Developing Web-Based Dynamic Negotiation towards Collective Innovation: the Open Contract Mechanism

Alessandro Avenali, Giorgio Matteucci and Fabio Nonino

Abstract The scope of this chapter is to propose the design and the implementation of an innovative web-based tool, the Open Contract Mechanism (OCM). OCM allows client companies and suppliers to dynamically and simultaneously bargain the clauses and the characteristics of distinct innovation contracts in general open innovation and collaborative crowdsourcing environments. The OCM can be implemented inside open innovation web-based platforms for evaluating offers and determining the winning one, and therefore guaranteeing valuable binding contract clauses.

1 Introduction

Internet and the web-based technologies have facilitated the collaborative approaches for innovation enabling the emergence of large networks of potential contributors (i.e. the so-called "crowd"). A high number of new innovative enterprises promoted innovation and allowed the collaboration of individuals and companies and the so-called crowdsourcing through open innovation web-based platforms [2]. The most recent innovation model is then the collaborative crowdsourcing (also referred to as the collective innovation), which integrates the crowdsourcing approach with the open innovation paradigm [3]. The collective innovation process involves three typologies of actors: (1) the individuals or companies forming the crowd (workers, solvers and know-how providers), (2) the

A. Avenali (✉) · G. Matteucci · F. Nonino
Department of Computer, Control and Management Engineering "Antonio Ruberti",
Sapienza University of Rome, Rome, Italy
e-mail: alessandro.avenali@uniroma1.it

G. Matteucci
e-mail: giorgio.matteucci@uniroma1.it

F. Nonino
e-mail: fabio.nonino@uniroma1.it

P. Spagnoletti (ed.), *Organizational Change and Information Systems*, Lecture Notes in Information Systems and Organisation 2, DOI: 10.1007/978-3-642-37228-5_21,

individuals or client companies (seekers) and (3) the open innovation intermediary platforms (OIPs) which act as brokers between the crowd and the seeker (e.g. Innocentive, NineSigma, CrowdSpring, etc.). The problem to reach an agreement on the quality and quantity of work to be performed among outsourcers and suppliers and on the value of the knowledge provided by solvers is typical to all markets and is often resolved through negotiation. But negotiation is not easily practicable in all situations. In open distributed markets, the work performed by any single worker is often vanishingly small. For this reason buyers' take-it-or-leave-it prices tend to predominate, and negotiations are rare [4]. Moreover open innovation and crowdsourcing requires web-based platform able to protect the intellectual property, but also to support the negotiation phase and the definition of terms of agreements in the contracts in order to motivate companies and individuals in participation both as seekers and solver and prevent disputes. A solution can came from patents, but patenting is costly and, in the collective innovation, the mechanisms for collecting and enforcing patent rights and licensing royalty income, are complex and impose high transaction costs. The user agreements and contracts should be used as a deterrent to opportunistic behaviour, while patents can be employed too late, i.e. in the final stages of the innovation process [1].

The objective of our research has been to design and to implement an innovative tool able to guarantee the solvers/providers that their ideas are protected in order to create a "win–win" scenario. The chapter presents the design (Sect. 2) and the implementation (Sect. 3) of the Open Contract Mechanism (OCM), an innovative web-based tool which allows innovation seekers and solvers to dynamically and simultaneously bargain the clauses and the characteristics of distinct innovation contracts in general open innovation and crowdsourcing collaborative environments. Finally, we draws conclusions (Sect. 4).

