

A Rule-Based Approach to Automatic Service Composition

Maria J. Santofimia, University of Castilla-La Mancha, Spain

Xavier del Toro, University of Castilla-La Mancha, Spain

Felix J. Villanueva, University of Castilla-La Mancha, Spain

Jesus Barba, University of Castilla-La Mancha, Spain

Francisco Moya, University of Castilla-La Mancha, Spain

Juan C. Lopez, University of Castilla-La Mancha, Spain

ABSTRACT

The incapability to foresee or react to all the events that take place in a specific environment supposes an important handicap for Ambient Intelligence systems, expected to be self-managed, proactive, and goal-driven. Endowing such systems with capabilities to understand and reason about context seems like a promising solution to overcome this hitch. Supported on the service-oriented paradigm, composing rather than combining services provides a reasonable mean to implement versatile systems. This paper describes how systems for Ambient Intelligence can be improved by combining automatic service composition and reasoning capabilities upon a distributed middleware framework.

Keywords: Ambient Intelligence Systems, Automatic Service Composition, Context, Distributed Middleware Framework, Service-Oriented Paradigm

1. INTRODUCTION

Self-management, pro-activeness, dynamism and goal-driven are some of the most challenging requirements that have to be faced when developing systems for Ambient Intelligence. Indeed, devising a strategy to address these issues has been one of the main concerns for researchers in the field for the last decades. The majority of the proposed solutions simply

address isolated aspects of the overall problem. Agent-based approaches are proposed as a mean to deal with specific and repetitive tasks, most of them targeted to the disabled people (Moraitis & Spanoudakis, 2007). Agent-based solutions have also been used to model the environment as a group of distributed services, provided by heterogeneous devices, as in Cabri, Ferrari, Leonardi, and Zambonelli (2005). Planning approaches have been proposed as a mean to enact goal-oriented behaviors (Amigoni, Gatti, Pincioli, & Roveri, 2005) while addressing device heterogeneity. Solutions based on sup-

DOI: 10.4018/jaci.2012010102

porting dynamic service composition have also been widely considered and implemented by means of different technologies, as described in Urbiet, Barrutieta, Parra, and Uribarren (2008). Nevertheless, adopting any of these approaches in an isolated manner is far from being the solution to address the ambiguity, uncertainty and incompleteness that characterize context information.

Systems for Ambient Intelligence have to deal with the service dynamism derived from the device mobility that characterizes such context. This feature makes unfeasible to count on pre-coded responses to ambient events since the available services cannot be known beforehand. Even when knowing a static set of services always available, the system response capabilities should not be constrained to these available services. On the contrary, responses should be automatically composed out of the available services, and based on a general description of what is required.

Automatic service composition has already been addressed from the web services perspective, as described in Medjahed and Bouguettaya (2005) or Maamar, Mostefaoui, and Yahyaoui (2005). However, although labeled as service composition, it is more appropriate to refer to this approach as service combination. It is important to remark the difference between service composition and combination. When composing services, it does not suffice with connecting a service output with a service input, as it is described in a static recipe. This task falls into what we consider service combination. Meanwhile, service composition requires the understanding of what services are capable of, and when required, the assignment of the service to a different purpose than originally intended. Nevertheless, supporting automatic service composition involves a deep understanding of the service semantics and the domain knowledge.

Domain models are intended to fill the gap between the perceived information and the meaning it has. Among the many different context modeling techniques (Strang & Linnhoff-Popien, 2004) ontologies stand out

