Negotiation Constellations – Method Selection Framework for Requirements Negotiation

Samuel Fricker[1,2] and Paul Grünbacher[3]

[1] University of Zurich, Department of Informatics
Binzmuehlestrasse 14, 8057 Zurich, Switzerland
fricker@ifi.uzh.ch
[2] ABB Switzerland Ltd., Power Systems
Bruggerstrasse 72, 5400 Baden, Switzerland
samuel.fricker@ch.abb.com
[3] Johannes Kepler Universität (JKU),
Institute for Systems Engineering and Automation
4040 Linz, Austria
paul.gruenbacher@jku.at

Abstract. Customers, product managers, project leaders, architects, engineers, and other stakeholders are negotiating requirements throughout the software lifecycle. Even-though fundamental for understanding requirements engineering, negotiation has not been as thoroughly studied as other facets of this engineering discipline. This paper casts requirements engineering into the landscape of negotiation by describing a framework for selecting tactics and methods for various negotiation constellations that can be encountered in a software organization. The framework opens perspectives that are essential for understanding the behavior of people involved in development projects, for understanding how development teams and stakeholders create mutually satisfactory solutions, and for giving tactical advice to practitioners.

1 Introduction

Software development is embedded in a complex network of stakeholders that include roles like customers, development managers, product managers, team leaders, architects, developers, testers, and maintainers [10]. The interplay between these stakeholders is a fundamental success factor, as every role brings essential knowledge, capabilities, and skills that are essential to design great new products.

However, designing appropriate requirements engineering processes for such complex stakeholder networks is still a major challenge [7]. For instance, there are multiple organizational interfaces at which requirements are engineered: it can be observed that stakeholders pursue their own objectives by trying to delegate the fulfillment of some goals while satisfying those of others [33]. This happens not only during early-phase requirements engineering, but also in design and change management activities throughout the whole development process [5,11].

The lack of approaches for tailoring requirements engineering to the structure of stakeholder networks and to the specific negotiations situations between these stakeholders

B. Paech et al. (Eds.): REFSQ 2008, LNCS 5025, pp. 37–51, 2008.

leads to misunderstandings and conflicts. As a consequence, development effort is wasted on insignificant features, rather than being invested on features that are most essential for stakeholder satisfaction.

In response to these challenges posed by complex stakeholder networks, this paper presents a framework for helping stakeholders to understand their *negotiation constellations* and for *selecting appropriate negotiation tactics and methods*. The proper selection of negotiation tactics and methods enables effective communication and acknowledgment of requirements, helps exploiting opportunities for stakeholder satisfaction by creating win-win situations, and establishes trust relationships that are important for development efficacy and high-impact development results.

Beyond its usefulness for practitioners, we hope that the framework will aid requirements engineering researchers to structure and understand the landscape of negotiation in requirements engineering. The framework references knowledge from the broad field of negotiation and identifies a number of research opportunities for understanding on how to handle requirements adequately in specific stakeholder constellations.

This paper presents the negotiation constellations framework and its implications on requirements engineering practice and research. The paper is structured as follows: Section 2 outlines background and related work. Section 3 presents the negotiation constellation framework. Section 4 illustrates the use of the framework. Section 5 discusses the presented work. Section 6 summarizes and concludes the paper.

2 Background and Related Work

This work has been motivated by challenges identified at ABB. It relates to a case where product managers coordinate distributed development teams with requirements that are derived from agreements with diverse stakeholders. Conflicts arise almost inevitably in such cases as project stakeholders pursue mismatching goals and try to influence each other [12,19]. For example, in a software product organization goals need to be considered from the market, partners, customers, users, company management, sales & marketing, research & innovation, consultants, development, and support [1,32]. Successful requirements engineering demands agreement on the requirements [15].

Key approaches that can be applied to reach such an agreement include analysis of viewpoints [14], stakeholder and goal modeling [15,33], and negotiation [6,12,16,31]. The negotiation process starts when the stakeholders communicate their goals. It ends when all have agreed to a specified contract [26].

There are two fundamental ways to manage this negotiation process with regard to how agreements are established in the stakeholder network. First, the process can be managed by a requirements engineer who elicits the positions and perspectives of stakeholders, documents them in a comprehensive goal model, facilitates the resolution of conflicts, and communicates the obtained global stakeholder agreement.

