Software Requirements Negotiation Using the Software Quality Function Deployment

João Ramires[1,4], Pedro Antunes[1,3], Ana Respício[2,3]

[1] LaSIGE – Large Scale Information Systems Laboratory
[2] Centro de Investigação Operacional
[3] Department of Informatics of the Faculty of Sciences of the University of Lisboa, Bloco C6, Campo Grande, 1700 Lisboa, Portugal
{paa,respicio}@di.fc.ul.pt
[4] Centro Nacional de Pensões, Av. República 102, 3º Lisboa, Portugal
joao.j.ramires@seg-social.pt

Abstract. We propose a groupware tool supporting the Software Quality Function Deployment approach to software requirements validation. The design challenge is to involve several stakeholders, having conflicting views and attitudes which may be difficult to reconcile, in the requirements validation. The adopted approach integrates collaboration and negotiation support. Negotiation models inspired the development of a set of mechanisms promoting integrative attitudes and avoiding distributive ones. Experiments with the tool revealed some usability problems, but also showed that it is convenient to use and beneficial promoting consensus.

1. Introduction

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 [2].

Quality Function Deployment (QFD) [3] is often used to implement TQM. QFD aims to define relationships and ultimately match the users’ and technical requirements [4]. QFD is used by manufacturing firms and has been lately applied to software production [2, 5]. 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 [6]). 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 [4]). The Software QFD (SQFD [5]) fills this gap in software engineering. The SQFD approach is considered part of the Capability Maturity Model (CMM) level 4 implementation [6].

Lecture Notes in Computer Science, vol. 3706, pp. 308-324. Groupware: Design, Implementation, and Use. H. Fuks, S. Lukosch and A. Salgado, Eds. Heidelberg, Springer-Verlag.

In a very simplified view, SQFD is a matrix of correlation values between requirements and specifications. This matrix is used in the following way [5]: (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," or 0, 1, 3 and 9 numeric values) is most often used [5]. 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 [4]; (2) using multi-criteria preference analysis to combine individual preferences into some utility function [7, 8]; and (3) in a meeting, where the stakeholders must negotiate their different opinions until a consensus is achieved [9].

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 [9].

One problem with the later approach is that, being based on meetings and a definite need to negotiate, the evaluation process may become time-consuming. According to [10], two consequences of increased participation in meetings are a decrease in response time and a decrease of total time available for decision making in organizations. This problem naturally increases with the size of the SQFD matrix.

We aim to develop a groupware tool supporting parallel work and facilitating consensus on the SQFD matrix, thus reducing the required amount of time to accomplish the task. The proposed groupware tool, which is named MEG, integrates SQFD support with collaboration, negotiation and argumentation support. MEG is at the same time a Group Support System (GSS) and a Negotiation Support System (NSS), thus falling in a category of tools commonly designated Group Decision and Negotiation Support System (GDNSS [11]).

More generally, we aim to research the integration of GSS and NSS functionality, addressing their differences in the conflict dimension. Regarding the low-conflict facet of the GSS perspective, we propose an approach to promote integrative attitudes and avoid distributive ones. Recognizing the high-conflict facet of the NSS perspective, the proposed approach also supports parallel negotiation processes, argumentation, bargaining and firm attitudes from the users.

The paper is structured in the following way. We start describing the requirements for integrating SQFD with collaboration and negotiation models in a coherent groupware tool. We then provide a detailed description of MEG. Finally, we describe two experiments with the tool, discuss the related work and present the obtained results.

2. SQFD and Negotiation – Requirements Definition

The SQFD matrix has cells correlating users' requirements with technical specifications. A correlation value measures the preference for the technical specification to fulfil the satisfaction of the requirement corresponding to a cell. We adopt the four-point scale with values 0, 1, 3 and 9. For reading convenience, correlation values will be referred as *C*. For example, in Figure 4, the requirement "Reply to mails easily" relates with the specification "Integrate external editor" with *C*=9, suggesting that this specification is strongly satisfying.

