7 Requirements Negotiation

Paul Grünbacher and Norbert Seyff

Abstract: Negotiation is regarded as crucial in many disciplines, and negotiation methods and tools are increasingly studied by requirements engineering researchers and practitioners. The objectives of this chapter are to motivate the need for negotiation in requirements engineering, to introduce fundamental concepts and terminology, and to provide an overview about negotiation research. We structure the existing research (a) by presenting a general negotiation process highlighting typical negotiation stages; (b) by introducing a framework covering important dimensions of requirements negotiation comprising the conflict resolution strategy, the collaboration situation of the stakeholders, and the degree of negotiation tool support; and (c) by discussing and classifying existing negotiation tools using the general process and framework.

Keywords: Negotiation, Negotiation process, Conflict resolution, Collaboration, Negotiation tools, Stakeholder win-win.

7.1 Introduction

Conflicts play an important role in software engineering although they are often neglected or badly handled by existing development methods. Conflicts arise almost inevitably as project stakeholders such as future system users, acquirers, developers, or maintainers frequently pursue mismatching goals [10]. For example, future system users are typically interested in many features, high level of service, or early availability. Acquirers focus on cost effectiveness, compliance with standards, or budget/schedule constraints. Developers typically want flexible contracts and stable requirements. Although studies show that conflict is extensive in software engineering [15], many existing methods neglect or do not explicitly address conflict handling and resolution. Nevertheless, negotiation techniques and tools have gained increased attention in software engineering research. As a result, methods and tools have been developed supporting the requirements negotiation process, some of them are also available commercially.

Software engineering is a highly collaborative process and identifying shared or opposed interests is a necessity for project success [41, 60]. The objectives of customers, users, or developers have to be understood and reconciled to develop mutually acceptable agreements [5]. This obviously does not mean that stakeholders will always agree. The result of negotiation is also to understand why stakeholders disagree. Identified disagreements represent major risks and need to be addressed by project management.

Requirements negotiation is not a one time episode in a project, but should be used early on and repeated in later stages [9]. In each cycle new stakeholders and new objectives have to be considered often leading to negotiations. In iterative software life cycles such as the spiral model [3] the achieved agreements are evolved into more detailed requirements, development plans, architectures, etc. The primary purpose of requirements negotiation is to identify and resolve conflicts among stakeholders. It contributes to the goal of defining feasible and mutually satisfactory requirements that accommodate all stakeholder goals and expectations [6, 41, 60]. Beyond this primary purpose, research and evidence from practitioners show further benefits:

Understanding project constraints. It has been shown in many studies that software projects often fail to meet critical project constraints such as budget and schedule [58]. Negotiation makes stakeholders aware of these constraints and supports finding solutions for meeting them.

Adapting to changes. Because of rapid chances of market competition, technology, personnel, etc. requirements (and sometimes even constraints) are highly volatile. As a result stakeholders are forced to frequently adapt to new situations. Negotiation helps to deal with such changes more easily as stakeholders are aware of existing issues and alternatives. Should agreements become obsolete they can be re-negotiated and revised to accommodate the evolving requirements and constraints.

Fostering team learning. Different stakeholders come to a project with their experiences, backgrounds, and expectations and bring their goals to the table. Developing requirements is a cognitive process, in which stakeholders collaboratively find out what has to be done [60] by understanding problems and domains, learning from other stakeholders, and by negotiating and discussing different viewpoints. Stakeholders share information and search for mutually beneficial solutions. Developers, for example, learn more about the customer's and user's world, while customers and users learn more about what is technically and economically feasible.

Surfacing tacit knowledge. People know more than they can ever tell. Tacit stakeholder goals, hidden assumptions and expectations often lead to problems in software projects. Negotiation supports people bringing hidden issues and assumptions to the table [27].

Managing complexity. Establishing software requirements is fraught with complexity. In a typical non-trivial project with 10+ stakeholders one has to deal with hundreds of individual goals, and dozens of issues and alternatives that need to be understood. Complex interdependencies among requirements and between requirements and related development artifacts are another source of complexity as described in Chap. 5. Further things complicating negotiations are cognitive overflows, conflicting strategies of negotiators, or unforeseen interventions by third parties [57]. Handling that complexity is supported by negotiation techniques [14].

Dealing with uncertainty. Specifying software requirements without negotiation is difficult, because users do not know exactly what they need and what is technologically feasible [4, 60]. Negotiation helps to reduce uncertainty by highlighting things needing attention and fosters a shared vision among stakeholders.

Finding better solutions. Without negotiation techniques stakeholders often try to persuade others to accept a suggested solution instead of jointly seeking for new solutions that are beneficial to all parties [52]. For example, the main disadvantage of sequential negotiation of issues is that trade-offs between issues cannot be considered adequately. Negotiation techniques help to see the full picture instead of dealing with issues sequentially, which can help to avoid suboptimal solutions.

The benefits of negotiation are obvious, and many researchers have pointed out its usefulness for requirements engineering [42, 46]. However, establishing a requirements negotiation process is not trivial and important issues have to be addressed: How can conflicts be identified? How can the identified conflicts be resolved? How can stakeholders find feasible alternatives? Who is in charge of the negotiation, the stakeholders themselves or a facilitator? How can the negotiation be supported with tools or other means? Requirements negotiation can make use of negotiation methods and tools from a wide range of disciplines and domains. Negotiation is a phase in the decision making process and there is a strong body of knowledge on decision making. Consequently, negotiation in group decisions have been investigated from multiple perspectives, such as decision theory [36], management theory and social sciences [19, 50, 59], organizational psychology [61], and game theory [49]. Giving an overview about the start-of-the-art in requirements negotiation is challenging, as a thorough discussion of all these aspect is certainly beyond the scope of this chapter. We therefore discuss the existing research from the perspective of software requirements negotiation instead of negotiation in general.

