

# Applying Social Interaction Theory to Negotiation Modeling: Design of E-negotiation System

Bo Yu<sup>1</sup> · Rustam Vahidov<sup>2</sup>

Published online: 6 June 2017  
© Springer Science+Business Media New York 2017

**Abstract** Modeling negotiations in a generic form is a persistent challenge in the design of electronic negotiation systems (ENS). ENS need to be sufficiently flexible to accommodate various types of mechanisms, as well as different roles of the participants. The current work applies social interaction theory to modeling negotiation. The purpose is to propose design principles and a model for ENS implementation. The model is presented using a representational framework for information system meta-artifacts in the spirit of design science research. System features and functions derived from the principles and model are exemplified in a system named PROSPER (a Platform Relying On Social Participation for E-market Realization).

**Keywords** Electronic negotiation systems · Social interactions · Design science research

## 1 Introduction

Negotiations allow involved parties to search for mutually acceptable agreements by exchanging offers. The outcomes of business negotiations have a significant effect on the parties' businesses, as well as their reputation and relationships with

business partners. While in many negotiation studies the focus is on the parties directly involved in an exchange, in reality there could be other significant roles that have influence on the process and outcomes of the negotiations. These roles may involve principals, agents, consultants, stakeholders, mediators, and others. In traditional face-to-face negotiations all of the involved parties follow their respective processes related to the central process of offer exchange.

The expansion of online commerce has led to the development of research in electronic markets (Beam et al. 1999). Electronic negotiations imply organic usage of computational and communication media in the negotiation processes. Much work has been done in enhancement of the electronic negotiations with the use of systems incorporating embedded exchange mechanisms, analytical tools, and software agents (Kersten and Lai 2007). The latter can automate some negotiation tasks, essentially introducing new roles in the negotiation process (Braun et al. 2006). Examples of such roles include negotiating agents that automate negotiation process (Jennings and Faratin 2001; Lee and Chang 2008), or advisor agents (Vahidov et al. 2014; Chen et al. 2005). Given the plethora of possible human, as well as software agent roles in negotiations, it would be natural to develop means to represent them explicitly in electronic negotiation systems (ENS). Such a system would allow to flexibly incorporate various roles following their respective processes to reflect a given negotiation situation. This is an issue of modeling at the heart of the design for e-negotiation systems.

Modeling negotiation plays pivotal role in both classical negotiation studies and electronic negotiation systems design. For example, the Invite system, which is a platform for ENS implementation allows modeling negotiations as state-based processes, thereby being able to implement different types of negotiation protocols (Kersten and Lai 2010). Alternative designs of ENS focus on various aspects of negotiation, such as

---

✉ Rustam Vahidov  
rustam.vahidov@concordia.ca

Bo Yu  
bo.yu@dal.ca

<sup>1</sup> Rowe School of Business, Dalhousie University, 6100 University Ave., Halifax, Nova Scotia B3H 4R2, Canada

<sup>2</sup> Department of Supply Chain & Business Technology Management, Concordia University, 1455 de Maisonneuve Blvd. W, Montreal, Quebec H3G 1M8, Canada

negotiation support, decision support, or agent-based negotiations. A desirable design goal would be to combine these various aspects into a generic form that could incorporate various types of negotiations. To this end the current paper proposes role-oriented modeling based on the theory of social interaction as a kernel theory (Markus et al. 2002). In addition, it demonstrates how to use design as a research method for developing information systems artifacts.

Development of such a generic IS artifact form will quite expectedly face a number of challenges. Negotiation as a robust and flexible mechanism has been studied in multiple disciplines. Each of the disciplines studies distinct aspects of negotiations, including negotiation problems, processes, and participants. Successful development of a universal model for negotiation systems that could incorporate these diverse aspects is a major challenge. Nonetheless, the paper aims at obtaining a form of negotiation model capable of abstracting negotiation instances from implementation details and organizing most negotiation issues in a coherent manner.

The current work proposes a way of modeling negotiation as social interaction. The feasibility of this approach is demonstrated by presenting a system named ‘PROSPER’ (a Platform Relying On Social Participation for E-market Realization). The work is developed in the spirit of “design science research” (Hevner et al. 2004; March and Smith 1995; Peffers et al. 2007; Vahidov 2012; Vaishnavi and Kuechler 2015) in that it progresses from kernel theory (theory of social interaction) to artifact development (March and Smith 1995). In order to introduce the developed prototype system, the current work adopts a framework of representing meta-artifacts (Vahidov 2012; 2006).

It needs to be clarified from the very beginning what PROSPER actually is. It is a meta-IT artifact rather than a concrete single system. It represents a family of possible ENS implementations, which adopt design principles outlined in this work that derive from the social interaction theory. It is represented in form of the layers of varying levels of abstraction. Each subsequent layer adds more specificity. Upper layers can be realized in a number of ways in lower layers. Thus, when we present a solution at a more technically specific layer, it must be realized, that this is one, but not the only way to design an ENS based on principles of social interaction. A preliminary sketchy description of PROSPER had been briefly introduced elsewhere (Yu and Vahidov 2014).

The paper proceeds with a literature review of past ENS developments. It then discusses some challenges in modeling negotiations. Subsequently, the social interaction theory is briefly introduced. The application of social interaction theory to negotiation modeling and ENS implementation is then proposed based on the framework introduced in (Vahidov 2006). The paper concludes with discussion of the implications of modeling negotiation as social interaction.

## 2 Electronic negotiation systems

The term Electronic Negotiation Systems refers to a family of systems that facilitate, support, or mediate negotiations involving two or more parties. ENS often adopt a web-based design and are deployed over the Internet (Kersten and Lai 2010, p363). They evolved mainly from two lines of research: 1) Decision Support Systems (DSS) and Negotiation Support Systems (NSS) (Lim and Benbasat 1992a) and 2) groupware (Kersten and Lai 2007).

DSS are systems designed to support complex decision making and problem solving. The key functionalities of a DSS include: (1) sophisticated and robust functions of accessing to and managing data, information, and knowledge, (2) problem identification and modeling, and (3) powerful and friendly user interface (Shim et al. 2002). Some DSS were designed to support negotiation tasks by adding functions such as conflict identification, management and resolution. The early NSS focused on supporting individual negotiators. Subsequently, other features were developed, so that NSS can facilitate negotiations, such as helping negotiators to search for potential agreements and conduct various kinds of analyses, e.g., equilibrium analysis. These efforts led to the emergence of NSS.

Groupware refers to a cluster of systems that are developed to facilitate group activities. Research in this cluster include computer-supported cooperative work (CSCW), group decision support systems (GDSS), group support systems (GSS), and meeting support systems (MSS). A main focus of these systems was to help group members to undertake some joint tasks, with or without analytical support functions. This is achieved through managing, facilitating, and modeling communication. The supported types of communications include synchronous vs. asynchronous, mediated vs. direct, and simple medium vs. multiple media. Many of these systems considered the function of managing and resolving conflicts as an extra feature. Other features that are relevant to negotiation and developed in these systems include the construction of joint problem representation, information sharing, identification of differences in users’ opinions, aggregation of individual choice, and generation of alternatives (Kersten and Lai 2010).

The above two research streams contributed to the development of ENS. ENS are concerned not only with the support of individual negotiators, but also the collective interaction and decision making. The negotiation processes need to be facilitated, managed, and supported. Lim and Benbasat (1992b) noted the minimum requirement by drawing insights to both lines of research and stated that an ENS requires the capabilities of NSS and DSS. It also needs to address issues of how negotiation can be effectively conducted through electronic communication channels between the negotiators.

Beam et al. (1999) introduced a conceptual framework for online negotiation systems. The framework includes

architecture layer (technical feasibility), rules layer (permissible behavior), and strategy layer (ways to win in deal-making). An overview of ENS reported in academic literature is given in Kersten and Lai (2010). The authors mention SimpleNS, a system that features a straightforward design of a negotiation system that allows users to exchange offers and messages. WebNS system offers advanced possibilities in addition to simple communication management (Yuan 2003). It allows parties to interact using multimedia and also use online video chat. It facilitates document management and allows for the presence of facilitator, which can provide advice to the parties.

A system reported in Schoop et al. (2003) called Negoisst allows for the exchange between the parties in B2B contexts. It manages exchange of messages as well as documents in a semi-structured manner. The goal of the message interchange is to finalize contracts. The system has two workspaces: the formal workspace, or “red area” for serious formal negotiations; and the informal space, or “green area” for preliminary exchanges, discussions, etc.

