International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:12, No:1, 2018

Definition and Core Components of the Role-Partner Allocation Problem in Collaborative Networks

J. Andrade-Garda, A. Anguera, J. Ares-Casal, M. Hidalgo-Lorenzo, J.-A. Lara, D. Lizcano, S. Suárez-Garaboa

***Abstract*—**In the current constantly changing economic context, collaborative networks allow partners to undertake projects that would not be possible if attempted by them individually. These projects usually involve the performance of a group of tasks (named roles) that have to be distributed among the partners. Thus, an allocation/matching problem arises that will be referred to as Role-Partner Allocation problem. In real life this situation is addressed by negotiation between partners in order to reach *ad hoc* agreements. Besides taking a long time and being hard work, both historical evidence and economic analysis show that such approach is not recommended. Instead, the allocation process should be automated by means of a centralized matching scheme. However, as a preliminary step to start the search for such a matching mechanism (or even the development of a new one), the problem and its core components must be specified. To this end, this paper establishes (i) the definition of the problem and its constraints, (ii) the key features of the involved elements (i.e., roles and partners); and (iii) how to create preference lists both for roles and partners. Only this way it will be possible to conduct subsequent methodological research on the solution method.

***Keywords*—**Collaborative network, matching, partner, preference list, role.

I. Introduction

NOWADAYS it is commonly accepted that, in a fast-moving globalized world, the alliance of organizations is an essential means for attending to customers. Organizations can reach many of their global goals by cooperation and, thanks to the advancements in information and communications technologies (ICTs); such cooperation takes the form of a collaborative network (CN). As defined in [1], a CN is constituted by autonomous organizations and/or individuals that are usually geographically distributed and heterogeneous. These entities, despite having different operating environments and purposes, collaborate to achieve common goals and address new business opportunities together. To this end, their interactions are supported by ICTs.

As indicated in [1], different types of CNs have emerged over time, such as virtual organizations, virtual enterprises, professional virtual communities, and collaborative laboratories (see e.g., [2], [3]). This demonstrates the increasing importance of this type of alliances.

In CNs, partners (individuals and/or enterprises) seek to undertake a business opportunity in a given field together; that is to say, they seek to start a new project together. At its core, a project is a set of tasks (here referred as roles) that have to be performed by the partners. The problem of distributing tasks among partners is an allocation/matching problem that will be called Role-Partner Allocation (RPA) problem. The goal is to find a matching M between roles and partners satisfying certain conditions.

Commonly, finding such a matching M in CNs is addressed in real life by negotiation between the partners interested in participating in the project. This takes a long time and is hard work. Partners have to present their arguments (i.e., knowledge, capabilities, personal preferences, availability, etc.) and discuss the situation in order to reach an agreement regarding the distribution of tasks between them. Even if certain information about the partners is stored, the process is complex because it implies human beings each one with its own personality, requirements, tastes, etc. In fact, as indicated in [4], a centralized matching scheme should be used to find such a matching M. This is because both historical evidence and economic analysis show that allowing participants involved in this type of problems to reach agreements by themselves without the help of a centralized matching scheme is not the best way to create successful allocations.

There are different centralized matching schemes to solve different types of matching problems. The RPA problem is an example of the two-sided matching problem [5]. Following the definition in [6], in this type of problems the input entities (in this case, the roles and the partners) are partitioned into two disjoint sets (in this case, the set of roles and the set of partners), and the aim is to pair elements of one set with elements of the other set subject to various criteria such as capacity constraints and preference lists.

The National Resident Matching Program (NRMP) [7] in the USA is perhaps the most representative example of a centralized matching scheme to solve a problem of this type. As indicated in [6], it handles the allocation of residents to hospitals, based on the preferences of residents over available hospitals, and the preferences of hospitals over residents. For that purpose, the NRMP employs an algorithm that finds a stable matching M of residents to hospitals that is resident-optimal (also known as resident-oriented), in that each resident obtains the best hospital that she/he could obtain in any stable matching. In a hospital-optimal (hospital-oriented) approach,

Javier Andrade-Garda, Juan Ares-Casal, and Marta Hidalgo-Lorenzo are with the Universidade da Coruña, Grupo ISLA, Dep. Computación, Fac. Informática, Elviña, 15071 A Coruña, Spain (e-mail: javier.andrade@udc.es, juan.ares@udc.es, marta.hidalgo@udc.es).