Second, the negotiation process can emerge out of the activities of stakeholders that perform the organizational roles they are assigned to. Instead of one large negotiation that involves all stakeholders, negotiation is carried out as a number of

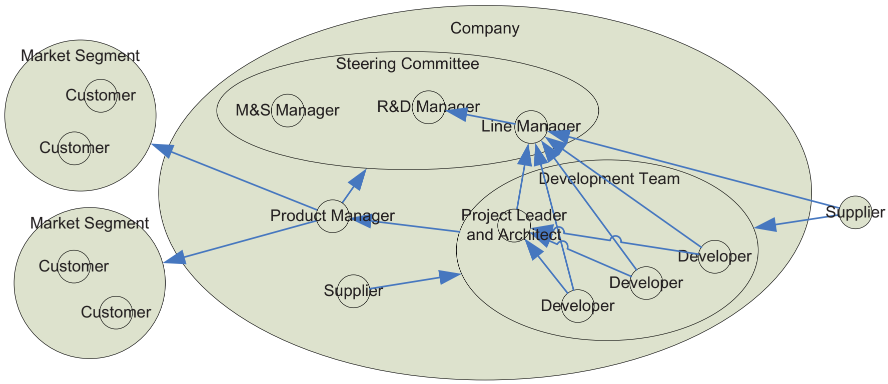


Fig. 1. Exemplary contract model of a software organization. Ellipses represent hierarchically nested groups of people or individuals. Arrows represent contracts that are agreed upon.

small-scale activities that are performed rather independently. This leads to a number of agreements between different stakeholders [7]. An example of such distributed negotiations is illustrated by the contract model shown in Fig. 1.

Independent of the process flavor, questions about the tactical approach and methodology appear in these different negotiation constellations. Requirements engineers needs to understand how to perform win-win negotiations, how to reach value-creating results, and how to deal with group dynamics. Stakeholders need to understand their role in the negotiation process and what they can and should do to achieve their objectives by influencing other stakeholders. Hence, the following issues need to be addressed:

- Correctly conceptualizing the negotiation constellation,
- Understanding the advantages and limitations of the constellation,
- Knowing the negotiation tactics and methods appropriate for the constellation,
- Identifying those stakeholders that need to be involved in negotiation, and
- Selecting and pursue the most appropriate negotiation approach.

The knowledge in the *negotiation constellations framework* assists stakeholders with these questions and provides negotiation advice. It also is used to improve requirements engineering processes by capturing, organizing and making available good negotiation practices and experiences.

The negotiation constellations framework is similar to reference models like CMMI for software process improvement [13], and the good practice guide for requirements engineering improvement [29]. It focuses, however, on requirements negotiation and adds criteria for selecting tactics and methods that are based on the situations in which they are applied. In contrast to other reference models, the negotiation constellations framework also supports capturing and structuring experience to support learning software organizations [28].

3 Negotiation Constellations

Understanding negotiation constellations is essential for efficiently finding good agreements among stakeholders. This section elaborates how the negotiation constellation framework advises practitioners and supports requirements engineering process improvement by describing its structure and use.

Negotiation is an interpersonal decision-making process to find a mutually acceptable agreement to a conflict [16,31]. Agreements can contain the planned realization of needs and objectives, the use of capabilities, the guarantee of financial or other backing, or the provision of knowledge [8].

A *negotiation constellation* is characterized by a number of facets that influence the selection of negotiation methods. Key facets include the characteristics of the negotiating parties, the relationships between these parties, and the negotiation object [31]. Other facets include the geographical distance between the parties [6] and their expected conflict behavior [30].

The *negotiation constellations framework* describes a taxonomy of negotiation constellations and provides specific advice for negotiation tactic, methodology, and experience for a given negotiation constellation. The framework was shaped to be relevant, simple, specific, and orthogonal. It contains knowledge that is useful for advising practitioners in a software development context. The number of taxonomic units is intentionally kept small. The decision criteria are simple and can be applied intuitively. The advice is given at a coarse level of granularity that still allows differentiating negotiation approaches. The number of fields in which the same negotiation tactics and techniques are found is minimized, however without compromising specificity.

The negotiation constellations framework has been defined with the following research process in collaboration with practitioners. Situations have been identified that require applying different negotiation tactics and techniques. These situations were then exemplified with stereotypical descriptions of software development organizations and relationships between various organizational roles. Finally, negotiation, requirements engineering and software engineering literature was studied to identify tactics and methods that adequately address the negotiation situations.

Subsection 3.1 describes commonalities of negotiation situations in a software engineering context. Subsection 3.2 describes the taxonomy of negotiation constellations. Subsections 3.3 and 3.4 describe tactical and methodological advice.

3.1 Common Negotiation Characteristics in Software Organizations

Negotiation has been studied in many different contexts, including product sales, employment contracts, personal affairs, politics, and peace keeping. Negotiation occurs 1) to agree on how to share or divide limited resources such as money, time and staff; 2) to create something new that neither party could do on its own; or 3) to resolve a conflict between parties. By choosing options other than negotiation, people may fail to achieve their goals, get what they need, or manage conflicts as smoothly as they might like to [16].

Negotiation in a software organization is special because a number of factors in the negotiation context are predetermined. This significantly reduces the variability of general negotiation situations and allows simplifying the negotiation constellation framework. The factors that are specific to software organizations concern the negotiation object, conflict management, and opportunities for renegotiations.

Bargaining over a *single issue* like a price is rare. Instead, people seek win-win results that occur when a mutually acceptable solution is sought. Win-win negotiation involves a *number of issues that are negotiated together*. For instance, a customer may want to reduce the price of a software solution or service, but this is typically negotiated together with other contractual elements like the scope of the solution or service. In other circumstances, people negotiate a set of concerns and objectives such as needs, requirements, and design decisions.

