Computing and Informatics, Vol. 25, 2006, 523–546

ADDRESSING THE CONFLICTING DIMENSION OF GROUPWARE: A CASE STUDY IN SOFTWARE REQUIREMENTS VALIDATION

Pedro Antunes

LaSIGE – Large Scale Information Systems Laboratory
Department of Informatics of the Faculty of Sciences of the University of Lisboa
Bloco C6, Campo Grande, 1749-016 Lisboa, Portugal
e-mail: paa@di.fc.ul.pt

João Ramires

LaSIGE – Large Scale Information Systems Laboratory
Centro Nacional de Pensões, Av. República 102, 1600-206 Lisboa, Portugal
e-mail: joao.j.ramires@seg-social.pt

Ana Respício

Centre for Operational Research
Department of Informatics of the Faculty of Sciences of the University of Lisboa
Bloco C6, Campo Grande, 1749-016 Lisboa, Portugal
e-mail: respicio@di.fc.ul.pt

Revised manuscript received 19 June 2006

Abstract. This paper addresses the conflicting dimension of groupware, seeking the reconciliation of two very different assumptions about the users' attitudes using groupware tools: users either collaborate or negotiate to reach consensus. We argue that groupware should integrate the full spectrum of attitudes occurring between these two extremes. The designed solution integrates content and process support in a coherent model supporting low and high conflict situations. Furthermore, we propose a set of benefits and resistances, developed at the user-interface level, aiming to influence users towards low conflict attitudes when interacting with groupware.

This approach was applied in a case study involving the development of a groupware tool supporting Quality Function Deployment for software requirements validation in a real-world organization. The case study indicated that the proposed approach was beneficial promoting consensus.

Keywords: Software requirements validation, groupware, conflict dimension, negotiation and collaboration support, integrative attitudes.

1 INTRODUCTION

Groupware is a technology genre offering support to groups of people applying their collective intelligence to achieve a common goal [18]. Most often associated to this definition, we find the assumption that collective intelligence should be applied in a collaborative way. Under this assumption, conflicts are necessarily lean and may be resolved by focusing the group on the development of shared representations of problems, issues and alternatives [4].

However, substantive conflicting situations may arise in work groups and, indeed, may even be beneficial to collective intelligence as, e.g., documented by research on the impact of Devil's Advocacy in group decision making [28, 31]. In order to manage these conflicting situations, groups tend to give more importance to third-party support to the process, which includes negotiation, arbitration, mediation or facilitation techniques [3]. Therefore, the type of support offered by groupware must be adapted to address the process concerns.

Unfortunately, the currently available groupware tools tend to adopt one of the above assumptions but not both [26]. The type of groupware designated Group Support Systems (GSS) adopts the collaborative assumption, offering varied ways to organize group information while avoiding process support. Often, these tools do not even support private information, required by some bargaining techniques such as deception [7]. On the other hand, the type of groupware designated Negotiation Support Systems (NSS) offers plenty functionality necessary to handle conflicting situations, such as bid support or visualization of utility spaces [27], but neglects shared contents creation.

Our major research goal is to develop integrated groupware support along the whole conflict dimension. The scenario is illustrated in Figure 1, where we identify the two fundamental assumptions of the conflict dimension and summarize their major implications to groupware design. The type of integration we envisage and explore in this paper assumes that groupware support should cope with very dynamic changes in the group context, which may evolve rapidly within the two extremes represented in Figure 1.

Our approach to integration consisted in investigating the integration of the Issue Based Information System (IBIS, [23]) argumentation model with a negotiation model. Furthermore, we also developed a strategy bringing groupware users from

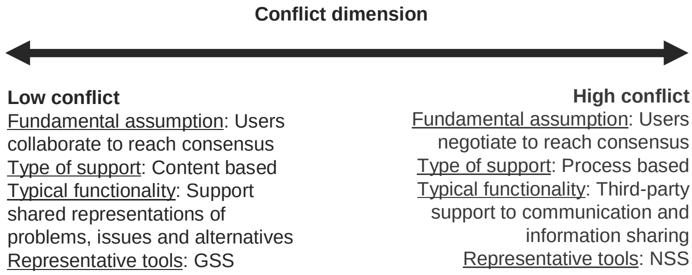

Fig. 1. The conflict dimension in groupware

the high conflict towards the low conflict side. This was accomplished by developing, at the user-interface level, benefits and resistances for, respectively, low conflict and high conflict attitudes taken by groupware users.

We demonstrate the proposed approach with a case study. The case study concerned the development of MEG, a groupware tool supporting the Quality Function Deployment (QFD, [1]) approach to software requirements evaluation in a real-world organization. Software requirements evaluation engages several stakeholders in the process of evaluating how far a software product under development accomplishes a set of previously established quality requirements. These stakeholders may have different perspectives over the software product and quality requirements, thus making this a rich and interesting context for exploring the conflict dimension in groupware.

This paper is structured in the following way. In the following section, we provide some background information about QFD and software requirements evaluation, necessary to understand the context and major design requirements behind MEG, the groupware tool developed in the case study. In Section three, we present an overview of several groupware tools addressing software requirements evaluation, showing that there is a clear dichotomy between low and high conflict features, and making the case for an integrated approach. In Section four we proceed with the discussion of the proposed integrated approach, discussing in particular the integration between the argumentation and negotiation models. We also define the strategy necessary to bring groupware users from high conflict towards low conflict situations. In Section five we describe in detail the MEG groupware tool. The qualitative results obtained from its preliminary use in the case study are presented in Section six. Finally, in Section seven we discuss the results obtained with this research and present some concluding remarks.

2 BACKGROUND INFORMATION ON QFD AND SOFTWARE REQUIREMENTS EVALUATION

Best engineering practices recommend that product quality should be addressed before and constantly evaluated during the product development. Furthermore, this vague notion of "product quality" should refer to concrete system attributes, addressing both the stakeholders' needs and technical activities necessary to deploy the product. This perspective is central in the Total Quality Management (TQM) trend adopted by many organizations pursuing excellence in software development [32].

QFD [1] is often used to implement TQM. QFD aims to define relationships between the users' and the technical requirements [29]. Although QFD was originally developed in manufacturing, it has more recently been adopted by software industry [32, 14]. In this later context, the fundamental value provided by QFD is focussing the software development process on the users' perspective: the Voice of the Customer (VoC, [2]). Although the traditional software development processes recognize the importance of the users, they do not offer simple mechanisms to verify the compliance with users' requirements through all development stages (lifecycle tracking, [29]). The Software QFD (SQFD, [14]) fills this gap in software engineering. The SQFD approach is also considered part of the Capability Maturity Model (CMM) level 4 implementation [2]. In a very simplified view, SQFD is a matrix of correlation values between requirements and specifications. This matrix is used in the following way [14]:

1. users' requirements are solicited to relevant stakeholders and placed in the left-hand side;
2. with the help from the stakeholders, the requirements are converted to technical specifications and placed at the upper side;
3. the stakeholders are then invited to complete the matrix with their perceived correlations;
4. a list of requirements priorities is defined; and
5. a list of technical specifications priorities is defined.

