2012 45th Hawaii International Conference on System Sciences

Self-Organized Service Management in Social Systems

E. del Val, M. Rebollo, V. Botti
Universitat Politècnica de València
Valencia, Spain
Email: {edelval, mrebollo, vbotti}@dsic.upv.es

***Abstract*—Humans create efficient social structures in a self-organized way. People tend to join groups with other people with similar characteristics. This is call homophily. This paper proposes how homophily can be introduced in Service-Oriented Multiagent Systems to create efficient self-organized structures in which agents are linked to similar agents, where the similarity is based on the set of services that each agent provides and the roles they play. The results show that a greedy method can be used to locate services in the network and that homophily, which links similar services together, can produce a significant improvement in the performance of the search process. A second contribution is the study of the adaptation of the agents to the number and the type of services demanded. The paper shows how, considering just local information and making local decisions to stay or leave the system, the network adapts itself to a known service distribution.**

***Index Terms*—Complex Networks, Social Systems, Service Discovery, Self-Adaptation**

I. INTRODUCTION

Paradigms for computing, such as peer-to-peer technologies, grid computing, or autonomic and large systems can be considered in terms of service provider and consumer entities or agents. Service-Oriented Multi-Agent Systems (SOMAS) can be described as one of these systems [1], [2], [3]. SOMAS are open and dynamic systems, where agents provide their functionality through services. In these environments, the available services change dynamically and service management is not an easy task [4].

Centralized mechanisms, such as registries or middle-agents, partially address this task [5], [6]. Weaknesses such as bottlenecks, coordination, or failures make centralized approaches inappropriate for coping with dynamic system requirements. Moreover, these mechanisms rely on global knowledge and this global knowledge is not usually available in open SOMAS because of the constant replacement of service providers. Hence, decentralized service management mechanisms are required in these systems.

By observing current society, human beings are able to create efficient social structures, in a self-organized way, without the supervision of a central authority [7]. These structures allow individuals to locate others in a few steps taking only local information into account. One of most salient properties present in these social networks is *homophily* [8], [9]. The idea behind this concept is that individuals tend to interact and establish links with similar individuals along a set of social dimensions (attributes such as religion, age, or education). Therefore, in a structure based on homophily, an individual has a higher probability to be connected to a more similar individual rather than to a dissimilar one. This criteria creates structures that facilitate the location task [10], [11], [12], [13], [14]. For this reason, homophily could be considered as a self-organizing principle to generate searchable structures. Homophily emerges from two mechanisms [15], [16]:

- individual preferences: individuals tend to interact with others who share similar attributes. This homophily is called *choice* or *individualistic homophily*. In general, these attributes can be considered static: the value does not change or it changes with a low frequency.
- social structures and dynamics, which make individuals more similar over time. This is called *induced* or *structural homophily*.

In this paper, we propose a decentralized service management system for SOMAS. The structure is a preferential attachment network [17] based on two types of homophily: *choice* and *structural*. *Choice homophily* is considered to build the system structure and to guide the search of services. *Structural homophily* is used in the system for self-adaptation of the agents to the system demand.

II. SYSTEM MODEL

The system is formed by a set of autonomous agents that offer their functionality through a set of semantic services. They have a reduced view of the global community: just a limited number of direct neighbors are known and the rest of the network remains invisible to them.

DEFINITION *1:* (System). For the aim of this proposal, the system is defined as a tuple (A, L), where $A = \{a_1, ..., a_n\}$ is the a finite set of autonomous agents that are part of the system and $L \subseteq A \times A$ is the set of links, where each link $(a_i, a_j) \in L$ indicates the existence of a direct relationship between agent a_i and a_j.
It is assumed that the knowledge relationship among agents is symmetric, so the network is an undirected graph.

An agent is a social entity that interacts with other agents in the system. It controls its own information about (i) the semantic services it offers, (ii) the role it plays in the organization, and (iii) local knowledge about its immediate neighbors. The agent is unaware of the rest of the agents in the system.

978-0-7695-4525-7/12 $26.00 © 2012 IEEE
DOI 10.1109/HICSS.2012.522

DEFINITION 2: (Agent). An agent $a_i \in A$ is characterized by a tuple of five elements $(R_i, N_i, st_i, \pi_i, \rho_i)$ where:

- $R_i = \{r_1, \ldots, r_m\}$ is the set of roles played by the agent;
- N_i is the set of neighbors of the agent, $N_i = \{a_p, ..., a_q\}$: $\forall a_j \in N_i, \exists (a_i, a_j) \in L$, and $|N_i| > 0$. It is assumed that $|N_i| \ll |A|$;
- st_i is the internal state of the agent;
- $\pi_i : st_i \rightarrow N_i$, is the neighbor selection function that returns the most promising neighbor to provide a service;
- $\rho_i : st_i \rightarrow \Psi$ is the adaptation selection function where Ψ is the set of finite adaptation actions of the agent.