A number of conflict resolution styles are differentiated in negotiation, depending on the negotiators interest in his own outcome and in the other negotiator's outcome [27]. This dual-concerns model is illustrated in Fig. 2.

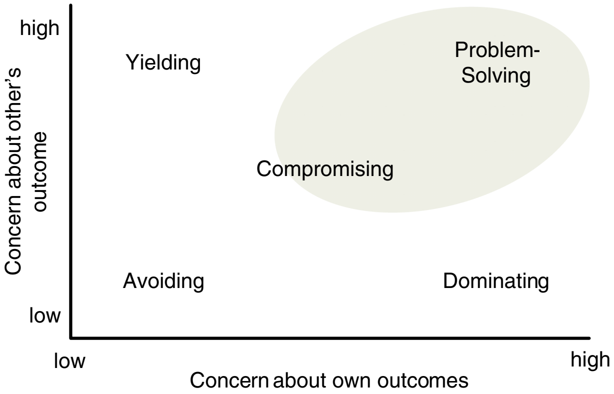


Fig. 2. Dual-concerns model of negotiation behavior [27]. The grey area highlights the conflict resolution style in a software organization centered on problem solving or compromising.

The conflict resolution style that should preferably be adopted in a software organization is *problem solving*, or *compromising* when consensus cannot be reached [23]. The issues that are negotiated in a software organization are complex: a synthesis of ideas is needed to come up with mutually satisfactory solutions, and time is available for such problem solving. Resources, skills and knowledge are possessed by different parties. Hence, commitment is needed from these other parties for successful implementation, with one party alone not being able to solve the negotiated problems. *Yielding* to another party should not be done because the issues negotiated are important, in the responsibility of the negotiators, and ultimately connected to their career. For the same reason, *avoiding* the other party is inappropriate. Finally, the other party should not be *dominated* because the negotiated issues are too complex and the negotiation partners have high degree of competence in their areas.

In software organizations, a number of *opportunities for renegotiation* are institutionalized. For example, change management processes are established to account for imperfect design and technology evaluation and planning. Hence, agreements are not carved in stone and may be changed. Still, the negotiators should be concerned about their reputation, because excessive and late use of renegotiation may severely weaken their position as an accepted negotiation partner.

The generic negotiation tactic in a software organization is integrative negotiation: be prepared, create value, and claim your share of the created value [31].

During *preparation* a negotiator[1] assesses his aspirations, his best alternative to a negotiated agreement (BATNA), and his reservation point at which he would stop the negotiations. He tries to elicit the same information from the negotiations partners by possibly revealing his aspirations, but without disclosing his BATNA and reservation point. In addition, he takes the situational factors into consideration that are described by the negotiation constellations framework.

Value creation can be achieved with creative conflict resolution. Good ideas can be identified when the negotiators trust each other, share information, and adjust the negotiation issues. Value can be created by capitalizing on differences in the valuation or preferences for goals, the forecast of the future, risk attitudes, time preferences, and capabilities. For example, a product marketing manager wanting to realize a number of product features may be faced with different design ideas by a development team of how such features can be implemented. The negotiation will cover a stage where the design ideas are created and evaluated by these parties.

In the late stage of a negotiation, the negotiators increasingly *claim value*. A negotiator claims value by a steadily improving its BATNA, anchoring the negotiation in the area of its aspirations, and planning for a sequence of concessions. To support value claiming, he can appeal to a number of facets to fairness, including equality, where all should get equal shares, equity, where the share is proportional to the party's contribution, and need, where share is proportional to the party's need.

Fig. 3 presents a model that explains the interrelationships of creating and claiming value in multi-issue negotiations [24].

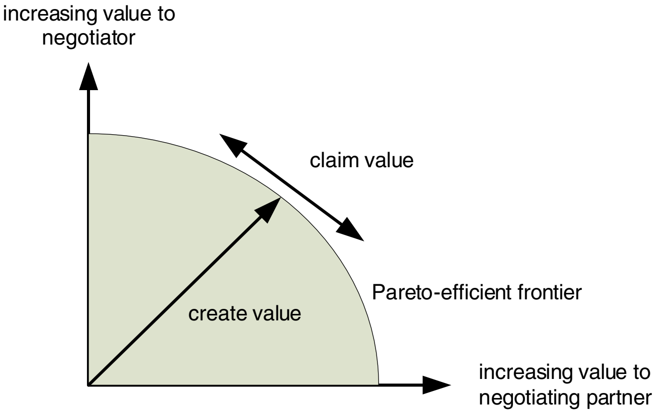

increasing value to negotiator
claim value
Pareto-efficient frontier
create value
increasing value to negotiating partner

Fig. 3. Conceptualization of creating and claiming value [16]. Maximal value is created when a point on the Pareto-efficient frontier is reached.

[1] For legibility reasons, we use the term 'he', but mean both sexes.