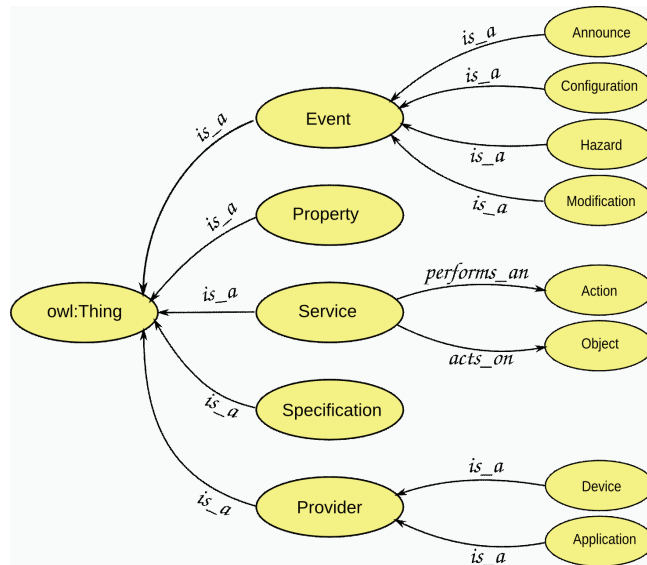
to be the more compelling approach to address domain models for Ambient Intelligence (Wang, Zhang, Gu, & Pung, 2004). The importance of providing an ontology for Ambient Intelligence is twofold. On the one hand, it unifies the vocabulary used to describe the domain knowledge. Moreover, interoperability among the different architectural elements of an Ambient Intelligence system draws on this unified vocabulary. On the other hand, providing an ontology about the domain knowledge makes it possible the generation of large bodies of knowledge logically related (Taylor, Matuszek, Klimt, & Witbrock, 2007; Mascardi, Locoro, & Rosso, 2010).

The work presented here provides a context ontology for supporting, not just the domain modeling but the reasoning and service composition tasks. Opting for a context reasoning technique over others is not easy, due to the wide range of available approaches (Bikakis, Patkos, Antoniou, & Plexousakis, 2007). The strengths and weaknesses of the different approaches depend on the target context and the technique used for context modeling. The proposal presented here, entrusts the context reasoning and inference tasks to a rule-based engine, mainly due to the proved performance of widely known engine rules, as well as for being compliant with ontology languages.

Finally, in seeking self-managed systems, the middleware framework plays an essential role in supporting both the aforementioned solutions, such as ontologies and reasoning techniques, and also in simplifying and abstracting the complexity and heterogeneity of both, network and device technologies.

The remainder of this paper is structured as follows. Section 2 undertakes the description of the semantic model for Ambient Intelligence, by depicting the most relevant aspects of the ontology. Section 3 advocates for a multi-agent system approach in order to inject certain features to the envisioned system, such as a goal-driven behavior, pro-activeness, or autonomy. Section 4 goes through the rule-based system, describing the fundamental principles of the reasoning systems, and how it is related to the

Figure 1. Ontology for a surveillance system



context ontology. Section 5 provides a methodology for evaluating the quality of the solutions proposed. Finally, Section 6 gathers the most relevant conclusions of the proposed approach.

2. AN ONTOLOGY FOR AMBIENT INTELLIGENCE CONTEXTS

Surveillance systems for enclosed spaces provide here the domain knowledge on which this work has been tested. Surveyed environments tend to be populated with a wide variety of electronic devices, such as video cameras, audio recorders, and different sort of sensors and actuators, distributed all around the area. Service providers fall into two groups: (1) the major group encompasses those services provided by the heterogeneous devices present in the environment and; (2) the second group comprises those services provided by applications running on computers, such as face detection and identification systems, tracking systems, or data gathering systems, just to name a few.

A detailed security policy describes the system responsibilities in terms of goals to be

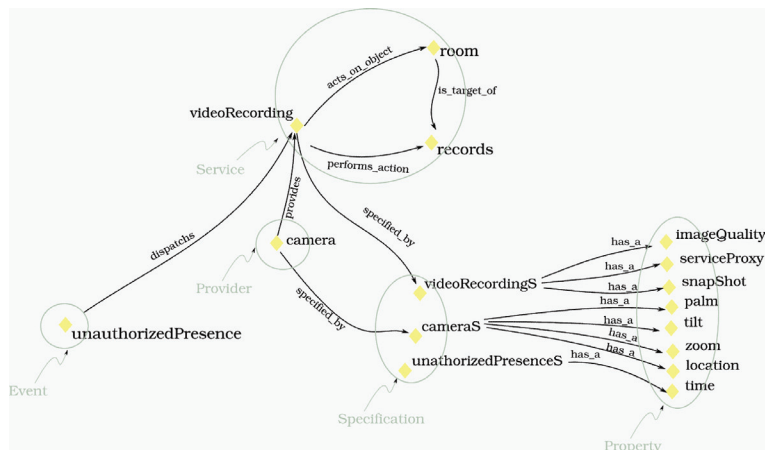
achieved or maintained. Events taking place in the context generate disturbances in the state of the environment. Some of these situations are contemplated in the security policy, where some hints about how to proceed are suggested. Contrary to what it might be expected, these events are not just constrained to hazards. In order to provide the system with understanding capabilities, a thorough characterization of events is required.

