_PROTOCOLO ANATEL_SIA - EF_29520_PROTOCOLO ANATEL_SIA_V1 - 17/12/2008

Question				Positions of other participants			My Positions		
ID	Prioridade	Estado	Nome	Data	Posição	Ressaltos	Data	Posição	Ressaltos
97	ALTO(A)	ABERTA	NO ITEM 12.3 "12.3. UC_SIA_29520_02_SOLICITAR ABERTURA DE TT PARA CLIENTE B2B" DA SESSÃO "CASOS DE USO" A REGRA DE NEGÓCIO RN4 NÃO ESTÁ SENDO CHAMADA EM LUGAR ALGUM.	17/12/2008 11:48	É ERRO LEVE	Acertar documentação.	19/12/2008 05:09	É ERRO LEVE	erro documentação
98	ALTO(A)	ABERTA	NO ITEM 12.3 "12.3. UC_SIA_29520_02_SOLICITAR ABERTURA DE TT PARA CLIENTE B2B" DA SESSÃO "CASOS DE USO" O REQUISITO NÃO ESTÁ SENDO ATENDIDO.	17/12/2008 07:10	É ERRO LEVE	Corrigir documentação de forma que a referencia seja realizada corretamente.			
99	ALTO(A)		NO ITEM 12.3 "12.3. UC_SIA_29520_01_SOLICITAR ABERTURA DE TT PARA CLIENTE B2B" NAS SESSÕES "CASOS DE USO" E "CARACTERÍSTICAS NÃO SUJEITAS A REVISÃO" NÃO ESTÁ SENDO ATENDIDO O REQUISITO DE "VISÃO GERAL DA SOLUÇÃO" CONTA O "PROCESSO DE ABERTURA DE TT PARA CLIENTE DE RETENÇÃO CONVERGENTE" O DIAGRAMA NÃO MOSTRA CRIAÇÃO DO TT NO SBL MKT.	19/12/2008 03:03	É ERRO LEVE	Erro de documentação			
104	ALTO(A)	EM NEGOCIAÇÃO	NO ITEM 12.3 "12.3. UC_SIA_29520_01_SOLICITAR ABERTURA DE TT PARA CLIENTE B2B" OS FLUXOS DE EXCEÇÃO EX1, EX2 E EX3 NÃO ESTÃO CLAROS OS MOTIVO DE SEREM EXCEÇÕES.						
109	ALTO(A)	EM NEGOCIAÇÃO							

Summary of positions
NÃO É ERRO: 0, É ERRO LEVE: 4, É ERRO GRAVE: 0, NULO: 0

Position
NÃO É ERRO

Minha Ultima Posição
NÃO É ERRO: X, É ERRO LEVE: -, É ERRO GRAVE: -

The proposals under consideration

Position:

- Not an error
- Serious error
- Light error

Fig 2. Registering the participants' positions.

	Reviewer 1	Reviewer 2	Reviewer N	
Proposal 1	1	1	0	Argument 1
Proposal 2	1	1	1	Argument 2
Proposal 3	0	0	1	Argument 3
Proposal 4	1	1	2	
		...		

Fig 3. Argumentation of divergent positions.

Figure 3 illustrates some possible outcomes of the collaboration-conflict process. A proposal should be negotiated when it receives different scores. The arguments associated with each position provide rational elements for the change of positions. Notice that in the illustrated example there are no arguments associated with proposal 2 (row 2) because the scores were consensual.

Figure 4 shows how arguments are inserted. For each proposal, the system shows its positions. When a position is added, the system opens a text box for writing an argument.