3.2 Characterization of the Negotiating Parties

To select appropriate negotiation tactics and methods, a negotiator needs to know his and his negotiation partner's constitutions. Fig. 4 shows the taxonomy of such constitutions, which is fundamental to the negotiation constellation framework.

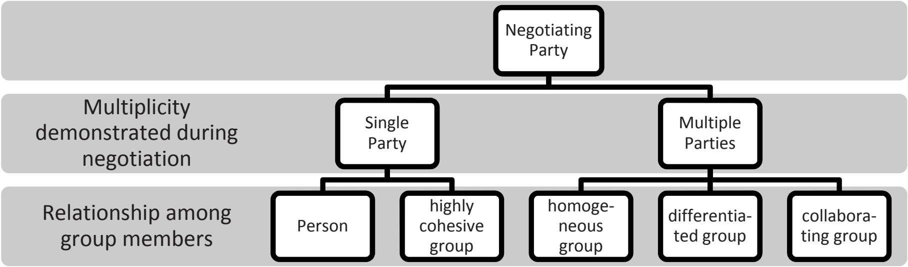


Fig. 4. Constitutions of negotiating parties shown with 'is-a-kind-of'-refinements

A *single party* is a person or a highly cohesive group of people. A single party has one set of aspirations, one BATNA, one reservation point, and one voice at the negotiation table. No internal fragmentation exists: there is neither intrapersonal conflict of the person nor interpersonal conflict in the cohesive group of people.

Typical roles of individual people in a software organization are line manager, product marketing manager, project manager, or architect. Examples of groups of people that appear as a single party at a negotiation table are a company in the role of a customer or supplier, management of a company when negotiating with employees, and a development team when negotiating with stakeholders.

The differentiation between a person or a highly cohesive group of people is not further used in the negotiation constellations framework. Both should use the same negotiation tactics and processes during a negotiation. It is likely, however, that in the course of software development, a group may recognize that it is not as cohesive as perceived initially. This can lead to a different negotiation situation and may require switching the mode of negotiation.

Multiple parties are a group of people that appears at one side of the negotiation table. In contrast to the single party, the constitution of the group is important. The group can consist of single parties or again other groups. The multi-party group is characterized by at least one of the following properties: the group members pursue different objectives, have different BATNA and reservation points, and have individual voices at the negotiation table. Since group members are differentiated, agreements made at the primary negotiation table should be ratified.

Typical examples of multiple parties are companies that make up a market, software users, management, a steering committee when negotiating with a project manager, an architecture team when negotiating with a product marketing manager, and a project team when negotiating with its project manager.

For the purpose of negotiation tactic and method selection, homogeneous groups, differentiated groups, and collaborating groups are distinguished. *Homogeneous* *groups* consist of members that have the same aspirations, BATNA and reservation point, but have individual voices. All members are willing to comply with negotiation results that are equal for everyone. A typical example is users within a user group.

A member of a *differentiated group* has, in addition to an individual voice, the desire to be different from the other group members. This leads to different aspirations, BATNA, and reservations points. Members of such a group are often competing with each other. A typical example is technology suppliers.

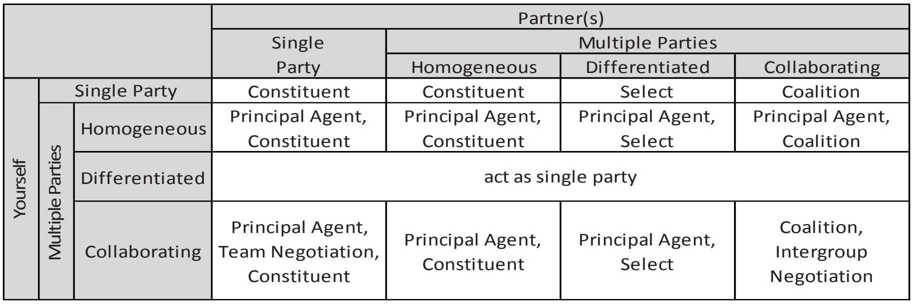
Members of a *collaborating group* also have individual voices, aspirations, BATNA, and reservation points. In contrast to the differentiated group, they seek an agreement that is satisfactory for every member. Rather than being in competition, the members of a collaborating group have different perspectives on the negotiation topic and have complementing aspirations, knowledge, networks, and capabilities.

3.3 Micro-level: Negotiation Tactics

The objective of the negotiation constellations framework is to help people in a software context to negotiate better. At a micro-level, the framework offers partisan tactical advice to a negotiator at the possible expense of his negotiation partner. Still, this advice is fair, because it is open for everyone. At the macro-level the framework offers methodological advice that helps all involved parties.

The *tactical negotiation constellation framework* differentiates between the negotiator who benefits from the advice and his negotiating partners. Both are involved in a negotiation that ultimately results in decisions about requirements, project plans, architectural design, and the like. The framework allows the negotiator to understand his negotiation constellation in terms of who he is and who the other is, and suggests tactical actions that strengthen his negotiation position.

The tactical negotiation constellations framework is shown in Fig. 5. The presented tactical advice is based on standard negotiation textbooks [31].