The organizational role of an agent is a semantic concept that is defined in a common ontology shared in the system. The role is related to the services that can be offered by the agent.

DEFINITION 3: (Role). A role $r_i \in R_i$ is defined by the tuple (ϕ_i, S_i) , where:

- ϕ_i is a semantic concept for the role;
- $S_i = \{s_1, \ldots, s_l\}$ is the set of services associated to the role. Each service is defined by the tuple $s_i = (I_i, O_i, P_i, Eff_i)$, where the components are the set of inputs, outputs, preconditions, and effects of the services, respectively. All of them are semantic concepts that can be defined in different ontologies.

Homophily is introduced to create s self-organized structure in which agents are linked to similar ones. *Choice homophily* (*CH*) is the factor that allows the agents to establish links with other agents and to redirect queries about services that they cannot offer. This homophily is based on the characterization of the services that the agents provide and the roles that are played by them. *Structural homophily* (*SH*) refers to how the structure in which the individuals are adapts itself to external conditions. The adaptation of each agent to the system conditions makes the structure of the system more efficient in fulfilling the service demand. Also, choice homophily is subdivided into two types: (i) *status homophily*, which is related to the formal or informal status similarity of the individuals (social status, status within an organization, or professional degree); and (ii) the *value homophily*, which is based on the similarity of shared attributes (such as gender, age, geographical location, and so on).

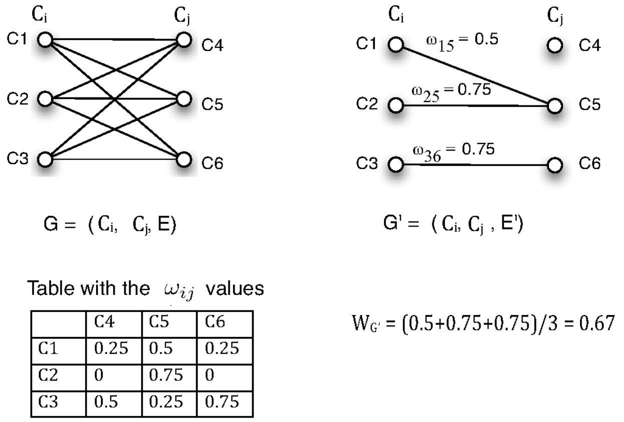
Matching these concepts with the agency-related concepts, status homophily can be identified with the semantic description of the role that an agent plays within an organization, whereas value homophily represents the individual characteristics of the agent.

DEFINITION 4: Choice homophily between two agents $a_i, a_j \in A$ in the system is defined as the linear combination of status and value homophily

$$CH(a_i, a_j) = \varphi * H_s(R_i, R_j) + (1 - \varphi) * H_v(S_i, S_j)$$

The φ parameter regulates the importance of the influence of roles (status homophily) or services (value homophily) in the total homophily of the agent with its neighbors.

The *value homophily* function $H_v(S_i, S_j)$ calculates the degree of matching between two set of services, where S_i and S_j are the sets of services provided by the agents a_i and a_j, respectively. In general, the level of matching between to sets of semantic concepts C_i and C_j is calculated through a *bipartite matching graph*. Let $G = (C_i, C_j, E)$ be a complete, weighted, bipartite graph that links each concept $c_i \in C_i$ to each concept $c_j \in C_j$. ω_{ij} represents the weight associated to the arc $e_i = (c_i, c_j) \in E$ between c_i and c_j as the semantic similarity between those concepts. Four degrees of matching can be identified: *exact, subsumes, plug-in*, and *fail* [18]. The match is considered as *exact*, if $c_1 \in C_i$ is equivalent to $c_2 \in C_j$ ($c_1 \equiv c_2$); *subsumes*, if c_1 subsumes c_2 ($c_1 \sqsupset c_2$); *plug-in*, if c_1 is subsumed by c_2 ($c_1 \sqsubset c_2$); and *fail*, otherwise. A value in the interval $[0, 1]$ is assigned to each degree of matching, where 1 represents an exact matching among the terms. The best match among concepts is obtained by calculating the maximum weighted bipartite matching, $G' = (C_i, C_j, E')$, where $E' \subseteq E$ are the edges that have the maximal value.