Assuming that several stakeholders work in parallel and select different *C* leads to a conflict situation addressed by our initial requirements:

R 1. *MEG will support the specification of different preferences for C, so that stakeholders may express their different views while working in parallel.*

R 2. *When alternative C have been proposed, MEG will support the negotiation of C until a final value is accepted by all stakeholders.*

Any negotiation requires information exchange. The SQFD model does not explicitly represent such information, but only the proposed *C*. One clear advantage of adopting groupware to implement SQFD is the possibility of extending the SQFD matrix with a shared memory component preserving the positions and arguments produced by the stakeholders. We adopted IBIS for that purpose. In Figure 1 we illustrate how SQFD and IBIS are combined to organize information pertaining to the negotiation of one SQFD cell [12]. The conflicting *C* generate an issue, for which there is an initial bid and a subsequent negotiation to reach an agreement. In that process, the stakeholders may add arguments to their preferences. This approach is articulated by the following requirement:

R 3. *MEG will handle multiple preferences for C as an issue, identifying an initial bid and positions against or in favour, and will allow the stakeholders to express their positions and arguments in favour of different C values.*

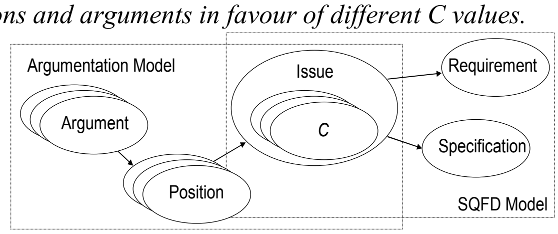

Fig. 1. Integration of SQFD and argumentation models for one SQFD cell

One further aspect related with SQFD and negotiation concerns the degree of information sharing supported by groupware. 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 [13]. On the contrary, negotiation assumes conflict between the parts involved, turning more difficult the creation of shared representations while increasing the process weight. The challenge then is that the integration of both perspectives creates some tensions between the support to the individual and group goals, and the support to shared representations and negotiation processes. These observations lead us to define a clear frontier between the stakeholders' hidden and shared knowledge [14]:

R 4. *MEG will not disclose the individual preferences of C, only their positions and arguments.*

We will now address the negotiation of *C* for one SQFD cell, considering that all cells are handled in the same way. Negotiation behaviour can be analysed according to two different strategies paradigmatic in negotiation research [15]:

- *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 [16], 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 [17], 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 will also follow the policy of favouring the integrative strategy.

According to [16], the switch to the integrative behaviour requires the combination of two conditions: an impasse and the willingness to engage in integrative behaviours. We investigate 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 defined by [18]:

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 [19]: 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 three fundamental requirements:

R 5. *MEG will favour Win-Win behaviours, which includes support to collaboration and compromise.*

R 6. *MEG will provide some resistance to Win-Lose and Lose-Lose behaviours, i.e. competition and avoidance.*

R 7. *MEG will be neutral about the Lose-Win behaviour, i.e. obliging.*

3. Design of the SQFD Negotiation Tool

In this section we describe our solution model. We start focusing on accomplish requirements R1, R2, R3 and R4.

Given a pair (requirement, specification), the SQFD matrix stipulates the correspondent 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).

When MEG starts, $SQFD_{rs} = 0, \forall r \in R, \forall s \in SP$, meaning that the default C is zero. Without loss of generality, in the following, we will consider a generic SQFD cell and the negotiation of the correspondent C.

The initial bidder is the first stakeholder specifying a non-zero C, while the value specified is the initial bid. MEG associates them to the cell through the concept of *ISSUE*, stated 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 preferences for C (requirements R1 and R2). All subsequent stakeholders attributing a value to the same 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 ($0 \le k \le 3$), and $c_j \in \{1,3,9\}$ is the j-th preference value S_i stated ($j \le k$).

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 (requirement R4), since individual preferences are kept undisclosed to the other stakeholders. Based on issues and preferences, MEG identifies the supporters and/or opponents to the initial bided (requirement R3). 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 (requirement R3), which confers them additional negotiation abilities. This means that for a stakeholder S_i it may be defined a tuple of arguments, defined by

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

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 experiments we relied upon generic roadmaps for quality assurance provided by software engineering literature.

