An Agent-based Negotiation Approach for Knowledge Exchange in Cloud Service Platforms

Sameh Abdalla[1], Bingquan Huang[2], Tahar Kechadi[3]

The Insight Centre for Data Analytics

University College Dublin (UCD)

Dublin, Ireland.

[1]sameh@ucd.ie, [2]bing.huang@ucd.ie, [3]tahar.kechadi@ucd.ie

***Abstract*— a number of cloud service platforms are targeting experts involved in vital decision-making activities that can lead, for example, governments to take precautions or healthcare professionals to perform specific operations. In order for these cloud platforms to operate there exist an underlying knowledge discovery engine that is employing a number of advanced data analytics techniques to derive insightful conclusions. The technical challenges facing designers and developers of such cloud architectures are increasing relatively with the sensitivity of the data in subject and the ranking of the addressees. In this paper, we investigate these challenges and examine the design approaches of these specific cloud service architectures. We propose an agent-based negotiation model for knowledge exchange among service architectures similar to those in subject.**

***Keywords*—Include Cloud Services, Software Agents, Negotiation Protocols, Multi-Agent Systems (MAS).**

I. INTRODUCTION

In Distributed Artificial Intelligence (DAI), a significant part of the research conducted is focusing on improving the level of cooperation achieved between agents that are located in distributed environments [1]. In an intersection between two of DAI's subtopics, Distributed Problem Solving (DPS) [4] and Multi-agent Systems (MAS) [11], our main focus comes in a place related to the quality, complexity, speed and amount of data or services that can be offered by these distributed agents with respect to the amount of resources utilized. Within this theme, several negotiation models were proposed by scholars aiming at the construction of a proper negotiation protocols, strategies, or tactics that agents may employ to reach mutually beneficial agreement, (e.g., the Strategic Negotiation presented in [6])

The art of negotiation in [7] and its attractive research arguments are always of great scholarly interest. From the literature of Multi-Agent Systems, many research efforts have been approaching differently the problem of resolving complex situations among interacting agents by means of self-organization as presented in [5], and others by means of argumentation [3], and also by means of cooperation as presented in [2]. However, negotiation, as another alternative for resolving complex situations among agents, is the focus of our research work.

In multi-agent systems, several research efforts are addressing the negotiation of agents in different contexts and, for different purposes. Literature contributions, such as those of [12, 10, 6], are presenting negotiation models that address specific situations wherein it is likely to have several heavyweight computing machines or robots interacting with each other. It is then assumed that by choosing and applying one of these negotiation models, the interactions among all involved entities will be driven to simplicity and efficiency in resolving complex situations.

In [8] a definition was given to a negotiation protocol in multi-agent systems as "*the public rules by which agents will come to agreements, including the kinds of deals agents can make and the sequence of offers / counter-offers that are allowed*". R. G. Smith in [10] showed ARPANET and similar protocols as examples of high-level negotiations while other low-level ones are required. The Contract-Net protocol R. Smith presented assumes the simultaneous operation of both; agents asking to execute tasks and agents ready to handle it. The asking agents broadcast a call for proposals, and the helping agents submit their offers and then one is granted the pending task.

In [9], a service-oriented negotiation model was presented to handle the interactions of autonomous agents operating in a business process management application, which is a client to server communications. However, Research efforts to come up with a negotiation protocol that increases the serviceability of agents in highly dynamic environments are few, which make our research effort one of the early steps taken to tackle agents' negotiation in a world of interconnected cloud services.

II. MODEL'S ABSTRACT SETTING

Different communities of agents may address different concerns in which some of these concerns may be the result of combining two or more sub-concerns. In our research, a concern is the abstract concept a community is continually supporting as long as certain objectives are better accomplished. Besides, a sub-concern is a generically narrower concept of a community's abstract concern, which is satisfied whenever a specific set of objectives is being achieved.

For instance, a group of robots in a warehouse might be concerned with placing all of the received objects in dedicated spaces, but a sub-concern emerges when a subgroup of these robots is concerned with organizing - *only* - the north part of this warehouse. A possible sub-subconcern occurs while two

978-1-4799-7749-9 /15/$31.00 ©2015 IEEE

robots of the north-part subgroup are concerned with organizing the red objects only. However, robots operating in a warehouse together with the robots operating in a nearby automobile manufacturer are forming a *society* of robots.