	C4	C5	C6
C1	0.25	0.5	0.25
C2	0	0.75	0
C3	0.5	0.25	0.75

$W_{G'} = (0.5+0.75+0.75)/3 = 0.67$

Fig. 1. Full connected Weighted Bipartite Graph G and resulting Maximum Weighted Matching Bipartite Graph G'

To calculate the *value homophily*, four bipartite graphs are defined, one for each one of the components of service s_i: inputs, outputs, preconditions, and effects. Let's explain the case of the inputs. The rest of the components are treated in the same way. Let $\overline{I_i} = \bigcup_{s_i \in S_i} I_i$ be the set formed by all the inputs of all the services s_i of the agent a_i; $G_I = (\overline{I_i}, \overline{I_j}, E)$ the weighted bipartite graph among the inputs of all the services S_i and S_j provided by agents a_i and a_j; and let $G'_I = (\overline{I_i}, \overline{I_j}, E')$ be the maximum weighted bipartite matching. Then $W_{G'_I}$ is defined as:

$$W_{G'_I} = \frac{\sum\limits_{\omega_{ij} \in E'_I} \omega_{ij}}{\max |\overline{I_i}|, |\overline{I_j}|} \tag{1}$$

the normalized total weight of the maximum bipartite graph G'_I. $W_{G'_O}$, $W_{G'_P}$, and $W_{G'_{Eff}}$ are similarly defined for outputs, preconditions, and effects, respectively.

DEFINITION 5: Value homophily between two agents a_i and a_j is defined as

$$H_v(S_i, S_j) = \alpha \left[\beta * W_{G'_I} + (1-\beta) W_{G'_O}\right] +$$
$$(1-\alpha)\left[\beta * W_{G'_P} + (1-\beta) W_{G'_{Eff}}\right] =$$
$$= \alpha \left[\beta \frac{\sum_{w_{ij} \in E'_I} w_{ij}}{\max|\overline{I_i}|, |\overline{I_j}|} + (1-\beta) \frac{\sum_{w_{ij} \in E'_O} w_{ij}}{\max|\overline{O_i}|, |\overline{O_j}|}\right] +$$
$$+(1-\alpha)\left[\beta \frac{\sum_{w_{ij} \in E'_P} w_{ij}}{\max|\overline{P_i}|, |\overline{P_j}|} + (1-\beta) \frac{\sum_{w_{ij} \in E'_{Eff}} w_{ij}}{\max|\overline{Eff_i}|, |\overline{Eff_j}|}\right]$$

The parameters α and β assign different weights to the components of the formula. The adjustment of $\alpha, \beta \in [0,1]$ allows varying how the parameters of the service are considered in the calculation of value homophily. The α parameter controls a data-driven homophily calculation (inputs and outputs) or a goal-driven (preconditions and effects) homophily calculation. The β parameter determines the importance of the intakes (inputs and preconditions) or the consequences (outputs and effects) in the homophily calculation.

The *status homophily* $H_s(R_i, R_j)$ in the system calculates the best match between the set of roles R_i and R_j played by the agents a_i and a_j. The match between two roles $r_i \in R_i$ and $r_j \in R_j$ is based on the distance between the semantic concepts ϕ_i and ϕ_j. The function presented by Fu et al. [19] is used to calculate the distance.

DEFINITION 6: Status homophily between two agents a_i and a_j is defined as the maximum semantic distance between the concepts ϕ_i and ϕ_j that describe the roles $r_i \in R_i$ and $r_j \in R_j$ for all possible pairs (r_i, r_j).

$$H_s(R_i, R_j) = \max_{r_i \in R_i, r_j \in R_j} (rmatch(\phi_i, \phi_j))$$

where

$$rmatch(\phi_i, \phi_j) = \begin{cases} 1 & \text{if path length = 0} \\ e^{(-\lambda(pl+pc))} * \delta & \text{if roles no siblings} \\ e^{(-\lambda(pl-d))} * \delta & \text{if roles siblings} \end{cases}$$

and

$$\delta = \frac{e^{\gamma dp} - e^{-\gamma dp}}{e^{\gamma dp} + e^{-\gamma dp}}$$

The status homophily considers: (i) the shortest path length between the role concepts ϕ_i and ϕ_j in the role ontology (*pl*); (ii) the depth of the roles in the ontology (*dp*); (iii) the number of the sibling nodes of each role (*d*), and (iv) the relationship 'parent-child' between roles (*pc*). λ and γ are parameters to control the influence of path length and depth respectively. The value obtained in the calculation of $H_s(R_i, R_j)$ ranges in the interval [0,1], where 1 indicates that the roles are the same.

III. COMMUNITY CREATION USING HOMOPHILY

Choice homophily establishes a measure of semantic similarity between two agents. When a new agent, a_i, arrives to the system, it establishes at least one link with another agent, a_j, that is already present in the network. The link between two agents is established taking into account the probability for the agent a_i to establish a connection with agent a_j ($P_l(a_i, a_j) = (1 - CH(a_i, a_j))^{-r}$), that is proportional to the *choice homophily* between the agents. To obtain the probability distribution the choice homophily between two agents should be divided by an appropriated constant ($\sum_{a_j}(1 - CH(a_i, a_j)^{-r})$).
The r parameter is a homophily regulator. When r is zero, the system shows no homophily, i.e, agents are not grouped by similar services. As r grows, links tend to connect agents with more similar services. Basically, r makes the system create communities with similar services [20].

