A Semantic Model for Actions and Events in Ambient Intelligence

Maria J. Santofimia[a], Scott E. Fahlman[b], Xavier del Toro[c], Francisco Moya[a], Juan C. Lopez[a]

[a] *Computer Architecture and Network Group, School of Computing Science. University of Castilla-La Mancha, Spain*
[b] *Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[c] *School of Industrial Engineering, University of Castilla-La Mancha, Ciudad Real, Spain*

Abstract

Event management and response generation are two essential aspects of systems for Ambient Intelligence. This work proposes handling these issues through the use of a semantic model for Ambient Intelligence which, under the umbrella of a philosophical and common sense optic, describes what actions and events are, how they are connected, and how computational systems should think about their meaning. This model entails an approach with which to both reason about and model context events and generate behavioral responses to those events, when necessary. The model supports this ad-hoc response generation by automatically composing services when those which are available do not meet the expected functionality (without requesting user intervention). An evaluation methodology is presented and illustrated with a case scenario, in which synthetic data has been generated to emulate events in which system response is expected. The evaluation of the system response is carried out on the basis of a vector of attributes.

Keywords: Common-sense reasoning, semantic model, multi-agent system.

1. Introduction

Self-management, pro-activeness, dynamism and goal-driven behavior are some of the most challenging requirements that have to be tackled when developing systems for Ambient Intelligence [1]. Despite the fact that devising a strategy with which to address these issues has been one of the main concerns for researchers in this field during the last decades, the majority of the solutions proposed to date remain far from the scenarios envisioned in [2].

An additional difficulty in developing Ambient Intelligence systems is that of having to cope with the wide range of device technologies present in these contexts, and the lack of continuity in device availability. These drawbacks have been addressed, with different levels of success, through the use of a wide variety of techniques, such as web services [3], middleware [4] [5], dynamic reconfiguration [6], agents [7], context modeling and reasoning approaches [8] [9], etc.

Nevertheless, in the authors' opinion, the autonomy expected from Ambient Intelligence systems can only be achieved by leveraging both common-sense knowledge and reasoning capabilities, rather than by focusing on implementational issues or the available technology. This work therefore starts from the premise that before tackling the specific requirements for Ambient Intelligence, it is first necessary to understand and model the nature of human agency. To this end, the approach followed in this work consists in the adoption and implementation, in the form of a computational model, of the conclusions concerning actions and events drawn from philosophical doctrine. The notion of event, action or service should not vary among systems, nor should they respond to the approach convenience used, as is evidenced in the models surveyed in [10]. Contrary to current practices, systems for Ambient Intelligence should be grounded on a solid base of a semantic model for actions and events.

Based on the enacted semantic model, and from a common-sense point of view, the problem of developing systems for Ambient Intelligence has to be tackled from two different perspectives: cognitive and behavioral. From the cognitive perspective, the problem can be addressed as an understanding problem. Comprehending a situation that takes place in a context might involve, for example, the inference of implicit, non-deterministic or delayed effects. A delayed effect of turning on a tap in a kitchen sink whose plug is in, will be a water overflow. From a behavioral perspective, the problem can be addressed as a planning problem of deciding what action to take in certain given circumstances. A common-sense strategy for planning and understanding, such as that presented in [11] would, therefore, appear to be the most compelling approach towards emulating the human-like rationality and reasoning capability.

In words of Doug Lenat et al. [12], the bottleneck of intelligent systems is reached when attempting to respond to unexpected situations, which, it should be noted, are the most common situations found in Ambient Intelligence contexts. The way in which people react to these unexpected situations provides an idea of the direction in which efforts should be addressed. Generally, when facing new situations, people tend to establish some similarities with past experiences, or resort to their general knowledge of how things work –the so called common-sense knowledge–, or even look for advice in books. Whatever the case may be, the authors of this paper believe that only Ambient Intelligence systems will be sufficiently flexible to support the scenarios envisioned in [2] when common-sense reasoning starts being considered as a structural part of such systems.

Understanding and modeling common-sense reasoning, in such a way that it can be automatically performed, is considered here as the key challenge that, once achieved, would allow systems for Ambient Intelligence to indeed be intelligent. In this paper, an implementation of the key issues[13] required for the automation of common-sense reasoning is also presented. Nevertheless, the main contribution of this work is founded on the proposition and formalization of a semantic model for actions and events in Ambient Intelligence, as a previous step towards common-sense reasoning and understanding. Section 2 provides a review of related and previous works. Section 3 describes the details of the proposed semantic model for actions and events. Section 4 provides the implementation details of the architectural approach adopted here. Section 5 presents a validation methodology to assess the goodness of the system response. Finally, Section 6 presents the conclusions drawn for this work.

