the interviews with three actual PD patients with their closest relatives, and the approach is applied in a patient as a case study. As future work, a specific ML, probably an agent-oriented one, is expected. It will be used for customizing MASs for particular PD patients.

Acknowledgements. This work has been done in the context of the project *Social Ambient Assisting Living - Methods* (SociAAL), supported by Spanish Ministry for Economy and Competitiveness, with grant TIN2011-28335-C02-01.

References

1. Arroyo, M., Finkel, L., Gomez-Sanz, J.J.: Requirements for an intelligent ambient assisted living application for parkinson patients. In: Corchado, J.M., Bajo, J., Kozlak, J., Pawlewski, P., Molina, J.M., Julian, V., Silveira, R.A., Unland, R., Giroux, S. (eds.) PAAMS 2013. CCIS, vol. 365, pp. 441–452. Springer, Heidelberg (2013)
2. Bickmore, T.W., Schulman, D., Sidner, C.L.: A reusable framework for health counseling dialogue systems based on a behavioral medicine ontology. Journal of Biomedical Informatics 44(2), 183–197 (2011)
3. Bukhari, A.C., Kim, Y.-G.: Integration of a secure type-2 fuzzy ontology with a multi-agent platform: a proposal to automate the personalized flight ticket booking domain. Information Sciences 198, 24–47 (2012)
4. García-Magariño, I.: Defining and transforming models of parkinson patients in the development of assisted-living multi-agent systems with INGENIAS. In: Corchado, J.M., et al. (eds.) PAAMS 2013. CCIS, vol. 365, pp. 460–471. Springer, Heidelberg (2013)
5. Gomez-Sanz, J.J., Fuentes, R., Pavón, J., García-Magariño, I.: Ingenias development kit: a visual multi-agent system development environment. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Demo Papers, pp. 1675–1676. International Foundation for Autonomous Agents and Multiagent Systems (2008)
6. Gupta, A., Ludäscher, B., Grethe, J.S., Martone, M.E.: Towards a formalization of disease-specific ontologies for neuroinformatics. Neural Networks 16(9), 1277–1292 (2003)
7. Riaño, D., Real, F., López-Vallverdú, J.A., Campana, F., Ercolani, S., Mecocci, P., Annicchiarico, R., Caltagirone, C.: An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. Journal of Biomedical Informatics 45(3), 429–446 (2012)
8. Wang, G., Wong, T., Wang, X.: An ontology based approach to organize multi-agent assisted supply chain negotiations. Computers & Industrial Engineering 65(1), 2–15 (2013)
9. Williamson, C., Simpson, J., Murray, C.D.: Caregivers' experiences of caring for a husband with parkinson's disease and psychotic symptoms. Social Science & Medicine 67(4), 583–589 (2008)

Integration of Self-organization and Cooperation Mechanisms to Enhance Service Discovery

Elena del Val, Miguel Rebollo, and Vicente Botti

Universitat Politècnica de València,
Camí de Vera s/n. 46022, València. Spain
{edelval,mrebollo,vbotti}@dsic.upv.es

Abstract. Agents self-organization and cooperation 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). However, the majority of the proposals for service discovery in distributed systems are only focused on the location task and do not 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 [14]. When a global view of the society is not available, these processes should be performed in a decentralized way without the supervision of any central authority. However, these tasks become 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 [5].

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 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.

J.-S. Pan et al. (Eds.): HAIS 2013, LNAI 8073, pp. 21–30, 2013.

2 Related Work

Search approaches commonly used in decentralized systems where all the entitites 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 the search process [13,18]. *Informed* approaches try to cope with this problem and consider local information to create and guide the search. The information is about their direct neighbors [3,10] or statistics from previous searches and it is stored in local registries [2].

Moreover, the majority of the proposals related to decentralized search or service discovery assume that all the entities that participate in the discovery process are cooperative. However, this not always happen in open societies where there is not a central entity that controls the behavior of the agents. 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 [11], indirect reciprocity [12], tags [15], or punishment [7] have been used. Many approaches that are 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 [6].

