Enhancing Decentralized Service Discovery through Structural Self-Organization

(Extended Abstract)

E. del Val[1], M. Vasirani[2] M. Rebollo[1], A. Fernandez[2]
[1]Universitat Politècnica de València
{edelval,mrebollo}@dsic.upv.es
[2] Universidad Rey Juan Carlos
{matteo.vasirani,alberto.fernandez}@urjc.es

ABSTRACT

The efficiency of service discovery in distributed systems relies on the collaboration of the agents and the structure of the relations established among them. Structural relations cannot be static, agents should be able to adapt their links as the domain conditions and their interests change. This self-organization considerably improves the performance of the service discovery process. We present a self-organization mechanism that facilitates the task of decentralized service discovery and improves its efficiency in dynamic environments. Each agent has local knowledge about their neighbors and the queries received during the discovery process. With this information, each agent is able to decide when it is more appropriate to modify its structural relations with its direct neighbors and what the most suitable acquaintances to replace them are.

Categories and Subject Descriptors

H. [**Information Systems**]

General Terms

Management, Performance

Keywords

Services, Self-adaption, Self-Organization, Similarity, Homophily

1. STRUCTURAL RELATIONS FOR SELF-ORGANIZING AGENTS

Structural relations define the set of agents with which an agent establishes a relation. One of the criteria considered to establish structural relations is *homophily* [3]. Homophily is present in many complex networks. The idea behind the homophily concept is that individuals tend to interact and establish links with similar individuals through a set of social dimensions. Therefore, at the structural level, communities of similar agents (provide similar services) are created in a decentralized way. The resulting system structure can be considered to be preferential attachment network, which grows according to a simple self-organized process [4].

Each agent in this structure has a local knowledge about its structural relations, as well as knowledge about the environment. Formally, the knowledge model of agent i is a tuple $< \mathcal{A}_i^T, \mathcal{A}_i^K, \mathcal{A}_i^R, \mathcal{C} >$, where $\mathcal{A}$ is the set of agents in a multiagent system and

- $\mathcal{A}_i^T \subseteq \mathcal{A}$ is the set of agents that agent i can communicate with.
- $\mathcal{A}_i^K \subseteq \mathcal{A}$ is the set of acquaintances of agent i. We point out that, in general, $\mathcal{A}_i^T \subseteq \mathcal{A}_i^K$, since an agent is always at least aware of the existence of its neighbors in the communication network.
- $\mathcal{A}_i^R \subseteq \mathcal{A}$ is the set of agents that agent i has a relation with.
- $\mathcal{C}$ is the set of possible categories (relation types). We consider that each agent is associated, at least, to one category that is related to the services it provides.

1.1 Self-Organization in Service Discovery

When an agent needs a service, since there is not a service discovery facilitator nor a registry to be queried, it sends a query q that contains the requirements for a provider agent t, which is unknown. The query contains the service description and its category. The receiver agent performs a matchmaking process of the query against the services it offers. If the best matching service has a degree above a certain threshold, then the search ends successfully. In the case of an unsuccessful matching, the query is forwarded to one of its neighbors $j \in \mathcal{A}_i^R$, which is the most similar neighbor to the target agent t considering the semantic closeness (homophily degree) to the desired service and the degree (number of connections) of the agent [4]. This process is repeated until an agent that offers a service that is 'similar enough' is found or when the TTL (Time To Live) of the query ends. The criterion of 'similar enough' is established by the agent that starts the service search process as a semantic similarity threshold.

Each time an agent forwards a query, it updates the information about its structural relations and adds its identification to the query along with its matching degree. If the query has been successfully solved, the agent that started the process adds the target agent t as an acquaintance agent.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

The target agent that has the required service propagates a message to the agents that participated in the search. Each agent analyzes the utility of its links. In the case that the agent has an acquaintance that has a higher utility than the current links, it decides to break useless structural relations with current neighbors and creates new ones with acquaintances.

The criterion to evaluate structural relations is based on their utility. In the context of service discovery, the *Utility* of a structural relation between agents i and j for a category c is defined as:

$$U^c_{i,j} = \frac{\#^c_i}{\#_i} \cdot m^c_j, \tag{1}$$

where $\frac{\#^c_i}{\#_i}$ is the ratio between the number of queries for service category c received by i and the total number of queries received by i so far, and $m^c_j \in (0, 1)$ is the average degree of match for queries of category c performed by agent j. How this degree of match is calculated is explained in more detail in [4].