The correlations may be expressed in several ways, although a four-point scale ("none", "weak", "medium" and "strong",) is most often used [14]. The selection of a correlation is a qualitative task, where the objective is to identify the most appropriate link between "what" will be implemented and "how" the implementation corresponds to the stakeholders' expectations. Since there are many stakeholders involved, it is natural that different values may be proposed, according to different perspectives about the system, interpretations of what is involved in system development, hidden agendas, etc. Three alternatives for obtaining SQFD correlations have been documented in the literature:

1. requesting individual responses and averaging the results, possibly using a moderation factor such as the relative importance attributed to each stakeholder [29];
2. using multi-criteria preference analysis to combine preferences into some utility function [19], [9]; and
3. setting up a meeting, where the stakeholders must negotiate their different opinions until a consensus is achieved [8].

Although there are differences between the first two approaches, their focus is on the individuals, while the later approach stresses the commitment of the whole group to the SQFD process. This later approach is considered beneficial for team building, increasing the involvement in product development, obtaining overall consensus about "what to do", and preserving momentum when the group changes [8]. One problem with the later approach is that, being based on meetings and a definite need to negotiate and obtain consensus, the evaluation process may become time-consuming. MEG was specifically developed to resolve this problem, supporting parallel work and facilitating consensus on the SQFD matrix, thus reducing the required amount of time to accomplish the task.

3 OVERVIEW OF RELATED GROUPWARE TOOLS USING THE CONFLICT DIMENSION

In Table 1 we analyze several groupware tools dealing with software requirements evaluation, using the low and high conflict assumptions previously discussed. EasyWinWin supports the negotiation of systems requirements [5] following the WinWin principle [6] that all stakeholders should win. The tool guides the users through a process where winning conditions are identified and negotiated until mutual agreements are obtained. We observe that EasyWinWin neglects negotiation support, mostly because it is based on a generic GSS tool (GroupSystems) and depends on a human facilitator to resolve conflicts and manage the decision process.

The MEDIATOR system combines several visualization and communication technology with the purpose to support negotiations by consensus seeking [20]. The system offers graphical representations of the problem, showing the individual points of view, and actively searches for a point of equilibrium, suggesting compromising solutions. The system also uses a set of dimensions to define goals and utility spaces with support from a human facilitator. As a negotiation support system, MEDIATOR is highly focussed on the negotiation process.

Hermes facilitates the solution of ill-structured problems by supporting argumentative discourse among decision makers [22]. Hermes offers some low and high conflict features, and therefore is closely related with our approach. Hermes organizes arguments using IBIS, assists the negotiation process with updated information about the process status, recommends possible solutions, and also searches for inconsistencies among the users' preferences. Contrary to our proposed approach, Hermes does not offer any strategy bringing users from the high conflict towards the low conflict side.

Virtual QFD [16, 17] is an Internet-based support system that facilitates the software requirements discussion based on the QFD approach. The collection of available tools includes discussion panels, VoC tables, evaluation panels and QFD matrixes. The system is strictly focussed on low conflict situations and depends on a human facilitator to manage data during meetings and resolve conflicts.

	Low conflict	**High conflict**
EasyWinWin	Uses GroupSystems Win-Win methodology	No (relies on human facilitator)
MEDIATOR	No	Defines goals and utility spaces (constructed by human facilitator) Identifies equilibrium point Suggests compromises
Hermes	Discussion forum, based on IBIS	Finds conflicts in discussion Advises on actions for conflict resolution
Virtual QFD	Web-based tools: discussion panels, VoC tables, evaluation panels and QFD matrixes Human facilitator manages data during meetings	No (relies on human facilitator)
Co-Decide	Multi-user spreadsheet extension Offers OLAP features	No

Table 1. Comparison of related groupware tools using the conflict dimension

Co-Decide supports multi-criteria decision making based on a multi-user extension of the single-user spreadsheet functionality [13]. The basic idea behind Co-Decide is to extend typical OLAP features supplied by spreadsheets to multiple users. Co-Decide does not support the negotiation process.

Table 1 supports our previous argumentation that most of the available groupware tools adopt either the low or high conflict assumptions, but not both, in the specific context of software requirements validation. Hermes is the most notable exception, because it has functionality typical to low and high conflict situations, but nevertheless does not assume an integrated and flexible perspective over the problem.

4 AN INTEGRATED APPROACH TOWARDS CONFLICT RESOLUTION

We will discuss the proposed integrated approach within the specific context of our case study: applying the SQFD technique to validate software requirements. As previously mentioned, the SQFD model is defined as a matrix of cells correlating users' requirements with technical specifications; where the correlation values measure the preferences for the technical specifications to fulfil the satisfaction of the corresponding requirements. For reading convenience, the correlation values will be referred as C.

MEG supports several stakeholders working in parallel and expressing their different preferences for C for each cell. This is an extension of the original SQFD model made possible by the groupware functionality, since the original model associates one single C to one cell. In Figure 2 we illustrate the extended SQFD model.

The extended model allows the stakeholders to construct a shared representation of the problem and its alternative solutions. We integrated the IBIS model in this shared representation to support the stakeholders' argumentation, since IBIS is adequate to represent and structure several issues, positions and arguments. In Figure 2 we illustrate how SQFD and IBIS are combined to organize the shared information pertaining to one cell. The approach regards the proposal of C in a cell as an issue, which may have associated positions and arguments. Note that the integration of SQFD and IBIS addresses the low conflict side of the conflict dimension, since it does not tackle the negotiation process necessary to resolve conflicts.

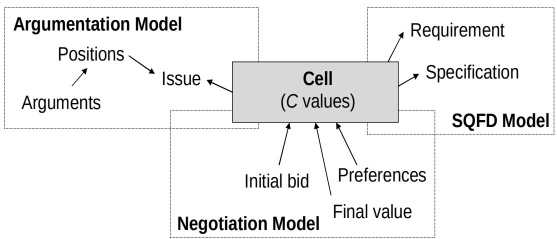

Fig. 2. Integration of the SQFD, argumentation and negotiation models

In order to address the high conflict situation, we integrate a negotiation model with the previous ones, as shown in Figure 2. The stakeholders, in accordance to their preferences, may propose several C values for a cell. The first occurring preference for C is classified as an initial bid and generates an issue. All subsequent preferences will be contrasted with the first one and automatically produce positions against or in favour of the issue, thus linking the negotiation and argumentation models. Of course, any C values that are equal to the initial bid are considered in favour while different values are classified as against. This means that, unlike what is common in IBIS, the issues and positions are not directly specified by the stakeholders but instead automatically derived by the system based on the initial bid and subsequent users' preferences. The occurrence of positions against an issue denotes a conflicting situation, for which a negotiation process is necessary to reach consensus. The negotiation process exists until a final C is accepted by all stakeholders.