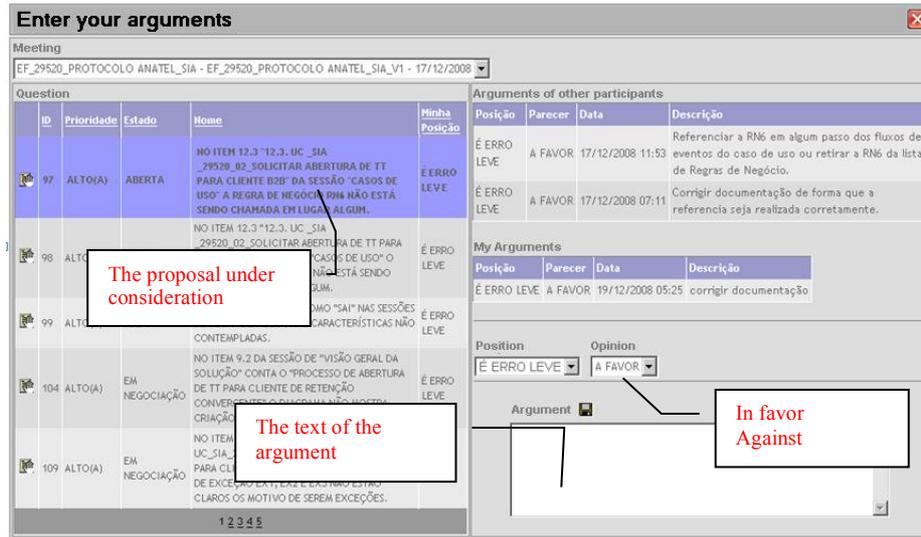


Fig. 4. Adding an argument to a divergent position.

When the session is completed, a summary is generated. This is shown in Figure 5. When assessing the results, the moderator is able to decide on the next steps. One alternative is giving the participants more time to analyze documents and code, and then scheduling another session. Another possibility is making a decision on the proposals that have not reached consensus.

Nome	Não e erro	É erro	É erro grave	Nulo	Estado	Posição final	Grav. consenso
A SESSÃO DE "FEATURES" ESTÁ CONFUNDINDO CONCEITOS DE CASOS DE USO, REGRAS ADERENTE AOS 3PS OJ).	0	4	0	0	FECHADA POR CONSENSO	É ERRO LEVE	ALTO(A)
PARA SOLICITAR ABERTURA DE	0	4	0	0	FECHADA POR CONSENSO	É ERRO LEVE	ALTO(A)
O BARRAMENTO SOA NÃO DEVE SER TRATADO COMO ATOR.	1	0	3	0	EM NEGOCIAÇÃO	MÉDIO(A)	ALTO(A)
QUESTIONÁRIO PADRÃO" DA SESSÃO "CASOS DE USO" DEVE CONTER UMA REGRA COM	0	1	3	0	EM NEGOCIAÇÃO	ALTO(A)	ALTO(A)
NO ITEM 12.1.5.1 "12.1. UC_SIA_18164_09_DISPONIBILIZAR QUESTIO	0	1	0	0	EM NEGOCIAÇÃO	ALTO(A)	ALTO(A)
NO ITEM 12.1.5.1 "12.1. UC_SIA_18164_09_DISPONIBILIZAR QUESTIO	0	2	2	0	EM NEGOCIAÇÃO	MÉDIO(A)	ALTO(A)
NO ITEM 12.1.5.1 "12.1. UC_SIA_18164_09_DISPONIBILIZAR QUESTIO	0	1	0	0	EM NEGOCIAÇÃO	ALTO(A)	ALTO(A)
NO ITEM 12.2 "12.2 UC_SIA_29520_01_SOLICITAR ABERTURA DE TT	0	0	0	0	BOCIAÇÃO	ALTO(A)	ALTO(A)
NO ITEM 12.2 "12.2 UC_SIA_29520_01_SOLICITAR ABERTURA DE TT PARA PRÉ-PAGO" DA SESSÃO "CASOS DE USO" O FLUXO DE BÁSICO:	0	0	0	0	BOCIAÇÃO	BAIXO(A)	ALTO(A)
2: ESTÁ ESCRITO "FLUXO ALTERNATIVO ALZ" E NÃO AL1.	0	0	0	0	BOCIAÇÃO	É ERRO LEVE	ALTO(A)
NO ITEM 12.2 "12.2 UC_SIA_29520_01_SOLICITAR ABERTURA DE TT PARA PRÉ-PAGO" DA SESSÃO "CASOS DE USO" O FLUXO DE BÁSICO:	0	0	0	0	BOCIAÇÃO	É ERRO LEVE	ALTO(A)
4 E 5: ESTÁ ESCRITO "BARRAMENTO ENVIAR".	0	0	0	0	BOCIAÇÃO	É ERRO LEVE	ALTO(A)

Fig 5. Summary of the FTR session.