The chapter is structured as follows: In Sect. 7.2 we review several definitions for requirements negotiation, define basic terminology, and present a general negotiation process highlighting typical negotiation stages. Section 7.3 introduces our framework covering important dimensions of requirements negotiation such as conflict resolution strategy, the collaboration situation of the stakeholders, and the level of negotiation tool support. The purpose of the framework is to help understand and classify existing and future research approaches and to increase awareness of the issues involved in defining and implementing requirements negotiation processes in practice. In Sect. 7.4, we use the framework to present examples of existing requirements negotiation approaches. Conclusions round out the chapter in Sect. 7.5.

7.2 The Negotiation Process

Negotiation is widely adopted and has been investigated by multiple disciplines. Consequently, there are different perspectives on negotiation and different aspects are emphasized [14, 16, 31, 47]. Negotiation is traditionally viewed as “the actual interactions among participants that lead to mutual commitment” starting “when participants begin communicating their goals, and ending (successfully) when all agree to a specified contract.” [52]

Other definitions have a slightly different flavor. Easterbrook [20] defines negotiation as “a collaborative approach to resolving conflict by exploration of the range of possibilities. It is characterized by the participants attempting to find a settlement which satisfies all parties as much as possible." The author emphasizes conflict as the fundamental reason for negotiation and points out that negotiation often involves some sort of compromise when saying that parties should be satisfied "as much as possible."

In another definition Curtis et al. [15] take a requirements engineering perspective when stating that "in general terms, requirements negotiation can be seen as an iterative process through which stakeholders make tradeoffs between requested system functions, the capabilities of existing or envisioned technology, the delivery schedule and the cost." Robinson and Volkov [52] argue that beyond the actual negotiation one should also consider pre- and post-negotiation phases as part of the negotiation process covering activities such as initial problem recognition, participant solicitation and communication, or solution maintenance. This broader view is also confirmed by different negotiation approaches. The negotiation support system Inspire [40], for example, uses the phases pre-negotiation, negotiation, post-settlement. The EasyWinWin negotiation approach is embedded in processes of preparing the actual negotiation and post-negotiation analyses and quality assurance [28]. The identification of stakeholders in EasyWinWin is covered by the win-win spiral model. Our discussion of the negotiation process follows these approaches and therefore discusses the general stages of pre-negotiation, negotiation, and post-negotiation.

7.2.1 Pre-Negotiation

Important activities of this phase are the definition of the negotiation problem, the identification and solicitation of stakeholders, the elicitation of goals from stakeholders, and the analysis of goals to find conflicts. The results of this phase are the issues and conflicts involved. According to [40] an issue is "a topic of discussion that is of particular interest in a negotiation. Each issue has a range of alternatives or options, one of which must ultimately be agreed upon by the negotiators in order to achieve a compromise."

Problem Definition. Before the actual negotiation can start it is important to identify the problem by analyzing the situation and defining the purpose of the negotiation. For example, in a software project the problem depends on both the overall objectives of the project and the current stage of the project. Early stage requirements negotiations involve high-level issues while later negotiation might focus on specific aspects or subprojects. Requirements gathered in early stages of a project express a wider range of possibilities in general terms and become more precise later on [22]. Defining the negotiation problem is essential for stakeholder identification and for adjusting the negotiation method and techniques.

Stakeholder Identification. The success-critical stakeholders have to be identified. Finding the people (or appropriate representatives) whose interests must be accommodated is often a challenging task itself [32, 56] but essential for the success of the requirements negotiation. The success-critical stakeholders are the people that can make agreements about requirements and can make those agreements stick. Identifying the right people can accelerate the negotiation process.

Goal Elicitation. Before conflicts can be identified stakeholders have to bring their individual goal to the table. A goal is an objective the system under consideration should achieve [43]. All success-critical stakeholders need to express their individual goals or the goals of people they represent. Depending on the identified problem and stakeholder characteristics such as role, domain knowledge, experience, etc. goals are formulated at different levels of granularity, ranging from high-level aspects such as general system capabilities, budgets, or schedules to lower level technical concerns such as development environments or target platforms. Many of the elicitation and prioritization techniques presented in Chaps. 2 and 4 support this activity.

Goal analysis. The elicited goals are examined to identify conflicts, i.e., by analyzing stakeholder goals and preferences. For example, there might be a conflict between the level of service required by users and budget constraints imposed by acquirers. Identifying conflicts is typically a manual process and relies on the knowledge and expertise of the involved stakeholders and the capabilities of the facilitator. Goal analysis does not only reveal conflicts among stakeholder goals but typically also reveals inconsistencies, risks, uncertainties, and hidden assumptions [27]. Prioritization techniques presented in Chap. 4 support this task.

Different authors have tried to automate or partially automate the task of understanding requirements conflicts. For example, Egyed and Grünbacher [21] recently presented an approach for identifying conflicts and cooperation among requirements based on software attributes and automated traceability. Another example of this kind of support are sophisticated visualization techniques to identify conflicting goals and requirements [33].

7.2.2 Negotiation

This phase involves the actual conduct of the negotiation and the definition of agreements. Based on the elicited goals and the identified conflicts stakeholders seek mutually beneficial solutions that are acceptable to all parties. This activity is about structuring issues and developing alternatives to solve problems, for example by exchanging offers and counteroffers, or proposing alternatives for mutual gain. After developing possible solutions stakeholders eventually agree on the "best" one. The explanation of possible solutions is a prerequisite before stakeholders can agree on a decision and requires the establishment of judgment criteria, a common set of rules agreed by all stakeholders [60]. If these rules are missing, the merits of different options will be inconsistent. It might therefore be necessary to carry out a preparatory negotiation session in order to agree on these judgment criteria.

Depending on the type of conflict and problem at hand different strategies can be adopted [48] for dealing with the conflicts (see also Sect. 7.3). This involves trade-offs in which stakeholders give up partly on some issues so as to gain on other issues, for example, by making concessions to ease gaining an agreement; problem-solving by identifying and adopting solutions that satisfy the goals of the parties; or persuading other negotiators to concede. Apparently, negotiators might also decide to drop out of a negotiation. Some authors have developed automated approaches for resolving conflicts. An example is the Oz system developed by Robinson and Fickas [51].