2. Related works

The multidisciplinary nature of the Ambient Intelligence paradigm entails a state-of-the-art review task that must be addressed from the different areas of knowledge that play a role in the paradigm. The work in [14] provides an excellent starting point from which to obtain an overall view of the technologies involved in Ambient Intelligence.

Due to the fact that this work is framed in a comprehensive approach to Ambient Intelligence, this section is intended to revise the fields of knowledge to which the propose solution is making contributions to. Besides, the justification for those contributions to be presented along with the proposed semantic model is due to the fact that they are a direct consequence of the proposed semantic model strengths.

2.1. Planning strategies

From the acting point of view, the planning theory contributes not only towards providing articulated responses by means of service composition, but also towards supporting the decision making of agents that exhibit goal-oriented behavior. However, literature in the field of planning for Ambient Intelligence is scarce, principally owing to the nonlinearity of problems that involve the exploration of a huge number of states. Some other features also contribute to this shortage: the nondeterministic effects of events, for example, make it impossible to determine whether picking up a slippery object will culminate in the object being dropped; those delayed effects, that occur a while after an event has taken place, lead people to foresee that if the kitchen sink has its plug in, then turning on the tap will cause the water to overflow. These are just a few of the features that make planning in Ambient Intelligence a non-trivial issue. Nevertheless, the need for planning strategies in Ambient Intelligence has already been stated in [15]. This work pays special attention to the device heterogeneity so characteristic of Ambient Intelligence contexts, and advocates the use of a distributed-centralized HTN-like approach (Hierarchical Task Network) [16]. In spite of the agreement on having to address device dynamism and heterogeneity, here, it is believed that these aspects should be tackled from the middleware perspective, rather than from that of the planner. The device heterogeneity should therefore remain a transparent matter for the planner, as is justified in the following section. In this respect, the use of agent approaches is also commonly accepted for auxiliary tasks. The work in [17] highlights the role assigned to a multi-agent system (MAS) architecture, acting as the context observer and regulator. The MAS assumes the responsibility of providing the planner with the required information about the context and the mechanisms with which to respond to it.

Although not directly applied but easily extrapolated to Ambient Intelligence, the use of a probabilistic search techniques [18] is presented in [19]. This work addresses the problem of task planning and action selection by means of a fuzzy-neural network approach combined with agent coordination and cooperation methods. Agents are trained to select the most appropriate action depending on the field configuration, changing their selections whenever the objects in the field adopt a different configuration. Another interesting approach, with applications in the Ambient Intelligence field, is that proposed in [20]. This approach adopts an HTN planning strategy which has been enhanced to fill the gap between real world environments and planning scenarios.

The Script-Based Task Planner (STP) resorts to a scrip structure to adapt the planning scenario to the real world environment. Finally, the work in [21] is also relevant. This work presents and discusses a planning strategy that seeks the optimal actions in partially observable stochastic domains, providing a firm foundation for planning in uncertainty conditions of actions and observations. An overall view of the planning strategies in distributed and cooperative circumstances is presented in [22]. The conclusions and suggestions drawn from this study set the basis for identifying the shortcomings of traditional Artificial Intelligence planning strategies, along with the strengths that can be used in a combined solution to the proposed planning approach. As will be stated later, the planning strategy proposed here is based on an HTN approach.

2.2. Cognition

From the cognitive perspective, planning depends strongly on knowledge and understanding competences. This dependency is grounded on the tight coupling that exists between knowledge and decision making. As stated by the authors of [23], there are certain pragmatic concerns about this relationship that do not have a unanimous answer. What does an agent need to know in order to perform a concrete action? When does an agent have to stop gathering information and make a decision? or at what point does an agent have to answer "I don't know what to do"? Answers to these questions imply the conviction that some degree of common sense is required. Please, refer to [24] for a thorough analysis of the most relevant knowledge-based planning techniques available in literature.

Automating common-sense reasoning has been one of the primary concerns for researchers in the Artificial Intelligence field. According to E. K. Muller, who in [13] provides a brief history of common-sense reasoning, the first work in this field dates from 1956. The main contributions to this field come from authors such as A. Newell, whose work is mainly concerned with the cognitive aspect of the Artificial Intelligence; M. Minsky, who has made enormous contributions to the domain of common-sense knowledge representation and reasoning [25] [26]; and finally D. Lenat, who in 1984 undertook the first real attempt to catalog common-sense knowledge, in the Cyc project [27].