The approach that we present in this paper is an informed algorithm that considers both types of local information in order to establish and modify the network structure and to guide the service discovery process. Initially, the structure is created based on the similarity of the resources provided by the agents. However, the environment conditions do not remain constant and in contrast to other proposals that do not consider dynamic environment conditions, 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 [17], 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 used in the area of Game Theory in networks 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 $A = \{i, ..., n\}$ 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 the semantic similarity of their attributes (i.e., the roles and the services of the agents) [4]. An agent is a social entity that interacts with other agents in the society. It controls its own information about (i) the semantic services it offers $S_i = \{s_i, \ldots, s_n\}$, (ii) the organizational roles it plays $R_i = \{r_1, \ldots r_m\}$, and (iii) an internal state st_i, that contains local information used by the self-organization and the

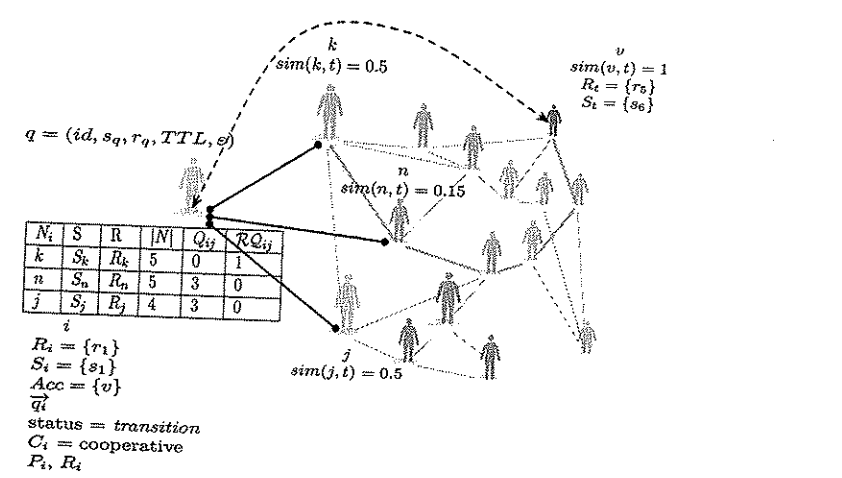


Fig. 1. An example of a decentralized service discovery system where agent i creates a query q that it has to forward to the most promising neighbor (i.e., the closest neighbor to the target).

cooperation promotion mechanisms. The information in the internal state related to the *self-organization* is:

- 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 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;
- $\vec{q_i} = [q_i^{r_1}, q_i^{r_2}, ...,]$ is the local service demand distribution (i.e., the number of queries that the agent receives about services offered by different organizational roles $r_1, r_2, ...$);
- the *status* of the agent. The status depends on the significance of the information an agent has. If an agent has an accurate view of the system, it is considered to be in a *stable* situation. When a new agent arrives to the system, or when it has outdated 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 related to the *cooperation* is:

- dc_i represents the degree of cooperation of agent i and ranges in the interval [0,1],
- C_i represents the behavior of agent i and can take two values: cooperative or not cooperative,
- Q_i is the number of queries that agent i forwarded,
- SQ_i is the number of queries that the agent i forwarded in successful discovery processes,
- RQ_{ij} is the number of queries from agent i that agent j refused to forward,

- $\mathcal{P}_i$ is the number of service requests attended to by agent i,
- $\mathcal{R}_i$ is the number of service requests sent by agent i.

The decision making process about self-organization actions or actions related to the promotion of cooperation is initiated when agent i generates a query $q = (id, s_q, r_q, TTL, \varepsilon)$, which contains the semantic description of the desired service s_q and the role r_q that the target agent should play (see Figure 1). A *target* agent profile t is created with the service and role specified in the query q. Agent i looks for a neighbor similar to t. If it finds a suitable neighbor, the service discovery process ends. Otherwise, the agent i forwards q to one of its neighbors $j \in N_i$. Specifically, q is forwarded to the agent that has semantic closeness to the *target* agent t and also has a high degree of connection. The selected agent j analyzes based on its payoff and the payoff of its neighbors if it is worthwhile forwarding the query. If j rejects forwarding the query, agent i updates the number of times that agent j rejects its request of forwarding ($\mathcal{RQ}_{ij}$) and based on this, agent i considers breaking its current link with j. If agent j does not cooperate, agent i has to select another neighbor that cooperates in the forwarding process. Once a cooperator neighbor is found, agent i forwards the query to it and updates its information about which of its links have been used. Agent i also updates the number of total queries it received (Q_i), and the number of queries about the role r_q ($\overrightarrow{q_i}[r_q]$). When the query reaches a suitable provider agent, all the participants in the service discovery receive a reward. The source agent adds the provider agent found to its set of acquaintances only if it does not already have an acquaintance that plays the role of the provider agent. Finally, the source agent updates its internal state st_i and analyzes the set of self-organization actions that it can carry out.