Agents have a greater probability of establishing connections with other agents if they provide similar services (value homophily) and play similar roles (status homophily) in the system. As a result of this behavior, communities of similar agents are created in a decentralized way. The system structure can be considered to be a preferential attachment network, which grows according to a simple self-organized process. The construction process of a growing network ensures that the oldest nodes have a higher probability of receiving new links [21], so the total number of neighbors an agent has will depend on its age. Therefore, agents with more connections are more likely to receive new connections than agents with fewer connections. Because the homophily condition is a probability function, it allows new agents not only to establish 'direct connections' between agents with similar attributes (services), but also between agents that are not similar. These connections are responsible for the small-world characteristics of the system, which allow agents to locate other agents efficiently by using only local information.

A. Semantic Distributed Search of Services

Agents should rely on local information for several reasons. One reason is to avoid a dependence on a single point of failure. Another reason is to avoid the effects of changes in the system structure. A third reason is that global information may not be available in open and dynamic systems.

In the context presented in this paper, the selected algorithm for service discovery in the system is an extension of the Expected-Value Navigation (EVN) algorithm [22], which is a greedy, mixed algorithm that uses degree and similarity. It has been modified to use the choice homophily as similarity measure that integrates role information with the service description. The algorithm performs as follows. When an agent a_i is looking for a target agent a_t that provides a required service s_t and plays certain role r_t, or when the agent a_i receives a query about a service that it cannot provide, it redirects the query to the most promising agent in its neighborhood. The most promising neighbor, $a_j \in N_i$, is the most similar neighbor to the target agent a_t that has the highest number of connections. The search process ends when 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 ε.

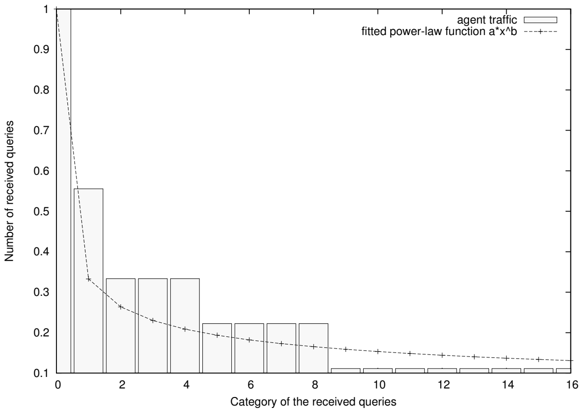


Fig. 2. Demand analysis in agent a_i. For each query received, a_i classifies it in a category. The x-axis shows the identified categories and the y-axis shows the number of queries of that category that a_i has received.

π_i is the selection function that calculates the most promising neighbor a_j of an agent a_i to reach the agent a_t (see Definition 2).

$$\pi_i(a_t) = \underset{a_j \in N_i}{\operatorname{argmax}} P_s(a_j, a_t) \tag{2}$$

For each neighbor a_j, $P_s(a_j, a_t)$ calculates the probability that the neighbor a_j redirects the search to the nearest network community where there are more probabilities to find the agent a_t. Equation 3 uses homophily-based factors (choice homophily CH and homophily regulator r) and degree-based factors (number of neighbors $|N_i|$) to explore the network. The results obtained, which demonstrate the validity of this approach, can be found in Section V-A.

$$P_s(a_j, a_t) = 1 - \left(1 - \left(\frac{(CH(a_j, a_t)}{\sum_{a_j \in N_i} CH(a_j, a_t)}\right)\right)^{|N_j|} \tag{3}$$

IV. STRUCTURAL HOMOPHILY AS A LOCAL SELF-ADAPTIVE METHOD

The concept of structural homophily is used to facilitate the decentralized self-adaptation of the system. This kind of homophily reflects in which proportion the services an agent supplies are similar to the system demand. In the system, each agent controls the queries that pass through it. The agent classifies each query into one category by calculating the degree of matching between the required service and one of its services that it uses as reference for the query classification. With this information each agent periodically analyzes its structural homophily in the system, that is, the agent determines how similar the services it offers are to the services demanded in the system. With this local information, the agent decides whether to continue in the system because its services are demanded, or leave it.

It is assumed that the traffic is modeled as a power-law distribution where there are always a few services that are the most demanded and the rest of services have a lower demand rate. Power-law and Zipf's law (a specific case of power-law) distributions are present in many aspects in Internet [23], [24]. Each agent, using a least squares method, fits the distribution of the data it has locally to a power-law curve $y = a \cdot x^b$ (see Figure 2). a and b parameters are estimated through the least squares method and their value depends on the data of the agent.

With this function, the agent evaluates whether or not the services demanded in the system correspond to the services that it offers. The agent substitutes the categories c_i of each one of the services it provides in the power-law function to get the estimation of the demand. The structural homophily of the agent ($SH(a_i)$) is the maximum value obtained among its services.