One further important aspect related with the integration of SQFD, argumentation and negotiation models concerns the degree of information shared between the stakeholders. As we mentioned previously, very often groupware assumes that the group has a shared goal and conflicts may be resolved with the support to shared representations of problems, issues and alternatives. On the contrary, a negotiation process assumes the conflict between the parts involved, turning more difficult the creation of shared representations. The integration of both perspectives thus creates some tensions between the support to individual goals and shared representations. These observations lead us to define a clear frontier between the stakeholders' hidden and shared knowledge [21] in the integrated model: the C expressed by the initial bid is made public, but the following individual preferences are kept undisclosed to the group; only positions and arguments are made public.

Having defined our integrated approach, we will now delineate the strategy necessary to bring the stakeholders from high towards low conflict. The strategy will be illustrated in the context of negotiating one cell, considering that all cells may be handled in parallel the same way.

The negotiation behaviour can be analysed according to two different strategies paradigmatic in negotiation research [25]:

integrative, where an agreement is found in an inventive and collaborative way, exchanging information about preferences and priorities, and seeking common gains – both parts win (Win-Win);

distributive, persuading the other part to accept an offer while disregarding the counteroffers – this is a game of wining and loosing (Win-Lose).

The negotiation process often follows a differentiation-before-integration pattern [15], where the negotiation starts with distributed behaviours until an impasse is reached, and then the participants switch to integrative behaviours to avoid failure.

Most academic and non-academic literature shows a bias towards the integrative strategy [24], because of two main reasons:

1. it represents a zero-sum solution, since the gains obtained by one party represent losses from the other; and
2. the fundamental values behind the integrative strategy – interpersonal trust, cooperation and search for mutually acceptable outcomes – are favoured by most scientists' value systems.

We also follow the policy of favouring the integrative strategy.

According to Harinck and Dreu [15], the switch to the integrative behaviour requires the combination of two conditions: an impasse and the willingness to engage in integrative behaviours. We propose another alternative: using groupware to foster users engaging in integrative behaviours. To accomplish this endeavour, we have to further explain the differences between the integrative and distributive strategies, based on the following set of negotiation attitudes [30]:

Competition – when one party tries to convince the other to accept a stake that is only favourable to self interests. This clearly corresponds to a Win-Lose attitude.

Collaboration – when both parties collaborate to maximize common gains (Win-Win).

Compromise – when both parties split the benefits. It is a satisfactory, although not necessarily an optimal result, since each party may not achieve all intended goals. This attitude leads to moderate Win-Win situations.

Obliging – when one party accepts a stake that is only favourable to the other party. This attitude occurs for several reasons, e.g. to close rapidly the process or simply because the issue is perceived as not important. This is usually considered a Lose-Win attitude. However, the literature reports that the obliging effects are unclear on the long run [12]: producing positive effects by eliminating conflicts, but on the other hand losing the opportunity to maximize common gains. In our context, we regard this attitude as neutral in terms of integrative/distributive behaviours. This view assumes that in SQFD parties engage in multiple negotiations, and thus the importance of a single Lose-Win is reduced.

Avoidance – when one or both parties decide to retreat. If there is a dependency on the negotiation process, this attitude frustrates the other's intentions (Lose-Lose strategy). It is also used, for instance, when one party seeks to use time pressure to own benefits (pursuing a Win-Lose strategy).

Based on the above attitudes, we adopted the following generic strategy:

1. favour Win-Win behaviours, which includes support to collaboration and compromise;
2. provide some resistance to Win-Lose and Lose-Lose behaviours, i.e. competition and avoidance; and
3. be neutral about the Lose-Win behaviour, i.e. obliging.

5 THE MEG GROUPWARE TOOL

This section discusses the application of the previously described integrated approach in MEG, a groupware tool supporting SQFD for requirements validation. MEG was implemented with MS Excel 2002, Access and Visual Basic 6.0 and adopts the client-server architecture shown in Figure 3. The SQFD matrix was implemented with an Excel spreadsheet and RTD technology. This SQFD matrix provides a global perspective of the multiple ongoing negotiations, each one related with a SQFD cell, for which the stakeholders much reach an agreement.

Although the stakeholders may interact with the SQFD spreadsheet to analyze the ongoing negotiations, they cannot set or modify the cell values directly in the spreadsheet. Those modifications are accomplished in a different tool, designated MEGCLIENT, which communicates with MEGSERVER, which in fact has the responsibility to modify the cell values. Beside resolving the common synchronization problems that occur with collaborative technology, MEGSERVER is also responsible for maintaining the consistency and persistency of the shared information. MEGSERVER uses an Access database for persistency support and relies on XML to communicate with the database.

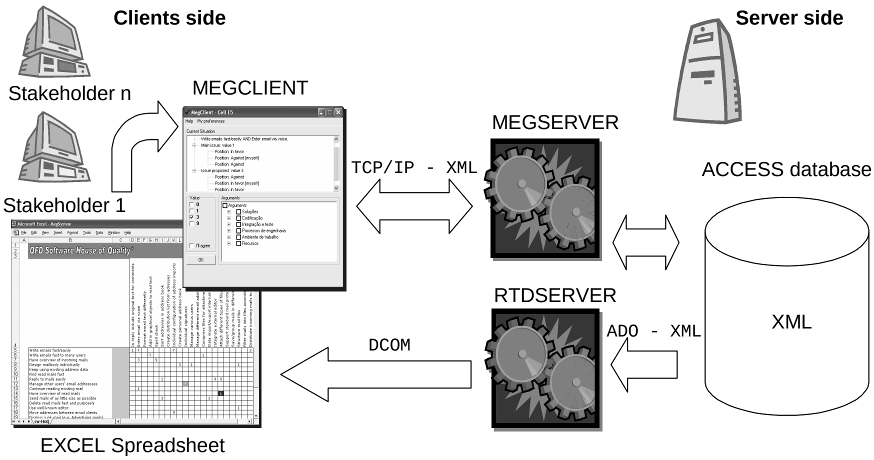

Fig. 3. System architecture

This architecture also includes an additional tool, designated RTDSERVER, which is responsible for updating the distributed spreadsheets whenever a modification in the shared information occurs. The communication between the RTDSERVER and the Excel spreadsheets relies on DCOM [10].