2 The Design of the Open Contract Mechanism

The design of OCM starts from a general open innovation and crowdsourcing scenario where an organization (from now on referred to as seeker) does not possess specific skills and technical knowledge (hereinafter referred to as know-how items) which are crucial for an innovation or for a part of NPD project (e.g. work, solutions to problems, patents, licenses, prototypes). Using the OCM tool the seeker can acquire these missing know-how items by posting a *challenge* inside an open innovation platform, where selected organizations and individuals (hereinafter referred to as solvers) are requested to submit bids to provide their work or know-how according to some conditions specified by the seeker. In particular, the seeker can specify the quantity demanded, the required quality, the terms of delivery, of warranty and of payment and clauses to safeguard possible intellectual property rights. Moreover, in order to select reliable solvers both in terms of owned know-how and economic performances, the seeker can impose agreements which strictly require solvers with suitable characteristics (special certifications, constraints in term of key financial ratios, etc.). To avoid solvers' post-contractual opportunistic behaviour, the agreements can be enforced by an appropriate set of penalties for not fulfilling the related undertakings. There exist different ways of exchanging skills and technical knowledge between the seeker and the solvers: sell of existing patents, licensing, development and patenting of the design of a crucial input described in terms of abstract and general ideas and/or a fixed amount of money to solve a specific technical problem. Furthermore the OCM allows the seeker and the solvers to bargain the features of an innovation contract (contract amendments) by applying an *open contract model*.

2.1 Design of the Seeker and Solvers Client Side

The seeker clicks on OCM tool which displays two sections, respectively labelled "My contracts" and "My bids". In the first section, the tool permits the seeker to create a new challenge to acquire know-how items. The challenge will allow the seeker to acquire all the specified know-how items from exactly one solver after a competition among all solvers invited by the seeker (possibly suggested by a suitable procedure based on the semantic engine of the OIP which selects potential solvers on the basis of the keywords representing the know-how items desired).

In the second section the seeker can:

- Monitor the current round of a challenge which the seeker himself has created. Since it is still running, the challenge is referred to as an *open challenge*.
- Look through a challenge created by him and that is ended. Since there are no more rounds, the challenge is referred to as a *closed challenge*.
- On the solver side the OCM displays a section where the solver:
- Attends the current round of an open challenge which a seeker has created and which the solver is taking part in.
- Attends the amending phase of an open challenge which the seeker has created and which the solver is taking part in.
- Attends the *boarding phase* (namely, the period when the invited solvers are accepting or rejecting the invitation to take part in the challenge) of a challenge and which the solver has been invited to take part in.
- Can investigate whether there are or not new challenges (still in the boarding phase) which the solver could be interested in. The tool allows the solver to contact the seeker in order to be invited before the boarding phase ends.

2.2 Process Description

The OCM functioning can be divided in three phases: (a) the creation, (b) the boarding and (c) the negotiation. During the *creation phase* the tool allows the seeker to define:

- The name of the challenge and a description of the required know-how items.
- A bunch of keywords related to every know-how item.
- The maximum number of rounds of the challenge. In every round the seeker and the solvers will bargain contract features. The challenge ends after a given maximum number of rounds. However, depending on the competition level among solvers, the challenge can end in a fewer number of rounds.
- The maximum running time of any round of the challenge. Each round automatically ends after a given maximum time.
- The timetable of the challenge, that is, the beginning and the end of the boarding phase, of the amending phase, and the beginning of the 1st round.

At the end of the creation phase the OCM asks the seeker to design the contract scheme (see Sect. 2.3). Once the contract scheme is definitively carried out, OCM allows the seeker to invite the solvers to take part in the challenge. When the list of all solvers to be invited is ready, the seeker can write a letter of invitation. Then, the tool automatically contacts all the solvers and communicates them the seeker's letter of invitation. Then the *boarding phase* starts and the OCM tool waits for the solvers answering to the seeker's invitation. The tool displays to any invited solver the list of the names of the open challenges which he has been invited to take part in and which are in the boarding phase. Once solver has clicked on the challenge he is interested in, the tool allows him to accept or to reject the invitation. At the end of the boarding phase, the OCM contacts the seeker providing the list of the solvers who have accepted the invitation. The tools don't provide this list to any solver so as to reduce collusion among solvers. Subsequently the *amending subphase* begins the tools waits for the solvers submitting their amendments. When the amending phase is over the seeker decides which amendments can be accepted and which ones have to be rejected. Afterwards the tool communicates the definitive formulation of the contract scheme to any solver and no information about which solvers have proposed the submitted amendments (so as to reduce collusion among solvers). Then, when it is the time for the beginning of the *negotiation phase*, the tool contacts the solvers and informs them for the beginning of the first round. For any current round the tool waits for the solvers submitting their contract bids. If any solver has definitely carried out his contract bids (or quit the challenge), or the maximum running time of the round is over, the tool stops the current round. Then the tool compute a *score* for every submitted contract bid (see Sect. 2.4). At the end of the negotiation phase the tool communicates:

- To the seeker, the details of the winning contract bid and of the solver.
- To the solver who has submitted the winning contract bid, the details of the winning contract bid.
- To all other solvers, the details of the winning contract bid and no information regarding which solver has submitted the winning contract bid (so as to reduce collusion among solvers).

