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., Finiel, L., Gonsa-Sanz, J.J.: Requirements for an intelligent ambien-
tent assisted living application for parkinson patients. In: Corchado, J.M., Baño, J.,
(2013)

2. Bickmore, T.W., Schulman, D., Sidner, C.L.: A reusable framework for health coun-
seling dialogue systems based on a behavioral medicine ontology. Journal of Biomed-
ical Informatics 44(2), 183-197 (2011)

3. Bokhari, A.C., Kish, V.G.: Integration of a secure type-2 fuzzy ontology with a
multi-agent platform: a proposal to automate the personalised flight ticket booking

4. García-Magariños, J.: Defining and transforming models of parkinsons patients in the
development of assisted-living multi-agent systems with INGENIAS. In: Corchado, J.M.,
(2013)

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., Loidlacher, B., Grothe, J.S., Martone, M.E.: Towards a formalization of
disease-specific ontologies for neuroinformatics. Neural Networks 16(9), 1277-1293
(2003)

7. Riesco, D., Reus, F., López-Vallverdu, J.A., Canamero, M., Escolano, S., Meocci,
knowledge to support clinical decisions for chronically ill patients. Journal of
Biomedical Informatics 45(3), 429-446 (2012)

assisted supply chains negotiations. Computers & Industrial Engineering 65(1), 2-11
(2013)

9. Williams, 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,
Camp 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 compo-
nents, 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 require-
ments [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 mech-
nisms 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 de-
mand to maintain or improve the performance of the service discovery process. The
mechanisms that promote cooperation when there are self-interested agents in the soci-
ety are based on local structural changes and the use of incentives.
Formal Model

The problem of cooperation during the service execution in the service-oriented system can be formulated as follows. Let us consider a network of service providers (SPs) where each SP has a set of available services. The goal of the network is to provide a set of services to the customers in an efficient manner. To achieve this goal, the network must coordinate the execution of services among the SPs. The coordination can be achieved through the use of a communication mechanism. The mechanism ensures that the services are executed in the correct order and the results are collected in the correct format. The communication mechanism is implemented using a distributed algorithm. The algorithm is designed to be robust and scalable.

The correctness of the communication mechanism is ensured by the use of a formal model. The model is a finite automaton that represents the execution of the algorithm. The automaton is defined by a set of states and transitions. Each state represents a configuration of the network, and each transition represents a change in the configuration. The model is used to verify the correctness of the algorithm. The verification is performed using a formal verification tool.

The formal model can be used to prove the correctness of the algorithm. The correctness of the algorithm is defined by a set of properties. The properties are defined using a formal specification language. The specification is used to describe the expected behavior of the algorithm. The specification is verified using a formal verification tool.

The formal model can be used to optimize the algorithm. The optimization is performed by analyzing the model and identifying the bottlenecks. The bottlenecks are identified by analyzing the transitions in the automaton. The transitions with the highest frequency are the bottlenecks. The optimization is performed by redesigning the transitions to reduce the frequency of execution.

The formal model can be used to simulate the algorithm. The simulation is performed by running the model on a computer. The simulation is used to test the algorithm and to identify any errors. The errors are identified by analyzing the execution of the model. The errors are corrected by redesigning the algorithm.

The formal model can be used to verify the security of the algorithm. The security is defined by a set of properties. The properties are defined using a formal specification language. The specification is used to describe the security requirements of the algorithm. The specification is verified using a formal verification tool.

The formal model can be used to verify the performance of the algorithm. The performance is defined by a set of metrics. The metrics are defined using a formal specification language. The specification is used to describe the performance requirements of the algorithm. The specification is verified using a formal verification tool.
## Chapter 5: Insurance and Capital Markets

### Table 5.1: Financial Instruments

<table>
<thead>
<tr>
<th>Financial Instrument</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stocks</td>
<td>Apple</td>
</tr>
<tr>
<td>Bonds</td>
<td>Treasury Bonds</td>
</tr>
<tr>
<td>Commodities</td>
<td>Oil</td>
</tr>
<tr>
<td>Derivatives</td>
<td>Options</td>
</tr>
</tbody>
</table>

### Subchapter 5.1: The Role of Financial Markets

In the context of the financial markets, the primary function is to facilitate the exchange of financial instruments. This allows for the efficient allocation of resources and the pricing of assets. The market participants include investors, issuers, and traders.

### Subchapter 5.2: Regulation and Oversight

The regulatory framework plays a crucial role in ensuring the stability of financial markets. This includes oversight by financial regulatory bodies, which monitor market operations and enforce compliance with regulations.

---

### Footnotes

2. This section is based on general principles and may not apply in specific legal or regulatory contexts.

---

### References


---

### Additional Resources

- [Regulatory Bodies: An Overview](https://www.fca.org.uk/about-us/what-we-regulate/financial-regulation/what-is-regulation)

---

*This document is a draft and is subject to further editing.*
Experiments

...
Agent Participation in Context-Aware Workflows

Jesús M. Fernández-de-Alba, Rubén Puentes-Fernández, and Juan Pavón

Facultad de Informática de la Universidad Complutense de Madrid
Avda. Complutense, s/n, 28040 Madrid, Spain
{jesusfernandezdealba, ruben, jppavon}@fciencias.ucm.es

Abstract. Smart environments assist users in the activities taking place in their influence area. 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 behaviours that adapt to complex and evolving environments and collaborate to reach certain goals. Most works in this area made 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 content 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

References