After this necessarily brief introduction to MEG, the remaining of this section is organized in the following way. Next, we describe in detail the integrated model implementation. Then we proceed with an explanation of the implemented benefits and resistances aiming to influence the stakeholders towards low conflict attitudes. Then we provide a brief illustration of the MEG functionality.

5.1 Integrated Model Implementation

We start describing the SQFD model details. Given a cell with a pair (requirement, specification), the SQFD model specifies the corresponding correlation value, $SQFD : R \times SP \rightarrow \{0, 1, 3, 9\}$, where R is the set of requirements, SP is the set of specifications and $\{0, 1, 3, 9\}$ is the set of feasible correlation values (a zero value corresponds to an empty cell). The $\{0, 1, 3, 9\}$ set is the most often used equivalence to the adopted "none", "weak", "medium" and "strong" correlations [14]. Initially, $SQFD_{rs} = 0$, $\forall r \in R$, $\forall s \in SP$, meaning that all cells are empty. Without loss of generality, in the following, we will consider a generic cell and the negotiation of the related C.

The initial bidder is the first stakeholder specifying a non-zero C, while the value specified is the initial bid. Our model associates the initial bid to an *ISSUE*, defined as

$$ISSUE = (initial - bid,\ initial - bidder)$$

where $initial - bid \in \{\, 1, 3, 9\}$, $\; initial - bidder \in ST$, and ST is the set of stakeholders.

The initial bid is public and its instantiation opens up the opportunity for other stakeholders to express their own preferences. All subsequent stakeholders attributing a value to the cell will be treated as supporters or opponents to the initial bidder. A stakeholder may instantiate more than one preference for C. $PREF(S_i)$ specifies the preferences' tuple of stakeholder S_i:

$PREF(S_i) = \langle c_1, \ldots, c_k \rangle$, where $S_i \in ST$, k is the tuple size ($0 \leq k \leq 3$), and $c_j \in \{1, 3, 9\}$ is the j^{th} preference value S_i stated ($j \leq k$).

Only stakeholders participating in the definition of C have a non-empty $PREF$ tuple ($k > 0$). These preferences are part of the hidden knowledge maintained by the system, since individual preferences are kept undisclosed to the other stakeholders. Based on issues and preferences, MEG identifies the stakeholders' positions as supporters and/or opponents to the initial bid. Whenever a stakeholder has a set of preferences compatible with the *ISSUE* (i.e. considering stakeholder S_i, at least one of the values in $PREF(S_i)$ is equal to the initial-bid), MEG registers a position in favour of the initial bid. When there is no such compatibility, MEG registers a position against. This is done by computing $POSITION(S_i)$ for all $S_i \in ST$, where

$$POSITION(S_i) = \begin{cases} In - Favour, \text{if } \exists j : PREF(S_i)_j = initial - bid \\ Against, \text{otherwise.} \end{cases}$$

Stakeholders are offered the possibility of attaching arguments to their positions, which confers them additional negotiation abilities. This means that for a stakeholder S_i a tuple of arguments may be defined, which are defined by

$$ARGUMENTS(S_i) = \langle Arg_1, \ldots, Arg_k \rangle, S_i \in ST, Arg_j \in Ontology, j \leq k, 0 \leq k.$$

An argument is a very short piece of text, such as "human factors" or "failure". MEG assumes the ontology necessary to implement this functionality has been previously supplied. The idea behind this approach is that the stakeholders do not have to write their own arguments; they can select relevant and meaningful ones from the ontology. We have not addressed this aspect in great detail, since the ontology varies from organization to organization. In our case study we relied upon generic roadmaps for quality assurance provided by software engineering literature.

Now we move our attention to the negotiation support, whose description is based on the states machine shown in Figure 4.

Our model assumes that, if there is at least one position against the initial bid, then there is a conflicting situation requiring a negotiation process. To handle that process, MEG deals with the concept of negotiation state. We consider an *Equilibrium* state, referred to by E, that is reached whenever there is no ongoing negotiation, either because:

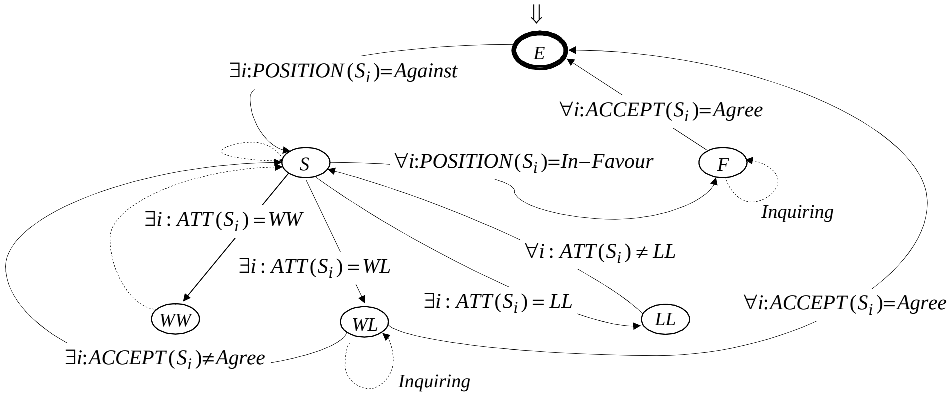

Fig. 4. Negotiation states

a) the cell has no preference assigned;

b) has one single C; or

c) there is no position against the current C.

A negotiation is successful whenever its end leads to state E, i.e., E is both the starting state and the only accepting state. All the other states (denoted by S, F, WW, WL and LL) are *negotiation* states. In Figure 4, plain arcs correspond to transitions associated with user actions, while dashed lines represent transitions related with system events (such as evaluating positions or attitudes).

The machine moves from E to S when at least one stakeholder has a position against the initial bid, thus starting a negotiation. F is reached when all positions against the initial bid have disappeared and the negotiation process is close to a final one. MEG then requires all stakeholders involved in the negotiation to explicitly agree to finish the process (thus moving to E). $ACCEPT(S_i)$ denotes this explicit acceptance of stakeholder S_i when inquired by MEG. If S_i agrees to finish the negotiation $ACCEPT(S_i) = Agree$ and, otherwise, $ACCEPT(S_i) = Not - Agree$.

The other states intimately relate with the integrative and distributive attitudes we have previously identified. Stakeholder S_i may take an attitude $ATT(S_i)$ of the following types: Win-Win (WW), Win-Lose (WL), Lose-Win (LW) or Lose-Lose (LL).