Automating commonsense reasoning is a task that requires a sufficiently expressive language, a knowledge base in which to store such a large amount of knowledge, and a set of mechanisms capable of manipulating this knowledge, so as to infer new information. Regarding the knowledge base, Cyc[27], ConceptNet [28], Scone [29], and WordNet [30] are by far the most evolved and successful approaches found in literature.

To date, Cyc has formalized the largest body of fundamental human knowledge. Cyc Corp is currently addressing its research efforts towards automating knowledge acquisition, either by interacting with people [31] or by making use of the already asserted knowledge, natural language understanding, and the knowledge published on the Internet [32]. In contrast to the property system approach followed by Cyc, ConceptNet resorts to acquiring knowledge from the general public. ConceptNet adopts a semantic network structure similar to WordNet. Nevertheless, when compared, ConceptNet claims to hold more informal, defeasible, and practical knowledge. It can also be argued that WordNet should not be listed along with common-sense reasoning systems like Cyc and Scone, since it is simply a large database for English lexicon.

Scone is an open-source knowledge based system written in Common Lisp. The main difference between this and other approaches lies in the way in which search and inference are implemented. Scone adopts a marker-passing algorithm [33] devised to be run in the NETL machine [34]. Despite the fact that these marker-passing algorithms cannot be compared with general theorem-provers, they are indeed faster, and most of the search and inference operations involved in common-sense reasoning are supported: inheritance of properties, roles, and relations in a multiple-inheritance type hierarchy; default reasoning with exceptions; detecting type violations; search based on set intersection; and maintaining multiple, immediately overlapping world-views in the same knowledge base.

2.3. Formal model for Ambient Intelligence

Since a semantic model is the main contribution of this work, particular attention should be paid to the works concerning this theme which have been proposed to date. Despite the recent efforts of the W3C to provide a standardized and formal model of the environment, traditionally, there has existed a lack of consensus regarding the conceptual entities that should be part of the model. The Delivery Context Ontology [35] proposed by the W3C does not suffice to address the context-centered view advocated in this work. On the contrary, it is characterized for adopting a device-centered approach, in which the focus is at capturing and modeling the *context of use*. Aside from the context of use, additional issues should be considered in order to characterize and model the changes that make the context evolve from one situation to a different one. These aspects, however, have not been considered in the Delivery Context Ontology.

In this regard, the low level details with which the Detail Context Ontology has described the *environment* concept is also responsible of its rigidness and the impossibility to adapt such ontology to different approaches, such as those focusing at users, user actions, or context events. This weakness has led to a situation in which each context aware or Ambient Intelligence framework proposes its own specific model. The majority of the approaches tend to oversee the role played by the modeling task, and the justification as to why a model is composed of certain concepts rather than others therefore tends to be overlooked. Among the concepts that should be modelled in a semantic model for Ambient Intelligence, solely the notion of context has been properly formalized by the work in [36]. Furthermore, based on the definition provided by Dey and Abowd with regard to the context notion, the work in [37] goes a step forward as to how the context notion should be handled. Three different levels of contexts are considered, partially ordered by sets. Whatever the cause may be, apart from the context concept and the Ambient Intelligence or context-awareness field, no relevant works concerning concepts such as actions and events have been taken which can be cited here.

Among the most promising projects for Ambient Intelligence, the contribution made by AMIGO[1], follows the same approach adopted in this paper: that only by achieving automatic service composition can self-sufficient context be leveraged. However, although agreeing with the role played by the semantic model, the authors of this

[1] http://www.hitech-projects.com/euprojects/amigo/index.htm

work disagree with the constituent entities of the model. The AMIGO semantic model understands context as being a physical context with different functional domains (i.e. PC, mobile, CE, and home automation). This project therefore proposes a complex structure of different ontologies, grouped in a modular manner [10]. The notion of action or event is ignored as it is the relationship of such concepts with the context devices.

Additionally, it is worth mentioning the novel approach undertaken by the mIO! project[2] for knowledge modeling. This project is devoted to address the achievements of Smart Spaces mainly through the use of mobile phones. This initiative pays a special attention to the key role that the *context* concept performs in determining and characterizing the situation that surrounds end users. As a result, the work in [38] presents an ontology network specially devoted to model the user's contextual knowledge. Despite the fact that the semantic model that they propose in [39] is intended to support the modeling task undertaken in Ambient Intelligence, it has to be noticed that it is addressed from a different perspective than the one presented here. It is a user and device centered perspective that it is not concerned about the external factors that are affectiong the context itself, and how it could evolve as a result of these external events. Besides, an additional reason that motivates the proposal of a new semantic model rather than using the ontology proposed by the mIO! project is grounded in the need for conciseness and simplicity in the number of concepts and relationships among them.

