Neurocomputing 176 (2016) 81–90

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Self-organization in service discovery in presence of noncooperative agents

E. del Val*, M. Rebollo, V. Botti

Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain

ARTICLE INFO

Article history:
Received 10 March 2014
Received in revised form
9 October 2014
Accepted 13 November 2014
Available online 12 May 2015

Keywords:
Self-organization
Cooperation
Complex networks
Service discovery

ABSTRACT

Self-organization and cooperation of agents in open societies play an important role in the success of the service discovery process. Self-organization allows agents to deal with dynamic requirements in service demand. Moreover, in distributed environments where service discovery is carried out by agents that only have a partial view of the system, cooperation with neighbors is a key issue in order to locate the required services. However, cooperation is not always present in open agent societies. With this motivation, we present a set of mechanisms that consider self-organization actions and incentives to adapt the structure of the society to the service demand and to promote a cooperative behavior among agents in open societies.

1. Introduction

Service discovery systems are deployed in dynamic environments where their components, features, and tasks do not remain constant. These systems are expected to perform well under many circumstances (i.e., when the number of available agents changes, or when the service demand varies with time). For that reason, it is important to take into consideration the inclusion of self-organization mechanisms in order to adapt the social underlying structure to environmental conditions and changes in the requirements [20]. When a global view of the society is not available, the organization process should be performed in a decentralized way without the supervision of any central authority. However, this task becomes even more difficult when there are self-interested agents that do not cooperate with others. In that case, if there are no mechanisms to deal with these agents and promote cooperation, the performance of the service discovery process could be seriously compromised [9]. The cooperation of entities that participate in a decentralized system and their self-adaptation to environment changes are required to obtain a good performance that provides benefits for all the participants in the context of distributed systems. Some of the scenarios where cooperation and self-adaptation are required are: wireless ad hoc networks where nodes rely on other nodes to forward their packets in order to reach the destination node, file sharing in P2P systems [24], streaming applications [16], discussion boards [11], on-line auctions [21], or overlay routing [4].

To illustrate the context where the self-organization and cooperation emergence mechanisms are going to be applied, let us present a service discovery scenario where the service discovery process is described as well as the situations where self-organization mechanisms are applied. Consider a network of services as a form of autonomic cloud computing system. This network contains different groups of semantic web services provided by software agents as part of an overlaying network. These services are provided by agents that, in some situations, should interact with each other to achieve a task that they cannot afford to do individually since they are not specialized in that area or because the task is too complex to be carried out by a single agent. For instance, if an agent i only has local knowledge about the services provided by agents k, n, and j and needs to locate a *payment service* in order to do a bank transaction, it should start a service discovery process to locate the agent v. The service discovery process will require a fewer number of steps if all the agents cooperate that if there are non-cooperative agents. It may happen that agent k and agent n may decide not to cooperate in forwarding messages which will damage the performance of the system. Moreover, if services provided by agent v are being frequently requested by agent i, agent i may consider to adapt its current links in order to reduce the number of steps from agent i to agent v.

In this paper, we present a combination of self-organization and cooperation mechanisms that agents use in order to maintain the performance of the service discovery process when there are changes in the service demand or when selfish agents appear. The self-organization mechanisms focus on how the relations between agents could be rearranged or how the agent population could be adapted according to the service demand to maintain or improve

* Corresponding author.
E-mail addresses: edelval@dsic.upv.es (E. del Val), mrebollo@dsic.upv.es (M. Rebollo), vbotti@dsic.upv.es (V. Botti).

http://dx.doi.org/10.1016/j.neucom.2014.11.085
0925-2312/

the performance of the service discovery process. The mechanisms that promote cooperation when there are self-interested agents in the society are based on local structural changes and the use of incentives.

The paper is structured as follows. Section 2 presents works related to the proposal presented in the paper. Section 3 describes the formal model of the system. Section 4 explains how the structure of the network can be modified through local decisions of agents in order to adapt itself to the service demand. Section 5 analyzes the combination of social plasticity and incentive mechanisms in the service discovery scenario to promote cooperation. Section 6 presents a set of experiments where we evaluate the performance of our proposal. Finally, Section 7 presents conclusions and final remarks.

2. Related work

Search approaches commonly used in decentralized systems where all the entities are considered to be equal and there is an arbitrary topology are based on *blind* or *informed* algorithms. *Blind* algorithms do not consider any information about resource locations and use flooding or random strategies that can overload the system with the traffic generated during the search process [19,26]. *Informed* approaches try to cope with this problem and consider local information to create and guide the search. The information considered is about their direct neighbors [3,15] or statistics from previous searches and it is stored in local registries [2].