Aurea Anguera is with the Technical University of Madrid, Carretera de Valencia, Km 7. 28031 Madrid, Spain (e-mail: aanguera@uei.upm.es).

Juan Alfonso Lara and David Lizcano are with the Distance University of Madrid, Vía de Servicio, 15, 28400 Collado-Villalba Madrid, Spain (e-mail: juanalfonso.lara@udima.es, davil.lizcano@udima.es).

Sonia Suárez-Garaboa is with the Universidade da Coruña, Grupo ISLA, Dep. Computación, Fac. Informática, Elviña, 15071 A Coruña, Spain (corresponding author, phone: +34981167000; fax: +34981167160; e-mail: sonia.suarez@udc.es).

each hospital obtains the best set of residents that it could obtain in any stable matching. A matching *M* is unstable if there are *a, b* elements that are not partners in *M* and prefer each other to her/his/its partner in the matching. Such a pair *(a, b)* is said to be a blocking pair for the matching. Thus, a matching is stable if it contains no blocking pair [8].

The NRMP manages a variant of the classical Hospitals/ Residents problem (HR) [9], [10]. There are many other generalizations of HR such as, for example, the School Choice problem [11], the Course Allocation problem [12], and the Student–Project Allocation problem [4]. In addition, examples of centralized matching schemes are used in situations as different as those of online matrimony in India [13] and the renowned case with patient-donor pairs [14]: a patient in need of a kidney and a donor (family, friend) who is willing to donate one.

Irrespective of the matching scheme used to solve an allocation problem, the problem itself and its core components have to be defined. Thus, for example, in the NRMP case, it was firstly necessary to define residents and hospital posts, together with their key features, in order to properly understand and manage these elements. It has also been necessary to define the problem constraints (i.e., one-to-one vs. one-to-many allocation, residents and hospital posts capacity constraints, etc.) and the mechanisms to compose the preference lists on both sides (i.e., how residents rank hospital posts and on what basis hospital posts prefer one resident to another). The same has to be done for the RPA problem in CNs. This is precisely the aim of the presented paper. Once the RPA problem and its core components are well defined, it will be possible to choose an existing matching scheme to solve it. It might even be necessary to develop a new one if none of the existing mechanisms fit the needs of the problem.

The rest of the paper is organized as follows. Section II introduces the RPA problem, Section III describes roles and partners and their key properties, and Section IV shows how to configure preference lists on both sides. Finally, Section V presents the most relevant conclusions and the future work.

II. The RPA Problem

Let $I \equiv \{i_1, i_2, \ldots, i_n\}$ with $|I| \geq 2$ denote the finite set of partners. A generic element in *I* is denoted by *i*. Each partner has a finite number of roles that she/he/it is willing to perform. Specifically, let $l(i)$ be the number of roles that *i* requires at least in order to join the project (the participation would otherwise not be profitable for the partner) and $u(i)$ the maximum number of roles that *i* is willing to play in the project. Let $R \equiv \{r_1, r_2, \ldots, r_m\}$ denote the finite set of roles. A generic element in *R* is denoted by *r*. It is also assumed throughout that the total number of roles is no less than the minimum total number of tasks required by the partners (i.e. $m \geq \sum_{i \in I} l(i)$). For each role there is a partially ordered preference list of partners such that element at position *x* is strictly preferable to element at position *x+1* or they are indifferent. Preferences of role *r* are denoted by $\alpha(r)$. Note $\alpha(r) \subseteq [i_1, i_2, \ldots, i_n]$. That is to say, not all partners have to be appropriate to play all roles. Likewise, for each partner there is a partially ordered preference list of roles such that element at position *x* is strictly preferable to element at position *x+1* or they are indifferent. Preferences of partner *i* are denoted by $\beta(i)$. Again, note that $\beta(i) \subseteq [r_1, r_2, \ldots, r_m]$. That is to say, not all partners are able, or want, to play all roles.