At a glance, Figure 1 depicts those entities that in one way or another play an important role in the Ambient Intelligence system. Encompassing these entities into a semantic model, and share it with the middleware, the system, the knowledge base, and the reasoning engine, is considered here as an essential requirement for automatic service composition.

However, the hierarchical representation of the domain model concepts is meaningless the relationships among these entities are not established.

For the sake of understanding, Figure 2 describes how the ontology is employed to model the context, under the circumstances of an unauthorized presence alarm. As depicted

Figure 2. Context ontology for an unauthorized presence alarm



in the figure, the yellow diamonds wrapped in circles represent the instances of a certain class, existing in the context. Furthermore, class properties are translated into the context model as relationships among instances, as for the Figure 2, the ontology property labeled as *dispatch*, models how an event occurrence dispatches the activation of one or several services. Finally, the omnipresent *Specification* class, associates valued name properties to the existing instances, featuring and enriching with meaning the instances and relationships in the context model.

Establishing the domain model sets the basis on top of which rests the constituent parts of the distributed architecture presented here, as detailed in the following sections. Additionally, the generation of the domain model has been accomplished by using OWL (McGuinness & van Harmelen, 2009). To this end, the tool Protégé has been used.

3. THE MULTI-AGENT SYSTEM

The implementation of agent-oriented approaches for building complex systems is progressively gaining credit (Jennings, 2001). The decomposability of complex systems simplifies the isolation of the composing subsystems, so

that each of them can be individually addressed. The collaboration aspect among agents is also an important asset in arguing for an agent-based approach.

The agent-oriented approach considers different sort of implementations as stated in Wooldridge (2000). Logic based architectures (deductive agents), reactive architecture (reactive agents), layered architectures (hybrid agents), and practical reasoning architectures (Belief-Desire-Intention agents). Among all these alternatives, the Belief-Desire-Intention model (BDI) has proved to be a powerful framework for building rational agents (Wooldridge, 2000) mainly because it offers the possibility to describe the agent behavior depending on the goals to be met, the knowledge it holds about the context, and a set of plans that assist the agent towards its goals. Indeed, the BDI model is intended to reproduce the process carried out when people take decisions to achieve a certain goal. Resorting to a BDI model of agency is going to enact the Ambient Intelligence system with the capability to supervise the environment and take part in it.

The multi-agent system implementation is supported in Jadex (Pokahr, Braubach, & Lamersdorf, 2005), an agent-oriented reasoning engine for rational agents. Instead of using formal logic descriptions, Jadex proposes the

use of two commonly known languages, such as Java and XML. The BDI agent is modeled by mapping the concepts of beliefs into Java objects, while desires and intentions are mapped into procedural recipes coded in Java that the agent carries out in order to achieve a goal. Additionally, the reasoning capability is highly dependent on the domain model used to describe the agent context. As mentioned in the previous section, this proposal resorts to OWL (McGuinness & van Harmelen, 2009) for generating the domain model that can be directly managed by the Jadex framework. The domain model states the context information that agents are able to handle and understand, which, in the end, determines their behavior.

3.1. Setting the Basis for the Composition Task

Despite the fact that agents have been a widely adopted solution for supporting service composition in the field of Web Services, its implementation in other fields of distributed systems has not been so prolific. On the contrary to Web Services that share a communication protocol such as SOAP, services deployed in pervasive environments are rarely capable of inter-working with the rest of services, mainly due to the heterogeneity of their implementation protocols. This drawback is addressed here by means of a middleware framework. In Villanueva, Villa, Moya, Santofimia, and Lopez (2009) a distributed object-oriented framework (DOBS, Distributed Object Based Services) is proposed, so as to overcome the problems appearing when the use of certain services involves managing different protocols. Adopting the proposed middleware architecture, and coupling it with the multi-agent system module proposed here, brings out an architecture that is able to be aware of all the services deployed, the properties that characterize them and how to interact with them.