			Partner(s)			
			Single Party	Multiple Parties		
				Homogeneous	Differentiated	Collaborating
Yourself	Single Party		Constituent	Constituent	Select	Coalition
	Multiple Parties	Homogeneous	Principal Agent, Constituent	Principal Agent, Constituent	Principal Agent, Select	Principal Agent, Coalition
		Differentiated	act as single party			
		Collaborating	Principal Agent, Team Negotiation, Constituent	Principal Agent, Constituent	Principal Agent, Select	Coalition, Intergroup Negotiation

Fig. 5. Tactical advice for different negotiation constellations. Section 4 exemplifies.

The advice can be read out from the negotiation constellations framework by consulting the cell that corresponds to the negotiator's perception of himself and of his partner. For example, if the negotiator is a single party, with multiple homogeneous partners, he can increase the value of what he gets or speed up the negotiations by influencing the partners through constituents.

While the provided advice is specific to the constitution of the both negotiators, the framework shows that some decisions depend only on the constitution of the negotiator, and other decisions on the constitution of the negotiation partner. For example, homogeneous parties can be influenced with a constituent, independently of the structure of the primary negotiator. Acting as a single party helps the negotiator who is in competition with peers, independent of the negotiating partner.

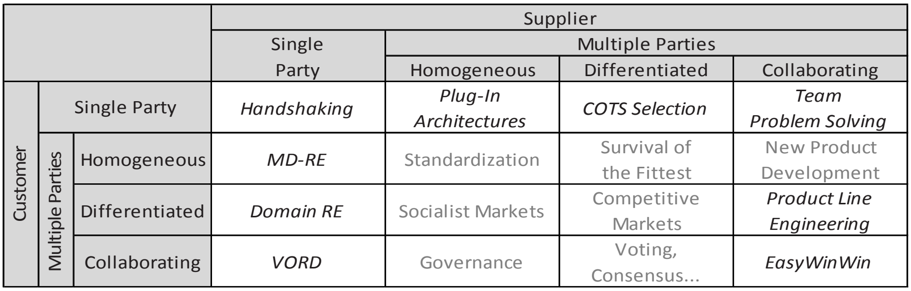
Table 1 explains the tactics suggested by the tactical negotiation constellations framework. A discussion of the advantages and risks of using the negotiation tactics can be found in standard textbooks [31].

Table 1. Explanation of negotiation tactics

Tactic	Explanation
Constituent	The use of peripheral players that have an indirect stake in the outcome to exert pressure on the other side.
Select	Stick to the party with the most promising outcome.
Coalition	Exert influence on outcomes by collaborating with a minimal but sufficient number of partners.
Principal Agent	Use an experienced agent to prepare or to run the negotiations on behalf of yourself.
Team Negotiation	Prepare and run the negotiations as a team to increase creativity and control of the negotiation.
Intergroup Negotiation	Control the conflicts that naturally appear in the confrontation of two or more groups.

3.4 Macro Level: Negotiation Methods

On a macro level, the negotiation constellations framework suggests methods and processes that maximize the value of the outcome and the satisfaction of the negotiators. The *methodological negotiation constellation framework* differentiates between generalized customer and supplier roles that engage in negotiations, without losing generality compared with the tactical framework. Fig. 6 shows the framework.



			Supplier			
			Single Party	Multiple Parties		
				Homogeneous	Differentiated	Collaborating
Customer	Single Party		*Handshaking*	*Plug-In Architectures*	*COTS Selection*	*Team Problem Solving*
	Multiple Parties	Homogeneous	*MD-RE*	Standardization	Survival of the Fittest	New Product Development
		Differentiated	*Domain RE*	Socialist Markets	Competitive Markets	*Product Line Engineering*
		Collaborating	*VORD*	Governance	Voting, Consensus...	*EasyWinWin*

Fig. 6. Methodological advice for different negotiation constellations. Italic entries refer to approaches from requirements or software engineering. The other entries describe metaphors for the negotiation constellations.

In addition to the self-assessment and the assessment of the negotiation partners, the negotiator analyzes the relationship to understand who is in a customer, supplier, or peer role. If he is in a customer role he places himself on a row, otherwise on a column. If some negotiators are peers, he and they together form a multiparty.

The method framework reflects in its basic form the state of knowledge. This implies that one, several, or no published methods can be identified for the various negotiation constellations. This advice should evolve by new research results and by the experiences made by those using it.

For example, for the one customer – one supplier constellation, one well-fitting method could be identified. The two parties will reach the best results if handshaking [24] is adopted for negotiation.

For the one customer – differentiated suppliers constellation, several methods could be identified. As long as the principles underlying these methods are not elaborated from the specific perspective of the negotiation situation, the negotiators have to select the best-fitting method. Such selection needs to be based on a refined understanding of the issues that are negotiated and the capabilities of the candidate methods. For example, to procure COTS software from candidate suppliers, a customer will employ one of the many supplier and COTS selection methods [17].