The state WW is reached whenever a stakeholder takes a WW attitude. In this case, MEG re-calculates the set of positions (returning to S) and, if the conflict has disappeared (thus moving to F), attempts to finish the negotiation. The WL state is reached whenever a stakeholder changes preferences in a WL attitude. Movements out of WL depend on the result of users' inquiry. Finally, the LL state is reached whenever a stakeholder adopts a LL attitude, a situation that requires MEG to suspend the cell negotiation until that attitude is revoked.

To understand the MEG functionality we also have to specify how these different attitudes are detected by the system. The specification is provided in Table 2.

Attitude	**Detection**
Win-Win	*PREF* became "closer" to *initial-bid*
Win-Lose	"Firm" option has been selected (see section 5.2 for explanation)
Lose-Win	*PREF* has been removed
Lose-Lose	"Block" option has been selected (see section 5.2 for explanation)

Table 2. Behavior detection

5.2 Influencing the Stakeholders Towards Low Conflict Attitudes

We previously declared the objectives to favour low conflict and resist to high conflict attitudes. We now describe in detail how we addressed these issues in MEG. Expressing the problem in more concrete terms, our objective is to facilitate Win-Win, be neutral about Lose-Win, and create difficulties to Win-Lose and Lose-Lose attitudes.

According to [11] an integrative strategy is founded on "principled negotiation":

1. separate people from problems;
2. focus on interests, not on positions;
3. create options for mutual gains; and
4. use objective criteria.

Our solution addresses these principles in the following ways:

- The stakeholders' identities are undisclosed. When a stakeholder originates an issue, position or argument, the information about who took that action is not displayed. This approach allows separating people from the problem.
- MEG does not show the stakeholders' preferred C, but only their positions relatively to the initial bid. This approach gives some latitude to changing positions and allows focussing more on interests than positions. MEG also allows the stakeholders to freely change their positions at any time during the negotiation.
- MEG creates opportunities for mutual gains by proposing a consensus value. The calculus of the consensus value is explained below.
- The ontology provides a standard mechanism for objectively arguing in favour or against an issue.

Whenever possible, under a conflicting situation, MEG proposes a consensus value for C that is obtained in the following way. The weight of stakeholder S_i in the negotiation is given by $SW(S_i) = 1 - (n_i \cdot 10^{-3})$, that decreases with n_i, the number of Win-Lose or Lose-Lose attitudes S_i has taken in the past. $UP(S_i, x)$ is the un-weighted preference for the correlation value $x \in \{1, 3, 9\}$ stated by S_i, while $WP(S_i, x)$ is the corresponding weighted preference, now considering the stakeholder weight in the negotiation. These values are given respectively by

$$UP(S_i, x) = \begin{cases} 1, \text{if } \exists j : PREF(S_i)_j = x \\ 0, \text{otherwise} \end{cases}$$

and

$$WP(S_i, x) = UP(S_i, x) \cdot SW(S_i).$$

$P(x) = \sum_{S_i \in ST} WP(S_i, x), x \in \{1, 3, 9\}$, computes the total preference for x expressed by the stakeholders. And finally, $CONSENSUS = c : P(c) = \max\{P(x), x \in \{1, 3, 9\}\}$ is the correlation value that obtained the highest number of occurrences in all the preferences' tuples, or Null if there is no such value.

In summary, we used majority voting, where votes are weighted according to the number of distributive attitudes taken during the system use. This approach is aiming at benefiting the stakeholders that take integrative attitudes. When a *CONSENSUS* value is obtained, MEG proposes it as a fair solution to the negotiation process, on par with the initial bid. MEG does not enforce the stakeholders to accept that value.

Now, we turn our attention to the mechanisms built in MEG to create difficulties to Win-Lose and Lose-Lose attitudes. MEG allows Lose-Lose attitudes using a "blocking" mechanism (mentioned in Table 2). Basically, the blocking mechanism allows one stakeholder to lead the negotiation to a suspended state (*LL*), so that the process stops until the stakeholder removes that condition or the SQFD task is concluded without consensus. To create some resistance to this attitude, MEG makes the user interaction with this mechanism difficult: the action is not easily accessible and several confirmations are required before activation.

MEG allows Win-Lose attitudes using a "firm" mechanism: one stakeholder may express to the others that he/she has a firm position about C. When this mechanism is activated, MEG informs all the other stakeholders and asks them if they accept that position or not (moving to state *WL*). In case all stakeholders accept, the negotiation process is finished, otherwise the negotiation continues. To create resistance to the usage of this mechanism, when MEG informs the stakeholders that someone has a firm position, it also informs about the total number of similar attitudes taken by that stakeholder. This information may influence the stakeholders not to accept firm positions from persons that have wield too many distributive attitudes in the past.

5.3 Illustration of MEG Functionality

Consider a set of three stakeholders: S1, S2 and S3. Figure 5 shows one SQFD matrix with several conflicting situations and ongoing negotiations.

Observe that some cells present correlation values (1, 3 and 9, since 0 corresponds to a blank cell), while some others show the symbols "?", "F" and "L". These

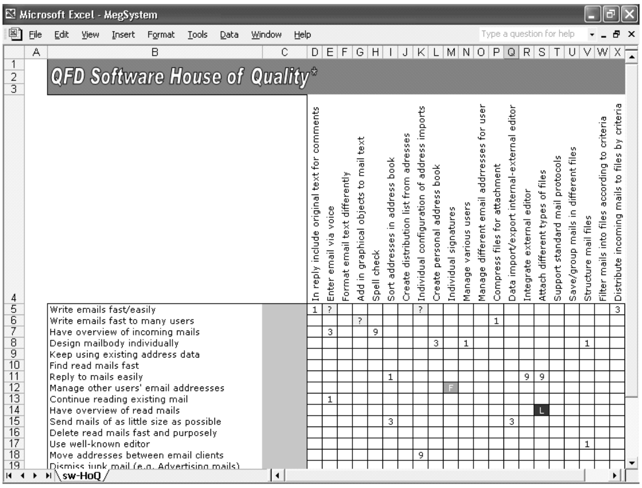

Fig. 5. SQFD matrix (example from [17])

symbols are shown when the cell is under negotiation. The "?" indicates that the process is ongoing; while "F" and "L" indicate that a user expressed a firm position and locked the cell, respectively. The users do not directly manipulate the SQFD matrix. Instead, MEGCLIENT is invoked whenever one user double clicks on a cell.

Time	**S1**	**S2**	**S3**	**Action**
t1	1			S1 selects $C = 1$
t2				S2 analyses cell
t3		3		S2 selects $C = 3$
t4				S3 analyses cell
t5			3	S3 selects $C = 3$
t6				MEG proposes $C = 3$
t7	1, 3			S1 adds $C = 3$ to selection
t8				MEG requests agreement

Table 3. Sequence of actions accomplished by S1, S2 e S3 and system events

We use the sequence of actions shown in Table 3 to illustrate how users interact with MEGCLIENT and the corresponding system reaction.