5. An Evaluation of the FTR Tool

An evaluation of the FTR tool was carried out and its results were compared with those obtained with a standard FTR. The evaluation was

conducted in a telecommunications company operating in Brazil. We use the fictitious name BTC to preserve the anonymity of both the company and the participants. The main purpose of the evaluation was to obtain qualitative insights about the collaboration-conflict model, the process and the FTR tool.

5.1. Experimental setting

BTC subcontracts several software companies to develop software artifacts. The subcontracted companies may be located in Brazil or abroad. Before formally concluding these contracts, all artifacts delivered by the subcontractors must be submitted to a quality assurance process that evaluates them against the specifications described in the contracts. Depending on the task complexity, quality assurance may demand considerable time and effort from both parties. The standard FTR has been used for a few years and all members of the quality assurance team considered that changes could be done to improve it without reducing quality, especially because the FTR were done face-to-face and often involved foreign subcontractors.

The standard FTR engages from five to eight people: up to four authors, three reviewers and a leader. For the evaluation sessions we planned a similar team. However, we had to define how to compare the standard FTR against our approach. In theory we had two alternatives:

- 1) Select an artifact, perform the standard and new FTR using two different teams, and then compare the results;
- 2) Assign the same team to two different but equivalent artifacts; and have the team successively apply the standard and the new FTR approaches.

Both alternatives had some constraints. We could not count on real subcontractors to play the authors' role due to the costs involved. We also did not have formal authorization from BTC to apply the new FTR approach in real reviews. But we still wanted to use real data in our evaluation. We thus adopted a variation of alternative 1: Select two recent artifacts and recover their FTR records, which were already concluded through the standard process. This corresponded to a post-hoc analysis of the FTR process.

After that, we rerun the FTR with the same two artifacts but using a different review team. From an experimental point of view, this corresponds to repeating samples with different subjects and experimental conditions (traditional FTR and our approach, using the FTR tool).

We then compared the results of both samples. We are aware of the limitations of this schema, but we preferred that to using, for example, a totally artificial setting, such as having students playing the FTR.

For the comparison we used three criteria: (1) number of proposals; (2) number of arguments; and (3) number of changed positions toward consensus. The comparison was directed by the assumption that higher these indicators were, higher was the quality of the reviewing process. The number of changed positions toward consensus was an indicator that deserved some further analysis, as discussed later.

5.2. Evaluation results

In the following description of the evaluation results we will refer to the artifacts as FE29520 and FE22520. First, it should be noted that the reviewers rejected them both. When comparing the results, we observed that the new FTR method resulted in increased numbers of arguments and changed positions towards consensus. This may be a sign that the FTR tool promotes higher levels of argumentation than traditional FTR.

A summary of the obtained quantitative results is reproduced in Tables 2 and 3. The standard FTR of FE29520 resulted in 4 proposals, 6 arguments and 2 changed positions. The new FTR (using the FTR Tool) resulted in 31 proposals. The first session resulted in 15 arguments and 3 changed positions, and the second session an additional 62 arguments and 1 changed position. Regarding the standard FTR of FE22520, we had 9 proposals, no arguments and no changed positions. The new process (FTR Tool), on the other hand, resulted in 23 proposals, 1 argument and also no changed positions. The results from FE22520 show that the participants (and in particular the leader) took the immediate decision to reject the functional specifications, which explains the absence of arguments.

Table 2. Indicators of FE29520 review

Indicators	Standard FTR	FTR Tool
Elapsed Time	3	8
Number of proposals raised	4	31
Number of sessions	3	2
Number of arguments placed by reviewers and authors	6	15 and 62 *
Number of changed positions towards consensus	2	3 and 1*

* First and second sessions, respectively

Table 3. Indicators of FE22520 review

Indicators	Standard FTR	FTR Tool
Elapsed Time	7	5
Number of proposals raised	9	23
Number of sessions	1	1
Number of arguments placed by reviewers and authors	0	1
Number of changed positions towards consensus	0	0

Apparently, the simplicity of the FTR Tool and the short training applied before the sessions were sufficient to accomplish the reviews without relevant problems. We noticed however, that the arguments were not always used as such. For instance, several comments were inserted as if they were arguments. Comments such as “I agree with the item above” are not real arguments but appeared as such. This may impact the above comparisons. Only about 40% of the arguments written by the team members were actually identified as real arguments. These difficulties are in line with those reported by Borges et al⁴⁴ on the use of a structured argumentation model.