One of the most intensively used and cited in academic research systems is Inspire (Kersten and Noronha 1999). The system design is organized around the process view of negotiations, including phases of pre-negotiation, negotiation, and post-negotiation. In pre-negotiation phase the system facilitated negotiation planning, in particular by allowing users to express their preferences using hybrid conjoint analysis. During negotiations the users were able to send each other structured offers, as well as freestyle messages. The users were also able to see negotiation history graphs expressed in terms of their utilities. In post-negotiation the system acted as a mediator and examined the agreements for Pareto-optimality, with the possible intervention if it found that the agreements could be improved.

Further developments in the generality of ENS design have led to the development of the Invite negotiation platform (Kersten and Lai 2010). The key purpose behind the platform design was to allow for flexible configuration of exchange mechanisms and protocols using the components of the platform. In other words, a variety of ENS types could be implemented on top of the infrastructure provided by Invite. Moreover, it allows for the implementation not only of negotiation systems, but also auctions. For example, for the study in Bellantuono et al. (2014) the authors have configured the platform to compare multi-bilateral negotiation vs. multi-attribute auction performance for the logistics services procurement scenario. Invite follows the principles of MVC (Model, View and Controller) design and allows for the flexible construction of negotiation protocols built separately from the view components.

The above systems focus more on the process of offer exchange, while assuming the mostly two-sided buyer-seller type of negotiations. However, for higher generality various roles of the negotiation stakeholders should be take into

account. For instance, in multi-lateral case many participants, playing similar role should arrive to the same agreement. An example includes multilateral negotiations between five countries regarding the status of the Caspian Sea (Rouhani et al. 2010; Sheikhmohammady et al. 2012). Other examples of roles include use of facilitators (Balachandran et al. 2011; Shyur and Shih 2015) or advisors (Vahidov et al. 2014) in ENS to help parties reach beneficial agreements. While some of the above work allow for such configurations, we argue that putting the roles and their interactions in the very basis of design will allow for more flexible and extendible negotiation system configurations.

There have been attempts to incorporate user personality characteristics in the design of ENS. A system called NegPlace had been designed to incorporate personality types of negotiators (Moura and Costa 2014). The users are given Myers-Briggs Type Indicator questionnaire, and the counterparts are informed about the personality traits of their opponents. The motivation behind this is that the negotiators will choose the right strategies knowing the opponent’s profiles. In other work design based on the use of negotiator’s personal and situational factors, including appearance, language, age and relationship to counterpart for generating optimal sentences to be included in negotiations has been advanced (Suzuki et al. 2017). While there could be many ways to capture personality aspects, in our work we tend to focus on roles as the way to ensure the generality of the proposed framework.

There have also been a bulk of research on automated negotiation approaches in e-business. While comprehensive review of agent-based negotiations are out of the scope of this paper, we will just mention few examples. In Szirbik (2002) a conceptual architecture enabling agent negotiations in the context of virtual enterprise had been devised. A multi-agent infrastructure design enabling flexible addition of new protocols have been proposed in Alfonso et al. (2014). In Huang et al. (2010) an intelligent negotiation agent architecture for B2C negotiation has been proposed, including such components as negotiator, searcher, manager, and others. In Lin et al. (2014) an agent hosting system design called GENIUS has been proposed that allows agents to be configured for negotiations with other agents or humans. Its architecture includes analysis, repository, logging, and simulation control modules. There have also been other agent-based approaches proposed, for instance, for service level agreement negotiations (Ventcinque et al. 2011; Zulkernine and Martin 2011), negotiations in supply chains (Guo et al. 2016; Hernández et al. 2014) and other areas. In 2011 first competition among negotiating agents (ANAC) has been held, where agents negotiated without a priori knowledge of the counterpart’s strategies and preferences (Baarslag et al. 2013).

In our view, in order to ensure generality, the ENS design should allow for the participation of both agents and humans

through the defined roles. In other words, the framework should not be aware whether the participants will be humans or automated agents. A given role, e.g. facilitator could be fulfilled either by human, or agent. This is why such components as agent tactics and strategies are not included in the design proposed here. At the same time, an ENS should be able to prescribe, enforce, or regulate the permissible actions of each acting party through roles in a negotiation instance.

The motivation for developing new design concept for ENS arises from several key considerations.

Firstly, the concept of ENS has roots in multiple disciplines. Negotiation research is an active research field, which emerged within multiple disciplines, including anthropology, social psychology, political sciences, economics, management, and law, among others (Bichler et al. 2003). Each discipline is concerned with specific aspects of negotiations relevant to that particular discipline. As a result the broadly defined field of negotiation research does not have a single established paradigm. There is no overarching framework to connect multiple lines of research in a coherent manner, despite the negotiation being the focal phenomenon of interest.

Secondly, ENS are complex systems. ENS need to seamlessly integrate a large number of features and functions, since the very concept derives from various types of systems. For instance, Kersten and Lai (2010) demonstrated four groups and seventeen types of functions that have been found in the past research on ENS. The four identified groups represent four important foci of an ENS: 1) communication, presentation and interaction, 2) problem and decision maker modeling, 3) offer formulation and concession making, and 4) process organization.

Thirdly, the new means to conduct negotiations have been emerging. The evolution of the research on negotiations and ENS shows a trend where innovative means for negotiations have been developing continuously. One of the most recent thrusts involves the application of software agents used in negotiations. Software agents may take various roles (Braun et al. 2006; Kersten et al. 2008), such as acting as negotiation assistants to support individual negotiators, or actively participating as negotiators (Kersten and Lai 2008).

Based on the above analysis, it can be concluded that the new developments in generic forms of ENS need to be able to flexibly accommodate a variety of roles, processes, and aspects of negotiations conducted electronically.

### 3 Negotiation modeling challenges

From the design perspective, it would be highly desirable to have a universal form that is fundamental enough to convey most aspects of negotiations, since such a form would be helpful guiding effective design for particular ENS. Past

attempts included protocol design, mechanism design, and system architecture, just to name a few (Kersten and Lai 2010; Lomuscio et al. 2003). However, such a universal form has been difficult to develop. The prior brief review showed that negotiation research is dispersed across multiple disciplines. The design and implementation of ENS often feed on multiple sources and adopt an inter-disciplinary approach. One has to keep in mind, though, that theories and findings from different disciplines are not always compatible.

Roughly speaking, the challenges of modeling negotiations arise from four perspectives. First, the composition of negotiations can vary in different ways, particularly in terms of the relations and roles of participants. Negotiations may take multiple forms, such as bilateral, multi-bilateral, or multi-lateral. Some facilitating parties may also be involved, e.g., mediators and arbitrators. Second, negotiators may follow heterogeneous processes according to adopted mechanisms, e.g., auctions or bargaining. Constraints may need to be applied to the participating parties in order to regulate their permissible behaviors. Third, negotiations could be directed towards solving different types of problems, such as value, ethics, dependency and conflicts. Fourth, negotiators may participate in a negotiation in different ways, e.g., with vs. without decision support, synchronous vs. asynchronous, and employing software agents or human participants. The requirements from the first two perspectives often prescribe desirable features of negotiation instances. In contrast, the last two mainly focus on the individual level. Such a division suggests that an effective method for modeling negotiations needs to successfully bridge the individual level issues with those on an instance or higher levels.

The past attempts have revealed that it is difficult to obtain a unified model that could integrate the extremely diversified negotiation aspects and approaches. The task of modeling could be simplified if one could consider negotiation process at the right level of abstraction that allows the generic negotiation model to be extended and diversified. Such a model could potentially integrate multiple aspects of negotiations and allow for flexible adaptation to different scenarios. From a very generic perspective, negotiation is about encounter and interaction. It is a quite general form of social interaction taking place in our daily lives. If one views negotiation as a form of social interaction, social interaction theory could serve as a basis for negotiation modeling. In terms of design science research perspective, it would serve as a kernel theory to guide the design of ENS (March and Smith 1995). The theory of social interaction includes a volume of literature containing knowledge about how individuals are connected with larger social structures and how they interact with each other in general. It was mainly developed from sociology, socio-psychology, and psychology disciplines (Kando 1977). Among its concepts, two notions, i.e., actor and role, are particularly useful in understanding how interactions are organized and how they take place in social contexts.