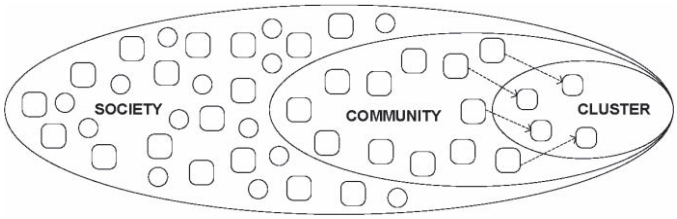

Figure 1**:** Agents' Society, Community, and Cluster.

Definition 1: Agents Society *is a set of agents located in a space wherein different interests' agents are encountering.*

As definition 1 outlines, and figure 1 depicts, when a group of agents come into a common space and, within this group; a number of agents are assigned to completing different abstract concerns, together they form what we call an agents' society. To better elaborate on this, we should think of an agents' society the same as we think of all robots in factories of a specific industrial zone. For example, the industrial zone in Milan has different factories that each has a number of operating robot agents, therefore, all agents in all factories of Milan are forming the Milan's society of industrial robot agents, even though each of these robots is having tasks with different natures to achieve.

A *society* can also be broken down into sub-societies, (i.e., communities), which we define in 2. So, within all factories of Milan, robots involved in car manufacturing, and those of washing machines production, together they are forming two different communities of robot agents, but yet they both belong to the Milan's society of industrial agents. However, the classification of societies and their communities are affected by the perspective a problem is tackled from. For instance, from a different perspective, industrial agents of north Milan can also be considered as a society by itself, and every set of similar robots can form a society's possible community. This classification can also be made according to robots colours, types, or names, and so on.

Definition 2: ***Agents Community*** *is a subset of an agents' society where a common interest is shared among all of its participants.*

In a community of agents, if a group of agents come into agreement about completing a sub-concern of their community's abstract concern, then we call this group an *agents' cluster*, (Definition 3). For an agents' cluster to be formed, the achievement of a common task must be shared among this cluster's parties. Meaning, this cluster's parties are uniting to achieve a task, (e.g., two agents in a warehouse: organizing boxes). However, for a union to occur, prospective agents must first agree on forming this union, and for agents to agree they must negotiate. Negotiation between agents in order to unite and become one of the possible clusters in a community is the broad scope of our model.

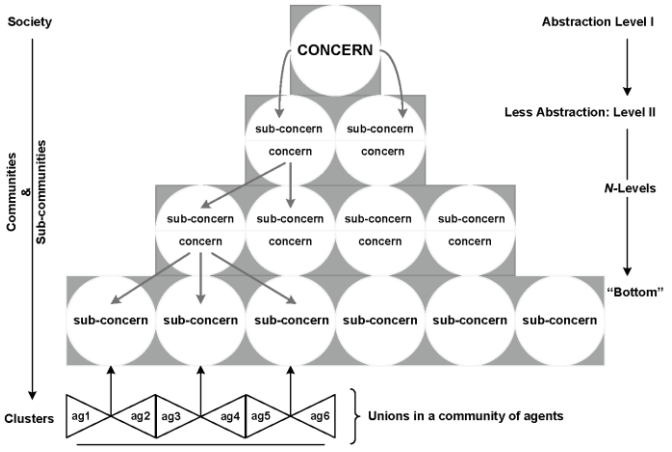

Figure 2: Community's concerns, sub-concerns and unions

The total number of concerns a specific society addresses can be descendingly placed on a pyramid of concerns. In figure 2 this concerns' pyramid has a society's very abstract concern on top of it, then this concern's sub-concerns in less-abstract levels. However, these sub-concerns can also be considered as abstract concerns for sub-communities that play different roles inside the larger community. Depending on the size and number of responsibilities a community has, breaking down the concerns into sub-concerns can be carried on within *N* levels of descending abstraction, until the least community of a society is defined.