If a relation with a neighbor is used to address requests of services of a certain category, then it is interesting for the agent to maintain the link. However, if a relation has not not being used, then the agent must decide whether or not to maintain it. The utility of an structural relation decays exponentially according to Eq. 2 [2]:

$$\tau^c_{i,j} = 1 - e^{-\rho \cdot U^c_{i,j}} \tag{2}$$

where $\rho \in (0, \infty)$ is an adjustable parameter and $U^c_{i,j} \in \mathbb{R}^+$ is the utility of the established relation between agent i and agent j for the category c .

Each agent i maintains a vector of values $\boldsymbol{\tau}_{i,j} = [\tau^1_{i,j} \ldots \tau^{|\mathcal{C}|}_{i,j}]$ for each one of its neighbors. An element $\tau^c_{i,j}$ of the vector represents, the probability of sending a query of category c through agent j. When agent i establishes a new relation of category $c \in \mathcal{C}$ with agent j, the corresponding value $\tau^c_{i,j}$ is also initialized to 1. New relations with some of the acquaintances can be formed. Thus, for every acquaintance $j \in \mathcal{A}^K_i - \mathcal{A}^R_i$, agent i maintains a vector of values $\boldsymbol{\eta}_{i,j} = [\eta^1_{i,j} \ldots \eta^{|\mathcal{C}|}_{i,j}]$ that represents the probability that agent i will establish a new relation of category $c \in \mathcal{C}$ with agent j. The probability of actually establishing a new relation with agent j is given by an Equation similar to Eq. 2.

2. EXPERIMENTS

We evaluated the influence of the proposed mechanism based on the utility functions for the evaluation of structural links with neighbors and acquaintances, and the criteria to each agent decides when it is more appropriate change current structural relations. We compared the results of our proposal with those obtained without using adaptation mechanisms and with a Q-learning algorithm called Weighted Policy Learner (WPL)[1].

Considering the number of changes in the structural relations between agents, the 'Utility' mechanism initially generates a high number of changes if we compare it with the 'WPL' mechanism. The 'WPL' follows a constant rate of changes, and the adaptation is slower than the 'Utility' mechanism. 'Utility' allows agents to only rewire links when the acquaintance links are significantly better than the current links. This makes agents change a reasonable number of structural relations. The success of the service discovery system is improved with both strategies (see Fig. 1). With both adaptation mechanisms, agents are able to create new links that connect with other agents that offer the most demanded services. The 'Utility' mechanism improves the success rate in the first two iterations. However, the WPL takes more time to achieve a success rate over 95%.

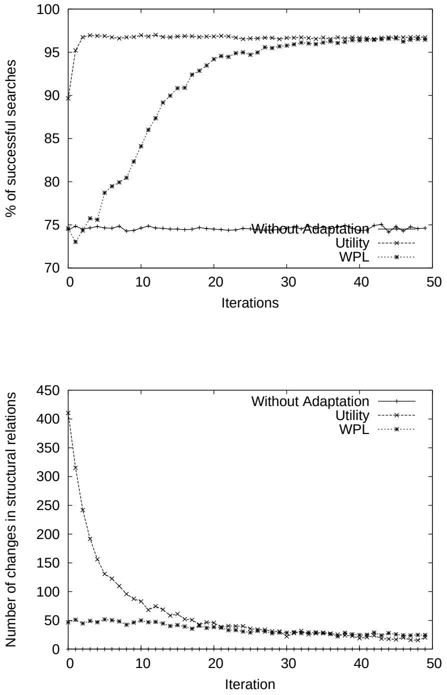

Figure 1: Evolution over time of system measures using Utility and WPL mechanisms. (Top) Percentage of queries that end successfully. (Bottom) Number of structural changes in the system.

3. ACKNOWLEDGEMENTS

Work partially supported by the Spanish Ministry of Science and Innovation through grants TIN2009-13839-C03-01, TIN2009-13839-C03-02 (co-funded by Plan E), CSD2007-0022 (CONSOLIDER-INGENIO 2010), FPU grant AP-2008-00601 awarded to E. del Val.

4. REFERENCES

[1] S. Abdallah and V. Lesser. Multiagent reinforcement learning and self-organization in a network of agents. In *6th AAMAS*, 2007.

[2] E. M. Jin, M. Girvan, and M. E. J. Newman. Structure of growing social networks. *Phys. Rev. E*, 2001.

[3] P. Lazarsfeld. Friendship as a social process: A substantive and methodological analysis. *Freedom and Control in Modern Society*, 1954.

[4] E. D. Val, M. Rebollo, and V. Botti. Self-Organized Service Management in Social Systems. In *45 HICCS*, 2012.