7.2.3 Post-Negotiation

In this phase stakeholders (or automated tools) analyze and evaluate the negotiation outcomes and suggest re-negotiation if necessary. For example, it can be determined if the current agreement satisfies the preferences of the counterparts and if a better solution would be possible for one negotiation party, without causing loss to the other side [37]. It can also involve quality assurance reviews of the negotiation results [28]. The importance of early quality assurance in RE is also emphasized in Chap. 8. Another important aspect of post-negotiation is to secure commitment of stakeholders over time. For example, by monitoring existing agreements and initiating re-negotiation in case agreements become obsolete due to new developments. Especially in iterative life cycle models [2, 3, 7] negotiation results need to be constantly evolved as new goals can always arise and potentially cause new conflicts [8]. Understanding the impacts of changing goals is typically non-trivial as also discussed in Chap. 6.

7.3 Dimensions of Requirements Negotiation

The negotiation process presented in the previous section defines the scope and purpose of activities relevant in requirements negotiation. It does, however, not address more specific aspects of negotiations. We therefore present a simple framework which describes important dimensions of requirements negotiation in more detail. By explaining the dimensions of the framework we give a survey of relevant research. The purpose of the framework is twofold: (a) It can be used for classifying and understanding existing negotiation approaches and tools by using well-defined and relevant dimensions; (b) it addresses issues important for organizations wishing to design and implement effective negotiation processes.

The dimensions of the framework address (1) the conflict resolution strategy, (2) the collaboration situation of stakeholders, and (3) the degree of negotiation tool support. The dimensions are derived by analyzing literature and negotiation tools from different fields. Although the chosen dimensions are important we do not claim that the framework is complete and covers all aspects relevant in requirements negotiations. Also, dependencies between the dimensions are not explicitly addressed. For example, a certain collaboration situation may imply certain conflict resolution strategies and specific kinds of negotiation support. The dimensions cover key questions in requirements negotiation: How are conflicts resolved? How do stakeholders collaborate? Which tools are used to support the process?

Conflict resolution strategy. Conflict is an inevitable part of system design and the reason for negotiation. The first dimension thus addresses the different conflict resolution strategies based on the conflict handling modes developed by Thomas [61] in the field of organizational psychology.

Collaboration situation. The second dimension addresses the collaboration setting defined by the location of stakeholders and the time of negotiation. Synchronous/co-located negotiations, where people work together face to face, are fundamentally different from asynchronous/dislocated forms of negotiations that make interaction more difficult. This dimension is informed by research done in CSCW (Computer Supported Cooperative Work) [35].

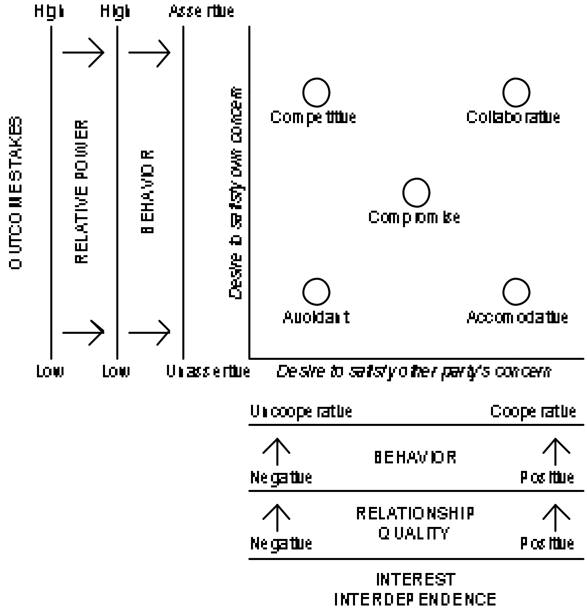

Fig. 7.1 Determinants of conflict behavior [1, 61]

Negotiation support tools. Negotiations can be supported with different kinds of tools ranging from manual guidelines to sophisticated tools and environments. Understanding these types and levels of automation is important to choose the appropriate level of support for a given situation. Authors in the field of negotiation support systems (NSS) have done research to classify the different options for tool support [34, 37, 44].

7.3.1 Conflict Resolution Strategy

Software engineering projects face conflicts of interests and needs in important decisions. Theoretically, such situations can be framed as mixed-motive, where parties experience partly common ground (joint goals and objectives of the project) but also face considerable differences in preferences for specific issues. It has been shown that conflict is not the exception but very common in group interactions. A study by Curtis et al. [15] reveals three major sources of conflict in software engineering: the thin spread of application domain knowledge; fluctuating and conflicting requirements; and breakdowns in communication and coordination. Conflicting requirements have many causes, including changes in the organizational setting and business environment. Also, software will be used by different people with different goals and needs. Further sources of conflicts listed by Easterbrook [20] include conflicts between suggested solution components; conflicts between stated constraints; conflicts between perceived needs; conflicts in resource usage; and discrepancies between evaluations of priority.

A well-known model of conflict behavior has been proposed by Thomas in the field of organizational psychology [61]. According to this model a stakeholders' orientation has two dimensions: the focus on satisfying their own concerns (unassertive, assertive) and the emphasis on satisfying the concerns of others (uncooperative, cooperative). Using the two dimensions one can define five dominant orientations of dealing with conflicts (see Fig. 7.1):

- *Competing (forcing)* involves an emphasis on winning one's own concerns at the expense of another, often leading to "win-lose" situations.
- *Accommodating (smoothing)* involves trying to satisfy the other's concerns without attention to one's own concerns. This can mean that one stakeholder is self-sacrificing and yielding to the other.
- *Collaborating (problem-solving)* focuses on satisfying the concerns of all parties to find alternatives that try to satisfy the concerns of all. The emphasis is on finding "win-win" situations.
- *Avoiding (withdrawing from)* a negotiation could be a result of indifference, denial, or apathy.
- *Compromising (sharing)* involves concessions to find a satisfactory middle ground.