For the differentiated customers – differentiated suppliers constellation, fitting methods are hard to find. Here the framework only indicates a metaphor for approaching the situation. For example, to describe the behavior of customers in a segmented market confronted with a number of software suppliers, the laws of competitive markets apply [21].

Table 2 references methods for those cells of the methodological negotiation constellations framework, for which methods could be identified. These methods represent the initial recommendations that are evaluated for the given negotiation constellation and adjusted as experience and improved state of knowledge suggest.

Table 2. Methods fitting the various negotiation constellations

Method	Short Description
Handshaking	The use of implementation proposals to control understanding of communicated requirements [24].
Plug-in Architecture	Software design for extensibility by defining consistent ways and means of third-party software integration [18].
COTS Selection	The use of criteria for selecting commercial off-the-shelf products for system development [3].
MD-RE	Market-driven requirements engineering addresses the management of requirements for a number of customers [25].
Domain-RE	Identify and analyze common and variable requirements [20].
Product Line Engineering	Develop software for heterogeneous needs [20].
VORD	The capture, analysis and resolution of different needs and ideas with viewpoints [14].
EasyWinWin	Multi-party requirements negotiation approach [2].
Team Problem Solving	Defining solutions to problems in a team [17].
Standardization	Establish a consistent technical specification for a number of players [22].
New Product Development	Coordination of roles for new product development [4].

4 Framework Use for Tactical and Methodological Advice

This section illustrates the use of the negotiation constellations framework based on the company described in Fig. 1. The illustration follows a narrative that is inspired by experienced practice. Careful empirical evaluation, however, is ongoing work.

The narrative and the contract models in Fig. 7 describe how various roles in the software organization perceive their negotiation context and use the negotiation constellations framework for advice on how to proceed tactically and methodologically. As such perception is highly personal, the decisions by the players represent just one of many possible courses of actions.

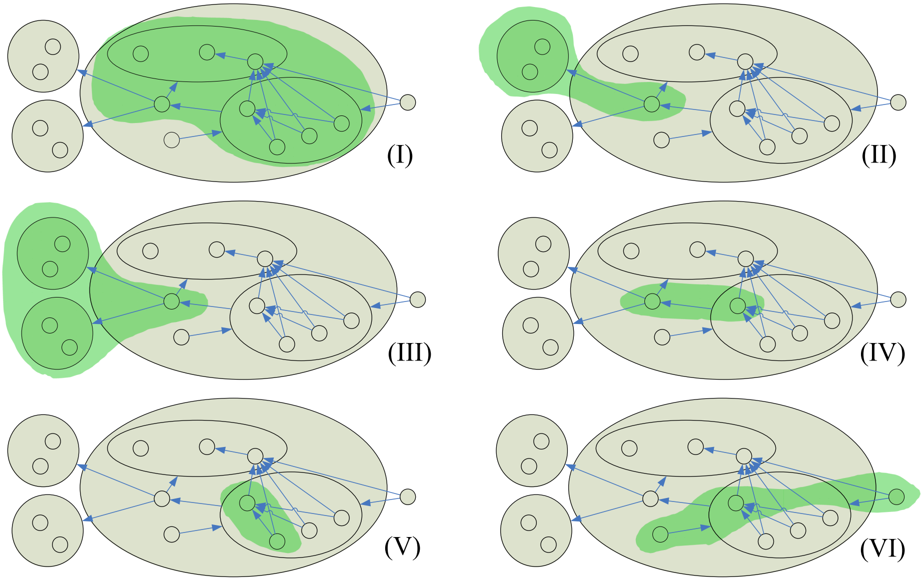


Fig. 7. Negotiation constellations, highlighted as shaded areas, in the organization described in Fig.1. The negotiation constellation framework provides tactical and methodological advice for such constellations.

(I) The *project leader and architect*, a member of the *development team*, is responsible for establishing architectural decisions that satisfy the needs represented by the stakeholders *product manager* and *steering committee* and for committing *developers* to implement the software according to these decisions. In this situation, he is confronted with a number of collaborating customers, the stakeholders, and a number of suppliers, himself and the *developers*. The negotiation constellations framework suggests using EasyWinWin as a methodology, building coalitions, and dealing with intergroup negotiation issues.

(II) The *product manager* is responsible to understand the company's *markets* and to identify requirements that best address the *customers*' needs. In this situation he may look at the market as a single *market segment* with homogeneous *customer* needs. Here he is well-advised with market-driven requirements engineering as a methodology.

(III) Alternatively, the *product manager* may identify multiple *market segments* with differentiated groups of *customers*. In this situation he is better advised to follow a domain requirements engineering approach for better understanding the variability of the needs of the different segments.

(IV) At some moment, the *product manager* has produced a software requirements specification that he hands over to the *project leader and architect* of the *development team*. The *development team* sees itself as a number of collaborating people and decides to use the *project leader and architect* as a principal agent, as suggested by the tactical negotiation constellations framework. The requirements hand-over situation, thus, is reduced to a negotiation between two individuals that is best addressed by handshaking with implementation proposals.