4 Self-organization Mechanisms

In order to make decisions about self-organization actions agents need to have an accurate local view of the service demand in the system. To evaluate the accuracy of its local view agents analyze its internal state. Initially, an agent is in a *transition* state. An agent in this state does not have reliable and sufficient information to be able to estimate what is the current service demand distribution in the system. In this state, an agent can reorganize its local view of the service demand distribution $\overrightarrow{q_i}$ taking into account the number of queries received about the services associated to each role. An agent changes its status from *transition* to *stable* when the correlation degree (ρ_i) between its local data about service demand in the society, $\overrightarrow{q_i}$, and an estimation of the service demand distribution is over a threshold. In our system, the estimation of the service demand distribution follows an exponential distribution. This type of distribution is present in many features of open systems such as Internet [1,8]. Specifically, we assume that the expected service demand distribution is $eDistr(x) = a \cdot e^{x \cdot b}$, where the x parameter represents a role identifier. We estimate the a and b parameters of this distribution using the least squares method and the data from $\overrightarrow{q_i}$. An agent turns back into the *transition* state if it detects a big enough deviation of the correlation degree at any moment. This usually happens when there are frequent changes in the service demand. Once the agent has analyzed its internal state, it is able to make decisions about self-organization actions.

Self-organization of the structural links. Agents are able to reason about whether or not maintain, reinforce or create new structural relations. Agents consider a *decay* metric associated to each link: $decay(Q_{ij}) = 1 - (1/(1 + \cdot e^{-(Q_{ij} - z)/y}$, where y is the slope, z is the displacement constant, Q_{ij} is the number of queries that arrived to agent i and were not forwarded through agent j. Each time agent i forwards a query, it updates the information about the traffic of its links. If the query is forwarded through agent j, the Q_{ij} is updated to 0. Otherwise, the Q_{ij} is increased by increments of 1. With the information provided by the *decay* function, agent i reasons about the benefit of maintaining its current links.

Population self-organization: leaving, remaining, or cloning. The analysis that evaluates whether it is worthwhile for the agent to remain in the system, clone itself, or leave the system takes the following parameters into account: (i)the number of queries received by the agent Q_i; (ii)the status of the agent; (iii)the structural similarity of the agent SH_i; (iv)the number of queries forwarded since the last analysis Δq^i; (v) the degree of correlation ρ_i.

In the context of service discovery, we define the concept of *structural similarity* SH_i as the degree of similarity between the services demanded in the system and the services provided by an agent in the system. This kind of similarity reflects how important an agent is to the system with regard to the current service demand. The structural similarity of an agent with respect the system dynamics is defined by the following function: $SH_i = a \cdot e^{r_i \cdot b}$, where r_i is the role of agent i that maximizes the following function: $r_i = \underset{x \in R_i}{\operatorname{argmax}}\, a \cdot e^{x \cdot b}$, where the a and b parameters are obtained from the local view of the service demand. SH_i ranges in the interval [0,1], where 1 indicates that the services the agent offers are required in the system, and 0 indicates that the services the agent offers are not being demanded in the system. The conditions to select the leaving or cloning actions are shown in Table 1.

Table 1. Parameters and conditions that agents use during the making decision process about self-organization actions. The parameters are: the number of queries received by the agent, the status of the agent, the structural homophily SH_i, the similarity of the neighborhood SimilarN(N_i), the increase in the number of queries received Δq^i, and correlation value ρ_i.