Figure 7.1 shows that choosing the best conflict handling strategy depends on factors such as the outcome stakes, the interdependence of interests, the relative power of parties, and their quality of relationship. For example, if the outcome stakes for a stakeholder is high (which is the case in many software projects) and people want to maintain a good quality of relationship, a collaborative conflict handling mode is preferred over accommodative behavior. Another model for comparing different negotiation styles has been proposed by Fisher and Ury [23]. The authors distinguish between soft, hard, and principled negotiation strategies. In the *soft strategy* the underlying assumption is that parties are willing to collaborate to seek mutually satisfactory agreements. Stakeholders cooperate in a consensus-oriented, problem-solving team process. In the *hard strategy* parties are seen as competitors that not necessarily want to arrive at a win-win situation. It can also be seen as an interaction of competing stakeholders, where conflicts are will occur inevitably. Instead of focusing on these two extremes Fisher and Ury propose a combined approach called *principled strategy* [23].

Table 7.1 Characteristics of soft, hard and principled strategies [23]

Soft	**Hard**	**Principled**
Participants are friends.	Participants are adversaries.	Participants are problem-solvers.
The goal is agreement.	The goal is victory.	The goal is a wise outcome reached efficiently and amicably.
Make concessions to cultivate the relationship.	Demand concessions as a condition of the relationship.	Separate the people from the problem.
Be soft on the people and the problem.	Be hard on the problem and the people.	Be soft on the people, hard on the problem.
Trust others.	Distrust others.	Proceed independent of trust.
Change your position easily.	Dig into your position.	Focus on interests, not positions.
Make offers.	Make threats.	Explore interests.
Disclose your bottom line.	Mislead as to your bottom line.	Avoid having a bottom line.
Accept one-sided losses to reach agreement.	Demand one-sided gains as the price of agreement.	Invent options for mutual gain.
Search for the single answer: the one they will accept.	Search for the single answer: the one you will accept.	Develop multiple options to choose from; decide later.
Insist on agreement.	Insist on your position.	Insist on using objective criteria.
Try to avoid a contest of will.	Try to win a contest of will.	Try to reach a result based on standards independent of will.
Yield to pressure.	Apply pressure.	Reason and be open to reason; yield to principle, no to pressure.

Table 7.1 compares the three strategies using a set of negotiation characteristics. The combined strategy focuses on four principles printed in bold in Table 7.1. These are separating the people from the problem; focusing on interests, not positions; generating a variety of possibilities before deciding what to do; and insisting that the result is based on some objective standard.

7.3.2 Collaboration Situation

The negotiation process discussed in Sect. 7.2 has to consider different collaboration situations depending on the time and place of interaction. For example, a team might decide to organize a face to face meeting for the definition of agreements, while the elicitation of preferences is carried out in a dislocated manner. The time of the negotiation and location of stakeholders have a strong impact on the actual interactions during a negotiation and pose additional challenges. The field of Computer-Supported Cooperative Work has developed the CSCW matrix, a simple classification scheme that distinguishes four different scenarios (see Table 7.2):

Table 7.2 Collaboration situations of negotiating stakeholders [35]

	Co-located	Dislocated
Synchronous communication	**Same time/Same place**	**Same time/Different place**
Asynchronous communication	**Different time/Same place**	**Different time/Different place**

Same time/Same place. Face to face meetings are still a common way to elicit and negotiate requirements. In requirements engineering, many approaches still work best or even necessitate continuous, synchronous team work [32]. Newer approaches such as agile methods strongly advocate face to face meetings. A popular example is the “on-site customer”, a practice in eXtreme Programming [2]. Especially when trying to resolve conflicts the richness of face to face interactions makes it easier to build trust and jointly seek for solutions. The facilitator guidelines of the EasyWinWin approach, for example, suggest to organize the “negotiation of agreements” activity as a face to face meeting to benefit from the richness of non-verbal cues, which make it easier to understand people and therefore to reduce negotiation time.

Different time/Same place. Organizing an entire negotiation with face to face meetings is typically not possible even if stakeholders are co-located at the same site. The duration of negotiations often exceeds the time of typical workshops and meetings are generally difficult to arrange due to time constraints. Also, information needed to take a final decision is often not available during a meeting. It is then necessary to carry out certain steps in an asynchronous manner, supported by shared workspaces allowing all stakeholders to contribute to ongoing negotiations and to keep track of the progress [26].

Same time/Different place. Even if it is impossible to bring together stakeholders in a face to face meeting, it is frequently possible to gather them at the same time, with some of them participating remotely. The use of audio and video conferencing provides a reasonable interaction bandwidth and the team benefits from same-time interaction. For example, group decision support systems have been successfully used to support synchronous/dislocated brainstorming or voting sessions [45].

Different time/Different place. Requirements engineering is increasingly carried out in an asynchronous and dislocated setting as more and more projects span globally or affect multiple organizations [12]. In such a situation advanced technology for collaboration is a necessity to allow stakeholders to contribute from different parts of the world. However, little research exists to investigate the impact of different time/different place interactions on the success of requirements negotiation. Damian et al. [18] have explored the role of facilitation in such a situation.

The four collaboration situations described by the CSCW matrix do, however, not address all important issues that impact requirements negotiations such as the number of stakeholders involved, the difference between multiple individual sites verses multiple group sites, as well as cultural differences among negotiating parties.

7.3.3 Negotiation Tool Support

The third dimension of our framework deals with the type and degree of tool support. Negotiations are often supported by traditional means such as guidelines and handbooks for facilitation as well as general meeting tools for all stakeholders such as whiteboards, flipcharts etc. [25]. The scale and complexity of real-world projects however suggest the use of more sophisticated forms of negotiation support ranging from software tools for communication to intelligent software agents. In a recent paper Kersten [37] provides an insightful classification for negotiation support tools:

Passive Support. Such tools provide an infrastructure for negotiation and support all different collaboration situations discussed above. They allow all parties involved to express their preferences, to communicate about ideas, offers and arguments, and to share intermediate and final results. Examples are email, chat, or multimedia rooms [17]. Passive systems do not support the production of content with hints and guidance.