There are proposals related to decentralized search or service discovery in distributed systems that assume that all the entities that participate in the discovery process are cooperative. However, this will not always happen in open societies. We consider an *open society* as a system where different and heterogeneous agents can join and leave the system. These systems are characterized by heterogeneity of participants, limited trust, different goals, and a high probability of non-conformance to specifications [6]. Approaches based on Game Theory have been widely used to explain mechanisms through which cooperation can emerge and be maintained in different scenarios. Depending on the context, mechanisms such as direct reciprocity [17], indirect reciprocity [18], tags [22], or punishment [12] have been used. Some approaches based on games assume well-mixed populations where everybody interacts with equal frequency with everybody else. However, real populations are not well-mixed. In real populations, some individuals interact more often than others; therefore, to understand the social behavior of the systems it is important to consider the social structure [10].

The approach that we present in this paper for service discovery is based on an informed algorithm that considers local information in order to guide the service discovery process as well as to self-organize the network structure (i.e., the social structure). Initially, the structure is created based on the similarity of the resources provided by the agents. However, the environment conditions do not remain constant. Therefore, in our approach, agents consider self-organization actions in order to maintain or improve the performance of the service discovery process when there are changes in the service demand. Unlike other proposals related to self-organization [25], in our proposal, we consider not only changes in the structure of the agents, but also changes in the population of the system. Moreover, we have considered strategies such as incentives and structural changes to promote cooperation during the service discovery process.

3. Formal model

Our proposal for agent society is modeled as an undirected network populated by a set of autonomous agents $A=\{i,\dots,n\}$ that establish relationships with other agents $L\subseteq A\times A$, where each link $(i,j)\in L$ indicates the existence of a direct relationship between agent i and agent j based on a probability that considers the semantic similarity of their attributes (i.e., the roles and the services of the agents). The role determines the type of services offered by the agent. For a more detailed description of how the structure of the network is created we refer the reader to [8]. An agent is a social entity that interacts with other agents in the society. It controls its own information about:

- The semantic services it offers $S_i=\{s_i,\dots,s_n\}$.
- The organizational roles it plays $R_i=\{cat_1,\dots,cat_m\}$ where each role is defined by a semantic concept. A role can contain a set of semantic service profile descriptions. Each role has associated a numeric index r_i.
- An internal state st_i that contains local information used by the self-organization and the cooperation mechanisms.

The following information contained in the internal state (st_i) is related to *self-organization* mechanisms:

- N_i is the set of direct neighbors agent i has a direct relationship with. For each neighbor $j\in N_i$, agent i has information about: the roles j plays, the services j offers, the degree of connection of j, and the number of times that a query that arrived to the agent i and was not forwarded through its neighbor j (Q_{ij}).
- Acc a set of acquaintances whose existence agent i is aware as a result of the discovery process but it does not have a direct relationship with.
- $\overrightarrow{q_i}=[q_i^{r_1},q_i^{r_2},\dots]$ is the local view of the service demand distribution (i.e., the number of queries that the agent receives about services offered by different roles r_1, r_2, …).
- *The status of the agent*: An agent can be in a *stable* or *transition* status. The status depends on the accuracy of the information about the service demand of the system an agent has. The degree of accuracy is determined by a correlation parameter ρ_i. This parameter establishes the relationship between the local service demand distribution ($\overrightarrow{q_i}$) and the expected service demand distribution. Power-law, Exponential, and Zipf's-law distributions are present in many features of Internet [1,13,5]. In our system, the exponential distribution has been considered as the function that models the service demand in the system, where there are always a few services that are the most demanded and the rest of the services have a lower demand rate. If an agent has an accurate view of the service demand (ρ_i close to 1), it is considered to be in a *stable* situation. When a new agent arrives to the system, or when it has information that introduces noise in its local environment, the agent is considered to be in a *transition* situation.

The information in the internal state of an agent (st_i) related to the *cooperation* is:

- dc_i represents the degree of cooperation of agent i and it ranges in the interval [0,1],
- $\mathcal{B}_i$ represents the behavior of agent i. The behavior of the agent can be cooperative or noncooperative and it is established taking into account the agent's payoff and the behavior of its neighbors.
- $\mathcal{F}_i$ is the number of queries that agent i forwarded,
- $\mathcal{SQ}_i$ is the number of queries that went through the agent i and finally arrived to the target agent,
- $\mathcal{RQ}_{ij}$ is the number of queries from agent i that a neighbor agent j refused to forward,
- $\mathcal{P}_i$ is the number of queries solved agent i,
- $\mathcal{C}_i$ is the number of queries created by agent i.