Social interactions take place in instances, each of them involving some parties participating in them. *Actor* is a notion that refers to the participants. Individual behaviors in social contexts are regulated by social structures, which are often shaped by laws, institutions, cultures, norms, relations, and other factors. *Role* is a key component that connects individuals with a social structure. On one side, role has its connections within a social structure. For example, students and teachers are roles that have their definition in a school context. On the other side, role is a behavior unit that prescribes the expected behaviors from the individuals who are associated with that role, e.g., a teacher needs to teach and students need to learn. Within the behavioral scope prescribed by the roles, individuals are able to exercise their freedom, e.g., making decisions, or applying influences. By using the concepts of actor and role, we can depict a social interaction instance. This is schematically represented in Fig. 1.

The combined use of the notions of role and actor enables modeling negotiation instances. For example, the main function of an ENS can focus on providing the overall facilities to manage the lifecycle of negotiation instances. Each instance can involve a group of actors. Actors can be offered as proxy accounts associated with actual users who participate in a negotiation instance. Roles in the system can be conceptualized and implemented as containers holding a group of permissible behaviors of actors. By attaching different types of roles to actors, a system could prescribe various sets of permissible behaviors to actors. By managing actors as delegated proxies that are connected with users, the system could regulate the permissible actions taken by individual users when they interact with other parties through the system. To summarize, the use of the two notions, i.e., role and actor, allows composing miniature social structures managed by an ENS.

Three examples are presented in Fig. 2 to show the variation of configuring negotiation instances and the flexibility of using the notion of role for modeling the miniature social structures. The first example shows a bilateral negotiation,

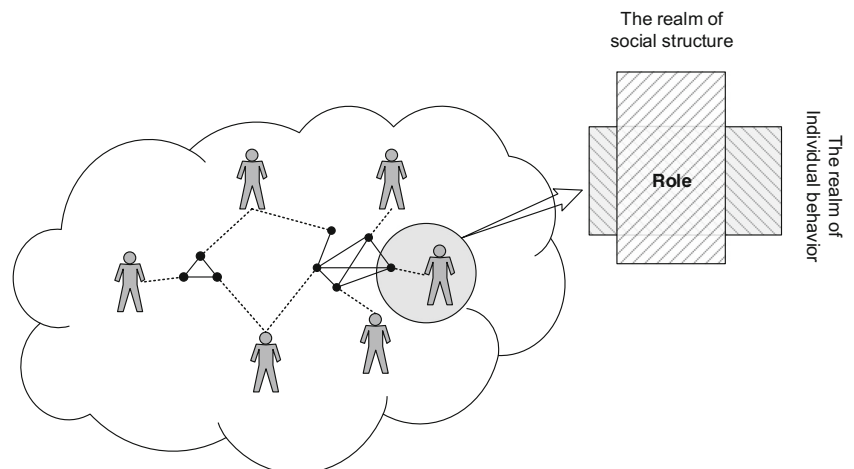
which has two parties including seller and buyer. The parties can exchange offers and decide whether to make an agreement. The second example is a mediated negotiation, in which the buyer and seller do not directly interact with each other. They interact with the mediator and decide whether to accept an offer proposed by the mediator. An agreement will be made when both parties accept the same offer. The third example is an English auction, which can be deemed as a special case of negotiation. The host of the auction (i.e., the seller) can even be absent from the interaction and leave the auction fully automated. It is quite intuitive to describe the configuration of the negotiation instance in terms of roles and the permissible actions of each role. The negotiations can be instantiated by coupling an actor with a role.

#### 4 Applying social interaction theory to ENS design

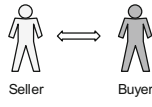
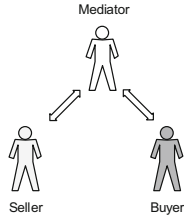
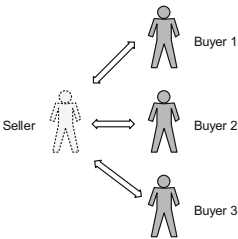
Design science research in information systems has been gaining considerable momentum lately (Beck et al. 2013). Design as a research method is natural in some disciplines such as engineering and computer science. In contrast to natural science which is concerned with describing and understanding natural phenomena, the focus of design oriented disciplines is on creating artifacts (Cross 2001). Using design and invention, these disciplines purposefully explore the possibilities of the non-existent. An important purpose of design-type research is to propose and verify meta-artifacts, which represent a class or form of artifacts that may have multiple applications or instantiations.

In the domain of negotiation research a recent publication by Yang et al. (2012) demonstrates the employment of “structured” design science research method. The word “structured” here means that the authors moved systematically from “kernel theory” to meta-artifact design. More specifically they proposed meta-design and meta-system solution for software negotiation agent based on the dual concern model (Rubin

**Fig. 1** The relation of individuals and social structure



**Fig. 2** Examples of negotiations

Bilateral negotiation		<ul style="list-style-type: none"> <li>-The seller can send offers to the buyer.</li> <li>-The buyer can send offers to the seller.</li> <li>-An agreement will be made if either the seller or the buyer accepts an offer sent by the counterpart.</li> </ul>
Mediated negotiation		<ul style="list-style-type: none"> <li>-The seller can reveal alternatives and preferences to the mediator.</li> <li>-The buyer can reveal alternatives and preferences to the mediator.</li> <li>-The mediator can elicit alternatives and preferences from the buyer and seller.</li> <li>-The mediator proposes offers to the buyer and seller.</li> <li>-An agreement will be made if both the seller and buyer accept an offer proposed by the mediator.</li> </ul>
English auction		<ul style="list-style-type: none"> <li>-The seller specifies the preferences of goods. The preferences need not be revealed to the buyers.</li> <li>-The seller can be absent from the auction.</li> <li>-The buyers submit bids to the auction.</li> <li>-The auction can automatically assess the bids and disclose the current best offer to all buyers.</li> <li>-The buyer who submitted the last best offer will win the auction when the deadline is reached.</li> </ul>

et al. 1994). In a similar fashion, current paper proceeds from the chosen kernel theory (social interaction) to deriving a model for the ENS meta-artifact. Additionally, the work employs a representational framework for comprehensively describing the relevant aspects of the proposed design concept.

A structural framework provided by Vahidov (2012) is adopted here for describing the ENS model. This framework has three main dimensions, i.e., Simonian, Aristotelian, and Feyerabendian dimensions. The Simonian dimension includes four perspectives from which a meta-artifact can be described, i.e., analytical, synthetic, technological, and implementation. The four perspectives align with Simon’s idea that artifacts can be designed by considering both external and internal environment. The Aristotelian dimension includes four categories, i.e., motivation, structure, behavior, and instantiation. The Feyerabendian dimension considers the alternative conceptualizations of a meta-artifact. The perspectives on Simonian dimension and categories on Aristotelian dimension can be used to construct a matrix, which helps decompose design objectives into more focused details (Table 1).

The matrix represented in the table had, in turn been inspired by Zachman’s framework (Zachman 1987) for information systems architecture. However, while the Zachman’s framework had a purpose of modeling concrete single information systems, here the focus is on families of systems. Thus, driven by a kernel theory (social interactions), the analytical layer aims at outlining the salient features and processes at a high level of abstraction. The layer below aims at representing synthetic solution concept. Note that, in general, a number of design conceptualizations may be developed to support the analytical perspective. Technological layer, in turn, proposes more specific, technology-based solutions to support the

synthetic perspective. The bottom layer corresponds to actual implemented system. Naturally, there could also be a number of ways to implement such system that fits the descriptions in the layers above. The framework, thus, can be viewed as a model with varying levels of abstraction, which allow it to be extended and diversified.

### 4.1 Analytical perspective

The focus of the analytical perspective is on representing the target meta-artifact as a set of relevant system characteristics and processes supported. It aims to highlight salient system features, capabilities, and processes that characterize the meta-artifact and distinguish it from other systems in the same category. Essentially, the analytical perspective captures meta-requirements for the type of system.

**Motivation** Negotiation is a robust social mechanism that is frequently used to resolve conflicts and dependencies between two or more parties. A negotiation often involves multiple issues with each issue having multiple (two or more) options. The participating parties have their own preferences over the issues and conflicting interests. They may also need to operate under constraint conditions (e.g., reservation levels for the issues). Issues, options, preferences, and constraint conditions are important elements of a negotiation problem, which can be conceptualized and represented in multiple forms.