Active facilitative support. Tools of this kind are capable of guiding the stakeholders towards an agreement, for example, by identifying situations for mutual gain. Such systems can aid the users in the formulation, evaluation, and solution of difficult problems. They also support concession-making and construction of offers, as well as the assessment of the process. Active negotiation support systems typically follow a negotiation process. Group decision support systems [45] fall in this category especially if the collaborative tools are integrated with facilitation guidelines [13].

Pro-active interventive support. These systems are additionally capable of coordinating the activities of stakeholders. For example, they critique their actions or suggest what agreement to accept. To provide such capabilities the systems access and use knowledge-bases and employ intelligent software agents that monitor the negotiation process and the negotiators' individual activities. An example is the Atin intelligent software agent augmenting the Inspire system (see Sect. 7.4.1) [39].

7.4 Examples of Negotiation Systems

Researchers and practitioners have been developing different types of negotiation systems supporting stakeholders in conducting a negotiation. However, some of them are particularly targeted at software requirements negotiation while most tools provide more general negotiation support. Examples of negotiation tools include DealMaker, Inspire, MeetingOne, Negoisst, SimpleNS, SmartSettle, and WebNS. In this section we use the negotiation process and framework to characterize existing negotiation support systems. We have selected four examples: Aspire is a pro-active negotiation support system supporting bilateral negotiations which is based on Inspire; EasyWinWin, a system targeted at software requirements negotiation; Negoisst, an electronic business-to-business negotiation system; and SmartSettle, a commercially available negotiation support system for complex negotiations.

7.4.1 Aspire

Aspire is a recent extension to the Inspire system and provides pro-active level support with the Atin software agent [39]. The agent advices the negotiators by analyzing an ongoing negotiation using rules derived from literature. This could, for example, involve warning the user about implications of actions he intends to undertake. The tool [37, 38] is a web-based negotiation support system supporting asynchronous, dislocated negotiations and is targeted at bilateral negotiations.

Aspire implements a three phase negotiation model comprising pre-negotiation, conduct of negotiation, and post-settlement. The key activities during the *pre-negotiation phase* are the analysis of the current situation regarding issues and options, and the identification of key stakeholders. In the pre-negotiation phase Aspire assists stakeholders in understanding the negotiation case by providing a detailed description of the initial situation. Stakeholders are invited to express their preferences regarding the issues and alternatives. During the *negotiation phase* the opponents exchange messages and offers to present their viewpoints. The negotiation ends when an agreement is achieved or one of the opponents stops the negotiation. Aspire supports the opponents by providing capabilities for sending messages and offers. Also, for analyzing the ongoing negotiation the two opponents can view a history of the negotiation processes, which is tracked by the tool. The *post-settlement phase* is used to analyze and evaluate the negotiation outcomes and if necessary to re-negotiate an already existing agreement. Based on the preference information entered in the pre-negotiation phase, Aspire determines if the current agreement satisfies the preferences of the counterparts. It checks if there is a better solution possible for one negotiation party, without loss to the other side. Aspire has a strong support for the solution generation stage by analyzing the negotiation and giving active hints.

7.4.2 Negoisst

The Negoisst system for negotiation has its focus on supporting business-to-business electronic commerce. Based on theories of communication and information systems it combines communication and document management [54]. Teams can use natural language to exchange semi-structured messages and jointly compose the terms of a complex contract. Negotiation systems for e-commerce transactions typically support general phases of business-to-business e-commerce: finding potential partners; negotiating and finding agreements; and fulfilling the contractual obligations [53]. In this context, the aim of the Negoisst system is to support the negotiation phase by providing intuitive, unambiguous, efficient, and process-oriented negotiation support between human negotiators. Using semi-structured message exchange the negotiators can choose from various message types to make intentions explicit. The Negoisst system provides the following types of messages, which also outline the negotiation process: request, offer, counter-offer, accept, reject, question, and clarification.

7.4.3 EasyWinWin

EasyWinWin is a requirements negotiation approach that combines the win-win spiral model of software engineering [9] with collaborative knowledge techniques and automation of a Group Support System. It is based on Boehm's negotiation model [11]. The individual objectives of stakeholders are captured as *win conditions*. Conflicts among win conditions, risks, and uncertainties are recorded as *issues*. *Options* are proposed to reconcile issues. *Agreements* are developed out of win conditions and out of options by taking into account the preceding decision process and rationale. EasyWinWin helps a team of stakeholders to gain a better and more thorough understanding of the problem and supports co-operative learning about others' viewpoints. It is an example of an active negotiation support system. The EasyWinWin requirements negotiation approach also includes steps for elicitation and analysis. For example, in a brainstorming step all stakeholders are invited to post their ideas. A facilitator analyzes the ideas and forms win conditions jointly with the team of stakeholders. EasyWinWin is based on a Group Support System (GSS). Within the vast number of groupware technologies Group Support Systems (GSS) focus on supporting group decision-making. A GSS is not just a single piece of software, but a collection of computer-based collaborative tools that a team may use to focus and structure their mental effort as they work together toward a goal. Extensive research in the lab and in the field reveals that, under certain circumstances, teams can use GSS to become substantially more productive than would otherwise be possible. Fjermestad et al. [24] provide an exhaustive compendium of GSS field research.

Typical examples of such tools are Electronic Brainstorming tools for support idea generation, group outlining tools for idea organization, or voting tools for idea evaluation. In EasyWinWin participants use a multi-criteria polling tool to prioritize win conditions regarding business importance and ease of implementation. The brainstorming capability is used to gather stakeholder interests. There is an electronic page for each stakeholder. Whenever a stakeholder contributes a comment to a page the system takes that page away and randomly replaces it with a different page containing comments from other stakeholders. As the activity progresses, the pages swap among the participants, picking up a new comment at each stop. This process tends to broaden the scope of the discussion, resulting in breadth, rather than depth. It is a useful way to identify many concepts in a short amount of time. The major area of application of EasyWinWin is software requirements negotiation. Teams use EasyWinWin throughout the development cycle to develop a shared project vision, high-level requirements definitions, detailed requirements for features, functions, and properties, requirements for transitioning the system to the customer and user. The goal elicitation aspect is strongly supported; the solution generation support is weaker and relies on the help of a facilitator. EasyWinWin follows mainly a collaboration-oriented conflict resolution strategy. There are no limitations with respect to the number of stakeholders and collaboration situations, although most groups have used EasyWinWin in same time (synchronous or asynchronous) settings. The level of tool support is active, the collaborative tools provide an infrastructure for negotiation and the negotiation model and the explicit process guide stakeholders.