Definition 3: ***Agents Cluster*** *is a subset of an agents' community wherein all parties have come into a mutually beneficial agreement that satisfies their predefined needs.*

Since the least sub-community in a chain of a larger communities is the one that has an abstract concern and a set of indivisible sub-concerns. Then - *at the bottom of our pyramid* - any set of sub-concerns that is linked to an earlier level concern are, together, reflecting a specific community where agents' clusters may exist. Depending on the number of involved agents, different sizes of clusters may exist within a community. However, in our model we consider a specific type of clustering in which only two agents are involved, which we call it a *Union*, which we define later in this section.

A. Agents Community, its Concern, and Sub-concern.

In this section we further elaborate on the relationship between a community's concern, its sub-concerns, members, unions, and correlated requests and tradeoffs.

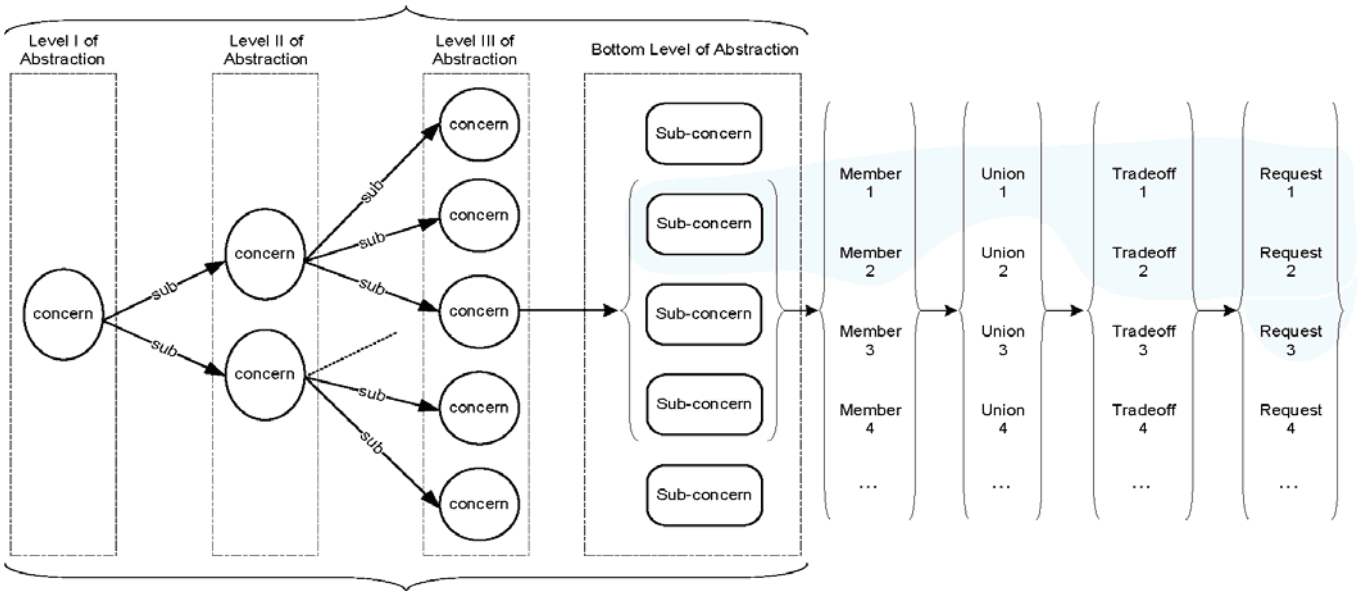

Figure 3: Community's concern, subconcerns, members, unions, requests and tradeoffs

As figure 3 depicts, every community's concern can be described by means of a number of smaller sub-concerns wherein each of these sub-concerns can be either directly assigned to an entity that is committed to satisfying it or, each sub-concern is in turn divided into a less-abstract set of sub-concerns. Therefore, assuming that there are ***N*** concerns, $Concerns = \{C_1,...,C_N\}$ that are distributed among an ***M*** levels of abstraction, $Levels = \{L_1,...,L_M\}$ in which every $l \in Levels$ represents a class of concerns that is more abstract than its subsequent one, then, elements of the **Concerns** are distributed among all **Levels**. Therefore, the number of concerns at L_1 is less than, or equal to, the number of concerns at L_2 , and the number of concerns at L_2 is less than, or equal to, the number of concerns at L_M .