Action	Num. Queries	Status	SH_i	SimilarN(N_i)	Δq^i	ρ_i
Leave	$1/e^{-(Q_i - d')/y}$	Stable	$<$	> 0		
Clone	$1/(1 + \cdot e^{-(Q_i - 2^{clones})/y}$	Stable	$>$	$< \|N_i\|$	> 0	$> \delta$

5 Incentives and Social Plasticity

In the proposed model, agents have cooperative or non-cooperative behavior. Cooperating in the service discovery scenario implies that an agent is going to: forward queries, request services, and attend to request about its services. If an agent has non-cooperative behavior, it means that the agent is going to act selfishly by requesting services and offering its services, but it is not going to forward the queries that it receives from its neighbors. We assume that each action in our model implies a cost and, in order to promote cooperation, the forwarding action has a reward if the search process ends successfully.

Agents in a neighborhood share information about their payoffs. An agent establishes its behavior based on its payoff and the payoff of its neighbors. An agent calculates its payoff as follows: $\mathcal{PO}(st_i) = \mathcal{SQ}_i \cdot sq - \mathcal{Q}_i \cdot q + \mathcal{P}_i \cdot p - \mathcal{R}_i \cdot r$, where $\mathcal{SQ}_i, \mathcal{Q}_i, \mathcal{P}_i, \mathcal{R}_i$ is the information of the internal state (st_i) of an agent; sq is the benefit obtained by the agents that participate by forwarding queries in a service discovery process that ends successfully; q is the cost of forwarding queries; p is the benefit obtained by the agents that provide a service; r is the cost of requesting a service.

The strategy followed by the agents in order to change their behavior is based on imitation [16]. Agents take into account the payoff of their direct neighbors to update their behavior. If an agent has a neighbor that obtains a higher payoff, the agent changes its behavior to the behavior of its neighbor.

When the number of cooperative agents is greater than the number of non-cooperative agents, non-cooperative agents are prone to change their behavior to cooperate since the probability that a query ends successfully is high, and, therefore, cooperation receives a reward if the discovery process ends successfully. However, when the number of non-cooperators is greater than the number of cooperators, cooperative behavior does not always emerge. In order to facilitate the emergence of cooperation in this scenario, in our proposed model, each agent also has the capacity to change its relationships as time passes based on a logistic function that depends on the number of times a neighbor has refused to forward one of its queries. An agent i maintains a counter per each direct neighbor j ($\mathcal{RQ}_{ij}$) that stores the number of times a neighbor rejected forwarding a query [6]. If a neighbor j decides to change its behavior and forwards queries, the agent updates its counter to 0.

With the combination of the social plasticity and incentives, non-cooperative agents lose connectivity, benefits, and influence in the neighborhood. As a consequence, they decide to change their behavior to the most promising behavior in the neighborhood, which is to cooperate.

6 Experiments

We analyzed the effects of using of self-organization and cooperation mechanisms in the discovery process. The tests were performed on a set of 10 undirected networks with an average degree of connection of 4. The degree of connection distribution follows an exponential distribution. The creation process of the network is described with detail in [4]. The networks were populated by 1,000 agents. The agents played one role and offered one semantic web service associated to this role. Initially, the agents were uniformly distributed over 16 roles, which were defined in an organizational ontology. The set of semantic service descriptions used for the experiments was taken from the OWL-S TC4 test collection[1].

All the agents in the system had the same probability of generating service queries. A query consisted of two features that characterize the required provider agent: the role