We move now our attention to the negotiation support, which description is based on the states machine displayed in Figure 2. MEG 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 cell state. We consider an *Equilibrium* state, referred by *E*, that is reached whenever there is no ongoing negotiation for that cell, either because: a) 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 2, plain arcs correspond to transitions associated with user actions, while dashed lines represent transitions related with system events (such as evaluating positions or attitudes).

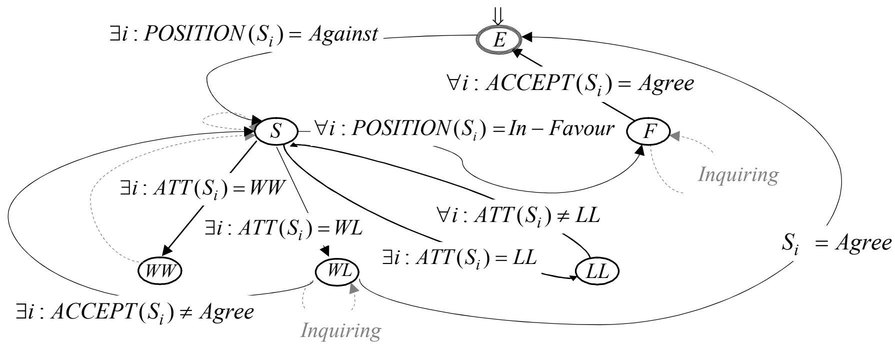

Fig. 2. Negotiation support (states machine)

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. 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 attitudes we have identified in requirements R5, R6 and R7. 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' inquire. 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 1.

Table 1. Behaviour detection

Attitude	**Detection**
Win-Win	*PREF* became "closer" to *initial-bid*
Win-Lose	"Firm" option has been selected (see section 4.2 for explanation)
Lose-Win	*PREF* has been removed
Lose-Lose	"Block" option has been selected (see section 4.2 for explanation)

3.1. Supporting the integrative approach

With requirements R5, R6, and R7 we declared the objectives to favour integrative strategies and resist to distributive strategies. We now describe how we addressed these issues. 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 [20] 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. Considering stakeholder S_i, the stakeholder weight 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 that stakeholder, while $WP(S_i, x)$ is the corresponding weighted preference, now considering the stakeholder weight in the negotiation. These values are given respectively by

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

$P(x) = \sum_{S_i \in ST} WP(S_i, x),\ x \in \{1,3,9\}$, computes the total preference for *x* expressed by the stakeholders. And finally, $CONSENSUS = \max_{x \in \{1,3,9\}}(P(x))$ 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 1). 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.

4. Implementation Details

MEG is a client-server tool implemented with MS Excel 2002, Access and Visual Basic 6.0. The system architecture is shown in Figure 3. The SQFD matrix was implemented with an Excel spreadsheet using RTD technology. Users may interact with the spreadsheet but cannot directly modify the cells. Those modifications are requested to MEGCLIENT, which communicates with MEGSERVER, which in turn maintains the shared information stored on an Access database. The database is accessed through XML. The RTDSERVER updates the distributed spreadsheets using DCOM. The interaction between Excel and RTDSERVER is explained in [21].