2.3 Contract Design

The tool allows the seeker to design the *contract schemes* to acquire all the desired know-how items and provide their details. In particular:

The seeker uploads files which can be applied (inserting links in the contract scheme) to design the clauses (i.e. ".pdf", ".avi") and enclosed in the contract. Any contract clause, which makes up the contract scheme, is a string.

Every contract clause can also be parametric. The seeker can insert parameters, that is, unspecified quantitative data of the clauses which every invited solver will specify through values selected among those the seeker has defined as feasible. In order to highlight which parameters are more crucial than others, the seeker assigns a weight to each introduced parameter. Therefore, if r is the overall number of parameters which characterize the contract scheme and $w_i \geq 0$ are the weight associated by the seeker to parameter i for $i = 1, \ldots, r$; the weights decided by the seeker are subject to the constraint $\sum_{i=1}^{r} w_i = 1$. For any parameter, the seeker has to specify if the parameter is ascendant (descendent), in the sense that the higher (the lower) the parameter, the better the resulting contract scheme for the seeker. The seeker has also to specify which values can be taken into account for it (the feasible values for the parameter); to do this, the seeker associates to any parameter i a lower bound lb_i, an upper bound ub_i and a minimum step ms_i. Consequently, if parameter i is ascendant the tool will show in the drop-down menu of the parameter all and only these values (ordered from the lowest value to the highest):

$$\begin{gathered} lb_i + 0 \cdot ms_i \\ lb_i + 1 \cdot ms_i \\ \vdots \\ lb_i + f_i \cdot ms_i \end{gathered}$$

where $f_i = \frac{ub_i - lb_i}{ms_i}$.

If the parameter is descendant the tool will show in a drop-down menu of the parameter all these numbers (ordered from the highest value to the lowest):

$$\begin{gathered} ub_i - 0 \cdot ms_i \\ ub_i - 1 \cdot ms_i \\ \vdots \\ ub_i - f_i \cdot ms_i \end{gathered}$$

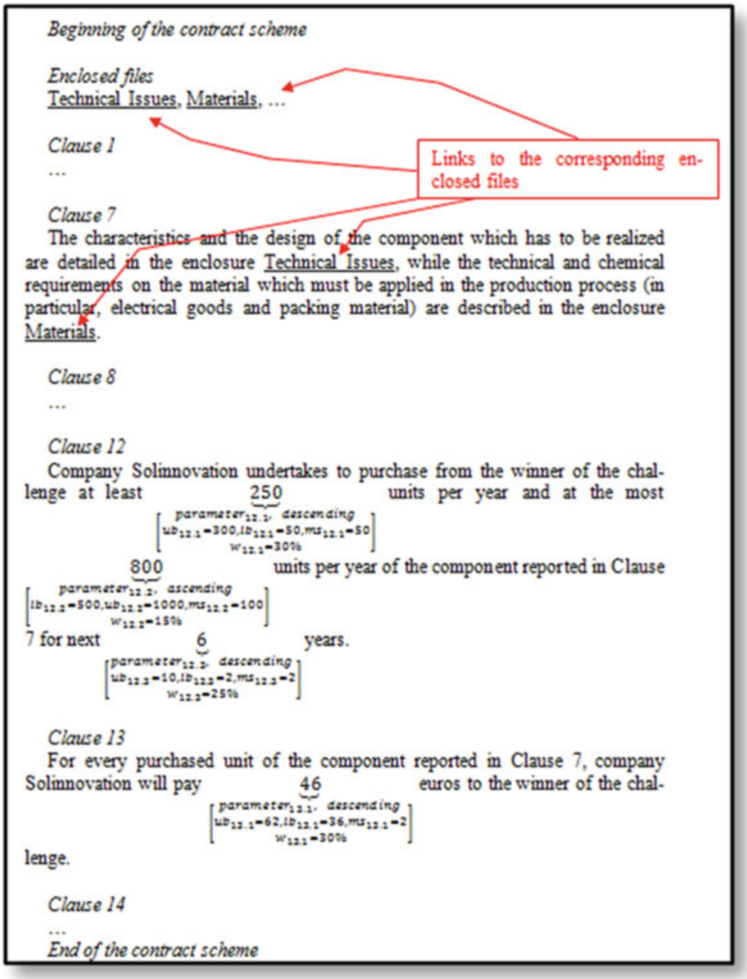
where $f_i = \frac{ub_i - lb_i}{ms_i}$.