It is the middleware responsibility to make it transparent the communication process, as well as to provide a common set of services that ease the development task and minimize

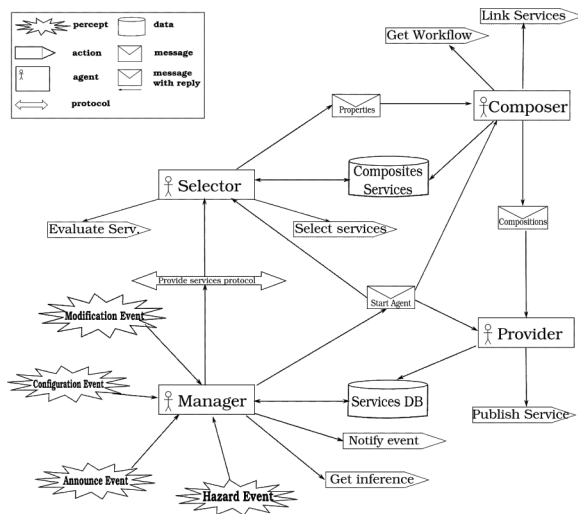
configuration efforts. Some of the most relevant common services provided in DOBS are an Abstract Service Discovery Framework (ASDF) (Villa, Villanueva, Moya, Rincon, Barba, & Lopez, 2008), security and QoS. Furthermore, this framework also holds an Information Model, specified as a taxonomy of services, with their attributes and events, for the sake of nomenclature homogeneity. Finally, it is also remarkable the fact that this framework supports a seamless integration of widely known subsystems such as UPnP, X10, or Bluetooth. Since it is out of the scope of this paper a deep description of this framework, refer to Villanueva, Villa, Moya, Santofimia, and Lopez (2009) for further details. To sum up, integrating the multi-agent system as a constituent part of the DOBS framework makes it possible to lay the foundations for heterogeneous distributed service composition.

3.2. The Multi-Agent System Service Composer (MASCS) Overview

Figure 3 depicts a comprehensive view of the proposed system. It is composed of four agents, known as *Manager*, *Selector*, *Composer*, and *Provider*. The adopted event-oriented middleware counts on several channels where services publish information. So, in order to be context-aware, the *Manager* agent subscribes to those channels. For instance, knowing that a restricted area is supervised by a specific presence sensor device, leads the manager agent to supervise the notifications published by this specific device in a specific channel. Along with the notification of activation, the presence sensor attaches to the message the time when the event took place and the identification of the presence sensor device that captured the event.

For being a BDI compliant MAS, agents are characterized with a set of beliefs, desires and intentions that under the Jadex framework are mapped to beliefs, goals and plans described using an XML-based nomenclature, as depicted in Listing 1. This code depicts how under the circumstances of an *unauthorized*

Figure 3. System overview diagram



Listing 1. Manager agent goals

```

<!--02. Intruder identification -->
<achievegoal name="intruder_identification">
  <parameter name="unauthorisedPresence" class="Event">
    <bindingoptions>beliefbase.eventTypes</bindingoptions>
  </parameter>
  <unique/>
</achievegoal>
<!--Plan intended to accomplish a biometric ID of the intruders -->
<plan name="get_biometric_ID">
  <body class="GetBiometricIDPlan"/>
  <trigger>
    <goal ref="intruder_identification"/>
  </trigger>
</plan>
    
```

presence, the goal *intruder identification* gets dispatched. The mean proposed to achieve this goal is by a *biometric identification*.

Obviously, hard-coding a plan to perform a biometric identification using a static set of services is of little help if those services are not available at the specific location where they are required. The idea behind the rule-based approach is to delegate to the reasoning engine the responsibility to automatically compose a service with the capability to perform some

specific actions upon some specific objects. Recall that services are presented in the ontology as the aggregation of actions and objects. Therefore, the Jadex plan will only contain a general description, adopting the form of a ternary element, containing the action to be performed, the object receiving the action and the expected result: (biometricIdentification, biometricFeature, person identity). The code listed in Listing 2 shows how the Jadex plans resorts to a rule-based planning strategy to seek

Listing 2. Manager agent plans

```

public void body()
{
    List P = new ArrayList<List>();
    List p = new ArrayList<List>();
    Planning pa = new Planning();
    P = pa.getPlan(P, "{identification}", "biometric feature", "{person identity}");
}

```

for the courses of actions that satisfy the ternary (action, object, result).