(V) To further progress in the implementation of the software, the *project leader and architect* conveys architectural decisions and distributes tasks to individual *developers*. Here the advice is again to use handshaking.

(VI) Finally, the *project leader and architect* sees opportunities to speed up development work with components that can be procured from an in-house or from an external *supplier*. Here he is confronted with a selection task where he adopts COTS-selection as a method.

5 Discussion

5.1 Practical Considerations

Section 4 has shown how the negotiation constellations framework can be used to provide tactical and methodological advice in practical situations. It helps a person or organization to conceptualize a negotiation constellation, to understand the advantages and limitations of the constellation, to know which tactics and methods are appropriate, and to identify the stakeholders that should be involved. Hence, the framework helps to exploit the strengths of the negotiation constellation and to understand its limitations.

As above illustration has shown, negotiation in a software context is not a one-shot activity. Rather, a sequence of overlapping negotiations is performed in practice. These negotiations are overlapping in time and in the people that are involved. Skilled negotiators do not act passively, but proactively try to shape the negotiation constellations in an attempt to strengthen their negotiation position for increasing the chances to achieve their objectives. The negotiation constellations framework may evolve into a valuable tool to support such reflections and is a basis for shaping and describing negotiation strategies.

As people and organizations enact the negotiation tactics and methods, they gain experiences, which can be reused [28]. The negotiation constellations framework provides a structure and means for such reuse. When advice has worked well, it is supplemented with experience data. When tactics or methods have been discovered that fit the negotiation constellation better in the specific negotiation constellation, previous advice is replaced by improved advice.

5.2 Implications on Research and Education

In addition to practical benefit, the negotiation constellations framework opens a number of perspectives for research and education. The framework provides a structured approach to transfer knowledge from the field of negotiation into requirements and software engineering. The table cells refer to specialized negotiation literature through the named tactics.

The framework organizes knowledge based on simple criteria that are relevant for practice. It is thus a basis to study the applicability of tactics and methods by comparing the organizational contexts which they apply to.

In the same line, the negotiation constellations framework helps to better understand limitations of current knowledge in requirements engineering. While all cells are relevant, for some negotiation constellations it is hard to find focused requirements or software engineering methods.

Finally, negotiation has the potential to act as a model of how requirements are communicated and transformed into design decisions. A better understanding of negotiation in the software context will lead to a better understanding of the co-evolution of requirements and design.

5.3 Limitations

The negotiation constellations framework has been designed for simplicity. This may be in conflict with the complexity of the real-world situations, where it is intended to be used. Experienced skillful negotiators act in a much more multi-faceted manner than the negotiation constellation framework suggests by adjusting to factors like geographical distance and negotiation style. Also a negotiator is typically embedded into a complex network of partners, which is not represented by the simple customer-supplier relationship of the framework. Still the negotiation constellation framework is a useful starting point for companies that wish to address requirements negotiation in a systematic manner.

The tactical and methodological advice that is suggested by the negotiation constellations framework is incomplete and requires adaptation to an organization. If consensus on the superiority of a given negotiation approach is not possible, the negotiation constellations framework needs to be tailored to parts of the company, or even to a single role. The framework would still be useful for providing advice and capturing experience, but a number of instances will need to be managed.

The fields of negotiation, requirements engineering and software engineering are evolving. This is an opportunity for the framework to mature, as more specific tactics and methods are discovered. With the evolving fields, the knowledge that is stored in the framework can be completed and improved.

The research on the negotiation constellation framework is still in progress. The limitations highlighted here can only be answered with careful empirical validation.

6 Summary and Conclusions

The negotiation constellations framework aims to contribute to more effective requirements engineering by capturing and structuring tactical and methodological advice that is tailored to the organizational context of a stakeholder. The framework can be used for reflecting on the negotiation constellation, identifying other stakeholders, and obtaining guidelines for reaching agreements that increase the value of the software being developed. It may also be used for process development by providing a structure for organizing tactics and methods and to capture experience.

The negotiation constellations framework builds on the tradition of reference models like CMMI to support tactical decision-making and method selection in the area of requirements negotiation. In this role, it can help to make essential knowledge from the field of negotiation accessible to requirements engineers and software professionals and to give insights into current requirements engineering knowledge.

The paper presents and exemplifies the structure and use of the negotiation constellations framework in practical situations. It further provides specific references to tactical and methodological knowledge that can be used as a starting point for software professionals that want to address negotiation systematically and for companies that decide to adopt the framework as part of their process improvement.

Future work should cover empirical studies of how the framework is used and evolved, and of what its effects are on software quality and on learning software organizations. One aspect of interest is the evolution of the stakeholder network that emerges as a result from following a strategy built on the tactics proposed by the framework. Evaluation and comparison of requirements engineering methods from the perspective of the described negotiation constellations will make these methods better accessible to practitioners and further supports method selection.