For example, let us assume that the know-how item is the production of a specific component which has been designed by capital company Solinnovation. The component is critical, but the expected demand is highly uncertain. The seeker Solinnovation proposes a challenge to negotiate the risk sharing between the seeker and the solver who will produce the component. Solinnovation defines a contract scheme (Fig. 1), where Clause 7 contains links to some enclosed files, while clauses 12 and 13 are parametric (in the example all and only the parameters of the contract scheme are those proposed by the seeker in these two clauses).

2.4 Bid Score Calculation

The score of a bid is computed as follows:

- Let l be the number of solvers who have submitted bids in the current round. Let d_k be the number of contract bids submitted by solver k in current round for



Beginning of the contract scheme

Enclosed files
Technical Issues, Materials, …

Clause 1
…

Clause 7
The characteristics and the design of the component which has to be realized are detailed in the enclosure Technical Issues, while the technical and chemical requirements on the material which must be applied in the production process (in particular, electrical goods and packing material) are described in the enclosure Materials.

Clause 8
…

Clause 12
Company Solinnovation undertakes to purchase from the winner of the challenge at least $\underbrace{250}_{\left[\begin{smallmatrix} parameter_{12.1},\ descending \\ ub_{12.1}=300, lb_{12.1}=50, ms_{12.1}=50 \\ w_{12.1}=30\% \end{smallmatrix}\right]}$ units per year and at the most $\underbrace{800}_{\left[\begin{smallmatrix} parameter_{12.2},\ ascending \\ lb_{12.2}=500, ub_{12.2}=1000, ms_{12.2}=100 \\ w_{12.2}=15\% \end{smallmatrix}\right]}$ units per year of the component reported in Clause 7 for next $\underbrace{6}_{\left[\begin{smallmatrix} parameter_{12.3},\ descending \\ ub_{12.3}=10, lb_{12.3}=2, ms_{12.3}=2 \\ w_{12.3}=25\% \end{smallmatrix}\right]}$ years.

Clause 13
For every purchased unit of the component reported in Clause 7, company Solinnovation will pay $\underbrace{46}_{\left[\begin{smallmatrix} parameter_{13.1},\ descending \\ ub_{13.1}=62, lb_{13.1}=36, ms_{13.1}=2 \\ w_{13.1}=30\% \end{smallmatrix}\right]}$ euros to the winner of the challenge.

Clause 14
…
End of the contract scheme

Fig. 1 Example of a contract scheme

$k = 1, \ldots, l$, and let $d = d_1 + \cdots + d_l$ be the overall number of contract bids submitted by all solvers in current round. Let $k(j)$ denote the solver who has submitted bid j. Every contract bid j for $j = 1, \ldots, d$ consists of an ordered list of values $\left(v_{1,j}, \ldots, v_{r,j}\right)$, where any $v_{i,j}$ for $i = 1, \ldots, r$ is the value offered, by solver $k(j)$, in contract bid j for parameter i. First, the OCM computes for $i = 1, \ldots, r$ both average m_i and standard deviations σ_i of the values offered for parameter i by all solvers. For $i = 1, \ldots, r$, the OCM computes:

$$m_i = \frac{\sum_{j=1,\ldots d} v_{i,j}}{d} \qquad \sigma_i = \sqrt{\frac{\sum_{j=1,\ldots d}\left(v_{i,j} - m_i\right)^2}{d}}$$