Consequently, the top level of abstraction $L_1 \in Levels$ contains one main concern. The bottom level of abstraction, contains the set of concerns, (i.e., sub-concerns), that cannot be broken-down. As a result, as shown in figure 3 every subset of concerns are associated with one concern of its earlier level of concerns, which is one possible bottom edge of a concerns' hierarchy. Hence, this bottom edge represents also a society's specific community, its main concern, and its members' sub-concerns.

Definition 4: ***Agents Union*** *is the agreement of two agents of the same community to fulfil part or all of each other's requests.*

B. Members, Unions and Tradeoffs.

In every community there is set of members, $Members = \{M_1,..,M_S\}$ in which a possible agreement, *Union* - Definition 4, can be reached. A $u \in Unions$ contains a subset of two **Members**. Members of every possible union will be trading off something that they have in order to eventually satisfy the sub-concern they are assigned to. Consequently, a set of tradeoffs is associated with all possible unions within a community.

Definition 5: ***Tradeoff*** *is the part, or the whole, of what an agent is ready to give in exchange of fulfilling one or more of service requirements.*

In order for a member to agree about trading off one or more of his belongings, and then establish a union with other members, he needs to guarantee his share of the union benefits. Therefore, a potential union partner will attempt to maximize his union benefits by negotiating the fulfilment of a set of requests with the other potential partner of the same union. The requests of this member, the requests of other potential union partners, and all requests of a community's possible unions' partner, together, they will form a community's set of requests, *Requests*.

Definition 6: ***Request*** *is a characteristic of the general demand an agent foresees fulfilled in a prospective union*

C. Tradeoffs vs. Requests

To better elaborate on these interconnected relations between Members, Concerns, Requests and Tradeoffs, we will assume that M_1 are lacking the time to achieve the delegated task. Consequently, M_1 attempts to negotiate the idea of establishing a union $U_1 \in Unions$ with M_2 , so that the time they both have will be enough to write a project proposal and increase research funds. In this situation, time is tradeoff $F_1, F_2 \in Tradeoffs$ both members will have to exchange with the fact of writing a project proposal.

On the other hand, M_1 and M_2 cannot tradeoff their time unless union benefits are guaranteed for each, (e.g., fund commission, promotion, PhD students). Therefore, a community's sub-concern is completed if both $M_1, M_2 \in Members$ fulfil their requests, such as $\{R_1, R_2, R_3\} \in$ **Requests** by means of joining possibly $U_1 \in Unions$ while $F_1, F_2 \in Tradeoffs$ are the union conditions.

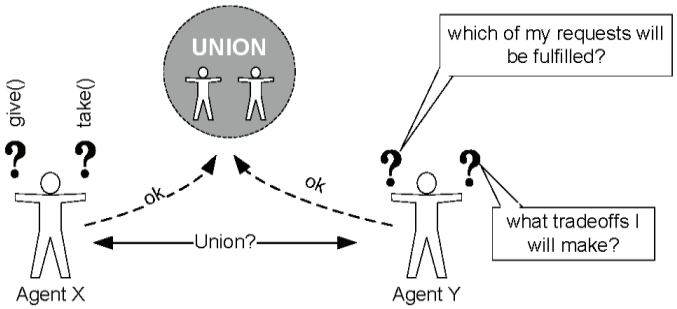

Figure 4: Unions, the negotiation issue of two agents.

Definition 7: ***Instance*** *is the specific concern an agent attempts to complete by means of negotiating the establishment of a union with one of the same community's agents.*

In order for us to keep our model focused on negotiation about a certain issue, we conclude this subsection by highlighting the distinction between any of the community's sub-concerns and the sub-concern that a specific agent addresses. We do that by using the word "*instance*" in Definition 7 - to refer to the single concern a specific agent attempts to complete while playing a certain role in a specific community, and "*instances*" to describe all the same agent's concerns. Consequently, an "*instance*" is a concern for an agent but a sub-concern for its community.