Negotiation processes are often regulated by social rules, laws, and institutions. These elements of our social structure often prescribe a procedural discourse that restricts the participants’ choices and actions. Sometimes, they also enforce

**Table 1** A representational framework for ENS

		Categories			
		Motivation	Structure	Behavior	Instantiation
Perspectives	Analytical	Major generic motivations	Salient properties and features of ENS	Processes supported by ENS	Mock-up prototype
	Synthetic	Motivation for the type of ENS solution as related to requirement	Overall logical organization (architecture) of ENS	The dynamic behavior of ENS; working principles	Working prototype that illustrates the concept
	Technological	Objective for the design	Refined structural design for the type of ENS	Refined methods, algorithms, interactions employed by ENS	Generic packages, frameworks, shells
	Implementation	Specific business motivations	Architecture design of implemented systems	Dynamic aspects of implemented system	Implemented functioning systems

certain forms of communication content that is permissible among the parties involved. Good examples can be found in the research of mechanisms design and comparison of negotiations and auctions (Kersten et al. 2006). Negotiations can be shaped in many aspects and configured in various forms. A common feature of different negotiations is that participating parties interact with each other and make their choices, which may lead to potential consensus.

When modeling negotiations, each party participating in a negotiation instance is essentially a placeholder that may accommodate different actual participants. Each participant may use various technologies and decision aides. From a system perspective, these are optional features, which can be extend as they are needed. Overall, it would be desirable that an ENS has the following characteristics:

- support of various compositions of negotiations
- support of multiple types of mechanisms
- support of multiple means of participating in negotiations
- ability for each party to use heterogeneous technologies or decision aids when participating in a negotiation instance.

**Structure** As mentioned earlier, the concept of ENS has largely evolved from areas of DSS and NSS on one hand, and groupware on the other. While the former are concerned with helping participants make decisions, the latter focuses more on enhancing the effectiveness of the communication and coordination within groups. Therefore, two important functions for ENS can be defined. First, an ENS needs to be a communicational tool, channel, or medium that supports structured or unstructured exchanges, including support of various compositions such as bi-lateral, multi-bilateral, or multi-lateral negotiations. Second, it may provide analytical features to support users in various stages of negotiations. It may also allow for multiple modes of participation (e.g. fully automated, semi-automated, and unassisted).

In regards to negotiation instances ENS should be able to:

- manage the life cycle of a group of negotiation instances
- create negotiation instances for different configurations.

**Behavior** Negotiation can be naturally modeled as a process, consisting of several stages. For instance, a model of negotiation process involving three stages, i.e., pre-negotiation, negotiation, and post-negotiation, had been proposed in the past (De Moor and Weigand 2004; Kersten and Noronha 1999). Conventionally, negotiation process is adopted as the unit managed by an ENS. In contrast, the current work argues that negotiation processes can be diverse and they should not serve as a back-bone unit managed by an ENS. Within the same

negotiation instance, participants may follow heterogeneous processes. In our approach negotiation instance is adopted as the key unit that bootstraps all configurations of other sub-components. In order to simplify the system structure, it will be helpful to restrict the main system function to managing the lifecycles of negotiation instances. The aspects such as negotiation protocol, participation, and negotiation problem are configurable with each negotiation instance. Overall the main function of an ENS is to provide functions for configuring, creating and managing instances. Negotiation instances can be diversified by providing different configurations.

**Instantiation** The PROSPER system adopts a popular design pattern, i.e., *configuration-deployment-execution*, to set the structural relation between the system and negotiation instances. The system manages a group of instances. Each negotiation instance bootstraps a group of components or features that are configured with it. The configuration can be represented in form of meta-data. The system also needs to manage several groups of meta-components that can be used when configuring negotiation instances.

The status of each negotiation instance can be changed by sequentially applying a set of operations. The operations include create, deploy, activate, and terminate. An un-terminated instance can apply backward operations, including deactivate, undeploy, and delete. The general life-cycle of a negotiation instance starts with creation and ends with termination. Fig. 3 shows the basic sequence of operations that can be applied to control the lifecycle of negotiation instances in the PROSPER system.

The system should provide functions for a system administrator to manage meta-components. The system manages four types of meta-components. The first type is negotiation case, within which the initial negotiation composition can be modeled, i.e., who are the participants, who can communicate with whom, what are the issues and options, and how to represent the negotiation problems. The second type is meta-instance actor. Meta-instance actors are used to create live instance actors, who will control the status of negotiation instances and the interaction of participating actors. An instance actor will invigilate all information sent and action taken by negotiators, when it needs to determine the progress of the negotiation. Some desirable control on the instance level (e.g., enforcing a protocol, censoring certain type of communication content, and applying some institutional rules) can be achieved by specifying the behavior of the instance actor. The

third type is meta-participant actor. Meta-participant actor is used to create live regular actors. Regular actors are used as a proxy associated with individual negotiation participants in a negotiation instance. The last type is meta-participation, which defines the supported means of participating in a negotiation. These concepts will be further refined in the following sub-sections.

## 4.2 Synthetic perspective

The analytical perspective specifies that the main function of an ENS is managing negotiation instances. By focusing on managing negotiation instances, the system becomes an environment within which various types of negotiations can be configured and handled in a coherent manner. There is a key challenge that needs to be resolved, i.e., how to robustly and flexibly configure negotiation instances. The synthetic perspective is about how to achieve such a design objective. By applying the social interaction theory, the PROSPER system is able to model negotiation instances in a form of miniature social interaction.

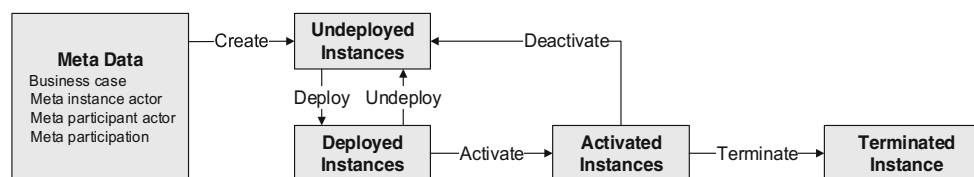
**Motivation** The prior sub-section has stressed that the main function of an ENS should focus on managing the lifecycle of negotiation instances. Each instance can bootstrap a set of configurations. An effective method of modeling negotiations needs to be capable of flexibly configuring negotiation instances on four aspects that have been reviewed in the sub-section as well:

- the compositions of a negotiation instance
- the protocols or mechanisms adopted in negotiation
- the modes of participation of each party
- the types of negotiation problems.

**Structure** The notions of actor and of role have been identified as the key modeling components, as they can be used to construct miniature social interaction instances. The discussion here will focus on how these two notions help resolve the four types of issues noted above.

First of all, a design pattern, actor, is adopted to represent the participating actors in negotiation, in order to make the composition of negotiation instances configurable. Technically, actors are live objects, which can send and respond to messages. Each negotiation instance will include

**Fig. 3** Negotiation instance lifecycle in the PROSPER system





an instance actor and multiple regular actors. Each regular actor represents a proxy account for each actual participant. The instance actor is used to control the process and status of the negotiation instance. Each regular actor maintains a private roster of its permissible contacts, i.e., other actors. A regular actor can send messages only to other regular actors included in its roster. Using actors and rosters, the system can construct a negotiation instance as a miniature social structure, within which actors can be connected in desirable ways. Actual negotiators can interact with each other by respectively controlling their regular actors as proxies within the negotiation instance. All messages sent by regular actors will go through the instance actor. When control of a negotiation instance is required, the instance actor can take actions to change the status of the negotiation instance or send messages to regular actors.

Second, permissible behaviors of actors can be configured in order to support different types of protocols or mechanisms. This design objective is achieved by introducing a new type of objects, i.e., roles. A protocol specifies behavioral conduct for multiple parties participating in a social interaction instance. The compliance of the protocol requires participating individuals to engage in their roles. For instance, some actions or messages have particular meanings and may lead to the transition of the negotiation status. A participant may choose to go for an agreement expressed as a message upon receiving an offer. However, negotiation participants should not simply treat an agreement as a regular message. They have to be able to correctly interpret the meaning of the message, as an agreement may lead to changes of negotiation status (e.g. termination). In some cases, a negotiation mechanism needs to prescribe various sets of permissible actions to different participants. For instance, bidders in an English price-based auction are required to submit bids with increasing price. A price that is lower than the current winning one will not be accepted. The types of messages and permissible actions can only be appropriately defined and understood given the adopted protocols or mechanisms.