[1] http://www.semwebcentral.org/projects/owls-tc/

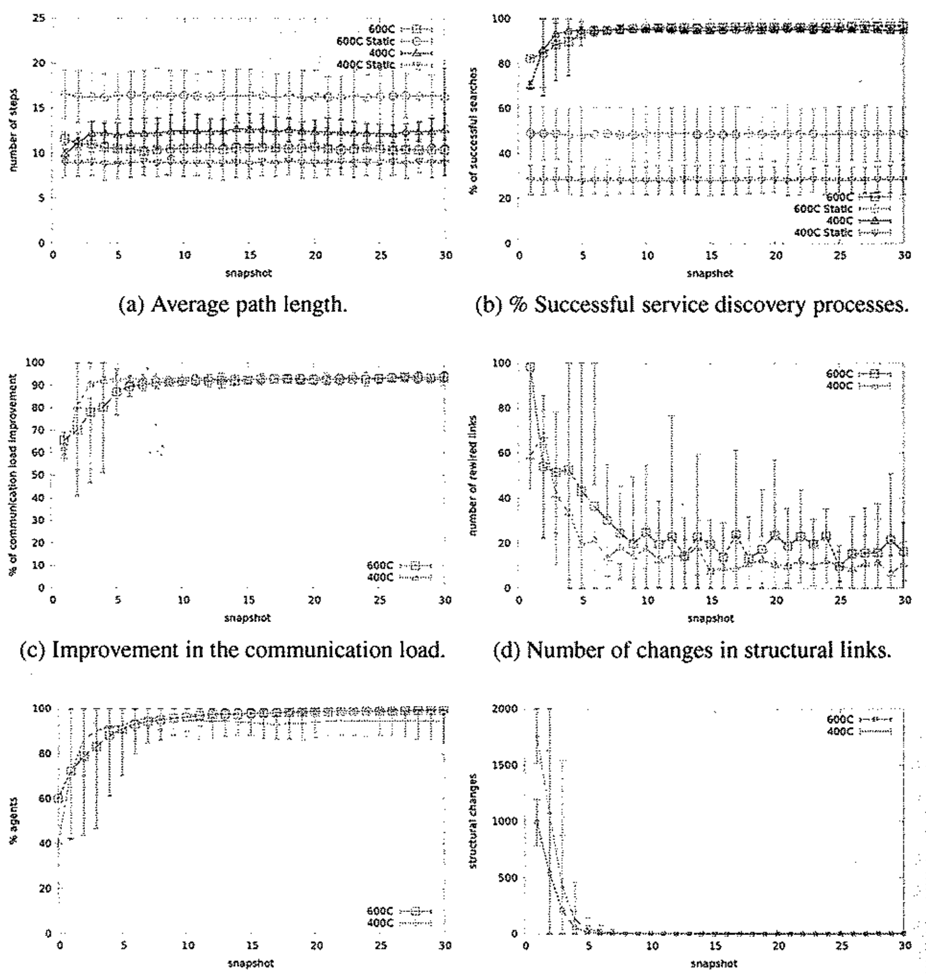
and the service. A query was successfully solved when an agent that offered a similar service (i.e., the degree of semantic match between the semantic service descriptions was over a threshold $\varepsilon = 0.75$) was found before the TTL ($TTL = 100$). Query distribution in the system was modeled as an exponential distribution. In the experiments, we made a snapshot of all the metrics every 10,000 queries in order to see the evolution of the system.

Specifically, the tests focused on a set of metrics that are meaningful for the analysis of the performance of the system and for the effects on the service discovery process when agents incorporate self-organization and cooperation mechanisms [9]. These metrics are:(i) average number of steps required to locate an appropriate agent that solves a query; (ii)% of queries that are solved before the TTL; (iii) communication load improvement (i.e., the system improvement comparing the number of exchanged messages during the service discovery process when adaptation mechanisms are exploited with respect to the number of exchanged messages when the network is not self-organized); (iv) Structural adaptive cost (i.e., the number of structural changes required to adapt the system: number of structural relations between agents that have changed during the service discovery process and number of agents that clone or leave the system during the service discovery process); (iv) evolution of cooperation in the system; (v) number of broken relationships as consequence of social plasticity.

In the experiments, the costs and benefits of the actions were: $q = 0.15$ (cost of forwarding action), $p = 0.5$ (benefit of providing a service), $r = 0.5$ (cost of asking for a service), and $sq = 0.30$ the reward of the forwarding action. The results were evaluated considering two different scenarios. In one scenario the number of initial cooperators in the network was 600. In the other scenario the initial number of cooperators was 400. We compare the results that we obtained using the proposed mechanisms with the results obtained in static networks.

Figure 2a shows the average number of steps in successful searches. In the case of 600 initial cooperators, the introduction of self-organization mechanisms considerably decreased the number of steps required to reach a suitable provider agent if we compare them with the steps required when the network was static and incentives were not considered. In the other scenario, where the initial number of cooperators was 400, the average number of steps increased if we compare it with a static network. This has sense since in static networks with 400 cooperators the only successful queries were those that were solved in the neighborhood of the agent that generated the query. In a dynamic network where mechanisms to adapt to the service demand and to promote cooperation were used, the number of queries solved is higher due to a query that could not be solved by a nearby agent could reach other agents that were far away, therefore the number of steps increased.