4. THE RULE-BASED REASONING SYSTEM

Among the existing implementations for rule-based reasoning engines, the proposal presented here counts on CLIPS (Riley, 2011). This engine is characterized for being quite light but powerful at the same time. Indeed, the work in Meditskos and Bassiliades (2007) presents a successful approach based on CLIPS for composing Web services. Nevertheless, the work presented here is not just constrained to Web services, but it is extended to run on any distributed service architecture.

The idea behind the adoption of a rule-based reasoning engine is to endow systems for Ambient Intelligence with the capability to automatically compose services. To this end, the reasoning engine needs to adopt the semantic model applied to the domain knowledge it holds. Regarding the contextual information, the manager agent that supervises the context asserts and retracts facts from the knowledge base whenever new events take place. The reasoning engine works on the basis of that information contained in the knowledge base, and an implementation of an HTN (Erol, Hendler, & Nau, 1994) planning-like rule. The assertion of a ternary fact of the form *action, object, result* dispatch the planning rule that seeks for the chain of actions that gets the expected result

from the given action and object. The planning rule implements Algorithm 1.

This algorithm strongly depends on how actions have been described. Adopting the approach presented in Santofimia, Fahlman, Moya, and Lopez (2010), actions are defined by describing the *state of the world* just before the action takes place and just afterwards.

The result of the rule dispatch is a new fact, the *plan* fact, being asserted (Listing 3). The plan is retrieved from the knowledge base and executed by the *composer* agent. Remark the fact that the adopted middleware add an abstraction layer that unifies the service instantiation service. The *composer* agent is relieved from having to deal with different protocols and technologies.

5. SYSTEM'S EVALUATION

The design of an evaluation methodology is a crucial issue to assess the degree of satisfaction of the end user, compare the performance of different alternatives and provide some feedback towards a process of continuous improvement and optimization. Nevertheless, this evaluation process entails a high degree of complexity since many different aspects, some of them highly subjective, are involved. The methodology here proposed is outlined in Figure 4.

According to Figure 4 the process of evaluation for each event generates a vector of attributes, which is mainly related to the services generated and their characteristics. This vector is then evaluated to obtain the fitness of

Algorithm 1. Planning (Π, A, O, R)

```

1:  $\pi = (A, O, R)$ 
2: if  $A$  is non-feasible then
3:   get all the actions  $A = (a_1, a_2, \dots, a_n)$  that have the
   same result  $A$ 
4:   while  $a_i$  is non-feasible do
5:     delete  $a_i$  from  $A$ 
6:   end while
7:   while only doable actions  $a_i$  does not have an
   equivalent target object do
8:     list all the objects  $Objects = (o_1, o_2, \dots, o_n)$  of
     action  $a_i$ 
9:     check if those  $o_i$  are equivalent to or can be  $O$ 
10:   end while
11:   Recursively call  $\pi = Planning(a_i, o_i, resultOf\ a_i)$ 
12: end if
13: Add  $\pi$  to  $\Pi$ 
14: Return  $\Pi$ 

```

the system's response by means of a rule-based approach, which is generated using human expertise and the end user expectations. Finally, the outcome of the evaluation process provides the grading of the system's response and generates statistics and time series for a deeper analysis.

The evaluation process described here can be implemented to be executed on-line in real-time, or off-line from captured data. Alternatively, it can be used at the design stage by implementing it in numerical simulations. The nature of the problem, which involves events and services, can be suitably addressed by means of discrete-event simulation tools (ref).

Furthermore the result of the evaluation process can also be very useful for other purposes such as condition monitoring. Fitness variations can alert of changes including device failures, vulnerabilities, environment and user habits variations that must be considered to redesign the system.

The key elements of the evaluation process are described in the following subsections.