III. The Negotiation Issue

In this section, we use the broad description of the model's setting presented in the earlier section to identify the specific community wherein our negotiation model can be applied. We define the issue which two agents of this community are going address in their negotiation, and probably agree on its realization. Depending on the sets of requests each negotiating agent is seeking to fulfil, a deal between two agents can be reached under different conditions. Therefore, we then link between an agent's possible situations of acceptance with its requests. We conclude this section by putting together the sets of requests of an agent and the tradeoffs it is ready to offer and, linking them with the different cases wherein a successful negotiation may occur.

Definition 8: ***Service-Centric Community*** *is the set of agents interacting with the intention to fulfil the abstract objective of acquiring a predefined service.*

A. The Service-Centric Community

In our model, we consider the negotiation between two agents that are members of the same service-centric community, which we outline in Definition 8. Therefore, all agents are aware of the community's abstract concern / provided service, (e.g., dating, or ridesharing, or bartering). Depending on the kind of service a community is concerned with, a *union* between two agents reflects the completeness of a unique community's sub-concern, (e.g., `date(john,sara)`).

Therefore, a community's sub-concern is created once an agent is searching for a union partner so that together they fulfil each other's' requests.

Definition 9: ***Head-Agent*** *is the central and managing member of a community, which is responsible of applying a community's common regulations.*

In a service-centric community, we assume the existence of a central agent that we call it a *head-agent*, Definition 9. This *head-agent* is the managing authority of a community, (e.g., a multi-agent platform).

For example, within a community of dating service, although all agents are seeking to get a date and somehow pay for it, yet; the *head-agent* will be responsible of putting all male agents in one category and doing the same for all female agents. Besides, the *head-agent* will ensure that any male agent that searches for a union partner is actually looking into the category of female agents, and the vice versa. The same applies for a ridesharing community. All ride-giver agents will be separated from ride-seeker agents, even though both categories contain agents of the same type. By "same type" we mean; all agents are searching to acquire a service and give something in return.

It is worth highlighting here that the reason in our model's setting we drifted toward the existence of an agents' managing entity that is centralized, (i.e., a head-agent (e.g., agents platform)), goes back to the fact that the service-centric community we address in our research is more of an industry-driven provisioning of a commercial service. Therefore, a complete control and monitoring over the behaviour and types of interactions software agents are performing is always necessary.

B. The Service-Centric Issue

In a service-centric community, two negotiating agents will discuss the formation of a mutually beneficial *union*. Meaning, a successful negotiation should lead agentX to employ a number of its capabilities (*Tradeoffs*) in order to satisfy a set of needs (*Requests*) agentY has, while agentY is doing the same for agentX.