The current study implements protocols or mechanisms in a distributive fashion. Instead of modeling a negotiation as a central process, the status and the progress of each negotiation process are jointly determined by the actions taken by the participating actors. Role can be also viewed as a map between the identifiable negotiation status and the permissible messages that an actor can send or actions that it is allowed to take. Certain actions or messages may cause the transition of negotiation status. A protocol or mechanism will require multiple roles, e.g., an auction will have the auctioneer and the bidder. Multiple participants may take the same role. Different types of roles can be registered with the system before they can be used. By using the concept of role, the behaviors of actors become configurable. When coupling with different roles, an actor can respond to the same message differently according to its role.

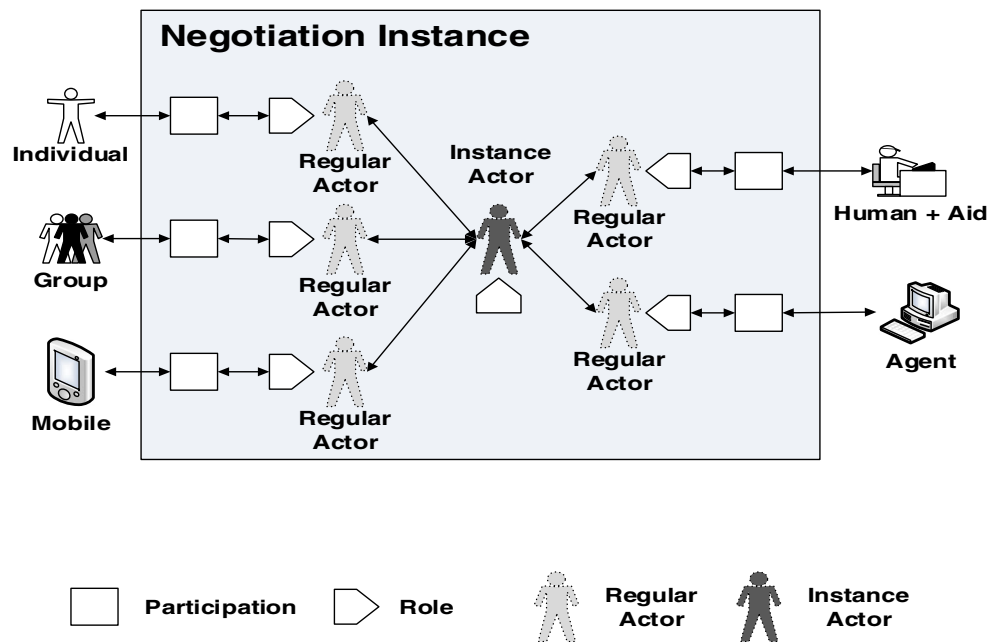
Third, the PROSPER system manages another class of objects that specify the means in which the actual negotiators can control their regular actors. In order to make the negotiation instance operable, it would be desirable to support multiple means. The system provides an abstract class from which different types of participation can extend. The PROSPER system is open to accept registration of new types of participation.

Fourth, the system provides an abstract class of individual problem space. Similar to participations, various types of problem spaces can extend this abstract class.

Fig. 4 shows an example of negotiation instance, which contains one instance actor and five regular actors. Each actor is coupled with a role. The selected roles will prescribe the permissible behaviors to negotiation participants, since participants have to act respectively through their actors. Similarly, the system can also register a group of participations, i.e., the means in which the actors can be accessed. By changing different types of participations, a negotiation instance can support various means of accessing the actors. The adopted participations by individual parties can be heterogeneous. The example shows that modeling negotiation as an instance of social interaction makes the configuration of negotiation very robust and flexible for handling a large variety of issues.

**Behavior** A generic process for managing negotiation instances was introduced in the prior sub-section of analytical perspective. Technically, the system must manage a group of meta-components that can be reused when configuring different negotiation instances in order to support the bootstrap process of a negotiation instance. The managed meta-components should be extendable as they are needed. The number of managed meta-components should not be limited. In order to achieve this objective, the PROSPER system creates objects from meta-data, which is a frequently used method to initialize a live component. The PROSPER system also adopts the factory design pattern when creating live components from meta-data. When a component is required, the system will first search in the registered factory list to find the right factory for the component by its type. Afterward, the system would feed the factory with meta-data and then obtain the live component from it. The factory takes care of the process of creating parts and assembling all the pieces into a live object that can be employed.

Similarly, meta-data can also be used in the bootstrap process of negotiation instances. The system can save the configuration of a negotiation instance into metadata, which contains the metadata of its components. The metadata indicate the appropriate factories from which the right components can be created. When the bootstrap process of a negotiation instance starts, the system will iteratively use the metadata of each component to identify the right factories, feed the factories with component metadata, and then obtain the live



**Fig. 4** Configuring a negotiation instance

components. After components are created, the bootstrap process would assemble them into a negotiation instance.

**Instantiation** A salient feature of the PROSPER system is the way in which it models negotiations as social interaction instances. It adopts two key concepts, i.e., actor and role, in order to achieve its design objective. The details are described here in order to demonstrate how these two concepts are utilized in the system.

Actors are reactive components that respond to messages. Actors are represented by threads with message pools. The thread will sequentially process messages in its message pool. The PROSPER system adopts an open source Groovy library for parallel computing, i.e., GParS (<http://gpars.codehaus.org/>) to implement actors. In some situations, an actor may need to take an initiative. In order to do so, the actor needs to respond to a heartbeat message generated by the system. The system broadcasts a heartbeat message to all live actors registered with the system.

Besides the heartbeat message, actors also respond to two more types of messages, i.e., regular message and control message. Regular message refers to the messages sent between regular actors, such as offers, agreement, and bids. Control messages are used to control the negotiation status and process. Instance actors may send control messages to regular actors to notify the changes of negotiation instance. For example, an instance actor may send an “End-Negotiation” control message to all participating regular actors to notify them that the instance is closed because an agreement has been reached. Both regular and control messages need to be defined in line with protocols or mechanisms.

Role in the PROSPER system is implemented as a class that contains a set of three message handlers. Each handler is used to respectively handle a type of message introduced above. An object of role may also include some extra properties that are required by protocols or mechanisms. Different types of roles can be defined when giving them different message handlers.

The PROSPER system can change the behaviors of an actor when a different role is attached to the actor. From the system perspective, roles can be used to prescribe or enforce some expected behaviors of the participants. The separation between actors and roles also helps to create a loose-coupled relationship between actors and protocols or mechanisms. The PROSPER system has only two generic types of actors, i.e., instance actor and regular actor. By coupling actors with different roles, actors are able to behave differently. In the meantime, mechanism and protocol design can be shifted to design for different types of roles, i.e., specifying the permissible behaviors for actors.

In order to enable robust configurations of negotiation instances, the PROSPER system manages three types of meta-components, i.e., meta-instance actors, meta-participants actors, and meta-participations. These meta-components are templates containing the configuration for instance actors, regular actors, and participations. Meta-components can be independently developed and then registered with the system. The registration of a meta-component requires specifying component name, matching factory, and values for its parameters. The factory will create components by using default parameters. Subsequently, the parameters can be overridden by customized parameters. This means that the same type of component can be diversified by specifying different parameters.

In addition to managing meta-components, the PROSPER system provides a means to model negotiation problem. A

system administrator can model negotiation problem by creating business cases. A business case contains negotiation issues and related options. It also contains the information on the initial composition of negotiation, i.e., who are involved and who can communicate with whom. Each participating party may have different types of problem spaces, e.g., modeled as utility-based or lexicographic.

The system provides a generic process to create individual problem spaces and stores them in form of meta-data. Three sets of meta-data are required: 1) a hierarchy of issues and options, 2) preference information on both level of issue and option, and 3) constraint conditions at space and issue level. The system will use these meta-data to create individual problem spaces for each participant. After instantiation, an individual problem space becomes an indexed space for packages. Each package is a set of options respectively selected for each issue. Currently, PROSPER system supports two types of problem spaces. One is a utility-based problem space, i.e., a rating-based space. The other one is a lexicographic type, i.e., a ranking-based space. Fig 5 shows these two types individual problem spaces in the system. The one on the left is a lexicographic space, which needs ranking of issues and options. The one on the right is a utility-based space, which needs rating of issues and options.