5.3. Questionnaires

The participants in the evaluation (those that used the FTR Tool) were requested to complete an open questionnaire about the tool. The answers to the questionnaire seem to indicate, in a general way, that the tool supports the dynamics of the collaboration-conflict model and promotes collaboration in FTR. A summary of advantages and disadvantages pointed by the participants is presented in Tables 4 and 5. Table 4 refers to the standard FTR while Table 5 refers to the FTR Tool usage.

Table 4. Advantages and disadvantages of standard FTR

Positive Aspects	Negative Aspects
Often a face-to-face meeting is more productive because people have difficulties in expressing themselves in writing - verbally is easier and faster - especially when it comes to a discussion where reasoning through arguments is necessary.	Meetings are not always possible because of the geographical distribution and the time involved. Also, difficulties documenting the meeting: what has been discussed and what has been resolved.
	Negotiation is difficult because there is no consolidation of ideas in written format.
	Poor use of time in meetings where one

	loses focus easily.
	Items considered irrelevant are not captured, but they contribute to a more clean and clear documentation.
The possibility of using drawings to explain an idea, facilitating the understanding and optimizing time usage.	There is no reliable history of assessments made by each participant.
	When it is necessary to return to a previous validation, there is no history of meetings and positions from each participant.
	Sometimes an item under discussion is forgotten. The questions raised during the review end up lost after several versions.

The participants pointed out the following main advantages: (1) the tool was easy to learn; (2) had clear rules; (3) managed knowledge evenly; and (4) preserved the argumentation history. Also, the support to asynchronous and geographically distributed meetings was identified as an advantage, though the face-to-face meetings ease understanding and offers more expressiveness.

Table 5. Advantages and disadvantages of using the FTR tool

Positive Aspects	Negative Aspects
Outcomes in one place, where all participants have access.	Not enough space to type an idea.
Participants may interact at the meetings at different times and without the need of being in the same place. It is a solution to the problem of dispersed teams.	The tool was unavailable during certain periods.
Negotiation was much faster because there was a consolidation of the points raised.	As each person works on her/his own schedule, sometimes the question you insert stays without any response for some time.
Less likely to shift the meeting focus.	
Uptake of irrelevant items that may contribute to a more clean and clear documentation that facilitates the next steps.	May hinder understanding, if the written communication is not clear.
The validations records of each participant are stored and this avoids	

bad communication.	
You can return to the validation process at any time.	Does not maintain version control. An issue may not be answered in the first round and may reappear later. But there is no explicit information that this issue has been treated earlier.
Contributes to equalize knowledge of everyone involved in the project.	
The formalization of the problems / issues / positions / arguments / results are recorded in a structured way, and it keeps meeting history.	There should be links between positions, arguments and evaluations from a single user, i.e., a user can assess all the involvement her/he had with an issue. This would make the tool more flexible and optimize the time to understand the issues.

It is important to emphasize that the participants, in general, valued the capability to register all arguments in an organized way. This seems to ease changing positions towards consensus and enriches the FTR as a whole.

One of the main problems identified in the standard FTR is that the review repeats itself several times without necessity, only because the reviewers' recommendations seem to be unnoticed by the authors. The FTR Tool was seen by the participants as a mechanism to overcome this problem.

Overall, the comments produced by the participants indicate that the desired objectives for the FTR are coherent with the collaboration-conflict model: supporting a continuum of collaboration and negotiation. The participants in the experiment indeed recommended the adoption of the FTR Tool in their organization.

6. Discussion

In Table 6 we summarize the various concepts involved in the collaboration-conflict model. The major distinctions concern the behavioral context, expected attitudes, computational support, incentives, contextual factors, quality criteria, and data elements. As the paper shows, the integration of such disparate concepts requires bridging information sharing with negotiation and argumentation. This was implemented in the FTR Tool through one common data element: argument. Arguments

contribute at the same time to build a common understanding of the problem and to bring forward different views and conflicting positions.