Figure 2b shows the effects of using self-organization and cooperation mechanisms on the success of the service discovery process. In general, the percentage of queries that ended successfully was improved with the inclusion of the mechanisms. This improvement was achieved in the first snapshots where the self-organization and the promotion of cooperation played an important role. The evolution of cooperation can be observed in Figure 2e.



(a) Average path length.

(b) % Successful service discovery processes.

(c) Improvement in the communication load.

(d) Number of changes in structural links.

(e) % of cooperative agents.

(f) Numer of social plasticity actions.

Fig. 2. Effects of the combination of self-organization and cooperation mechanisms in the service discovery process. We considered two scenarios. In one scenario the number of initial cooperators in the network was 600. In the other scenario the initial number of cooperators was 400.

Figure 2c shows the improvement in the communication load. This measures the system improvement comparing the number of exchanged messages during the service discovery process when adaptation mechanisms are exploited with respect to the number of exchanged messages when the system is not self-organized. In general, it can be observed that the number of messages required in a service discovery process was reduced.

Figure 2d shows the number of structural relations that agents change in order to improve the system performance. The results show that the self-organization mechanism

allowed agents to be aware that there was a change in the service demand; therefore, they realized that structural changes were needed to adapt some of their links according to a new service demand. This fact can be observed in the first five snapshots where the number of rewired links was greater than in the following snapshots. In the configuration of 600 initial cooperators, the number of rewired structural relations in the initial snapshots was greater than in the configuration with 400 initial cooperators. This is due to in the network with more cooperator agents, agents had more information about the service demand since more searches ended successfully. As the number of cooperators in the network increased this difference between both initial configurations was reduced.

Figure 2f shows the number of structural changes that were done by the agents in order to isolate non-cooperative agents. In the first 10 snapshots, the number of structural changes was higher than the following snapshots. This fact is because structural changes were used by the agents when the majority of their neighbors were non-cooperators. After the first 10 snapshots, cooperation emerged and in order to maintain the network connected agents only used incentives to promote cooperation.

7 Conclusions

Our proposal addresses the problem of self-organization and cooperation of agents in order to deal with the service discovery when service demand changes or selfish agents appear in open societies. Agents include *self-organization* mechanisms in order to adapt the underlying structure of the agent society to changes in the service demand. Agents replace their relationships with neighbors that are not being used with new structural relations with acquaintances. Agents are also able to estimate whether or not they are playing an important role in the society through the calculation of their structural similarity. With this information, agents decide to remain, leave, or clone themselves in order to adapt the population to the service demand. We also include the use of incentives and social plasticity in order to promote and maintain *cooperation* in the society. Incentives influence the behavior of other agents and promote cooperation. Moreover, social plasticity allows agents to change their structural relations based on the degree of cooperation of their neighbors. We evaluated the integration of the proposed mechanisms through a set of experiments taking into account the effects on the average path length, the percentage of successful searches, the improvement in communications, and the cooperative behavior. The results show that the proposed mechanisms improve the service discovery performance increasing the success, reducing the path length, and increasing the number of cooperators in the agent society.

Acknowledgements. This work is supported by TIN2011-27652-C03-01 and TIN2012-36586-C03-01 projects and FPU AP2008-00601 granted to E. del Val.