DEFINITION *7:* (Structural Homophily) defines the relative importance of an agent based on the services it has served and the queries it has redirected as the value associated to the category c_i of the most demanded service $s_i \in S_i$:

$$SH(a_i) = a \cdot c_i^b$$

where c_i is the category that maximizes the following function $c_i = \underset{x}{\operatorname{argmax}}\, a \cdot x^b$

The second contribution of this paper shows how *structural homophily* helps the network to adapt its structure to the service demand using local information without centralized coordination. Each agent is able to decide autonomously about two structural aspects. The first one is whether it should continue, replicate, or leave the system. The second one is related to the creation of new links as a result of the discovery process and the decay of the existing ones.

A. Disconnection or cloning in the network

Each agent decides, independently of the rest of the system, when it is appropriated to analyze its situation. This decision is based on a probabilistic function that relies on the number of queries that the agent receives. This function is a sigmoid that ranges between [0,1],

$$d_{a_i}(q_i) = 1 - \frac{1}{1 + l \cdot e^{\frac{-(q_i - m)}{n}}}, \tag{4}$$

where q_i is the number of queries that arrived to the agent a_i. The parameters l and m are the displacement, and n the steepness. These parameters are configurable and could be adjusted by the agent.

When the function $d_{a_i}(q_i)$ returns a value close to 1, the agent makes an estimation about the demand of its services.

This estimation is based on the structural homophily defined previously (see Definition 7) and it considers choice homophily and the local traffic evolution. This estimation is included in the adapting selection function ρ_i of the agent (see Definition 2).

The basic decisions an agent can take in the proposed model are: to stay in the network or to leave because its services are not demanded. Other actions such as the adaptation of its own services could be taken by the agent, but are not considered in this work. The probability of both facts depends on the structural homophily calculated, therefore $P_\psi(\text{stay}) = SH(a_i)$ and $P_\psi(\text{leave}) = 1 - SH(a_i)$. Agents can be saturated if the number of queries that they receive increases. As a result, agents that decide to stay, can be 'cloned' with a probability that depends on the increment of the traffic Δt managed by the agent $P_\psi(\text{clone}) = 1 - f(x) = 1 - \frac{1}{1+e^{\Delta t}}$. By combining these behaviors, the possible actions an agent can take are the result of the adapting function ρ_i are $\Psi = \{leave, continue, replicate\}$, with probabilities:

$$\begin{aligned} P_\psi(\text{leave}) &= 1 - SH(a_i) \\ P_\psi(\text{continue}) &= P_\psi(\text{stay} \cap \overline{\text{clone}}) = SH(a_i)f(x) \\ P_\psi(\text{replicate}) &= P_\psi(\text{stay} \cap \text{clone}) = SH(a_i)(1-f(x)) \end{aligned}$$

Consequently, an agent will (i) *leave* the network if it is not important for the system (its services are not demanded or it is badly located); (ii) *replicate* itself if it considers that it is relevant for the network and it has received a significant increment in the number of queries that it receives; and (iii) *continue* otherwise (it is relevant but the number of attended queries remains nearly constant).

B. Links decay

Besides the decision of continue, replicating, or leaving, an agent can also decide about the convenience of maintaining links or create new ones. An agent establishes new links with other agents as a result of the service discovery process. Moreover, each agent decides to maintain all the links it has because all are being used to forward queries or remove some of them because they are not being used. To take this decision, the agent uses a decay function that evaluates the probability of maintaining a link considering the number of queries forwarded through it. The lower number of queries are forwarded using a link, the higher probability it has to remove this link. This function is a logistic (sigmoid) function similar to the formula defined previously (Equation 4). Here, the q_i is the number of queries that arrived to the agent and were not forwarded through the link l_i. Each time an agent receives a query, it updates the information about the traffic of the link. If the query is forwarded through the link l_i, its q_i is updated to 0. Otherwise, the number of queries q_i is incremented.

V. EXPERIMENTS

In order to evaluate the system proposal for decentralized service management, several tests have been proposed. The first set of experiments are centered on an equilibrium system where the topology does not change (new agents and links are not created). In these experiments, the influence of choice homophily in the system structure and in the search process is evaluated. The second set of experiments is focused on the analysis of the system adaptation to the service demand considering the structural homophily.

The experiments have been done in a set of synthetic networks with a preferential attachment structure with 1,000 agents, an homophily regulator factor $r = 1.5$ and $k = 2$ as average degree. We consider that two agents have the maximum degree of similarity when $CH(S_i, S_j) < \varepsilon$, that is, the homophily value between them is under a specific threshold $\varepsilon < 0.01$. The effect of this parameter is that the set of agents is divided into a limited number of communities. Agents are initially distributed uniformly over these categories.