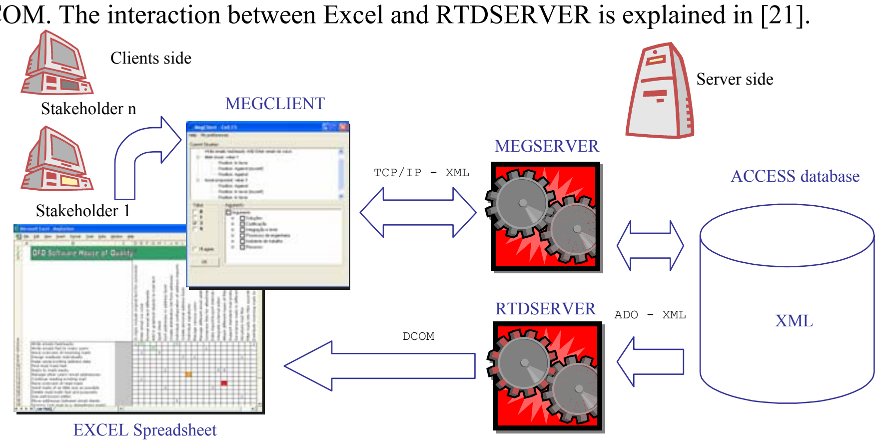

Fig. 3. System architecture

4.1. Illustration of MEG functionality

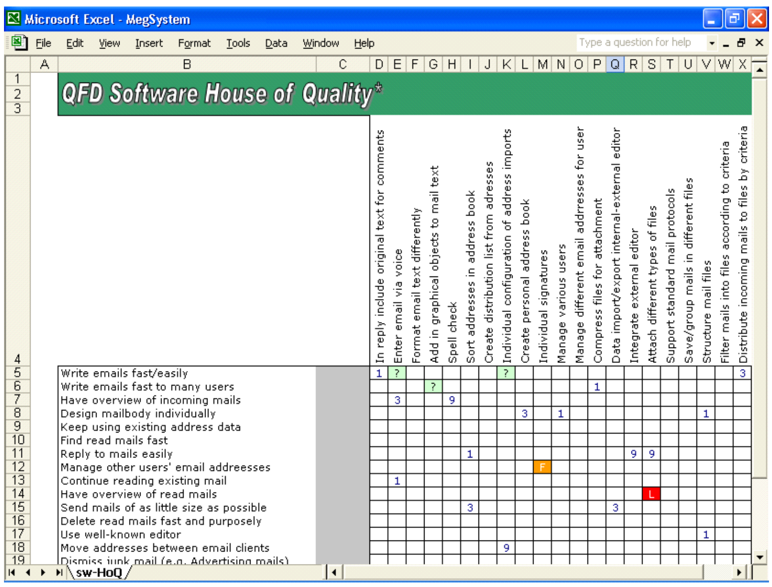

Fig. 4. SQFD matrix (example from [1])

Consider a set of three stakeholders: S1, S2 and S3. Figure 4 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 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.

Table 2. Sequence of actions accomplished by S1, S2 e S3 and system events

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

We use the sequence of actions shown in Table 2 to illustrate how users interact with MEGCLIENT and the correspondent 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 5-8 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 5). S2 does not agree with the correlation and 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). 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 7).

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 8). All users agree and the negotiation process finishes. The value 3 finally appears in cell E5.

5. Evaluation

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 in a large company; and (D) analyst/programmer in statistics and operational research.