When creating a negotiation instance, the PROSPER system needs to assemble components created from four types of meta-data i.e., business case, meta-instance actor, meta-

participant actor, and meta-participation. After the initialization, the configuration can be further customized. The system also provides a method to export a well-configured instance into a text based meta-data, i.e., template. By using templates, the system can create instances in bulk volumes.

### 4.3 Technological perspective

The analytical and synthetic perspectives deal much with theoretical and conceptual content of a meta-artifact. The feasibility of the meta-artifact needs to be in line with more concrete or available solutions and technologies. These issues need to be addressed from the technological perspective. Classical scientific research aims at demonstrating the validity of a theory by providing evidence. Similarly, design-type research needs to demonstrate the feasibility of meta-artifacts by showing their connections with other artifacts, solutions, and materials that are available, tangible, and understandable. Building such connections also helps to evaluate the meta-artifact in terms of vital criteria, such as optimality and efficiency.

**Motivation** Multiple objectives can be desirable from the technological perspective. Optimality and efficiency are two frequently used criteria when building a system. However, these two criteria are often difficult to determine, as it is not rare in practice that the plan and scope of a software project will change

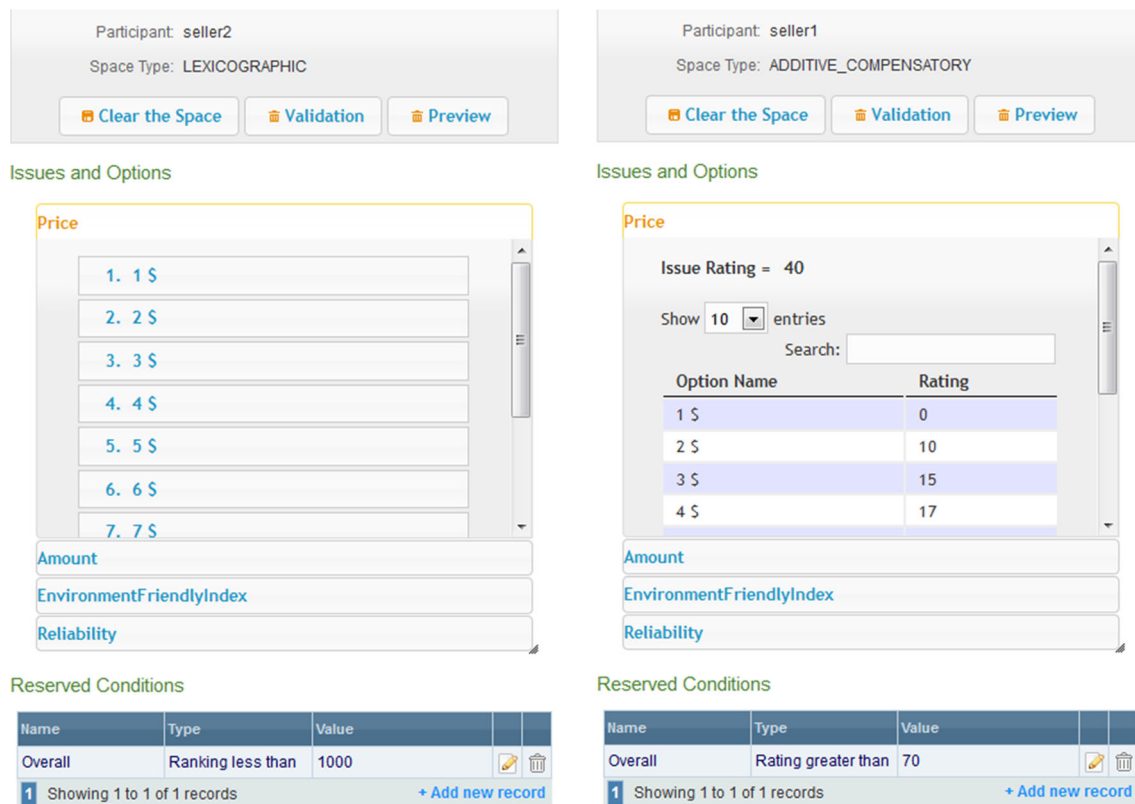


Fig. 5 Two examples of individual problem spaces

during its development process. Agile software development encourages the use of existing components, frameworks, and software. Being agile in system development becomes much desirable when optimality and efficiency is difficult to achieve. The motivation behind the PROSPER system from the technical perspective is to be agile, in that its development tries to use available frameworks, components, and software.

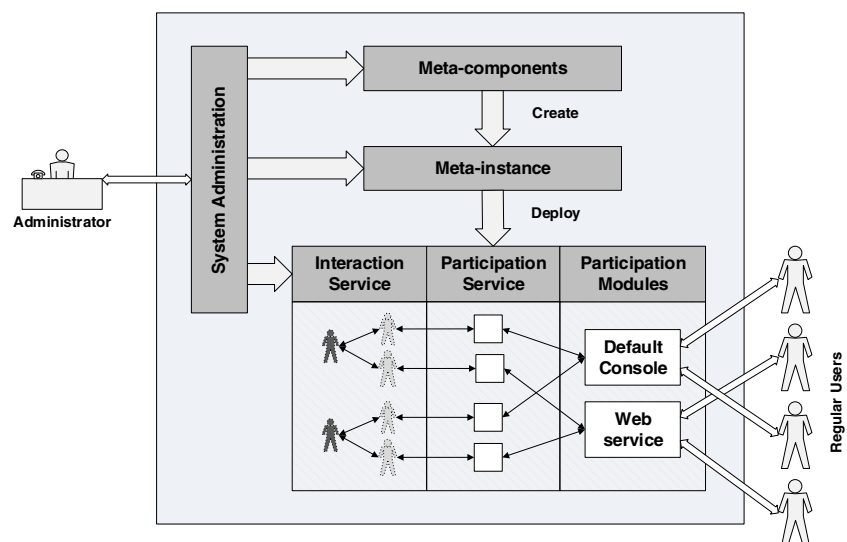
**Structure** PROSPER adopts a group of frameworks, components and plugins in its development. It uses Grails (<http://www.grails.org>) as its main framework. Grails is a well-established agile web application development framework. It supports both Java and Groovy as its main programming languages. Grails is empowered by Spring (<http://spring.io>) and Hibernate (<http://www.hibernate.org>), which are two industrially tested frameworks. Spring is known for its robustness and flexibility for configuring and providing Java objects by using the inverse of control techniques. Hibernate has been broadly recognized as an effective tool for object/relational mapping. GPar, a software package of Groovy parallel system (<http://gpars.codehaus.org/>), is used to implement actors. Actors can respond to in-coming message or heartbeats generated by the system. Activiti (<http://www.activiti.org/>) is used as a business process engine. jQuery, jQuery UI, and several plugins (<http://jquery.com/>) are used on top of Groovy Server Pages shipped with Grails to enhance the usability.

The PROSPER system supports two types of users, i.e., administrators and regular users. Regular users are those who participate in a negotiation instance. Regular users are able to access the system and control their actors through the participation modules, e.g., default web console and web services. The PROSPER system provides a set of system administration functions, through which an administrator can

manage meta-components, create and deploy negotiation instances, and monitor the execution of negotiation instances. The negotiation instances are created from meta-components described in forms of meta-data. Two sets of live objects are registered to the system after they are created. The actor objects are registered with the interaction service. Live participation objects are registered with the participation service. Each of these two services has a map, with which the system can use names to search and access the live objects that are respectively associated to each negotiation instance. The system structure is shown in Fig. 6. In reality there would also be other, ancillary services, such as registration, login, and credentials retrieval. However these are not included in the figure so that the focus stays on the essential structure.

**Behavior** As mentioned earlier, the PROSPER system manages three types of meta-components, including meta-instance actor, regular actor and participation. Meta-components are used to further configure negotiation instances. Each meta-component contains a set of parameters and values in forms of meta-data. The PROSPER system manages two forms of negotiation instances, i.e., meta-data and live forms. Similar to meta-components, negotiation instances need to be configured by specifying their composition in terms of meta-components. The configuration of negotiation instances would persist in meta-data as well. Live negotiation instances are created in a deployment process. The deployment process is a bootstrap procedure in which the PROSPER system iteratively creates live components of a negotiation instance by referring to the appropriate meta-components and related factories. The live objects are assembled and associated with each other according to the specification.