On the basis of the above, the following restrictions can be defined for an RPA problem instance and a matching *M*:

1. For each role *r*, $0 < |\alpha(r)| \leq |I|$. Every role has to admit at least one partner (otherwise the distribution of tasks is not possible), but it has not to admit all partners as acceptable players. That is to say, incomplete preference lists in the role side are allowed but not empty preference lists.
2. For each partner *i*, $0 < |\beta(i)| \leq |R|$. To be taken into consideration, a partner has to want to/be able to play at least one role but she/he/it does not have to want to/be able to play all the roles. Thus, incomplete (but not empty) preference lists on the partner side are allowed.
3. Each role *r* strictly prefers $\alpha(r)[x]$ to $\alpha(r)[x+1]$ or is indifferent between them. The roles' preference lists are partially ordered because they include ties. That is to say, a role does not have a strict preference between two or more partners because they are just as good for the job.
4. Each partner *i* strictly prefers $\beta(i)[x]$ to $\beta(i)[x+1]$ or is indifferent between them. The partners' preference lists also include ties. That is to say, a partner could equally prefer to play two or more roles.
5. For each partner *i* and role *r*, if $r \in \beta(i)$ then $i \in \alpha(r)$. A partner cannot want to play a role if she/he/it is an unacceptable player for the role (e.g., she/he/it does not have the required knowledge). Note that the reverse is not true. That is to say, $i \in \alpha(r)$ does not mean that $r \in \beta(i)$. In other words, *i* could be an acceptable (even highly desirable) partner for playing *r* but *i* does not want to play *r* at that moment (e.g. due to problems of time availability, budget problems, different current personal preferences).
6. If $M(i)$ is the set of roles assigned to *i* after the allocation process, then $0 < l(i) \leq M(i) \leq u(i) \leq |R| - \sum_{j \in I} l(j), i \neq j$. That is to say, each partner has to take the responsibility of a minimum number of $l(i)$ roles (with $l(i)$ at least 1; that is, a piece of project) and a maximum of roles such that each other partner *j* has the possibility to play a minimum of $l(j)$ roles.
7. If $M(r)$ is the set of partners playing role *r* after the allocation process, then $M(r) = 1$. Each role represents a piece of project intended to be played by one partner. If no partner in *r*'s preference list wants/is able to play *r* it will be necessary to redesign the role, probably splitting it into different new roles, together with the consequent re-

allocation process.

In summary, the RPA problem (i) is a two-sided allocation problem, (ii) is a many-to-one allocation problem where one partner can play many roles, (iii) accepts upper and lower quotas in the partners' side, and (iv) admits incomplete preference list with ties in both sides; all of which fulfilling restrictions 1 to 7.

III. Partners and Roles

The concepts of partner and role have been used for describing the RPA problem, but a detailed specification of these two concepts is required in order to adequately understand and manage them.

In the RPA problem context, it is crucial to properly represent and store the relevant information about the involved partners and roles in order to be sure that the decision making process works with up to date, reliable, and sufficient information. Moreover, the selected information representation scheme has to be general enough to represent any type of information since CNs are needed in any business domain and to face multidisciplinary projects. Knowing this, the general and domain independent representation scheme presented in [15], [16] is proposed and adapted in this paper for defining partners and roles in the RPA problem. On this ground, partners are said to have two key features: goals (i.e., capabilities) and associated abilities. In brief, goals are the specific objectives of a given partner, meaning the specific activities that it is capable of performing (e.g. project management is a goal of partner A; that is, partner A is capable of managing projects). Associated abilities, on the other hand, are partner's additional goal-related capabilities (e.g. partner A can complement her/his/its goal of project management with the associated abilities of quality oriented attitude and good social skills). An associated ability does not exist separately; it is always linked to one or more goals for improving and/or completing them. So, goals are related to specific partners' capabilities, whereas associated abilities are related to other interesting aptitudes/knowledge that might be associated with such capabilities. On the other hand, roles are also said to have two key features: the obligations and the skills_weights pair. The obligations refer to the activities represented by the role (e.g., a role called Database Development is designed with the obligations of SQL, Oracle and Store Procedure). Thus, if a partner wants to play the role she/he/it has to comply with its obligations; that is to say, the partner must have the required knowledge. The skills refer to the desired but not necessary conditions for a player to obtain the role (e.g. decision making, time management, critical thinking). Like what happens with the partners' abilities, each skill for a given role will be linked with one or more specific obligations, as it improves and/or completes them. Each skill has a given weight (e.g. from 1 to 5) that represents its importance for the associated obligation. This way, there are primary skills—very relevant to the obligation—with high weight values, and secondary skills—not very important to the obligation—with low weight values (e.g., decision making has a weight of 2 for the obligation SQL of the role Database Development, and critical thinking has a weight of 4). This structure of partners and roles is shown in Fig. 1.