References

1. Adamic: Zipf's law and the internet. Glottometrics 3, 143–150 (2002)
2. Basters, U., Klusch, M.: RS2D: Fast adaptive search for semantic web services in unstructured P2P networks. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 87–100. Springer, Heidelberg (2006)
3. Bianchini, D., Antonellis, V.D., Melchiori, M.: Service-based semantic search in p2p systems, pp. 7–16. IEEE Computer Society, Los Alamitos (2009)
4. Del Val, E., Rebollo, M., Botti, V.: Enhancing Decentralized Service Discovery in Open Service-Oriented Multi-Agent Systems. In: JAAMAS, pp. 1–30 (2013)
5. Doran, J.E., Franklin, S., Jennings, N.R., Norman, T.J.: On cooperation in multi-agent systems. The Knowledge Engineering Review 12, 309–314 (1997)
6. Eguiluz, V.M., et al.: Cooperation and emergence of role differentiation in the dynamics of social networks. American Journal of Sociology 110, 977 (2005)
7. Hauert, C., Traulsen, A., Brandt, H., Nowak, M.A., Sigmund, K.: Via Freedom to Coercion: The Emergence of Costly Punishment. Science 316(5833), 1905–1907 (2007)
8. Huberman, B.A., Adamic, L.A.: The nature of markets in the www. Technical report (1999)
9. Kaddoum, E., Raibulet, C., Georgé, J.-P., Picard, G., Gleizes, M.-P.: Criteria for the evaluation of self-* systems. In: Proc. of the SEAMS, pp. 29–38 (2010)
10. Kontominas, D., Raftopoulou, P., Tryfonopoulos, C., Petrakis, E.G.M.: $\mathcal{DS}^4$: A distributed social and semantic search system. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) ECIR 2013. LNCS, vol. 7814, pp. 832–836. Springer, Heidelberg (2013)
11. Nowak, M.A.: Five Rules for the Evolution of Cooperation. Science 314(5805), 1560–1563 (2006)
12. Nowak, M.A., Sigmund, K.: Evolution of indirect reciprocity by image scoring. Nature 393(6685), 573–577 (1998)
13. Ouksel, A., Babad, Y., Tesch, T.: Matchmaking software agents in b2b markets. In: Proc. of the 37th Annual Hawaii International Conference on System Sciences, HICSS 2004 (2004)
14. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: State of the art and research challenges. Computer 40, 38–45 (2007)
15. Sigmund, K.: Sympathy and similarity: The evolutionary dynamics of cooperation. Proceedings of the National Academy of Sciences 106(21), 8405–8406 (2009)
16. Strang, D., Macy, M.W.: In search of excellence: Fads, success stories, and adaptive emulation. American Journal of Sociology 107, 147 (2001)
17. Wang, L.: Sofa: An expert-driven, self-organization peer-to-peer semantic communities for network resource management. Expert Syst. Appl. 38(1), 94–105 (2011)
18. Zhong, M.: Popularity-biased random walks for peer-to-peer search under the square-root principle. In: Proc. of the 5th International Workshop on Peer-to-Peer Systems (2006)

Agent Participation in Context-Aware Workflows

José M. Fernández-de-Alba, Rubén Fuentes-Fernández, and Juán Pavón

Facultad de Informática de la Universidad Complutense de Madrid
Avda. Complutense, s/n. 28040 Madrid, Spain
{jmfernandezdealba,ruben,jpavon}@fdi.ucm.es

Abstract. Smart environments assist users in the activities taking place in their influence areas. These activities are occasionally part of workflows and have multiple physical or computational participants playing different roles. The system has to monitor the development of the activities, and to take the necessary actions for them and the workflow to reach a certain end. These tasks largely depend on obtaining data from sensors, inferring the proper information from those data, and using actuators consequently. The context-aware paradigm pursues helping to develop these applications. In certain situations, computational participants need to take complex decisions. Agents are a convenient way to describe entities with sophisticated and flexible behaviors that adapt to complex and evolving environments and collaborate to reach certain goals. Most works in this area make use of agents for infrastructure-related or domain-specific tasks, whereas this research proposes patterns to integrate agents on top of an existing context-aware architecture in order to exploit its capabilities to improve functionality. A case study on guiding a user along a path illustrates this approach.

Keywords: software agent, software architecture, context-awareness, workflow management, ambient intelligence, ambient-assisted living.

1 Introduction

Ambient Intelligence (AmI) makes use of different technologies (e.g., location, identity, movement, face or speech recognition), integrated into a myriad of devices. These information sources combined are rich enough to support context-aware systems that adaptively solve high-level tasks minimizing the need of explicit interaction with users. Such tasks frequently involve activity recognition and assistance in business workflows. These workflows may involve multiple actors, including systems and users, which require coordination. The adaptation here implies performing tasks according to the actual setting regarding, for instance, resources and user configurations. A correct evaluation of the setting relies on systems making a proper interpretation of the available data, and using inferred context to fill in the missing information needed by their services.

The previous adaptation requires an infrastructure that solves abstract representations of existing tasks into runtime processes that drive the sensors and

J.-S. Pan et al. (Eds.): HAIS 2013, LNAI 8073, pp. 31–40, 2013.