A. Search Performance

The first set of experiments analyzes the performance of the proposed system for service management. The behavior of different algorithms has been evaluated with a set of 5,000 queries. The difference among them is how the most promising neighbor is selected in each step. These algorithms are: (i) *random*: a search process using random walks (a neighbor is randomly chosen); (ii) *degree*: a search process using only degree information (the neighbor with the highest degree is selected) [25]; (iii) *similarity*: a search process using only similarity information (the most similar neighbor is chosen) [26], [27], [28]; and (iv) *EVN*: mixed search process using a combination of degree and similarity [11]; and (v) *role-based EVN*: selection based on degree and $CH(a_i, a_j)$..

Figure 3 compares the results obtained with these algorithms in networks with different values of φ. When $\varphi = 0$, the networks have been built considering only value homophily information. The consideration of status homophily ($\varphi > 0$) improves the results obtained by the role-based EVN algorithm. In the networks built based only on semantic service information ($\varphi = 0$), small agent communities specialized in certain type of services emerge and there are only a few connections between different communities. These features make complicated the navigation from one community to other. Therefore, the path lengths obtained by the search strategies are longer. In the networks built based on a combination of organizational and service information, the communities that emerge have a higher number of agents and are highly connected between them. This fact makes easier the navigation between communities. Therefore, the path lengths obtained in the search process are shorter. Organizational information is useful to guide the search.

B. Self-Adaptation

This second set of experiments shows how the network is able to adapt itself to the distribution of the queries making local decisions, considering local information, and without exchanging information.

Initially, agents are uniformly distributed in the system. It is assumed that the request distribution in the system follows

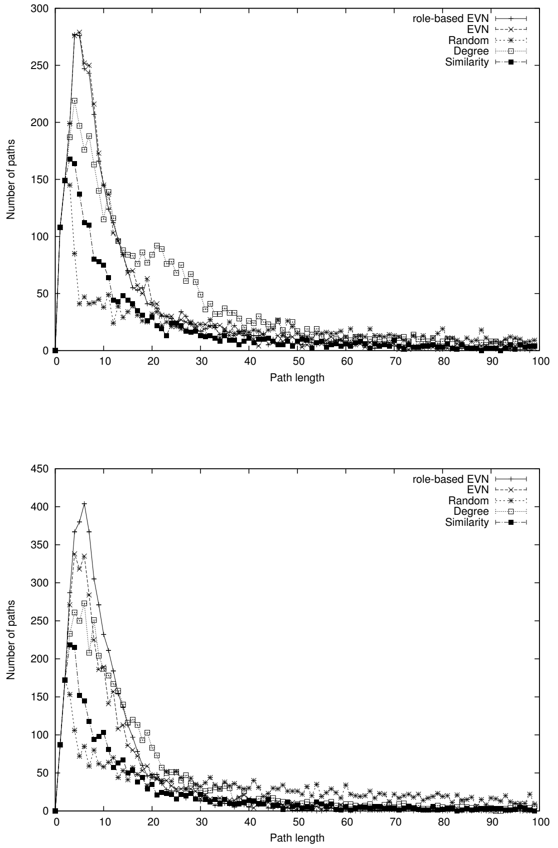


Fig. 3. Search Performance in networks with φ = 0 and φ = 0.5.

a power-law function. Therefore, after the adaptation of each agent in the system, the agent distribution should follow the same power-law distribution. The results are normalized to compare the size of the network with the distribution of the queries. Graphics in Figures 4 and 5 shows the initial distribution (bars), the agent distribution obtained after a set of queries, and the query distribution.

In Figure 4 it can be observed how the agents adapt themselves to the system demand without considering links decay. Each agent, when it considers that it has received enough queries, decides to continue in the system, replicate itself, or leave the system. Most of the agents that offer services that are not demanded decide to leave the system. Agents with more demanded services continue in the system and replicate themselves. The agents with the category 19 are the agents that offer the less demanded services. Therefore, they receive the less number of queries and the adaptation decision takes more time (see Figure 4). The system self-adaptation contributes to improve the performance of the searches. The rate of successful searches increases, and the path length decreases.

In Figure 5 the agents beside deciding about continuing, replicating or leaving, they create links as a result of the discovery process and evaluate the utility of their links in order to maintain or break them. The agents adapt themselves to the service demand distribution. Nevertheless, the global adaptation of the system is not as good as the the adaptation without considering the creation of new links and the link's decay. The main reason is that new links connect different communities of agents and this reduces the traffic in the network. With less traffic, the agents need more time to adapt themselves properly to the service demand distribution. The self-adaptation also contributes to improve the search performance. The disadvantage of this scenario is that the mean path increases. This happens when many agents create too many links with different communities and eliminate links with similar agents. This condition could introduce noise in the traffic that affects to search process. There should be a balance between long links with dissimilar communities and similar communities to improve the search success and reduce the path's length.