- The OCM defines the new weights of the parameters ($\overline{\mathrm{w}}_{\mathrm{i}} = \mathrm{w}_{\mathrm{i}}$ if parameter i is ascendant and $\overline{\mathrm{w}}_{\mathrm{i}} = -\mathrm{w}_{\mathrm{i}}$ if parameter i is descendant).
- Then, the tool sets the score of contract bid j equal to:

$$sc_j = \sum_{i=1}^{r} \overline{\mathrm{w}}_i \frac{v_{i,j} - m_i}{\sigma_i}$$

Among all bids which have been stored so far (i.e. from the beginning of the challenge until this round), the tools select as winning the contract bid with the highest score (ties are broken randomly).

3 The Open Contract Mechanism (OCM) Tool

The OCM has been implemented as a standalone web-based tool which can be integrated inside an OIP. The overall architecture is depicted in Fig. 2.

OCM has been implemented using a web service approach where a number of HTTP verbs have been developed implementing the basic operations that are needed in order to create, manipulate and negotiate contracts. The communication between the client and the server through HTTP is performed with the exchange of appropriate JSON documents. Everything in the OCM data model is built around the concept of the contract. Inside the contract, a number of metadata are defined, e.g. information regarding the seeker, a list of solvers that the seeker decides to call to participate and a number of parameters that constitute the negotiable values of the contract. These entities are related to the contract creation and boarding phases. Finally, for the modelling of the negotiation phase, the concept of round is introduced. The final element is the scheduler. The various phases and the rounds of the negotiation are defined by a starting and ending time provided by the seeker. At the beginning of each round specific actions have to be performed, e.g. the change of the contract status from creation to boarding and from boarding to negotiation. These kinds of actions are performed by the scheduler in pre-defined time intervals based on the values provided by the seeker during the contract

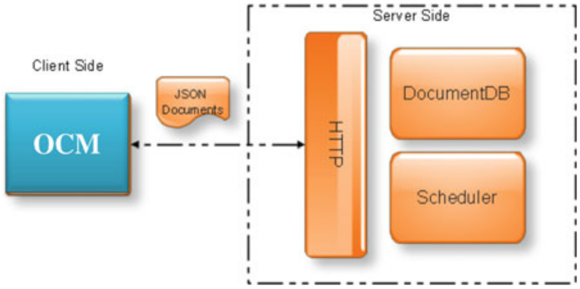


Fig. 2 The open contract mechanism overall architecture

creation. A relational database is used to store the jobs. The current version of the OCM consists of the three distinct phases described below.

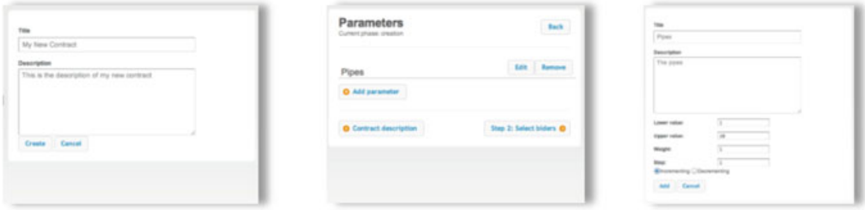
3.1 Creation Phase

During this phase the seeker defines all the defined parameters of a contract and invites the potential solvers who would participate in the negotiation of the specific contract. The contract creation phase consists of four distinct sub-phases:

1. Creation of a new contract with a title and a brief description (Fig. 3–left).
2. Definition of an arbitrary number of parameters that will be negotiated for the contract together with their attributes (Fig. 3–center).
3. Selection of the potential solvers who will participate in the negotiation.
4. Definition of the number of rounds the negotiation phase will have, when the boarding phase will start and end and the duration of each negotiation round.