Looking at this focal point, it was striking to find out that in the evaluation the FTR Tool generated more arguments than the standard FTR. The responses to the questionnaires also emphasize that the participants considered arguments as important meeting elements, allowing them to reason and consolidate the discussion while avoiding bad communication.

Table 6. The collaboration-conflict model.

	Collaboration	Conflict
Behavioral context	Single-party, collaborating	Differentiated, homogeneous
Expected attitudes	Collaborative	Conflictive
Computational support	Information sharing	Negotiation, argumentation
Incentives	Awareness, visualization, knowledge exchange, contribution, consensus	Conflict detection, preferences, settlement spaces, detection of malicious acts
Contextual factors	Expertise, involvement	Level of conflict, status differences
Quality criteria	Efficiency, contribution	Efficiency, flexibility
Data elements implemented by the FTR tool	Proposals, arguments, decision, final scores	Positions, arguments, scores Automated positions

Although these results are promising, we are aware that we need more experiments to claim that computer support may increase argumentation, and also that argumentation may increase the quality of review meetings. The qualitative insights obtained with the experiments show that such causal relationships should be further investigated, and also indicate that the increased number of arguments might be related to the increased number of proposals. One possible interpretation is that the collaboration-conflict model might promote constructive conflict, since conflicting positions may be accompanied with alternative proposals. This interpretation is inline with the observations from D'Astous et al,¹ although in that case no technology support was used.

We also observe that the validation in a real-world setting provided some insights not possible when using students or artificial settings, but on the other hand limited the number of samples and the level of control over the evaluation setting. In any case we are aware that we need more sessions with more variety of artifacts and participants to consolidate our conclusions.

We finally note that of the three quality criteria considered by our study - efficiency, flexibility and contribution - only contribution seems to have been affected by the FTR Tool. Future experiments may be set up to

evaluate the impact of technological incentives specifically focused on improving efficiency and flexibility.

7. Conclusions

We developed a collaboration-conflict model for software review meetings and a tool to support it. The collaboration-conflict model brings together very distinct behavioral contexts, expected attitudes, computational support, incentives and quality criteria. The research allowed us to understand how to bring together these elements. The developed collaboration-conflict process integrates information sharing with negotiation and argumentation, linking various data elements such as decisions, proposals, positions, arguments and scores.

Two evaluation sessions were carried out in a telecommunications company that adopts a global software development strategy. We compared the results of four review meetings, two using the standard review process and two using the tool described in this paper. The quantitative and qualitative results provide some insights about the reviewers' behavior facing the somewhat contradictory process of collaboration-conflict.

First, the evaluation data indicates that the developed tool is capable to support software reviews with some advantages over the standard process. Second, the evaluation shows that the pivot data element in the collaboration-conflict model is the argument, as it integrates the collaborative and the negotiated aspects of the tool functionality.

And third, the evaluation also allowed us to identify some points that may constitute subject for future research. An important challenge is to evaluate the causal relationships between technology use, increased argumentation and improved decision quality. Another challenge is validating the positive relationships between proposals and arguments, delineating what may be designated as "constructive conflict". And finally, this research also gives some positive indications towards extending the collaboration-conflict model to other collaborative tools and applications.

The research described in this paper contributes to information systems development in two main ways. One is raising attention, articulating the problems and describing a technical solution for integrating collaborative and conflicting behaviors in computational support. The other one is contributing to the development of technology-designed incentive mechanisms, which influence human behavior and process quality through

information structures that promote positive and discourage negative values.

Acknowledgments

This work was partially supported by grants No. 479374/2007-4 and 567220/2008-7 from CNPq (Brazil), and grant PTDC/EIA/102875/2008 from FCT (Portugal).