VI. CONCLUSIONS

The aim of this work is to investigate how the integration of different areas such as SOMAS and social networks provide the necessary tools to build a decentralized service management system. This system is based on homophily: a sociological concept that is present in many human networks. These networks are created in a decentralized way without the supervision of any entity. One of the most interesting characteristics of these networks is that having only information from their immediate contacts, individuals can reach other individuals in only a few steps. For this reason, homophily has been introduced in our system. Two types of homophily have been considered: choice and structural homophily. The former is based on static information related to services provided by the agents and the roles played by them. The latter is based on the service demand in the system. Choice homophily is used to create a preferential attachment system where agents have more probability of establishing links with other agents that share static attributes with them (such as services and roles) than with dissimilar agents. As the experiments demonstrate, the resultant structure allows agents to reach other agents that offer a required service in a few steps. Of the set of typical strategies used in decentralized environments (degree, similarity or random), the strategy that takes into consideration choice homophily between agents to lead the search obtains better results. Also, the system is able to adapt itself to the service demand, in a completely decentralized way. To do that, each agent calculates its structural homophily. With this information, each agent decides to leave or remain in the system, depending on whether or not its services are required. The experiments demonstrate (i) that homophily is a good criteria to structure agent communities based on similar services, increasing the performance of service discovery in decentralized environments, and (ii) that structural homophily is a good strategy for adapting the system agent distribution to the service demand.

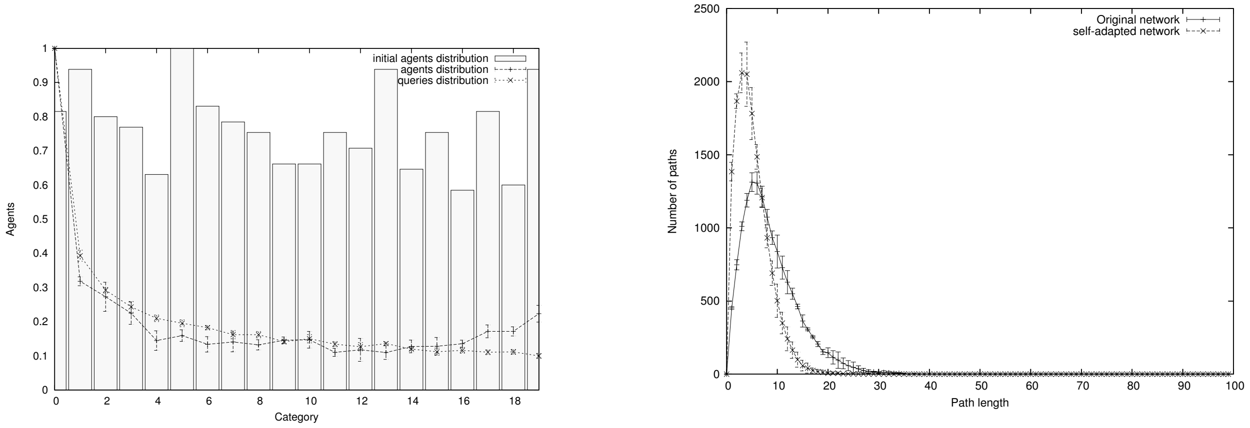


Fig. 4. (Left) Agents distribution in self-adapted networks; and (Right) search performance in self-adapted networks after 15,000 queries.

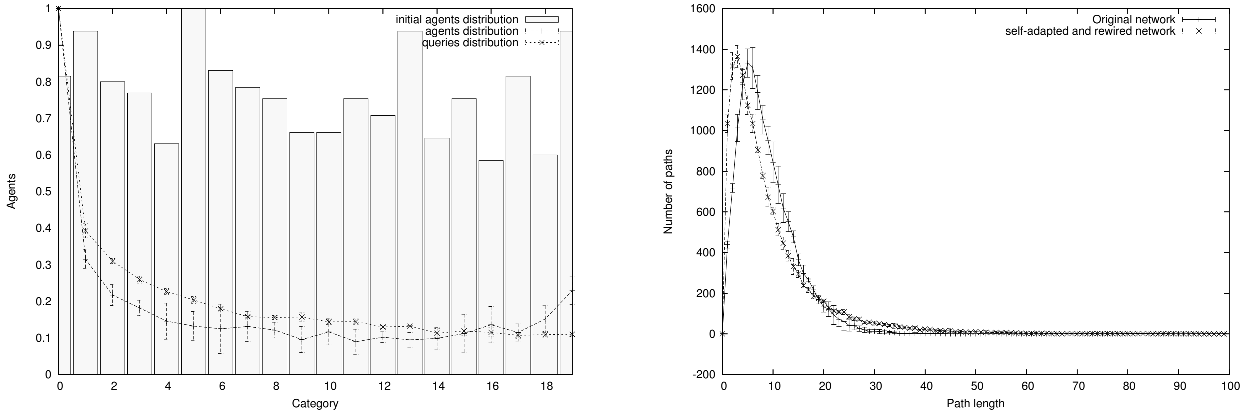


Fig. 5. (Left) Agents distribution in self-adapted networks with link decay; and (Right) Search performance in self-adapted networks with link decay after 15,000 queries.