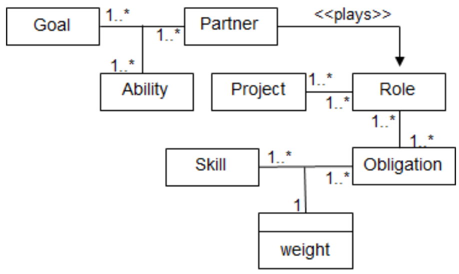

Fig. 1 Information structure (adapted from [16])

IV. Preference Lists

It is also required to specify how to create preference lists, both for roles and partners. In this regard, the information structure presented in Section III is used to establish these preference lists as described in the following.

A. Roles' Preference Lists

A role is not a human being (nor a set of human beings, like an enterprise), so it has no personal preferences or individual tastes. Thus, the preference list $\alpha(r)$ of each role *r* must be created on the basis of objective and stable criteria. In this case, in the ideal situation, each role would be played by the partner with the goals and abilities that better fulfil the role requirements. For that, the general proposal in [16] for building holonic virtual enterprises is adapted here for the RPA problem in CNs. Thus, the criteria used to configure the roles' preference lists are: (i) the matching between the obligations of the role and the partners' goals, and (ii) the matching between the skills_weights pairs of the role and the partners' abilities. On this basis, the following can be established:

- Only partners fulfilling all the obligations required by the role can be feasible players for the role (i.e. members of the preference list of that role). Think for example of a role that requires expert knowledge of game theory as an obligation. Defining this item as an obligation means that it is absolutely necessary to achieve the objectives of the role. Otherwise, it would be defined as a skill with an associated weight. Formally: $i \in \alpha(r) \Leftrightarrow obligations(r) \subseteq goals(i)$.
- Once the feasible partners are calculated for each role, they must be ranked in the preference list of the role. It should be remembered that partially ordered preference lists with ties are allowed. What makes the difference between the feasible partners is their contribution in terms of abilities. Formally:
  - o Let *i* be a feasible player for role *r*, *g* a goal of *i* such that $g = b$ with $b \in obligations(r)$, and let w_g be the sum of the weights associated with the skills in $B = skills\_weights(b) \cap abilities(g)$ (i.e., w_g represents the added value that *i* provides to *r* due to *g*), then

$w = \sum_{g \in goals(i)} w_g$ is used to position partner i in the preference list of role r. Thus, the partners with the highest value of w will be at the top of the list.

B. Partners' Preference Lists

Partners are human beings (or sets of human beings, like enterprises), so that besides their goals and abilities they may also have personal preferences and individual tastes. Moreover, even having the capacities (i.e., the goals) required by a role and being personally interested in playing the role, at a moment in time a partner could have other limitations that inhibit her/him/it to play the role (e.g., lack of time, budget and financial problems, infrastructure problems, staffing problems). As a consequence, the creation of the preference lists on the basis of objective and stable criteria is not always possible or desirable. Thus, in this paper three ways are proposed to build each partner's preference list:

- The Automatic Approach: This approach is possible when the partner has no personal preferences. The partner only has to specify those roles to which she/he/it has special limitations (e.g., lack of time, budget problems). The system is the one in charge of identifying and sorting the feasible roles in the preference list of the partner. The specification of those limitations can be done before or after the search process. In the first case, the unfeasible roles are not taken into consideration from the beginning. Otherwise, all the theoretically feasible roles would enter the process and they would make it onto the preference list. After that, the partner has to specify which roles she/he/it is not able to play at that moment, these roles have to be deleted from the list, and their adjacent items in the list have to be moved in order to occupy the empty positions. In any case, the following can be established:
  - Role r is feasible for partner i if and only if $obligations(r) \subseteq goals(i)$.
  - Role r_1 is better for partner i than role r_2 if and only if i is a better partner for role r_1 than for role r_2.
- The Personal Approach: In this case, the choice and prioritization of feasible roles rests entirely with the partner. She/he/it makes it on the basis of her/his/its personal preferences and individual tastes (e.g., motivation, relationship between the role and other tasks in hand, economic interests).
- The Mixed Approach: The preference list is partially built by the partner, and partially by the system. The partner places certain roles at the top and/or tail of the list. Thus, a partner could specify the roles that she/he/it prefers the most and/or those that she/he/it likes least to play. Note that these roles do not necessarily coincide with the roles which theoretically seem best/less suited for the partner. In other words, the partner can organize the two ends of the list using the personal approach and delegate to the system the prioritization of the remaining roles through the automatic approach.