References

1. P. D'Astousa, F. Détienne, W. Visser and P. Robillard, Changing Our View on Design Evaluation Meetings Methodology: A Study of Software Technical Review Meetings, *Design Studies*, **25**(6) (2004) 625-655.
2. P. Antunes, J. Ramires and A. Respício, Addressing the Conflicting Dimension of Groupware: A Case Study in Software Requirements Validation, *Computing and Informatics*, **25** (2006) 523-546.
3. A. Hevner, S. March, J. Park and S. Ram, Design Science in Information Systems Research, *Management Information Systems Quarterly*, **28**(1) (2004) 75-105.
4. T. Dyba and T. Dingsøyr, Empirical Studies of Agile Software Development: A Systematic Review, *Information and Software Technology*, **50** (2008) 833-859.
5. A. Mockus, R. Fielding and J. Herbsleb, Two Case Studies of Open Source Software Development: Apache and Mozilla, *ACM Transactions on Software Engineering and Methodology*, **11**(3) (2002) 309-346.
6. E. Veenendaal, *Standard Glossary of Terms Used in Software Testing* (International Software Testing Qualifications Board, 2007).
7. M. Fagan, Design and Code Inspections to Reduce Errors in Program Development, *IBM Systems Journal*, **15** (1976) 182-211.
8. P. Johnson, Reengineering Inspection, *Communications of ACM*, **41**(2) (1998) 49-52.
9. J. Esser, Alive and Well after 25 Years: A Review of Groupthink Research, *Organizational Behavior and Human Decision Processes*, **73**(2-3) (1998) 116-141.
10. R. Baron, So Right It's Wrong: Groupthink and the Ubiquitous Nature of Polarized Group Decision-Making, *Advances in Experimental Social Psychology*, **37** (2005) 219-253.
11. C. Schwenk and J. Valacich, Effects of Devil's Advocacy and Dialectical Inquiry on Individuals Versus Groups, *Organizational Behavior and Human Decision Processes*, **59** (1994) 210-222.
12. S. Fricker and P. Grünbacher, Negotiation Constellations – Method Selection Framework for Requirements Negotiation, *Requirements Engineering: Foundations for Software Quality* (Springer, 2008), pp. 37-51.
13. J. Holt and C. DeVore, Culture, Gender, Organizational Role, and Styles of Conflict Resolution: A Meta-Analysis, *International Journal of Intercultural Relations*, **29**(2) (2005) 165-196.
14. K. Thomas, Conflict and Conflict Management: Reflections and Update, *Journal of Organizational Behavior*, **13** (1992) 265-274.
15. R. Vetschera, Preference Structures of Negotiators and Negotiation Outcomes, *Group Decision and Negotiation*, **15**(2) (2006) 111-125.