Using MEGCLIENT to modify the SQFD cell E5, S1 selects $C = 1$. Since the cell was previously empty, MEG creates an issue with 1 as initial bid and propagates it through the system. 1 will appear in E5 for all stakeholders. Figures 6 illustrate interaction of S2 with MEG.

Afterwards, S2 decides to analyse E5, double clicking E5 to open MEGCLIENT. The issue is displayed, showing the proposed correlation but without identifying S1 as initial bidder (Figure 6/left-up). S2 does not agree with the correlation and

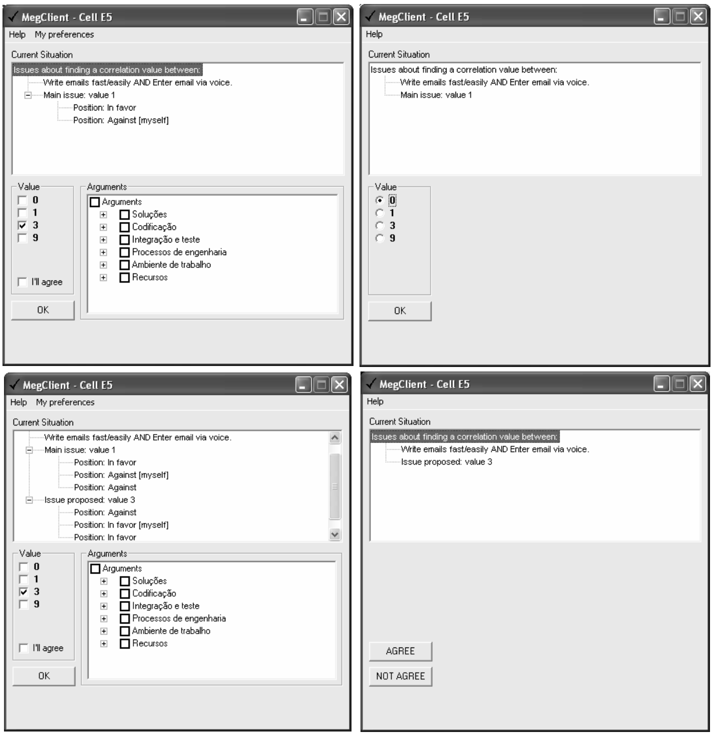

Fig. 6. MEGCLIENT for S2 in t2 (left-up), in t3 (right-up), in t6 (left-bottom) and in t8 (right-bottom)

selects C=3. MEG recognizes two conflicting proposals for E5 and initiates a negotiation process. The preferences list is constructed with one supporter (S1) and one opponent (S2) to the initial bidder (Figure 6/right-up). Note that the identity of the supporters and opponents is undisclosed. Furthermore, a "?" appears in E5.

S3 decides to enter the negotiation and proposes C=3. MEG recalculates the preferences, to come with one supporter and two opponents to the issue. MEG also analyses if there is a consensus value. Since no previous "firm" or "block" positions have been used, the obtained consensus value is 3 ($P(1) = 1$, $P(3) = 2$ and $P(9) = 0$). Therefore, MEG proposes 3 to the stakeholders (Figure 6/left-bottom).

S1, analysing the consensus value, decides to adopt a compromising attitude and adds 3 to the range of accepted correlations. MEG realizes there is one possible agreement on 3 and requests confirmation from all stakeholders (Figure 6/right-bottom). All users agree and the negotiation process finishes. Value 3 finally appears in cell E5.

6 EVALUATING THE APPROACH

MEG was evaluated in two pilot experiments involving two stakeholders each. The participants had the following background:

(A) more that 30 years experience in software development and requirements negotiation with outsourcing organizations;

(B) 6 years experience in systems analysis;

(C) project coordinator; and

(D) analyst/programmer in statistics and operational research.

The experiments were accomplished in the context of a governmental agency responsible for the national pensions system. The project concerned the introduction of a new formula for computing pensions. The goal set for the pilot experiments was to construct and evaluate the SQFD matrix designated as "House of Quality" (HoQ). The HoQ correlates preliminary lists of user and technical requirements, so that priorities can be set early in the project. The HoQ was specified in the following way. We interviewed stakeholder A, who is deeply knowledgeable about the problem context. His recommendations allowed us to specify the user and technical requirements. We followed ISO/IEC 9126 to finally structure the quality requirements:

Functionality	Apply new formula Integrate with current formulas in the pension application
Reliability	Detailed contingency plan
Usability	Provide adequate training Document new functionality Clearly define modifications to existing processes
Maintenance	Add new functionality with minimum operational modifications
Portability	Detailed migration plans

Table 4 presents the list of technical requirements that was specified by the authors based on the recommendations of stakeholder A.

The resulting SQFD is therefore an 8x24 matrix with 192 correlation values. Each pilot experiment started with a brief tutorial about MEG, which took approximately about 15 minutes. Then, the SQFD was negotiated by a pair of stakeholders until a consensus was obtained. During the experiment, whenever necessary, additional help about the MEG functionality was provided by one of the authors, which participated in the process as observer.

Beyond obtaining the SQFD matrixes with correlations, we requested the participants to fill up a questionnaire with open questions about the most positive and negative aspects about MEG usage, as well as specific questions concerning the MEG functionality and usability.

The evaluation results allowed us to arrive at several important conclusions. Regarding MEG functionality, *the system is convenient to use* (the available functions

Functionality	Calculus of pensions for person P Calculus and demonstration of pensions for entities E1 and E2 Store data according to user profile Client can operate in different OS Display specific legislation used in calculus User authentication
Reliability	Complete a transaction cycle without execution errors Continue operation if data is not available Continue operation if write error Use secondary server if main server fails Recover operations completed before a power failure
Usability	Organize items logically in screen Provide online explanations of calculus Provide online help Alert that new functionality is available
Maintenance	Show how to realize calculus Support configuring codes and parameters Use several modules Reuse some modules Facilitate data access in every screen Data must look the same in every printer
Portability	Export data to Excel Use install rules from internal doc PPP/2004 Use portability rules from internal doc 002/2004

Table 4. List of technical requirements

are appropriate for the task at hand), *the system is accurate* (the results reflect the participants' opinions), and *the embedded consensus approach, complemented by majority voting, is beneficial*. Individual positive comments (in Figure 7) have strengthened these main features. For instance, comments about understanding the overall positions from others (participant B), revising own positions and ease finding agreements (participant A) reinforce the convenient use of the system. Integrating negotiation attitudes with the QFD promotes system accuracy (participant D), as well as the efficiency of the underlying negotiation model (C). Additionally, the stakeholders agree that the results reflected their opinions. Keeping others positions undisclosed (B), knowing arguments from others and revising own positions (A) support the ease to reach consensus.