7.4.4 SmartSettle

SmartSettle is a negotiation support system that uses the Internet to enable the interaction among project stakeholders with conflicting objectives that wish to reach an agreement. A facilitator is required to model the problem and to represent preferences in way that can be used by the adopted optimization algorithms. SmartSettle uses a joint session area to compose a Framework for Agreement with natural language messages. Preferences can be represented using satisfaction graphs. The SmartSettle negotiation process further uses optimization algorithms to transform conflicting objectives into fair and efficient solutions and to generate suggestions before an agreement is reached. After a tentative agreement is reached, SmartSettle looks to improve the situation by fairly distributing gains to both parties. The use of these built-in optimization algorithms leads to solutions maximizing the mutual satisfaction for all stakeholders.

A facilitator guides stakeholders through the stages of the SmartSettle process, including the following stages: Prepare for negotiation, qualify interests (the elicitation of stakeholder objectives and draft of framework for agreement), qualify satisfaction (preference elicitation), establish equity (suggestion of solutions and acceptance of tentative agreement), maximize benefits (refinement of preferences including optimization), and secure commitment.

In Sect. 7.2 we discussed a general negotiation processes and explained important activities done during pre-negotiation, the actual conduct of the negotiation, and during post-negotiation. Table 7.3 shows that specific implementations of this general process emphasize different stages. For example, Negoisst provides a strong message model supporting the actual negotiation. EasyWinWin supports both pre-negotiation and negotiation activities but its negotiation model is less rigorously enforced.

Table 7.3 Comparison of negotiation tools

Dimension/Tool	*Aspire*	*Negoisst*	*EasyWinWin*	*SmartSettle*
Specific implementations of negotiation process	**Pre-Negotiation** * Negotiation preparation	**Pre-Negotiation** * Define categories for negotiation	**Pre-Negotiation** * Define negotiation purpose, negotiation topics, and glossary of terms * Identify success-critical stakeholders * Elicit win conditions * Prioritize win conditions * Reveal issues & constraints	**Pre-Negotiation** * Negotiation preparation * Qualify interests * Qualify satisfaction
	Negotiation * Conduct of negotiation (offers and counter-offers)	**Negotiation** * Conduct of negotiation (request, offer, counter-offer, accept, reject, question, clarification)	**Negotiation** * Identify issues and options * Negotiate agreements	**Negotiation** * Establish equity * Maximize benefits
	Post-Negotiation *Post-settlement	**Post-Negotiation** * Definition of contract	**Post-Negotiation** * QA reviews * Win-win spiral model iterations	**Post-Negotiation** * Secure commitments
Conflict resolution strategy	Competing	Competing	Collaborative compromising	Competing compromising
Collaboration situation	* Different time – different place	* Different time – different place	* Same time – same place * Same time – different place	* Different time – different place
Negotiation support	Pro-active interventive	Active facilitative	Active facilitative	Active facilitative

Similarly, differences can be seen in the conflict handling dimension: Aspire supports a conflict-oriented approach where two stakeholders can exchange offers and counters, whereas EasyWinWin emphasizes a collaborative conflict resolution based on problem-solving by a team. The chosen negotiation tools support different time/different place interaction with the exception of EasyWinWin, which is weaker in this respect and assumes synchronous interaction in most of its negotiation steps. With respect to the degree of negotiation tool support, Aspire is the only tool that can be classified as pro-active interventive as its Atin agent continuously monitoring negotiations and giving guidance to stakeholders.

7.5 Conclusions

In this chapter our aim was to give an overview of the state-of-the art by explaining important negotiation steps; introducing a three-dimensional framework that covers the conflict resolution strategy, the collaboration situation of stakeholders involved, and the degree of negotiation support; and by discussing existing negotiation approaches in the context of this framework. Beyond its value for classifying existing and future research the purpose of the framework is to assist practitioners to understand important issues when implementation negotiation processes. Although some progress has been made in the area of requirements negotiation by researchers and practitioners, there are still many open issues requiring further research. The discussion of the requirements negotiation dimensions already defined some candidate areas. In particular, investigating the complex interdependencies between the dimensions leads to some interesting questions. For example, finding the most effective negotiation processes for a given negotiation problem, expected conflict behavior, collaboration situation, and adopted tools. For the future, we expect several developments for requirements negotiation which pose some interesting research challenges:

Scalability. Researchers have been developing numerous methods and tools supporting negotiations. Often, these systems are applicable to small problems only and do not scale up to real-world situations which are characterized by many stakeholders and many issues (which is the case in most real-world software projects).

Integration of fields. Software engineering researchers have been developing approaches, often not aware of research going on in the NSS community. While pragmatic approaches such as EasyWinWin work quite well in real-world settings, complementing it with techniques and tools from the NSS community would be beneficial. We hope to see the better integration approaches from different fields.

Novel tools. New technological developments will result in more sophisticated negotiation support. For example, mobile computing enables stakeholders to participate in negotiations in new collaboration situations more easily. First prototypes of such tools have already been developed [55].

Multi-stakeholder distributed systems. A further challenge comes from the fact that more and more applications, especially those that are developed and deployed over the web, represent so-called multi-stakeholder distributed systems, "... in which subsets of the nodes are designed, owned, or operated by distinct stakeholders." [30] These nodes are often designed or operated in ignorance of one another or with different, possibly conflicting goals. Negotiation approaches will become even more important in such a context as the requirements placed by diverse stakeholders are often ephemeral and conflicting. Furthermore, details about the elements of such a dynamic system are largely unknown to single stakeholders and outside their sphere of control [29].