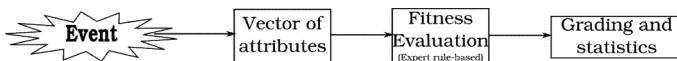
5.1. Vector of Attributes

The vector of attributes must gather all the valuable information regarding the services provided for a certain event. In the system presented it is particularly important to assess the benefits that composite services will bring. The vector of attributes could consist of the following elements: event start time; event duration; type of event; services provided including the type of service (basic or composite), the number of basic or composite services provided, the service response times, and other particular characteristics.

5.2. Fitness Evaluation

Once the vector of attributes has been generated for a certain event, the system response has to be evaluated. The approach adopted here is to create a set of rules designed to establish the criteria and quantify the fitness of the system response. According to the type of event, there are certain user expectations to be fulfilled such as: response time, information provided by the system and finally how successfully the situation has been handled. The various aspects to be graded by the rules for each type of event are: number of basic services provided; the

Figure 4. An Evaluation Process



Listing 3. Action class in CLIPS

```

(defclass ACTION
  (is-a USER)
  (slot ID
    (type SYMBOL)
    (default ?NONE))
  (slot roles
    (type SYMBOL))
  (slot before-context
    (default ?NONE))
  (slot thorough-context
    (default ?NONE)))
  
```

number of composite services provided; the ratio between composite services and basic services; the response time for each service; the usefulness of the services provided according to the type of event.

The grades given according to the rules have to be properly weighted and added to obtain the fitness of the response given by the system in response to the event.

5.3. An Example

A simplified example is presented here to illustrate the evaluation process described. The example is a surveillance application, where the scenario considered is a room containing a presence sensor and a camera. The image of the camera can be processed by a face recognition software application.

In this example there is only one type of event to be considered, human presence in the room. Regarding the services provided, the basic ones are the state of the presence sensor, the video streaming from the camera and the face recognition output; while the composite services are combinations of them according

to the rule-based system implemented for automatic service composition. In this case, two composite services have been considered. One of them notifies the security staff when the sensor detects presence and automatically provides the video images. The other composite service also launches the face recognition application and provides its output.

For each event a vector of attributes is generated, which contains the most relevant information about the services provided. Then the fitness evaluation is performed by grading each individual service according to a set of rules that establish the level of availability and accomplishment of the service. Finally, the overall fitness of the system response is obtained by multiplying the grades of each service by a weight and adding them all, as in equation 1. The weight of each service is established according to how satisfied is the end user is. Composite services have higher weight than basic ones since they better fulfill user needs.

$$F = \frac{\sum_{n=1}^N \omega_n \cdot g_n}{\sum_{n=1}^N \omega_n} \cdot 100 \tag{1}$$

Figure 5. Simulation results

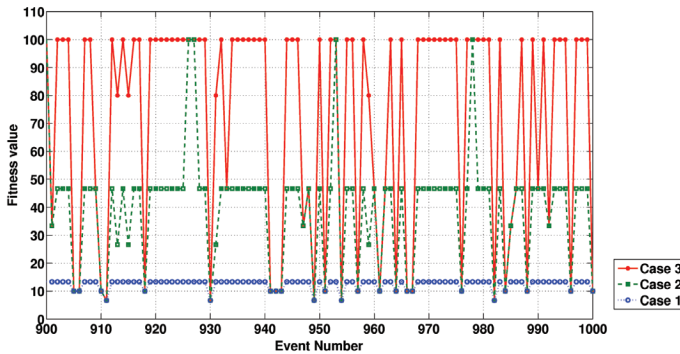


Table 1. Simulation results

CASE	Mean of F	Std dev of F	% of Comp Srv.
1	12.55	1.937	0
2	42.31	20.33	75.4
3	77.21	35.93	75.4

Where F is the fitness and N is the total of number of services provided, and are the weight and grade of the nth service respectively. The value of F is normalized in the range 0 to 100. The example described has been simulated using probabilistic distributions to model the availability, rate of success and response time of the services previously described. Three different cases have been considered: in the case 1 only basic service are provided; in the case 2 basic and composite services are provided. The face recognition application has a rate of success of 10%; in the case 3 basic and composite services are provided. The face recognition application has a rate of success of 90%.

A simulation with 1000 events has been run for each case. Figure 5 shows the result obtained for the three different cases.