Concerning usability, the experiences indicated that the *participants could understand the working logic of MEG* and that *they easily learned to deal with MEG functionality, as well as with the negotiation operating process*. However, it was pointed out that the system is difficult to use by non experienced users. Another drawback relates with bad performance (A and B), but that is caused by DCOM and therefore does not impact negatively the proposed approach. Several minor functional and user interface details were also raised by the stakeholders, for instance,

Positive Aspects	**Participants**
Easy finding point of agreement	A
Knowing arguments from others to evaluate and eventually revise my position	A
Better understanding of the overall ideas from stakeholders	B
The "current situation" closes every time a change is made, which is positive because it obliges to read modifications	B
The system does not show how many others have confirmed their positions	B
The negotiation model is efficient, although for top management it should be more graphical	C
The integration of negotiation attitudes with the QFD affords obtaining reliable results	D
Negative Aspects	
The system is slow	A, B
Unusable by common users	A
It is more intuitive to qualify correlations by names than numbers	A
The situation where all stakeholders are in favour but one does not press the option "I agree" is confusing, because the consensus was rejected but all were in favour	A
If 2 stakeholders obtain an agreement, that value goes to a cell. If another pair negotiates a different value, the initial pair is not informed	B
The information shown in "current situation" should be presented graphically	C
The graphics should be more intuitive. For instance, it is more intuitive for a manager to see that there are N stakeholders in favour or against a value	C
The value obtained by consensus by a group of stakeholders may be substituted by another group of stakeholders without notifying the first one	D

Fig. 7. Positive and negative aspects of MEG utilization

about the use of the "I agree" button, renegotiation of cells overriding previous consensus, the absence of graphical information, and difficulties in obtaining summary view of the negotiation processes. Again, these comments should be used in future versions of MEG but do not reflect any significant issues about the core design decisions.

In both experiences, a consensus was reached in a short period of time. The stakeholders have rarely taken distributive attitudes. Another interesting conclusion stated by the participants was that the stage of learning the system utilization gave them new insights on negotiating software requirements. Although outside of our research scope, we may envisage that the possibility of taking integrative attitudes in the negotiation promotes a deeper understanding of the process. Therefore, it may well contribute to reduce its inherent complexity.

Finally, we would like to emphasise that although only four individuals participated in the experiments, their deep understanding of the problem area and application domain give credit to the benefits achieved.

7 DISCUSSION AND CONCLUSIONS

This paper proposes a groupware approach supporting a broad range of different levels of conflicting attitudes. We argue that most of the existing groupware is positioned in one of the sides of conflict dimension – either on the side of low conflict or on the side of high conflict. These perspectives involve different strategies to reach participants agreement, some of them of opposite nature, which become deeply rooted in the system functionality. With the intent to fill the gap between these single opposite perceptions, our proposal is to integrate strategies from both of them.

As a case study, we applied this combination of negotiation and argumentation in the development of MEG, a groupware tool supporting SQFD and software requirements validation.

MEG allows users to construct a shared representation of problems, alternative solutions and arguments, and at the same time engage into multiple parallel negotiations, expressing their preferences, firm positions and other barriers to negotiation. This functionality is orchestrated by the combination of argumentation and negotiation models. Although MEG was developed in a specific context, we believe the integrated argumentation and negotiation models may be applied to many other circumstances where users concurrently manipulate a large set of shared objects and must identify and resolve potential conflicting views over these objects.

Attitude	**Design decision**
Win-Win	• Keep identities undisclosed (separating people from the problem) • Hide individual preferences, showing positions only (giving latitude to change positions) • Support multiple individual preferences and automatic agreement within the set (avoiding unnecessary conflicts) • Propose consensus value (based on users' preferences and majority voting) • Use ontology to support argumentation, reducing the levels of conflict by exchanging standard messages, meaningful in the domain, instead of free text (objective argumentation)
Win-Lose and Lose-Lose	• Reduce accessibility, making options difficult to access and requesting unnecessary confirmations by the users • Maintain an historical record of these attitudes and display it to users when relevant (closing negotiation) • Associate a cost to these attitudes and calculate the consensus value taking in consideration that cost (weighted preferences)
Lose-Win	• Preferences may be removed at any time (immediate positive effects by eliminating conflict) • Preferences may be set again at any time, even to reinitiate a closed negotiation (long-term opportunity to maximize common gains) • Support multiple negotiations (reducing the impact of single Lose-Win situations)

Table 5. Set of design decisions stimulating integrative attitudes

Moreover, a unique characteristic of our approach is that it attempts to stimulate users to assume integrative attitudes based on the set of subtle design decisions, most of them with impact on the user-interface level, as presented in Table 5.

One interesting outcome from the combination of low and high conflict support is that the resulting tool offers more latitude and flexibility handling group strategies: the tool supports low-conflict collaborative situations, but is also capable to cope with increased levels of conflict in a flexible way.

Although more testing is needed (models' integration should be further evaluated in other contexts), the results from the pilot experiments indicated that the integration approach is beneficial for reaching consensus.

Acknowledgements

This paper was partially supported by the Portuguese Foundation for Science and Technology, Projects POSI/EIA/57038/2004 and POCTI/ISFL/1/152.

REFERENCES

[1] Akao, Y.: Quality Function Deployment: Integrating Customer Requirements into Product Design. Productivity Press, Cambridge, Massachusetts, 1990.

[2] Antoniol, G.—Gradara, S.—Venturi, G.: Methodological Issues in a CMM Level 4 Implementation. Software Process Improvement and Practice, Vol. 9, 2004, No. 1, pp. 33–50.

[3] Antunes, P.—Ho, T.: The Design of a GDSS Meeting Preparation Tool. Group Decision and Negotiation, Vol. 10, 2001, No. 1, pp. 5–25.

[4] Bichler, M.—Kersten, G.—Strecker, S.: Towards a Structured Design of Electronic Negotiations. Group Decision and Negotiation, Vol. 12, 2003, No. 4, pp. 311–335.

[5] Boehm, B.—Grünbacher, P.—Briggs, R.: Developing Groupware for Requirements Negotiation: Lessons Learned. IEEE Software, Vol. 18, 2001, No. 3, pp. 46–55.

[6] Boehm, B.—Ross, R.: Theory-W Software Project Management Principles and Examples. IEEE Transactions on Software Engineering, Vol. 15, 1989, No. 7, pp. 902–916.