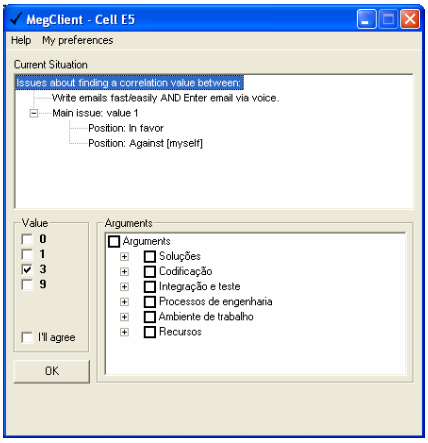

Fig. 5. MEGCLIENT in t2 for S2

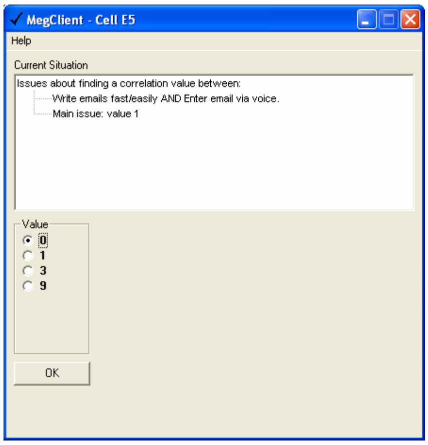

Fig. 6. MEGCLIENT in t3 for S2

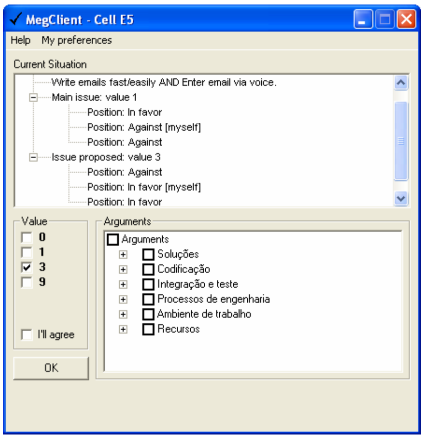

Fig. 7. MEGCLIENT in t6 for S2

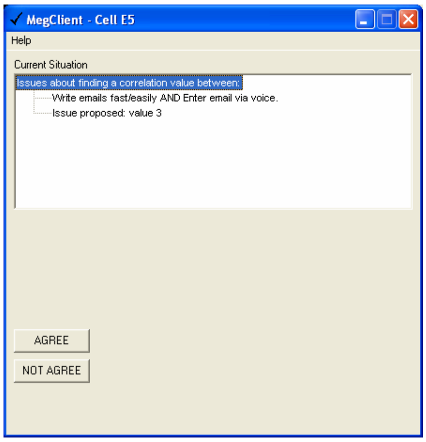

Fig. 8. MEGCLIENT in t8 for S2

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

The following list of technical requirements was specified by the authors based on the recommendations of stakeholder A:

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

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 the authors, which participated in the process as observers.

Beyond obtaining the SQFD matrixes with correlations, we requested the participants to fill up a questionnaire with questions about the MEG functionality and usability, as well as open questions about the most positive and negative aspects. The following quantitative results were obtained:

Functionality	**1 (<)**	**2**	**3**	**4**	**5 (>)**
Convenience (the available functions are appropriate for the task at hand)				4	
Precision (the obtained results reflect your opinions)			2	2	
Agreement (you agree with the consensus and majority voting approach)				3	1
Usability					
Compreension (your effort to understand the application logic)			2	1	1
Learning (your effort to learn how to use the application)			2	1	1
Operability (your effort to control the negotiation process)			2	1	1

The list of positive and negative aspects was as follows:

Positive	**Participant**
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	
Very slow	A
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
System is slow	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

Combining the quantitative and qualitative results obtained from the questionnaire, we arrived at the following conclusions about MEG:

- **The system is considered difficult to use**. The three criteria related with usability had the lowest score in the questionnaire. This situation is reinforced by negative comments from stakeholders A and C. Two stakeholders (A and B) also considered that the system had bad performance (this situation is caused by DCOM).
- **All stakeholders agreed that the system is convenient to use**. This position is reinforced by several positive comments about understanding the overall positions from others, revising own positions and ease finding agreements.
- **All stakeholders agreed that the consensus approach, complemented by majority voting, is beneficial**. The agreement criterion was the one that received the highest score, reinforced by positive comments about the ease to reach consensus.

Several minor functional and user interface details were also raised by the stakeholders, for instance, about the use of the "I agree" button, renegotiation of cells overriding previous consensus, and difficulties in obtaining summary view of the negotiation processes. These comments should be used in future versions of MEG but do not reflect any significant issues about the core design decisions made.

6. Related Work

Table 3. Comparison between MEG and related systems

	GSS	**NSS**	**Argumentation**	**Facilitator**
MEG	Supports parallel activities Shared memory component	Bid support Supports private preferences	Based on IBIS Uses pre-defined ontology	No

	Proposes consensus value			
Easy-WinWin [22]	Uses GroupSystems Follows Win-Win methodology	No	No	GroupSystems requires facilitator
MEDIATOR [23]	No	Defines goals and utility spaces Identifies equilibrium Suggests compromises	No	Facilitator aids the construction of shared representation
Hermes [24]	Discussion forum	Updates process status Recommends solutions Finds inconsistencies	Based on IBIS	No
Virtual QFD [1, 25]	Web-based tools: discussion panels, VoC tables, evaluation panels and QFD matrixes	No	No	Facilitator manages data during meetings
Co-Decide [26]	Multi-user spreadsheet extension Offers OLAP features	No	No	No

In Table 3 we show a comparison of MEG with related systems using four criteria: (1) support to GSS features; (2) support to NSS features; (3) argumentation support; and (4) dependence on the facilitator.