Table 1 shows the mean value and standard deviation of the fitness for each case. Additionally, it shows the percentage of events where composite services are provided.

6. CONCLUSION

This paper points out some of the most common difficulties found when developing systems for pervasive or ambient intelligence environment. The lack of a common framework supporting the service development and the limited functionalities provided by these services are considered to be two of the key issues towards self-sufficient systems. This work thinks of the automatic service composition as an essential step for being able to develop systems with the capability to take decisions, respond to events, and handle common day-life situations.

To this end, a combination of multi-agent system and a rule-based reasoning engine is proposed as an approach to automatic service composition. Nonetheless, this combination requires a middleware platform, providing the foundations for communications, discovering, activation, parsing, QoS, and so on. On the basis of this middleware framework, and modeling the domain problem by means of an ontology,

the multi-agent system deals with events taking place in the environment by means of the services deployed in the environment as well as their compositions that have been automatically composed and evaluated by means of the rule-based reasoning engine.

REFERENCES

- Amigoni, F., Gatti, N., Pinciroli, C., & Roveri, M. (2005). What planner for ambient intelligence applications? *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, 35(1), 7–21.
- Bikakis, A., Patkos, T., Antoniou, G., & Plexousakis, D. (2007). A survey of semantics-based approaches for context reasoning in ambient intelligence. *Proceedings of the IEEE*, 11(1), 14–23.
- Cabri, G., Ferrari, L., Leonardi, L., & Zambonelli, F. (2005). The LAICA project: Supporting ambient intelligence via agents and ad-hoc middleware. In *Proceedings of the 14th International Conference Enabling Technologies* (pp. 39-46). Washington, DC: IEEE Computer Society.
- Erol, K., Hendler, J., & Nau, D. (1994). HTN planning: Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence* (Vol. 2).
- Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4), 35–41. doi:10.1145/367211.367250
- Maamar, Z., Mostefaoui, S. K., & Yahyaoui, H. (2005). Toward an agent-based and context-oriented approach for web services composition. *IEEE Transactions on Knowledge and Data Engineering*, 17(5), 686–697. doi:10.1109/TKDE.2005.82
- Mascardi, V., Locoro, A., & Rosso, P. (2010). Automatic ontology matching via upper ontologies: A systematic evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 22, 609–623. doi:10.1109/TKDE.2009.154
- McGuinness, D. L., & van Harmelen, F. (2009). *OWL web ontology language*. Retrieved March 15, 2009, from <http://www.w3.org/TR/owl-features/>
- Meditskos, G., & Bassiliades, N. (2007). A semantic web service discovery and composition prototype framework using production rules. In *Proceedings of the European Semantic Web Symposium Workshop OWL-S: Experiences and Future Developments*, Innsbruck, Austria.
- Medjahed, B., & Bouguettaya, A. (2005). A multilevel composability model for semantic web services. *IEEE Transactions on Knowledge and Data Engineering*, 17(7), 954–968. doi:10.1109/TKDE.2005.101
- Moraitis, P., & Spanoudakis, N. (2007). Argumentation-based agent interaction in an ambient-intelligence context. *IEEE Intelligent Systems*, 22(6), 84–93. doi:10.1109/MIS.2007.101
- Pokahr, A., Braubach, L., & Lamersdorf, W. (2005). Jadex: A BDI reasoning engine. In R. Bordini, M. Dastani, J. Dix, & A. E. F. Seghrouchni (Eds.), *Multi-agent programming* (Vol. 15, pp. 149-174). New York, NY: Science + Business Media.
- Riley, G. (2011). *CLIPS: A tool for building expert systems*. Retrieved from <http://clipsrules.sourceforge.net/>
- Santofimia, M. J., Fahlman, S. E., Moya, F., & Lopez, J. (2010). A common-sense planning strategy for ambient intelligence. In *Proceedings of the 14th International Conference on Knowledge-based and Intelligent Information and Engineering Systems* (pp. 193-202).
- Strang, T., & Linnhoff-Popien, C. (2004). *A context modeling survey*. Retrieved from <http://citeseer.ist.psu.edu/strang04context.html> =0pt
- Taylor, M., Matuszek, C., Klimt, B., & Witbrock, M. (2007). Autonomous classification of knowledge into an ontology. In *Proceedings of the Twentieth International FLAIRS Conference* (pp. 140-145).
- Urbietta, A., Barrutieta, G., Parra, J., & Uribarren, A. (2008). A survey of dynamic service composition approaches for ambient systems. In *Proceedings of the Ambi-Sys Workshop on Software Organisation and Monitoring of Ambient Systems* (pp. 1-8). ICST, Brussels, Belgium, Belgium: ICST.
- Villa, D., Villanueva, F., Moya, F., Rincon, F., Barba, J., & Lopez, J. (2008). Asdf: an object oriented service discovery framework for wireless sensor networks. *International Journal of Pervasive Computing and Communications*, 4(4), 371–389. doi:10.1108/17427370810932141