V. Conclusion

When a project is undertaken in the context of a CN, the distribution of roles (i.e., project tasks) among partners must be automated by means of a centralized matching scheme since the success of the project depends largely on the suitability of this allocation. If it does not (i.e., if the involved partners construct an allocation by approaching one another directly) partners have to present their arguments (i.e., knowledge, capabilities, personal preferences, availability, etc.) and discuss the situation. This can lead to mistakes, disputes, or misunderstandings that may obstruct the negotiation. But even when this is not so and the negotiation process is executed normally, the vast quantity of information involved and the fact that partners are human beings with their own strengths and weaknesses may result in poor agreements.

In this paper the problem of allocating roles to partners (i.e., the RPA problem) is defined as a many-to-one two-sided matching problem with upper and lower quotas in the partners' side, and incomplete preference lists and ties on both sides (i.e., roles and partners). In addition, the key properties of partners and roles are specified, allowing the definition of the process to create preference lists on both sides. Therefore, this paper provides the necessary elements for future work. In this case, future work implies the search for a matching mechanism to solve the RPA problem fulfilling its requirements and constraints. If none of the existing mechanisms is valid for the RPA problem, a new one will be defined. In any case, the allocation process will be simpler and safer than direct negotiation.

References

[1] L. M. Camarinha-Matos, and H. Afsarmanesh. "Collaborative networks: a new scientific discipline," *Journal of Intelligent Manufacturing*, vol. 16, no. 4-6, pp. 439–452, Oct. 2005.

[2] T. Vuletic, R. I. Whitfield, W. Wang, A. Duffy, S. Gatchell, H. Prins, and M. Leer-Andersen. "Improving the creation and management of collaborative networks within the European maritime sector: An operational collaboration model for the European maritime sector," *Journal of Industrial Information Integration*, In Press, Available online 8 June 2017.

[3] D. Wang, K. K. Yan, J. Rozowsky, E. Pan, and M. Gerstein, "Temporal Dynamics of Collaborative Networks in Large Scientific Consortia," *Trends in Genetics*, vol. 32, no. 5, pp. 251-253, May 2016.

[4] D. J. Abraham, R. W. Irving, and D. F. Manlove, "The Student-Project Allocation Problem," in *Algorithms and Computation*, *Conf. Rec. 2003 International Symposium on Algorithms and Computation*, pp. 474-484.

[5] D. Gale. "The Two-Sided Matching Problem: Origin, Development and Current Issues," *International Game Theory Review*, vol. 3, no. 2-3, pp. 237–252, 2001.

[6] D. J. Abraham, R. W. Irving, and D. F. Manlove, "Two algorithms for the Student-Project Allocation problem," *Journal of Discrete Algorithms,* vol. 5, no. 1, pp. 73–90, March 2007.

[7] National Resident Matching Program (NRMP). Available: http://www.nrmp.org/. Accessed: 08/09/2017.

[8] R. W. Irving. "Stable marriage and indifference," *Discrete Applied Mathematics,* vol. 48, pp. 261–272, 1994.

[9] D. Gusfield, and R.W. Irving. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, 1989.

[10] D. F. Manlove. *Algorithmics of Matching Under Preferences*. World Scientific, 2013.

[11] A. Abdulkadiroğlu, and T. Sönmez. "School Choice: A Mechanism Design Approach," *American Economic Review,* vol. 93, no, 2, pp. 729-747, June 2003.

[12] F. Diebold, H. Aziz, M. Bichler, F. Matthes, and A. Schneider. "Course

Allocation via Stable Matching," *Business & Information Systems Engineering,* vol. 6, no. 2, pp. 97-110, April 2014.

[13] M. Morge, and E. Piette. "Practical Application of Matchmaking Problem: Trainee Allocation for Teachers," *Lecture Notes in Artificial Intelligence*, vol. 8473, 2014, pp.195-206 [*Proc. 2014 International Conference on Practical Applications of Agents and Multi-Agent Systems*, Salamanca, Spain].

[14] A. E. Roth, T. Sönmez, and M.U. Ünver, "Kidney Exchange," *Quarterly Journal of Economics*, vol. 119, pp. 457- 488, 2004.

[15] J. Andrade, J. Ares, R. Garcia, M. A. Martinez, J. Pazos, S. Rodriguez, and S. Suarez. "A new approach for the validation of conceptual holonic constructions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 2, pp. 318-331, March 2013.

[16] J. Andrade, J. Ares, R. García, M. A. Martínez, J. Pazos, and S. Suárez. "A Game Theory Based Approach for Building Holonic Virtual Enterprises," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 2, pp. 291–302, Feb. 2015.