Handling cultural differences. Negotiation is a complex decision process which is influenced by political, psychological, sociological and organizational aspects and cannot be formally represented. For example, there is currently only limited understanding of the impact of corporate and national culture on requirements negotiation. Some approaches exist [40], but we have mostly only tacit expertise and anecdotal evidence. A research challenge is to develop negotiation processes, techniques, and tools that better understand and handle the impact of corporate and national culture.

References

1. (2002) Workbook on international negotiation. Netherlands institute of international relations Clingendael, 69p.
2. Beck K (1999) Extreme programming explained: Embrace change. Addison-Wesley
3. Boehm BW (1988) A spiral model of software development and enhancement. IEEE Computer. 21(5): 61−72
4. Boehm BW (2000) Requirements that handle IKIWISI, COTS, and rapid change. IEEE Computer. 33(7): 99−102
5. Boehm BW (2000) Spiral development: Experience, principles and refinements. Han-sen WJ, Editor, CMU/SEI-00-SR-08
6. Boehm BW, Abi-Antoun M, Port D, Kwan J, Lynch A (1999) Requirements engineering, expectations management, and the two cultures. In: Proceedings of IEEE International Symposium on Requirements Engineering, pp.14−22
7. Boehm BW, Bose P (1994) A collaborative spiral software process model based on Theory W. In: Proceedings of Conference on the Software Process, pp.59−68
8. Boehm BW, Bose P, Horowitz E, Lee MJ (1994) Software requirements as negotiated Win conditions. In: Proceedings of IEEE CS 1st International Conference on Requirements Engineering. Colorado Springs, Colorado, USA
9. Boehm BW, Egyed AF, Kwan J, Port D, Shah A, Madachy R (1998) Using the Win-Win spiral model: A case study. IEEE Computer. 31(7): 33−44
10. Boehm BW, Port D, Al-Said M (2000) Avoiding the software model-clash spiderweb. IEEE Computer, pp.120-123
11. Boehm BW, Ross R (1989) Theory-W software project management: Principles and examples. IEEE Transactions on Software Engineering, 15(7): 902−-916
12. Bose P, Zhou X (1999) WWAC: WinWin abstraction based decision coordination. In: Proceedings of International Conference on Work activities Coordination and Collaboration. San Francisco, California, United States: ACM Press, pp.127−136
13. Briggs RO, de Vreede GJ, Nunamaker JF (2003) Collaboration Engineering with ThinkLets to pursue sustained success with group support systems. Journal of Man-agement Information Systems, 19(4): 31−63

14. Briggs RO, Grünbacher P (2002) EasyWinWin: Managing complexity in requirements negotiation with GSS. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS-35.02). Big Island, Hawaii

15. Curtis B, Krasner H, Iscoe N (1988) A field study of the software design process for large systems. Communications of the ACM, 31: 1268−1287

16. Damian D (2001) Negotiation behavior and group interaction in face-to-face and distributed requirements negotiations: four case studies. In: Proceedings of the 6th Australian Workshop on Requirements Engineering. Sydney, Australia, pp.22−31

17. Damian D, Eberlein A, Shaw M, Gaines BR (2000) Using different communication media in requirements negotiation. IEEE Software. 17(3): 28−36

18. Damian DE, Eberlein A, Shaw MLG, Gaines BR (2003) An exploratory study of facilitation in distributed requirements engineering. Requirements Engineering Journal 8(1): 23−41

19. Deutsch M (1973) The resolution of conflict. Yale University Press, New Haven

20. Easterbrook S (1991) Handling conflict between domain descriptions with computer-supported negotiation. Knowledge Acquisition: An International Journal, 3: 255−289

21. Egyed A, Grünbacher P (2004) Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help. IEEE Software, November/December, pp.50−54

22. Fickas S, Feather M (1995) Requirements monitoring in dynamic environments. In: Proceedings of 2nd IEEE International Symposium on Requirements Engineering, pp.140−147

23. Fisher R, Ury W (1983) Getting to yes: Negotiation agreement without giving in. New York. Penguin Books

24. Fjermestad J, Hiltz R (2000) Case and field studies of group support systems: An empirical assessment. In: Proceedings of 33rd International Hawaii Conference on System Science, January, Mauii, Hawaii, 1: 4−7

25. Galin A, Gross M, Gosalker G (1993) E-negotiation versus face-to-face negotiation. What has changed - If anything? , Tel Aviv University: Tel Aviv, Accessed on 3rd December 2004, http://www.recanati.tau.ac.il/research/IIBR/obhr/amira_miron.doc

26. Grünbacher P, Braunsberger P (2003) Tool support for distributed requirements negotiation. In: Cooperative methods and tools for distributed software processes. De Lucia A, Gall H (Eds.) FrancoAngeli: Milano, Italy, pp.56−66.

27. Grünbacher P, Briggs RO (2001) Surfacing tacit knowledge in requirements negotiation: Experiences using easy WinWin. In: Proceedings of 34th Hawaii International Conference on System Sciences, 3-6 January, Maui, Hawaii, Vol.1, pp.1024

28. Grünbacher P, Halling M, Biffl S, Kitapci H, Boehm BW (2004) Integrating collaborative processes and quality assurance techniques: Experiences from requirements negotiation. Journal of Management Information Systems, 20(4): 9−29

29. Grünbacher P, Stallinger F, Maiden NAM, Franch X (2003) A negotiation-based framework for requirements engineering in multi-stakeholder distributed systems. Requirements Engineering and Open Systems (REOS). Monterey, CA, Accessed on 3rd December 2004, http://www.cs.uoregon.edu/~fickas/REOS/

30. Hall RJ (2002) Open modeling in multi-stakeholder distributed systems: requirements engineering for the 21st Century. In: Proceedings of 1st Workshop on the State of the Art in Automated Software Engineering. U.C. Irvine, Institute for Software Research