16. M. Poblet, P. Casanovas, J. López-Cobo and N. Casellas, Odr, Ontologies, and Web 2.0, *Journal of Universal Computer Science*, **17**(4) (2011) 618-634.
17. A. Lodder and J. Zeleznikow, Developing an Online Dispute Resolution Environment: Dialogue Tools and Negotiation Support Systems in a Three-Step Model, *Harvard Negotiation Law Review*, **10**(1) (2005) 287-336.
18. R. Briggs and P. Gruenbacher, Easywinwin: Managing Complexity in Requirements Negotiation with Gss, *Proceedings of the 35th Hawaii International Conference on System Sciences*, Hawaii, 2002.
19. P. Antunes and T. Ho, The Design of a Gdss Meeting Preparation Tool, *Group Decision and Negotiation*, **10**(1) (2001) 5-25.
20. D. Damian, A. Eberlein, M. Shaw and B. Gaines, An Exploratory Study of Facilitation in Distributed Requirements Engineering, *Requirements Engineering*, **8** (2003) 23-41.
21. P. Tedesco, Marco: Building an Artificial Conflict Mediator to Support Group Planning Interactions, *International Journal of Artificial Intelligence in Education*, **13** (2003) 117-155.
22. M. Halling, S. Biffel and P. Grunbacher, An Economic Approach for Improving Requirements Negotiation Models with Inspection, *Requirements Engineering*, **8** (2003) 236-247.
23. S. Alshattawi, G. Canals and P. Molli, Concurrency Awareness in a P2p Wiki System, *The 2008 International Symposium on Collaborative Technologies and Systems*, California, USA, 2008.
24. F. He and S. Han, A Method and Tool for Human–Human Interaction and Instant Collaboration in CSCW-Based Cad, *Computers in Industry*, **57** (2006) 740-751.
25. N. Shi, W. Tang, T. He and Y. Ye, Real-Time Conflict Detection Using Prior Knowledge and Context in CSCW Plant Design System, *Proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design*, Melbourne, Australia, 2007.
26. H. In, D. Olson and T. Rodgers, Multi-Criteria Preference Analysis for Systematic Requirements Negotiation, *Proceedings of the 26th Annual International Computer Software and Applications Conference*, Oxford, England, 2002, pp. 887-892.
27. M. Jarke, T. Jelassi and M. Shakun, Mediator: Toward a Negotiation Support System, *European Journal of Operational Research*, **31**(3) (1987) 314-334.
28. L. Yesilbas and M. Lombard, Towards a Knowledge Repository for Collaborative Design Process: Focus on Conflict Management, *Computers in Industry*, **55**(3) (2004).
29. N. Karacapilidis and D. Papadias, Computer Supported Argumentation and Collaboration Decision Making: The Hermes System, *Information Systems*, **26**(4) (2001) 259-277.
30. J. Ramires, P. Antunes and A. Respicio, Software Requirements Negotiation Using the Software Quality Function Deployment, *Groupware: Design, Implementation, and Use. 11th International Workshop, Criwg 2005, Porto De Galinhas, Brasil, September 2005* (Heidelberg, Springer-Verlag, 2005), pp. 308-324.
31. T. Rodgers, D. Dean and J. Nunamaker, Increasing Inspection Efficiency through Group Support Systems, *Proceedings of the 37th Hawaii International Conference on System Sciences*, Hawaii, 2004.
32. P. Ipeirotis, F. Provost and J. Wang, Quality Management on Amazon Mechanical Turk, *Proceedings of the ACM SIGKDD Workshop on Human Computation*, New York, USA, 2010.
33. A. Kittur and R. Kraut, Beyond Wikipedia: Coordination and Conflict in Online Production Groups, *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, Savannah, Georgia, USA, 2010.

34. C. Jensen and W. Scacchi, Collaboration, Leadership, Control and Conflict Negotiation Processes in the Netbeans.Org Open Source Software Development Community, *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, Hawaii, USA, 2005.
35. W. Kunz and H. Rittel, *Issues as Elements of Information Systems*, Institute of Urban and Regional Development, University of California at Berkeley (1970).
36. J. Conklin and M. Begeman, Gibis: A Hypertext Tool for Exploratory Policy Discussion, *ACM Transactions on Office Information Systems*, **6**(3) (1988) 303-331.
37. M. Lubars, Representing Design Dependencies in an Issue-Based Style, *IEEE Software*, **8**(4) (1991) 81-89.
38. M. Koegel, H. Naughton, J. Helming and M. Herrmannsdoerfer, Collaborative Model Merging, *Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion*, Reno, Nevada, USA, 2010, pp. 27-34.
39. V. Dubrovsky, S. Kiesler and B. Sethna, The Equalization Phenomenon: Status Effects in Computer-Mediated and Face-to-Face Decision-Making Groups, *Human-Computer Interaction*, **6**(2) (1991) 119-146.
40. H. Raiffa, J. Richardson and D. Metcalfe, *Negotiation Analysis. The Science and Art of Collaborative Decision Making* (Cambridge, Harvard Business Press, 2002).
41. L. Weingart, R. Bennett and J. Brett, The Impact of Consideration of Issues and Motivational Orientation on Group Negotiation Process and Outcome, *Journal of Applied Psychology*, **78**(3) (1993) 504-517.
42. J. Fjermestad and S. Hiltz, An Assessment of Group Support Systems Experimental Research: Methodology and Results, *Journal of Management Information Systems*, **15**(3) (1999) 7-149.
43. J. Herbsleb, Global Software Engineering: The Future of Socio-Technical Coordination, *International Conference on Software Engineering: Future of Software Engineering*, Minneapolis, 2007, pp. 188-198.
44. M. Borges, J. Pino, D. Fuller and A. Salgado, Key Issues in the Design of an Asynchronous System to Support Meeting Preparation, *Decision Support Systems*, **27**(3) (1999) 271-289.