REFERENCES

[1] M. N. Huhns, "Agents as web services." *IEEE Internet Computing*, pp. 93–95, 2002.

[2] M. N. Huhns, M. P. Singh, M. Burstein, K. Decker, E. Durfee, T. Finin, L. Gasser, H. Goradia, N. Jennings, K. Lakkaraju, H. Nakashima, V. Parunak, J. S. Rosenschein, A. Ruvinsky, G. Sukthankar, S. Swarup, K. Sycara, M. Tambe, T. Wagner, and L. Zavala, "Research directions for service-oriented multiagent systems," *IEEE Internet Computing*, vol. 9, pp. 65–70, 2005.

[3] F. M. T. Brazier, J. O. Kephart, H. V. D. Parunak, and M. N. Huhns, "Agents and service-oriented computing for autonomic computing: A research agenda," *IEEE Internet Computing*, vol. 13, 2009.

[4] N. Bisnik and A. Abouzeid, "Modeling and analysis of random walk search algorithms in p2p networks," in *Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems*. IEEE Computer Society, 2005, pp. 95–103.

[5] UDDI, "Evolution of uddi, white paper available," `http://www.uddi.org/`, 2002.

[6] M. Klusch, B. Fries, and K. Sycara, "Automated semantic web service discovery with owls-mx," in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, ser. AAMAS '06. New York, NY, USA: ACM, 2006, pp. 915–922.

[7] H. Holzer, "Search method used by unemployed youth," *Journal of Labor Economics*, vol. 6, pp. 1–20, 1988.

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

[9] M. McPherson, L. Smith-Lovin, and J. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, 2001.

[10] Watts, Dodds, and Newman, "Identity and search in social networks," 2002.

[11] Ö. Simsek and D. Jensen, "Decentralized search in networks using homophily and degree disparity," in *IJCAI*, 2005, pp. 304–310.

[12] J. Kleinberg, "Complex networks and decentralized search algorithms," in *ICM*, 2006.

[13] S. Currarini and F. Vega-Redondo, "Search and homophily in social networks," *World*, no. 24, pp. 1–32, 2010.

[14] H. Bisgin, N. Agarwal, and X. Xu, "Investigating homophily in online social networks," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, ser. WI-IAT '10. Washington, DC, USA:

IEEE Computer Society, 2010, pp. 533–536. [Online]. Available: http://dx.doi.org/10.1109/WI-IAT.2010.61

[15] M. McPherson, L. Smith-Lovin, and J. Cook, “Birds of a feather: Homophily in social networks,” *Annual Review of Sociology*, 2001.

[16] D. Centola, J. C. Gonzalez-Avella, V. M. Eguiluz, and M. San Miguel, “Homophily, cultural drift, and the co-evolution of cultural groups,” *Journal of Conflict Resolution*, vol. 51, no. 6, pp. 905–929, December 2007.

[17] A. L. Barabasi and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, October 1999.

[18] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, “Semantic matching of web services capabilities,” 2002.

[19] P. Fu, S. Liu, H. Yang, and L. Gu, “Matching algorithm of web services based on semantic distance,” in *WISA*, 2009.

[20] Kleinberg, “The small-world phenomenon: An algorithmic perspective,” in *ACM STC*, 2000.

[21] S. Dorogovtsev and J. Mendes, *Evolution of Networks*. Oxford U.Press, 2003.

[22] Şimşek and Jensen, “Navigating networks by using homophily and degree,” *Proceedings of the National Academy of Sciences*, 2008.

[23] Adamic, “Zipf’s law and the internet,” *Glottometrics*, 2002.

[24] B. A. Huberman and L. A. Adamic, “The nature of markets in the www,” Tech. Rep., 1999.

[25] S. Xiao and G. Xiao, “On degree-based decentralized search in complex networks,” *CoRR*, 2006.

[26] H. Zhang, W. B. Croft, B. Levine, and V. Lesser, “A multi-agent approach for peer-to-peer based information retrieval system,” in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, ser. AAMAS ’04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 456–463.

[27] Y. Upadrashta, J. Vassileva, and W. Grassmann, “Social networks in peer-to-peer systems. paper presented at the,” in *38th Hawaii International Conference on System Sciences*, 2005, pp. 3–6.

[28] D. Bianchini, V. D. Antonellis, and M. Melchiori, “Service-based semantic search in p2p systems,” *European Conference on Web Services*, vol. 0, pp. 7–16, 2009.