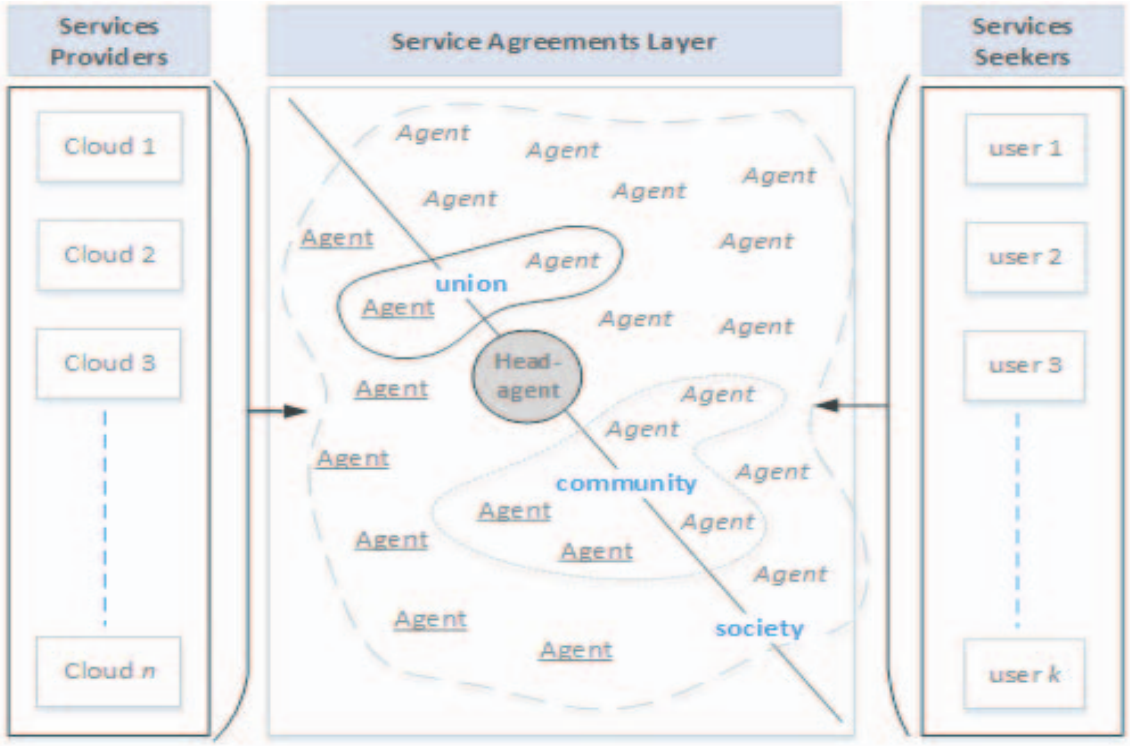

Figure 5: Cloud Service Providers and Software Agent Societies.

For two agents to complete their *instances*, they must unite. For a *union* to occur, agents must tradeoff something that they are capable of providing with the completeness of these *instances*. In order for two agents to agree to tradeoff something they have, they must first be persuaded with the benefits of this prospective union while being involved in a negotiation session - Definition 10.

Definition 10: ***Negotiation Session*** *is the time space in which two agents are negotiating the formation of a union.*

An agent gets into a *negotiation session* following the *head-agent*'s task of applying the community's common rules. However, all agents of a specific community are having their own description of the service they are searching to acquire and what they are willing to give in return, (e.g., if I get a blond or curly female from 20 to 25 I would give either a dinner or flower).

For every agent, this general description is broken down into a set of requests and tradeoffs (Definition 6 & Definition 7) and eventually an agent negotiates with its potential union partner the possibility to satisfy a set of requests with respect to the associated tradeoffs.

C. *Agent's satisfaction and its Instances*

Depending on the nature of requests each agent in a service-centric community is searching to fulfil, the service a community makes available to its members may have different forms. From an agent perspective, the different forms a service takes correspond to different levels of satisfaction an agent may attain while acquiring a service.

The highest level of satisfaction an agent could possibly attain is associated with the fulfilment of a specific set of requests. The lowest level of satisfaction an agent may reach to - *not satisfied* - corresponds to the situation where none of the requests' subsets can possibility be fulfilled.

In our model, an agent's *instance* can take different forms wherein each reflects a different level of satisfaction an agent may attain. Reaching each of these forms is associated with the fulfilment of a different subset of agent's requests. Therefore, the total number of forms and agent *instance* may take correspond to a large set of different requests. The optimal form of an agent's instance, and its highest level of satisfaction, is obtained when a specific set of requests is fulfilled; a *key-set*.

Definition 11: ***Key-set*** *is a specific subset of requests that an agent attempts to fulfil in order to obtain the optimal form of its instance.*

IV. Cloud Services and Software Agents

The formalization we presented earlier can be used in the context of cloud service provision and acquisition by making software agents represent providers and users of different cloud services. As shown in figure 5, different users and service providers are represented by software agents that meet in a common interaction space where, for example, $Agent_1$ representing $User_1$ is attempting to find the cloud service provider that best fit to the preferences of the user it represent.

The role of the *head-agent* will then be responsible of managing the interactions of all agents belonging to the overall society. It is assumed here that different cloud services are managed by different providers and, a range of users are also searching for a range of diverse services. However, whether it is a provider or a potential user they still belong to the *Society* who its members are looking forward to fulfil the concern of closing a "*cloud service agreement*"'.