**Fig. 6** The system structure of the PROSPER system



**Fig. 7** An example of template for instance configuration

```

{
  "metaInstanceActorName": "defaultAlternativeOfferInstanceActor",
  "businessCaseName": "TestingBilateralExample",
  "params": "{heartBeatsPerRound:30}",
  "participants": [
    {
      "name": "seller2",
      "metaParticipantActorName": "defaultAlternativeOfferParticipantActor",
      "participantParams": "{}",
      "metaParticipationName": "AO_DEFAULT",
      "participationParams": "{ heartBeatsPerResponse:3,
                             problemSpaceType:ADDITIVE_COMPENSATORY,
                             concessionCoefficient:0.5,
                             baseValue:0.2}",
      "caseParticipantName": "seller2",
      "contacts": "buyer1,buyer2",
      "side": "seller"
    },
    ...
  ]
}

```

There are four important types of live components in each negotiation instance, including actor, role, individual problem space, and participation. Roles are closely coupled with actors, while individual problem spaces are closely coupled with participations. Participations are used to specify the participation modules through which regular users can control their

actors. The participation modules control user interface. When a regular user takes actions through user interface, her actions are applied through the participation modules to her controlled actor. If the controlled actor needs to inform its user, it will send the information to the participation component, which will show the information to the appropriate user.

Instance Information

Instance name: Instance 1  
 Business case: TestingBilateralExample  
 Date Created: 2013-11-01 16:25:28 EDT

When to activate: 2013/11/02 16:25:28  
(yyyy/mm/dd hh:mm:ss)

When to interact: 2013/11/02 16:25:28  
(yyyy/mm/dd hh:mm:ss)

When to end interaction: 2013/11/02 16:25:28  
(yyyy/mm/dd hh:mm:ss)

When to terminate: 2013/11/02 16:25:28  
(yyyy/mm/dd hh:mm:ss)

Activated: false  
 Terminated: false

Delete
Save as Template
Deploy

Participants
The Parameters of Instance Actor

The Information of Participants

	Participant Name	Meta Participant Actor	Case Participant	Meta Participation	Date Created		
	seller1	default_AO_PActor	TestingBilateralExample.seller1	AO_DEFAULT	2013-11-01		
	buyer1	default_AO_PActor	TestingBilateralExample.buyer1	AO_DEFAULT	2013-11-01		
	buyer2	default_AO_PActor	TestingBilateralExample.buyer2	AO_DEFAULT	2013-11-01		
	seller2	default_AO_PActor	TestingBilateralExample.seller2	AO_DEFAULT	2013-11-01		

1 Showing 1 to 4 of 4 records
+ Add new record

**Fig. 8** The instance management functions of the PROSPER system

The PROSPER system uses both database and text to store meta-data. Meta-data saved in database are easy to indexed and searched. Live objects serialized in text-based meta-data are saved in the form of JSON (JavaScript Object Notation, <http://json.org>). A Java software package is adopted to serialize objects into to JSON text. An example of the JSON text is presented in Fig. 7. The text is a structured snippet of instance template. When instantiating an instance, the PROSPER system will iteratively look up the right factories by using the meta-component names contained in the snippet. Afterward, the factories will create components and assemble parts into an instance. The PROSPER system manages a class of factories, each maps to a registered meta-component. The deserialization process will use these names to look up factories and then produce the live objects. In addition to meta-component names, both `businessCaseName` and `problemSpaceType` are important fields that determine which and what type of individual problem space will be created for individual participants.

**Instantiation** The PROSPER system has been implemented based on the social interaction theory. Fig 8 shows an example negotiation instance and demonstrates some management functions of the PROSPER system.

## 5 Conclusion

Modeling negotiations is important as it may influence both how we see negotiation and how we can manage it. The field of ENS is a research arena, where various lines of research meet. These different lines of research are not always compatible and sometimes are in conflict with each other. It would be desirable to have a generic form of modeling negotiations. By employing two key notions, i.e., actor and role, from social interaction studies, the current work proposes a generic form in which negotiation can be modeled and managed in ENS. In summary, our approach models negotiation as a miniature social interaction instance. We found that the notions of actor and role are helpful when abstracting negotiation into instances, which are able to bootstrap a variety of configurations of diversified negotiations. Our approach is robust and flexible. By doing so, the functions of a ENSs can be simplified to focusing on the management of negotiation instances and related reusable meta-components. Our approach is replicable in other ENSs design.

The current study adopts design as a research method and uses a framework for representing information systems meta-artifacts to present the ENS called PROSPER. The PROSPER system demonstrates the feasibility of the proposed design principles and model of ENS implementation. Modeling negotiation as social interaction has at least two benefits. First, it

reduces the complexity of the design of ENS, as different types of negotiation instances and their lifecycles can be managed in a coherent manner. The system manages a set of negotiation instances which can be configured, deployed and executed. Each instance is composed as a miniature social structure by adopting the notions of role and actor, within which actual users can interact with each other. Second, it offers a flexible, robust, and open approach to handle a large variety of negotiation issues. Negotiation instances managed by PROSPER can bootstrap a broad variety of configurations. They can also be serialized to or deserialized from meta-data. This design allows the configurations of negotiations to be easily extended.

## References

- Alfonso, B., Botti, V., Garrido, A., & Giret, A. (2014). A MAS-based infrastructure for negotiation and its application to a water-right market. [journal article]. *Information Systems Frontiers*, 16(2), 183–199. doi:10.1007/s10796-013-9443-8.
- Baarslag, T., Fujita, K., Gerding, E. H., Hindriks, K., Ito, T., Jennings, N. R., et al. (2013). Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence*, 198, 73–103. doi:10.1016/j.artint.2012.09.004.
- Balachandran, B. M., Gobbin, R., & Sharma, D. (2011). Development of a Multi-issue Negotiation System for E-Commerce. In J. Watada, G. Phillips-Wren, L. C. Jain, & R. J. Howlett (Eds.), *Intelligent Decision Technologies: Proceedings of the 3rd International Conference on Intelligent Decision Technologies (IDT' 2011)* (pp. 429–438). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Beam, C., Segev, A., Bichler, M., & Krishnan, R. (1999). On negotiations and deal making in electronic markets. [journal article]. *Information Systems Frontiers*, 1(3), 241–258. doi:10.1023/a:1010050311741.
- Beck, R., Weber, S., & Gregory, R. W. (2013). Theory-generating design science research. [journal article]. *Information Systems Frontiers*, 15(4), 637–651. doi:10.1007/s10796-012-9342-4.
- Bellantuono, N., Ettore, D., Kersten, G. E., & Pontrandolfo, P. (2014). Multi-attribute auction and negotiation for e-procurement of logistics. [journal article]. *Group Decision and Negotiation*, 23(3), 421–441. doi:10.1007/s10726-013-9353-7.
- Bichler, M., Kersten, G., & Strecker, S. (2003). Towards a structured design of electronic negotiations. *Group Decision and Negotiation*, 12(4), 311–335.
- Braun, P., Brzostowski, J., Kersten, G., Kim, J., Kowalczyk, R., Strecker, S., et al. (2006). E-negotiation Systems and Software Agents: Methods, Models, and Applications. In J. Gupta, G. Forgyonnie, & M. Mora (Eds.), *Intelligent Decision-Making Support Systems: Foundations, Applications and Challenges* (pp. 271–300). Berlin: Springer.
- Chen, E., Vahidov, R., & Kersten, G. E. (2005). Agent-supported negotiations in the e-marketplace. *International Journal of Electronic Business*, 3(1), 28–49. doi:10.1504/IJEB.2005.006387.
- Cross, N. (2001). Designerly ways of knowing: Design discipline versus design science. *Design issues*, 17(3), 49–55.
- De Moor, A., & Weigand, H. (2004). Business negotiation support: Theory and practice. *International Negotiation*, 9(1), 31–57.
- Guo, W., Li, W., Zhong, Y., Lodewijks, G., & Shen, W. (2016) Agent-based negotiation framework for agricultural supply chain supported