Comparing EasyWinWin and MEG, we observe the former neglects negotiation support. EasyWinWin uses generic GSS tools (GroupSystems' brainstorming, categorizing and voting tools) with two major consequences: dependence on the facilitator to manage the technology; and limited support to parallel activities. EasyWinWin follows the Win-Win principle [27] that all stakeholders should win, and guides the users through a process where winning conditions are identified and negotiated until mutual agreements are obtained.

MEDIATOR is strictly a NSS where problem representation evolves by sharing individual points of view and searching for a point of equilibrium. The system uses a set of dimensions to define goals and utility spaces. The evolution of utility spaces is displayed in matrix or graphical form. Like MEG, compromising solutions can be suggested. The system supports a facilitator who aids in the construction of a shared problem representation.

Hermes is the system more closely related with MEG: it organizes arguments using IBIS, assists the negotiation process with updated information about the process status, recommends possible solutions, and also searches for inconsistencies among users' preferences. However, it offers limited GSS support, and the adoption of a discussion forum makes it inadequate for handling a large number or requirements.

Virtual QFD basically supports data sharing before and during meetings using the Web. Available tools include discussion panels, VoC tables, evaluation panels and QFD matrixes. Unlike MEG, a facilitator is required to manage data during meetings.

Finally, Co-Decide is a multi-user extension to a single-user spreadsheet. The basic idea behind Co-Decide is to extend typical OLAP features to multiple users. Unlike MEG, Co-Decide does not support the negotiation process.

With this necessarily brief overview we show that the fundamental characteristic of MEG is bringing together several characteristics of GSS and NSS, in particular support to shared and private data, parallel work, bidding and argumentation.

7. Discussion

Table 4. Combining GSS and NSS perspectives

NSS (high-conflict)	**GSS** (low-conflict)
- Bargaining - Firm positions - Negotiation blocking	- Multiple parallel negotiations - Stimulate integrative attitudes - Avoid distributive attitudes - Ontology based argumentation

MEG improves the effectiveness of SQFD combining functionality attributed to GSS and NSS (Table 4). A unique characteristic of MEG is that it attempts to stimulate users to assume integrative attitudes based on a set of subtle interventions at the user-interface level, underpinned by models of negotiation processes. Experimented solutions included: 1) reducing the accessibility to Win-Lose and Lose-Lose attitudes, making difficult the access to associated buttons and requesting unnecessary confirmations; 2) associating a cost to Win-Lose and Lose-Lose attitudes and showing that cost to the group; 3) supporting Win-Win and Lose-Win attitudes, avoiding focus on definite values, facilitating position changes and multiple choices; 4) promoting Win-Win attitudes, recommending a consensus value based on majority voting. MEG also supports ontology based argumentation. The objective is to reduce the levels of conflict by exchanging standard messages, meaningful in the domain, instead of free text.

One interesting outcome from this combination of GSS and NSS functionality is that the resulting tool offers more latitude and flexibility handling group strategies: the system supports low-conflict collaborative situations, but is also capable to cope with increased levels of conflict in a flexible way. The results from the pilot experiments indicate that the approach is considered beneficial for reaching consensus.

Acknowledgements

This paper was partially supported by the Portuguese Foundation for Science and technology, Project POSI/EIA/57038/2004.

References

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

[2] R. Zultner, "Tqm for Technical Teams," *Communications of the ACM*, vol. 38, pp. 79-91, 1993.

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

[4] A. Stylianou, R. Kumar, and M. Khouja, "A Total Quality Management-Based Systems Development Process," *The DATA BASE for Advances in Information Systems*, vol. 28, pp. 59-71, 1997.

[5] S. Haag, M. Raja, and L. Schkade, "Quality Function Deployment Usage in Software Development," *Communications of the ACM*, vol. 39, pp. 41-49, 1996.