Based on the formalization we presented in earlier sections, a mutually beneficial agreement can be reached between two software agents and it is then called a *Union*, as shown in figure 5. However, an agreement, or the negotiation process of reaching one, may involve more than one service provider agent and a set of service seeker agents too, and it is then called a *Cluster*. Together, all software agents representing available cloud service providers or seekers are then forming what we defined earlier as *Agents Society*.

The negotiation process software agents engage in to reach agreements is governed by a negotiation model, (i.e., *protocol and strategies*), that takes users and providers requirements and preferences into account. This negotiation model is not discussed in this paper; however, it is currently a work in progress. The purpose of our model will be to maximize the benefits of the interacting agents while considering the limitations of the cloud technologies involved and, the nature of the users they represent.

V. Conclusions

We introduced our view on agents' *Society*, *Community*, and agent *Clusters* that can be applied to software agents repressing users in cloud services. Then we introduced our notion of *Concerns* that can represent the cloud service itself, and the fact that a concern can also be perceived as a sub-concern from less abstract level of the society tree, and so on until a sub-concern is linked to a member of a *society* that ends at that level of abstraction: then it is that member's concern and, therefore, his responsibility to complete. Then we gave an example of mapping concerns onto society's levels.

Within the context of our negotiation model, we introduced our approach of defining a *Request*, a *Tradeoff*, and a *Union* between two agents, which we linked to a *Society's* concerns and levels of abstraction afterwards. Then, we defined what we call the *Head-Agent*, which we assume to be responsible of putting together members of a service-centric community in order to get their service requests fulfilled.

Within a *service-centric community*, each member has a set of requests to be fulfilled and in return a set of *tradeoffs* must be made. Therefore, we then linked a members' different satisfaction levels, (i.e., *views*), with the types of tradeoffs he/she ready to make and, the different subset of requests that will consequently be fulfilled. A member attempts to fulfill any of his *views* by negotiating the establishment of a *Union* with one of the same community's members.

References

[1] A. H. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, San Mateo, CA, USA, August 1988.

[2] J. E. Doran, S. Franklin, N. R. Jennings, and T. J. Norman, "On cooperation in multi-agent systems." *The Knowledge Engineering Review*, vol.12,no.3,pp.309–314, 1997.

[3] P. M. Dung, "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games." *Artificial Intelligence*. vol.77,no.2,pp.321–357, 1995.

[4] E. H. Durfee, V. R. Lesser, and D. D. Corkill, "Trends in cooperative distributed problem solving." *IEEE Transactions on Knowledge and Data Engineering*, vol.1,no.1,pp.63–83, 1989.

[5] M. P. Gleizes, A. L´eger, E. Athanassiou, and P. Glize. "Abrose: Selforganization and learning in multi-agent based brokerage services." In *IS&N 1999: Intelligence in Services and Networks Paving the Way for an Open Service Marke*t, pp.41–54, Barcelona, Spain, Springer. April 1999.

[6] S. Kraus. *Strategic negotiation in multiagent environments*. MIT Press, Cambridge, MA, USA, September 2001.

[7] H. Raiffa. *The Art and Science of Negotiation*. Belknap Press of Harvard University Press, Cambridge, MA, USA, March 1985.

[8] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. MIT Press, Cambridge, Massachusetts, July 1994.

[9] C. Sierra, P. Faratin, and N. R. Jennings."A service-oriented negotiation model between autonomous agents." In *Collaboration between Human and Artificial Societies, Coordination and Agent-Based Distributed Computing*, pp.201–219, London, UK, 1999. Springer-Verlag.

[10] R. G. Smith. "The contract net protocol: High-level communication and control in a distributed problem solver." *IEEE Transactions on Computers*, vol.12,pp.1104–1113, 1981.

[11] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, New York, NY, USA, June 2002.

[12] G. Zlotkin and J. S. Rosenschein, "Negotiation and task sharing among autonomous agents in cooperative domains." In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp.912–917, San Mateo, CA, 1989. ACM.