- by third party logistics. In *Computer Supported Cooperative Work in Design (CSCWD), 2016 I.E. 20th International Conference on, 2016* (pp. 584–589): IEEE.
- Hernández, J. E., Mula, J., Poler, R., & Lyons, A. C. (2014). Collaborative planning in multi-tier supply chains supported by a negotiation-based mechanism and multi-agent system. *Group Decision and Negotiation*, 23(2), 235–269.
- Hevner, A., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.
- Huang, C.-C., Liang, W.-Y., Lai, Y.-H., & Lin, Y.-C. (2010). The agent-based negotiation process for B2C e-commerce. *Expert Systems with Applications*, 37(1), 348–359. doi:10.1016/j.eswa.2009.05.065.
- Jennings, N. R., & Faratin, P. (2001). Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10, 199–215.
- Kando, T. M. (1977). *Social interaction*. Saint Louis: The C. V. Mosby Company.
- Kersten, G., & Lai, H. (2007). Negotiation Support and E-negotiation Systems: An Overview. *Group Decision and Negotiation*, 16(6), 553–586. doi:10.1007/s10726-007-9095-5.
- Kersten, G., & Lai, H. (2010). Electronic Negotiations: Foundations, Systems, and Processes. In D. M. Kilgour & C. Eden (Eds.), *Handbook of Group Decision and Negotiation Vol. 4* (pp. 361–392). Netherlands: Advances in Group Decision and Negotiation: Springer.
- Kersten, G. E., Chen, E., Neumann, D., Vahidov, R., & Weinhardt, C. (2006). *On Comparison of Mechanisms of Economic and Social Exchanges: The Times Model*. Paper presented at the Dagstuhl Seminar: Negotiation and Market Engineering, Dagstuhl Nov. 12–17, 2006.
- Kersten, G. E., Kowalczyk, R., Lai, H., Neumann, D., & Chhetri, M. B. (2008a). Shaman: Software and Human Agents in Multiattribute Auctions and Negotiations. In H. Gimpel, N. R. Jennings, A. Ockenfels, & C. Weinhardt (Eds.), *Negotiation, Auctions, and Market Engineering Vol. 2* (pp. 116–149). Springer Berlin Heidelberg: Lecture Notes in Business Information Processing.
- Kersten, G. E., & Lai, H. (2008b). Negotiation Support and E-negotiation Systems. In F. Burstein, & C. Holsapple (Eds.), *Handbook on Decision Support Systems I* (pp. 469–508, International Handbooks on Information Systems): Springer Berlin Heidelberg.
- Kersten, G. E., & Noronha, S. J. (1999). WWW-based negotiation support: design, implementation, and use. *Decision Support Systems*, 25(2), 135–154. doi:10.1016/S0167-9236(99)00012-3.
- Lee, C.-F., & Chang, P.-L. (2008). Evaluations of Tactics for Automated Negotiations. [journal article]. *Group Decision and Negotiation*, 17(6), 515–539. doi:10.1007/s10726-008-9109-y.
- Lim, L.-H., & Benbasat, I. (1992a). A Theoretical Perspective of Negotiation Support Systems. *Journal of Management Information Systems*, 9(3), 27–44. doi:10.1080/07421222.1992.11517966.
- Lim, L.-H., & Benbasat, I. (1992b). A theoretical perspective of negotiation support systems. *Journal of Management Information Systems*, 9(3), 27.
- Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K., & Jonker, C. M. (2014). Genius: an integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 30(1), 48–70. doi:10.1111/j.1467-8640.2012.00463.x.
- Lomuscio, A. R., Wooldridge, M., & Jennings, N. R. (2003). A classification scheme for negotiation in electronic commerce. *Group Decision and Negotiation*, 12(1), 31–56.
- March, S. T., & Smith, G. F. (1995). Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15, 251–266.
- Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A Design Theory for Systems that Support Emergent Knowledge Processes. *MIS Quarterly*, 26(3), 179–212.
- Moura, J. A., & Costa, A. P. C. S. (2014). Incorporating personal style into a Negotiation Support System. In *Group Decision and Negotiation 2014 GDN 2014: Proceedings of the Joint International Conference of the INFORMS GDN Section and the EURO Working Group on DSS, 2014* (pp. 95): EWG-DSS.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77.
- Rouhani, O., Madani, K., & Gholizadeh, S. (2010). Caspian Sea Negotiation Support System. In *World Environmental and Water Resources Congress 2010* (pp. 2694–2702): American Society of Civil Engineers.
- Rubin, J. Z., Pruitt, D. G., & Kim, S. H. (1994). *Social conflict: Escalation, stalemate, and settlement*: McGraw-Hill Book Company.
- Schoop, M., Jertila, A., & List, T. (2003). Negoisst: a negotiation support system for electronic business-to-business negotiations in e-commerce. *Data & Knowledge Engineering*, 47(3), 371–401. doi:10.1016/S0169-023X(03)00065-X.
- Sheikhmohammady, M., Hipel, K. W., & Kilgour, D. M. (2012). Formal analysis of multilateral negotiations over the legal status of the caspian sea. [journal article]. *Group Decision and Negotiation*, 21(3), 305–329. doi:10.1007/s10726-010-9195-5.
- Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., & Carlsson, C. (2002). Past, present, and future of decision support technology. *Decision Support Systems*, 33(2), 111–126.
- Shyr, H.-J., & Shih, H.-S. (2015). Designing a multi-issues negotiation support system based on prospect theory. *Information Sciences*, 322, 161–173. doi:10.1016/j.ins.2015.06.014.
- Suzuki, K., Tamada, H., Doizaki, R., Hirahara, Y., & Sakamoto, M. (2017). Women's Negotiation Support System—As Affected by Personal Appearance Versus Use of Language. In W. Chung & C. S. Shin (Eds.), *Advances in Affective and Pleasurable Design : Proceedings of the AHFE 2016 International Conference on Affective and Pleasurable Design, July 27–31, 2016, Walt Disney World®, Florida, USA* (pp. 221–230). Cham: Springer International Publishing.
- Szirik, N. (2002). A negotiation enabling agent based infrastructure: composition and behavior. [journal article]. *Information Systems Frontiers*, 4(1), 85–99. doi:10.1023/a:1015346808770.
- Vahidov, R. (2006). Design Researcher's IS Artifact: a Representational Framework. In *1st International Conference on Design Science Research in Information Systems and Technology, DESRIST'2006 Claremont, CA, February 24–25, 2006*.
- Vahidov, R. (2012). *Design-Type Research in Information Systems: Findings and Practices*: IGI Global.
- Vahidov, R., Chen, E., & Kersten, G. (2014). Experimental assessment of agent-supported electronic negotiations. *International Journal of Human Computer Interaction*, 29(11), 764–774. doi:10.1080/10447318.2013.777825.
- Vaishnavi, V. K., & Kuechler, W. (2015). *Design science research methods and patterns: innovating information and communication technology*: Crc Press.
- Venticinque, S., Aversa, R., Di Martino, B., Rak, M., & Petcu, D. (2011). A Cloud Agency for SLA Negotiation and Management. In M. R. Guarracino, F. Vivien, J. L. Träff, M. Cannatoro, M. Danelutto, A. Hast, et al. (Eds.), *Euro-Par 2010 Parallel Processing Workshops: HeteroPar, HPCC, HiBB, CoreGrid, UCHPC, HPCF, PROPER, CCPI, VHPC, Ischia, Italy, August 31–September 3, 2010, Revised Selected Papers* (pp. 587–594). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Yang, Y., Singhal, S., & Xu, Y. (2012). Alternate strategies for a win-win seeking agent in agent-human negotiations. *Journal of Management Information Systems*, 29(3), 223–256.
- Yu, B., & Vahidov, R. (2014). Modeling negotiation as social interaction for ENS design: The PROSPER approach. In *Joint International Conference on Group Decision and Negotiation* (pp. 168–175): Springer.

- Yuan, Y. (2003). Online negotiation in electronic commerce. *International Journal of Management Theory and Practices*, 4(1), 39–48.
- Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal*, 26(3), 276–292.
- Zulkemine, F. H., & Martin, P. (2011). An adaptive and intelligent SLA negotiation system for web services. *IEEE Transactions on Services Computing*, 4(1), 31–43. doi:10.1109/TSC.2010.44.

**Bo Yu** is an Assistant Professor in Management Information Systems of Rowe School of Business at Dalhousie University, Halifax, Nova Scotia,

Canada. He received his Ph.D. in Management Information Systems from Concordia University, Montreal, Canada. He has published in Journal of Organizational Computing and Electronic Commerce and a number of conferences, including HICSS, DSI, and AMCIS.

**Rustam Vahidov** is a Professor of Business Technology Management at John Molson School of Business, Concordia University (Montreal, Canada) and holds Royal Bank of Canada Professorship in Inter-organizational Governance of IT. He received his PhD from Georgia State University. Dr. Vahidov has published papers in a number of journals, including Journal of MIS, Decision Support Systems, Information and Management and others.