[6] G. Antoniol, S. Gradara, and G. Venturi, "Methodological Issues in a Cmm Level 4 Implementation," *Software Process Improvement and Practice*, vol. 9, pp. 33-50, 2004.

[7] 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 (COMPSAC'02), Oxford, England, 2002.
[8] P. Chuang, "Combining the Analytic Hierarchy Process and Quality Function Deployment for a Location Decision from a Requirement Perspective," *Advanced Manufacturing Technology*, vol. 18, pp. 842-849, 2001.
[9] V. Bouchereau and H. Rowlands, "Quality Function Deployment: The Unused Tool," *Engineering Management Journal*, pp. 45-52, 2000.
[10] V. Vroom and A. Jago, *The New Leadership: Managing Participation in Organizations*. Englewood Cliffs: Prentice-Hall, 1988.
[11] E. Bellucci and J. Zeleznikow, "A Comparative Study of Negotiation Decision Support Systems," Thirty-First Annual Hawaii International Conference on System Sciences, Kohala Coast, Hawaii, 1998.
[12] Y. Reich, "Ai-Supported Quality Function Deployment," *Artificial Intelligence in Economics and Management*, pp. 91-106, 1996.
[13] M. Bichler, G. Kersten, and S. Strecker, "Towards a Structured Design of Electronic Negotiations," *Group Decision and Negotiation*, vol. 12, pp. 311-335, 2003.
[14] J. Johannessen, J. Olaisen, and B. Olsen, "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, pp. 153-168, 1997.
[15] R. Lewicki and J. Litterer, *Negotiation*. Homewood, Illinois: R. D. Irwin, 1985.
[16] F. Harinck and C. Dreu, "Negotiating Interests or Values and Reaching Integrative Agreements: The Importance of Time Pressure and Temporary Impasses," *European Journal of Social Psychology*, vol. 34, pp. 595-611, 2004.
[17] R. Lewicki, S. Weiss, and D. Lewin, "Models of Conflict, Negotiation and Third Party Interventions: A Review and Synthesis," *Journal of Organizational Behavior*, pp. 209-252, 1992.
[18] K. Thomas and R. Kilmann, *Thomas-Kilmann Con.Ict Mode Instrument*. Escondido, California: Blanchard Training and Development, 1974.
[19] R. Friedman, S. Tidd, S. Currall, and J. Tsai, "What Goes around Comes Around: The Impact of Personal Conflict Style on Work Conflict and Stress," *The International Journal of Conflict Management*, vol. 11, pp. 32-55, 2000.
[20] R. Fisher and W. Ury, *Getting to Yes: Negotiating Agreement without Giving In*. New York: Penguin Books, 1983.
[21] P. Cornell, "Building a Real-Time Data Server in Excel 2002," Microsoft Corporation 2001.
[22] B. Boehm, P. Grünbacher, and R. Briggs, "Developing Groupware for Requirements Negotiation: Lessons Learned," *IEEE Software*, vol. 18, pp. 46-55, 2001.
[23] M. Jarke, T. Jelassi, and M. Shakun, "Mediator: Toward a Negotiation Support System," *European Journal of Operational Research*, vol. 31, pp. 314-334, 1987.
[24] N. Karacapilidis and D. Papadias, "Computer Supported Argumentation and Collaborative Decision Making: The Hermes System," *Information Systems*, vol. 26, pp. 259-277, 2001.
[25] G. Herzwurm and S. Schockert, "Virtual Product Development," Proceedings of the Fifth International Symposium on Quality Function Deployment and the First Brazilian Conference on Management of Product Development, Belo Horizonte, Brazil, 1999.
[26] M. Gebhardt, M. Jarke, and S. Jacobs, "A Toolkit for Negotiation Support Interfaces to Multi-Dimensional Data," *SIGMOD Conference*, pp. 348-356, 1997.
[27] B. Boehm and R. Ross, "Theory-W Software Project Management Principles and Examples," *IEEE Transactions on Software Engineering*, vol. 15, pp. 902-916, 1989.