Villanueva, F., Villa, D., Moya, F., Santofimia, M., & Lopez, J. (2009). A framework for advanced home service design and management. *IEEE Transactions on Consumer Electronics*, 55(3), 1246–1253. doi:10.1109/TCE.2009.5277984

Wang, X., Zhang, D., Gu, T., & Pung, H. K. (2004). Ontology based context modeling and reasoning using owl. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops* (pp. 18-22). Washington, DC: IEEE Computer Society.

Wooldridge, M. (2000). *Reasoning about rational agents*. Cambridge, MA: MIT Press.

Maria J. Santofimia received the degree of Technical Engineer in Computer Science in 2001 from the University of Córdoba (Spain); the Master's degree on Computer Security from the University of Glamorgan (Wales, UK) in 2003; and the degree of Engineer in Computer Science in 2006 from the University of Castilla-La Mancha (Spain). She received the PhD degree from the University of Castilla-La Mancha (Spain) in 2011. She is currently working as Teaching Assistant and she is member of the Computer Architecture and Networks Research Group (ARCO) at the Univeristy of Castilla-La Mancha.

Xavier del Toro García received the degrees of Technical Engineer on Industrial Electronics and Engineer in Automatic Control and Industrial Electronics in 1999 and 2002, respectively, from the Universitat Politècnica de Catalunya (Spain). He received the PhD degree from the University of Glamorgan (UK) in 2008. From September 2005 until October 2006 he was a Marie Curie Research Fellow at Politecnico di Bari (Italy). Since 2008 he is working as a researcher in the University of Castilla-La Mancha (Spain). His research interests include power electronics, renewable energy sources, energy storage systems and power quality.

Felix J. Villanueva received the Computer Eng. Diploma from the University of Castilla-La Mancha (UCLM) in 2001. In 2009 he obtained the PhD degree from the UCLM, where he is now working as associate professor. His research interests include wireless sensor networks, ambient intelligence and embedded systems.

Jesus Barba received the MS and PhD degrees in Computer Engineering Diploma from the University of Castilla-La Mancha (UCLM), Spain, in 2001 and 2008 respectively. He is working as Teaching Assistant with the Department of Information and Systems Technology since 2001. His research interests include SoCs, HW/SW integration and reconfigurable systems.

Francisco Moya received his MS and PhD degrees in Telecommunication Engineering from the Technical University of Madrid (UPM), in 1996 and 2003 respectively. In 1999 he joined UCLM and currently holds an Associate Professor position. His research interests include heterogeneous distributed systems, automatic service composition, electronic design automation, and their applications to large-scale domotics and systems-on-chip.

Juan C. Lopez received the MS and PhD degrees in Telecommunication (Electrical) Engineering from the Technical University of Madrid in 1985 and 1989, respectively. From September 1990 to August 1992, he was a Visiting Scientist in the Department of Electrical and Computer Engineering at Carnegie Mellon University, Pittsburgh, PA (USA). His research activities center on embedded system design, distributed computing and advanced communication services. From 1989 to 1999, he has been an Associate Professor of the Department of Electrical Engineering at the Technical University of Madrid. Currently, Dr. López is a Professor of Computer Architecture at the University of Castilla-La Mancha where he served as Dean of the School of Computer Science from 2000 to 2008. He is and has been member of different panels of the Spanish National Science Foundation and the Spanish Ministry of Education and Science, regarding the Information Technologies research programs. He is member of the IEEE and the ACM.