3.2 Boarding Phase

During the boarding phase, when the user submits the newly created contract, two scheduled jobs are created: one for the initialization of the phase and one for its conclusion. Both the date and time for these two states are provided by the seeker


Title: My New Contract
Description: This is the description of my new contract

Parameters
Current phase: creation
Pipes

Title: Pipes
Description: The pipes
Lower value: 1
Upper value: 10
Weight: 1
Step: 1
● Incrementing ○ Decrementing

Fig. 3 Contract creation, definition of parameters and of the number of rounds (screenshots)

during the creation phase. When the scheduled job is fired, a number of e-mails are sent to the candidate solvers. The status of the solver has three potential states, (a) pending, (b) accepted and (c) rejected. Initially the system assigns the state pending. By either accepting or rejecting the participation, the user alters the status either as accepted or rejected. At the end of boarding phase, another scheduled job is fired and the invited solvers are scanned. If their status is still in pending state, it is changed into rejected, while if no one accepted the invitation, then the current state of the contract is changed into closed and the whole procedure ends. If one or more potential solvers have accepted the invitation, the status of contract changes into negotiation.

3.3 Negotiation Phase

During the negotiation phase the actual process of negotiating a contract is performed. Initially the seeker is able to create a new negotiation round (Fig. 4–center). When this happens a new scheduled job is created and scheduled using the round duration provided by the seeker at the creation phase. During the creation of a new negotiation round, notifications are sent to the solvers so they can submit new offers for the current negotiation round. The solvers can create new offers using the OCM interface. After deciding about the values of each parameter, they are able to submit their offers (Fig. 4–right). At the end of the round the scheduled job is fired and a number of actions occur; initially the system parses all the offers together with any past winning offers and calculates the various scores based on the scoring algorithm (see 2.4).

On the basis of these scores the winner is retrieved and then the system sends e-mails and notification to both the seeker and the solvers. The mail to the seeker contains all the details regarding the last negotiation round together with the information about the current winner. The solvers receive an e-mail which contains only the information of the winning contract but without the credentials of the winning solver. At this point the seeker can initiates again a new negotiation round unless all the solvers have left the negotiation phase: in this case the winner is found and the contract status changes to closed.

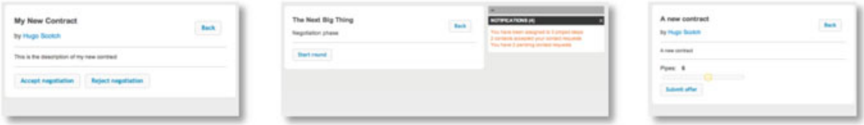


Fig. 4 Notification for participating in the negotiation of a contract, initiation of a new negotiation round and creation and submitting an offer (screenshots)

4 Conclusions

The open contract mechanism can be implemented inside open innovation web-based platforms, or directly used by the companies. OCM can become a powerful tool for evaluating offers and determining the winning one, and therefore guaranteeing valuable binding contract clauses. Moreover it can enhance confidence in decision making and flexibility in dealing with negotiation issues developing the company's ability to confront with the complexity of the collective innovation in open innovation web-based platforms. On the basis of several simulations run up to now using the software prototype developed, we found improvements achieved in negotiation performance in terms of reduction of purchasing costs and increase of quality and quantity of know-how items acquired.

Acknowledgments The research leading to these results has received funding from the European Community's Seventh Framework Programme managed by REA-Research Executive Agency http://ec.europa.eu/research/rea [(FP 7—SME 2008—2)] under grant agreement n [243593], project COLLECTIVE—Emerging communities for collective innovation: ICT Operational tool and supporting methodologies for SME Associations. For details, see www.collectiveproject.eu.

References

1. Avenali, A., Matteucci, G., & Nonino, F. (2010). *Contract clause dynamic bargaining and procurement design*. In: Proceedings of the International Public Procurement Conference 26–28 Aug 2010, Seoul, Korea.
2. Battistella, C., & Nonino, F. (2012). "What drives collective innovation?—Exploring the system of drivers for motivations in open innovation, web-based platforms". Information Research, Vol. 17, No. 1, paper 513, Univ. Sheffield, Dept. Information Studies, Western Bank, Sheffield, UK.
3. Chesbrough, H. (2003). *Open innovation: The new imperative for creating and profiting from technology*. Boston: Harvard Business School Press.
4. Horton, J.J. & Zeckhauser, R.J. (2010). *Algorithmic wage negotiations: Applications to paid crowdsourcing*. In: Proceedings of CrowdConf, 4 Oct 2010, San Francisco, CA.