References

1. Alexander, I., Robertson, S.: Understanding Project Sociology by Modeling Stakeholders. IEEE Software 21(1), 23–27 (2004)
2. Boehm, B., Grünbacher, P., Briggs, R.: Developing Groupware for Requirements Negotiation: Lessons Learned. IEEE Software 18(3), 46–55 (2001)
3. Cechich, A., Piattini, M., Vallecillo, A.: Component-Based Software Quality. LNCS, vol. 2693, pp. 99–127. Springer, Heidelberg (2003)
4. Cooper, R.: Winning at New Products: Accelerating the Process from Idea to Launch. Perseus Publishing (2001)
5. Curtis, B., Krasner, H., Iscoe, N.: A Field Study of the Software Design Process for Large Systems. Communications of the ACM 31(11), 1268–1287 (1988)
6. Damian, D., Eberlein, A., Woodward, B., Shaw, M., Gaines, B.: An Empirical Study of Facilitation of Computer-mediated Distributed Requirements Negotiations. In: 5th Intl. Symposium on Requirements Engineering (2001)
7. Doerr, J., Paech, B., Koehler, M.: Requirements Engineering Process Improvement Based on an Information Model. In: 12th IEEE Intl. Requirements Engineering Conference (2004)
8. Foa, U., Foa, E.: Resource Theory of Social Exchange. General Learning Press (1975)
9. Fricker, S., Gorschek, T., Myllyperkiö, P.: Handshaking between Software Projects and Stakeholders Using Implementation Proposals. In: 13th Intl. Working Conference on Requirements Engineering: Foundation for Software Quality (2007)
10. Glinz, M., Wieringa, R. (eds.): IEEE Software Special Issue on Stakeholders in Requirements Engineering 24(2) (2007)
11. Gorschek, T., Svahnberg, M.: Requirements Engineering in Practice: Studies of Six Companies. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 405–426. Springer, Heidelberg (2005)

12. Grünbacher, P., Seyff, N.: Requirements Negotiation. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 143–162. Springer, Heidelberg (2005)
13. Humphrey, W., Snyder, T., Willis, R.: Software Process Improvement at Hughes Aircraft. IEEE Software 8(4), 11–23 (1991)
14. Kotonya, G., Sommerville, I.: Requirements Engineering with Viewpoints. Software Engineering Journal 11(1), 5–18 (1996)
15. van Lamsweerde, A., Darimont, R., Letier, E.: Managing Conflicts in Goal-Driven Requirements Engineering. IEEE Transactions on Software Engineering 24(11), 908–926 (1998)
16. Lewicki, R., Barry, B., Saunders, D.: Essentials of Negotiation. McGraw-Hill, New York (2007)
17. Lumsdaine, E., Lumsdaine, M.: Creative Problem Solving. IEEE Potentials 13(5), 4–9 (1994)
18. Marquardt, K.: Patterns for Plug-ins. In: Manolescu, D., Voelter, M., Noble, J. (eds.) Pattern Languages of Program Design 5, pp. 301–336. Addison-Wesley, Reading (2006)
19. Ovaska, P., Rossi, M., Smolander, K.: Filtering, Negotiating and Shifting in the Understanding of Information System Requirements. Scand. J. of Information Systems 17(1), 31–66 (2005)
20. Pohl, K., Böckle, G., van der Linden, F.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer, Heidelberg (2005)
21. Porter, M.: Competitive Advantage: Creating and Sustaining Superior Performance. The Free Press (1985)
22. Poston, R.: Software Standards. IEEE Software 1(2), 87–94 (1984)
23. Rahim, M.: Rahim Organizational Conflict Inventories: Professional Manual. Consulting Psychologists Press (1990)
24. Raiffa, H., Richardson, J., Metcalfe, D.: Negotiation Analysis: The Science and Art of Collaborative Decision Making. The Belknap Press of Harvard University Press (2007)
25. Regnell, B., Brinkkemper, S.: Market-Driven Requirements Engineering for Software Products. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 287–308. Springer, Heidelberg (2005)
26. Robinson, W., Volkov, S.: Supporting the Negotiation Life Cycle. Communications of ACM 41(5), 95–102 (1998)
27. Rubin, J., Pruitt, D., Kim, S.: Social Conflict: Escalation, Stalemate and Settlement. McGraw-Hill, New York (1994)
28. Schneider, K., von Hunnius, J., Basili, V.: Experience in Implementing a Learning Software Organization. IEEE Software 19(3), 46–49 (2002)
29. Sommerville, I., Sawyer, P.: Requirements Engineering: A Good Practice Guide. Wiley, Chichester (1997)
30. Thomas, K.: Conflict and Conflict Management. In: Dunette, M. (ed.) Handbook of Industrial and Organizational Psychology, pp. 889–935. Rand McNally College Publishing Company (1976)
31. Thompson, L.: The Mind and Heart of the Negotiator. Pearson Prentice Hall, London (2005)
32. van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: Towards a Reference Framework for Software Product Management. In: 14th IEEE Intl. Requirements Engineering Conference (2006)
33. Yu, E.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: 3rd IEEE Intl. Symposium on Requirements Engineering (1997)