31. Herlea DE (1998) Computer supported collaborative requirements negotiation. In: Proceedings of KAW'98. Banff, Alberta, Canada, Accessed on 3rd December, 2004, http://ksi.cpsc.ucalgary.ca/KAW/KAW98/herlea/

32. Herlea DE (1999) User participation in requirements negotiation. ACM SIGGROUP Bulletin. 20(1): 30−35

33. In H, Roy S (2001) Visualization issues for software requirements negotiation. In: Proceedings of Computer Software and Applications Conference, pp. 10−15

34. Jelassi MT, Foroughi A (1989) Negotiation support systems: An overview of design issues and existing software. Decision Support Systems, 5: 167−181

35. Johansen R (1988) Groupware: Computer support for business teams, New York. The Free Press

36. Keeney RL, Raiffa H (1976) Decisions with multiple objectives: Preferences and value tradeoffs. J. Wiley & Sons, NY

37. Kersten G (2004) E-negotiation systems: Interaction of people and technologies to resolve Conflicts. In: Proceedings of 3rd Annual Form on Online Dispute Resolution 5-6 July, Melbourne Australia

38. Kersten G, Noronha SJ (1997) Negotiation via the World Wide Web: A cross-cultural study of decision making. An Interim Report, Access on 3rd December 2004, http://www.iiasa.ac.at/Publications/Documents/IR-97-052.pdf

39. Kersten GE, Lo G (2003) Aspire: Integration of Negotiation Support System and Software Agents for E-Business Negotiation. International Journal of Internet and Enterprise Management, 1(3): 293−315

40. Kersten GE, Noronha SJ (1999) Negotiations via the World Wide Web: A cross-cultural study of decision making. Group Decision and Negotiations, 8(3): 251−279

41. Kotonya G, Sommerville I (1996) Requirements engineering with viewpoints. Software Engineering Journal, 11: 5−18

42. Lamsweerde Av (2000) Requirements engineering in the year 00: A research perspective. In: Proceedings of the 22nd International Conference on Software Engineering. Limerick, Ireland, pp.5−19

43. Lamsweerde Av (2001) Goal-oriented requirements engineering: A guided tour. In: Proceedings of International Conference on Requirements Engineering'01 Tutorial Notes

44. Lim LH, Benbasat I (1992-93) A Theoretical Perspective of Negotiation Support Systems. Journal of Management Information Systems, 9(3): 27−44

45. Nunamaker JF, Briggs RO, Mittleman DD, Vogel DR, Balthazard PA (1997) Lessons from a dozen years of group support systems research: A discussion of lab and field findings. Journal of Management Information Systems, 13(3): 163−207

46. Nuseibeh B, Easterbrook S (2000) RE: A Roadmap. In: Proceedings of 22nd International Conference on Software Engineering, Special Issue: ACM-IEEE, pp.37−46

47. Park J, Port D, Boehm BW (1999) Supporting distributed collaborative prioritization. in Software Engineering Conference, pp. 560−563

48. Pruitt DG, Carnevale PJ (1993) Negotiation in social conflict. Buckingham. Open University Press

49. Rapoport A (1974) Game theory as a theory of conflict resolution. D. Reidel Publ. Co., Dordrecht, Holland

50. Robbins S (1989) Organizational behavior: Concepts, controversies and applications. 4th edition, Prentice Hall, NJ

51. Robinson WN, Fickas S (1994) Supporting multi-perspective requirements engineering. In Proceedings of IEEE Conference on Requirements Engineering, pp.206−215
52. Robinson WN, Volkov V (1998) Supporting the negotiation life cycle. Communications of ACM, 41(5): 95−102
53. Schmid B, Lindemann M (1993) Elements of a reference model for electronic markets. In: Proceedings of the 31st Hawaii International Conference on System Sciences, IEEE Computer Society Press, pp.193−200
54. Schoop M, Jertila A, List T (2003) Negoist: a negotiation support system for electronic business-to-business negotiations in e-commerce. Data & Knowledge Engineering, 47(3): 371−401
55. Seyff N, Grünbacher P, Maiden NAM, Tosar A (2004) RE Tools Go Mobile. In: Proceedings of the 26th IEEE International Conference on Software Engineering (Research Demo), IEEE Computer Society Press.
56. Sharp H, Finkelstein A, Galal G (1998) Stakeholder identification in the requirements engineering process. In: Proceedings of 10th International Workshop on Database & Expert Systems Applications. Florence, Italy, pp.387−391
57. Souren P (2001) Collective memory support in negotiation: A theoretical framework. In: Proceedings of the 34th Hawaii International Conference on System Sciences, pp.1−8
58. Standish Group (2001) Extreme CHAOS report. The Standish Group, 196 Old Townhouse Road, West Yarmouth, MA 02673 -- http://www.standishgroup.com
59. Strauss A (1978) Negotiations: Varieties, contexts, processes and social order. Jossey-Bass Publishers, San Francisco, CA
60. Sutcliffe AG (2002) User-Centred Requirements Engineering. Springer, London
61. Thomas K (1976) Conflict and conflict management. In: Handbook of industrial and organizational psychology. Dunnette MD (Ed.) Rand McNally College Publishing Company, Chicago, pp.889−935

Author Biography

Paul Grünbacher is an Associate Professor at Johannes Kepler University Linz, Austria and a research associate at the Center for Software Engineering (University of Southern California, Los Angeles). He studied Business Informatics and holds a Ph.D. from the University of Linz. Paul's research focuses on applying collaborative methods and tools to support and automate complex software and system engineering activities such as requirements acquisition and software inspections. He is a member of ACM, ACM SIGSOFT, and IEEE.

Norbert Seyff is a Research Assistant at the Johannes Kepler University Linz, Austria where he received his Master's degree (Dipl.-Ing.) in Computer Science. Within the scope of his ongoing Ph.D. research Norbert is developing and evaluating innovative methods and tools supporting mobile stakeholders and analysts in acquiring and negotiating requirements.