[7] Boles, T.: Deception and Retribution in Repeated Ultimatum Bargaining. Organizational Behavior and Human Decision Processes, Vol. 83, 2000, No. 2, pp. 235–259.

[8] Bouchereau, V.—Rowlands, H.: Quality Function Deployment: The Unused Tool. Engineering Management Journal, Vol. 10, 2000, No. 1, pp. 45–52.

[9] Chuang, P.: Combining the Analytic Hierarchy Process and Quality Function Deployment for a Location Decision from a Requirement Perspective. The International Journal of Advanced Manufacturing Technology, Vol. 18, 2001, No. 11, pp. 842–849.

[10] Cornell, P.: Building a Real-time Data Server in Excel 2002. Technical report, Microsoft Corporation, 2001.

[11] Fisher, R.—Ury, W.: Getting to Yes: Negotiating Agreement Without Giving In. Penguin Books, New York, 1983.

[12] Friedman, R.—Tidd, S.—Currall, S.—Tsai, J.: What Goes Around Comes Around: The Impact of Personal Conflict Style on Work Conflict and Stress. The International Journal of Conflict Management, Vol. 11, 2000, No. 1, pp. 32–55.

[13] Gebhardt, M.—Jarke, M.—Jacobs, S.: A Toolkit for Negotiation Support Interfaces to Multi-dimensional Data. ACM SIGMOD Record, Vol. 26, 1997, No. 2, pp. 348–356.

[14] Haag, S.—Raja, M.—Schkade, L.: Quality Function Deployment Usage in Software Development. Communications of the ACM, Vol. 39, 1996, No. 1, pp. 41–49.

[15] Harinck, F. —Dreu, C.: Negotiating Interests or Values and Reaching Integrative Agreements: The Importance of Time Pressure and Temporary Impasses. European Journal of Social Psychology, Vol. 34, 2004, No. 5, pp. 595–611.

[16] Herzwurm, G.—Schockert, S.: Virtual Product Development. In Proceedings of the Fifth International Symposium on Quality Function Deployment and the First Brazilian Conference on Management of Product Development, Belo Horizonte, Brazil, 1999.

[17] Herzwurm, G.—Schockert, S.—Dowie, U.—Breidung, M.: Requirements Engineering for Mobile Business Applications. In Proceedings of the First International Conference on Mobile Business, Athens, Greece, 2002.

[18] Hiltz, S.—Turoff, M.: The Network Nation: Human Communication Via Computer. The MIT Press, Cambridge, 1978.

[19] In, H.—Olson, D.—Rodgers, T.: Multi-criteria Preference Analysis for Systematic Requirements Negotiation. In Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC '02), Oxford, England, 2002, pp. 887–892, IEEE Computer.

[20] Jarke, M.—Jelassi, T.—Shakun, M.: Mediator: Toward a Negotiation Support System. European Journal of Operational Research, Vol. 31, 1987, No. 3, pp. 314–334.

[21] Johannessen, J.—Olaisen, J.—Olsen, B.: Information Management in Negotiations: The Conditions Under Which It Could Be Expected That the Negotiation Partners Substitute a Competitive Definition of the Situation for a Cooperative One. International Journal of Information Management, Vol. 17, 1997, No. 3, pp. 153–168.

[22] Karacapilidis, N.—Papadias, D.: Computer Supported Argumentation and Collaborative Decision Making: The hermes System. Information Systems, Vol. 26, 2001, No. 4, pp. 259–277.

[23] Kunz, W.—Rittel, H.: Issues as Elements of Information Systems. Technical report, Institute of Urban and Regional Development, University of California at Berkeley, 1970.

[24] Lewicki, R.—Weiss, S.—Lewin, D.: Models of Conflict, Negotiation and Third Party Interventions: A Review and Synthesis. Journal of Organizational Behavior, Vol. 13, 1992, pp. 209–252.

[25] Lewicki, R.—Litterer, J.: Negotiation. R.D. Irwin Inc., Homewood, Illinois, 1985.

[26] Ramesh, B.—Mohan, K.: Integrating Group Decision and Negotiation Support Systems with Work Processes. In IEEE Computer Society, editor, Proceedings of the 34^{th} Hawaii International Conference on System Sciences, Maui, Hawaii, 2001.

[27] Rubenstein-Montano, B.—Malaga, R.: A Co-evolutionary Approach to Strategy Design for Decision Makers in Complex Negotiation Situations. In Proceedings of the 33^{rd} Hawaii International Conference on System Sciences, Hawaii, 2000.

[28] Schwenk, C.—Valacich, J.: Effects of Devil's Advocacy and Dialectical Inquiry on Individuals Versus Groups. Organizational Behavior and Human Decision Processes, Vol. 59, 1994, pp. 210–222.

[29] Stylianou, A.—Kumar, R.—Khouja, M.: A Total Quality Management-based Systems Development Process. The DATA BASE for Advances in Information Systems, Vol. 28, 1997, No. 3, pp. 59–71.

[30] Thomas, K.—Kilmann, R.: Thomas-Kilmann Conflict Mode Instrument. Xicom, Tuxedo, NY, 1974.

[31] Valacich, J.—Schwenk, C.: Devil's Advocacy and Dialectical Inquiry Effects on Group Decision Making Using Computer-mediated Versus Verbal Communication. Organizational Behavior and Human Decision Processes, Vol. 63, 1995, No. 2, pp. 158–173.

[32] Zultner, R.: Tqm for Technical Teams. Communications of the ACM, Vol. 38, 1993, No. 10, pp. 79–91.

Pedro Antunes is associate professor at the University of Lisboa. His research interests address the design, development and assessment of collaborative technologies, focusing in particular on complex socio-technical interactions such as electronic meetings and group decision and negotiation. He has published more than 70 papers on these topics and participated in more than 40 conference program committees. He is Steering Committee Member of CRIWG – International Workshop on Groupware. Since 2000, he has also been project evaluator and reviewer for the European Commission in several key actions: eHealth, collaborative working environments, new methods of work, and education and training.

João Ramires received his Masters Degree in 2004 from the University of Lisboa, advised by Pedro Antunes. His research work focussed on the study and development of collaborative requirements negotiation tools. He works for the National Pension Center of the Portuguese Public Administration.

Ana Respício is auxiliary professor at the University of Lisbon. She is a member of the Centre for Operational Research and of the Group of Studies in Social Simulation (GUESS). Her research interests include decision support in complex scenarios, optimization, and simulation systems. She has been involved in several research and development projects with industry, having gained experience in the design of decision support systems integrating multi-level decisions and avoiding conflict. She has published several papers in conferences and journals. She is a member of Ifip-WG 8.3 and of the EuroWG on DSS.