The idea behind the work proposed here is to combine, into a semantic model, the minimal set of concepts that are present and relevant to the different layers that composed an architecture for Ambient Intelligence (hardware layer, communication layer, and service layer) as the one advocated here. In this sense, this minimal core can afterwards be enriched with further and more high-level details at the different layers. However, the core concepts composing the semantic model has to remain simple and common to all of them.

The work in [40] echoes the need for standardizing the semantic model proposed for context-aware or Ambient Intelligence, independently of the domain considered. This approach adopts a strategy based on answering questions such as: Who are the participants in the interaction?; Where does the interaction take place?; When does the interaction take place?; What does the interaction describe?; or how is context captured and accessed in the interaction? Once again, the proposed model fails to justify why these issues, rather than others, address these questions and should be reflected in the semantic model.

Some other projects resort to semantic models for different purposes: the work in [41] proposes a semantic model for services with the intention of supporting service discovery tasks in pervasive contexts; in [42] the semantic model, proposed as an OWL ontology, shares contextual information with agents; and finally, the authors of the work in [43] believe that the solution to supporting interoperability among devices populating the contexts lies in stating a common terminology. The semantic model is responsible for sharing such a common terminology.

[2] http://www.cenitmio.es/

Despite the existing differences concerning the concepts that should be present in the semantic model, when it comes to the modeling language, the use of OWL has become a common practice. The work in [44] describes a promising approach for dynamic service composition, in which the ontological model is represented by means of semantic graphs. The representation technology makes an important impact on the approach used to reason about the context. Please, refer to[45] for an extensive survey of context modelling and reasoning techniques.

3. A semantic model for actions and events

The definition of context given by Dey and Abowd in [36] was that of: "*context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves*". However, the work presented here, refers to context in the same way that J. Allen's refers to the concept of *world* in [46]. This concept is represented by a set of descriptions of both the static and dynamic aspects of the world, therefore modeling what is known about the past, present, and future. By using the J. Allen nomenclature, the static aspects of the world are easily captured as *properties* while dynamic aspects are captured by *occurrences*.

Since the main element of the Ambient Intelligence paradigm is context, and context dynamics are to be described by means of *occurrences*, it is obvious that actions and events (such as the constituent elements of occurrences) should be part of the semantic model. Actions and events have commonly been treated as being equivalent, or as having the slight difference of considering actions as events which have been intentionally generated [47]. On the contrary, there are some other theories that support the opposite belief. For example, the work in [48] argues that actions are not events. The main argument supporting this dissociation lies in considering actions and their agents as being inseparable or correlative [49]. The theory of action for multi-agent planning [50] also advocates for this distinction, although it hints that actions are accomplished by agents in their endeavor to achieve a goal.

Davidson's theories, particularly those regarding the philosophy of action, also identify actions with events, as is argued in [51]. Actions are described as a combination of two views. On the one hand, actions can be seen as causal explanations of body movements and on the other hand, actions can also be seen as the justifying reason that leads the action to take place. Davidson considers events to be equivalent to actions. The sole difference is that when an action is considered as an event, it is re-described in terms of its effects.

The semantic model proposed here adopts the Davidsonian view of actions and events. It should be highlighted that Cyc, through its language CycL, represents actions and events using a Davidsonian approach. Actions are described as events but are carried out by an agent. The approach implemented in Scone, as will be explained later, extends the work in [52] so as to include the notion of primary reasons for an action, along with its temporal and locational aspects.

At this stage, almost all the concepts contained in the semantic model have already been presented. However, we have not yet stated what "*semantic model*" signifies.

From a computing perspective, it is considered to be an agreement on how to interpret the knowledge represented in the knowledge base. Semantic models therefore ensure common interpretations of shared knowledge. They are also an essential requirement when there are different instances handling the same knowledge. Every holder is expected to extract the same meaning or conclusions from the represented knowledge. Having said that, the following subsections provides a more formal expression of the proposed semantic model, along with the justification of why the proposed semantic model is considered to be common-sense compliant.

3.1. Semantic model description and formalization

The proposed semantic model is depicted in Figure 1. Apart from the concept of action and event, some other relevant entities also exist whose semantics must be modelled. It is obvious that Ambient Intelligence cannot conceive existence without services. The service concept therefore exhibits a core position in the proposed model. Services can be described in terms of the actions that they can undertake and the objects or things receiving such actions.

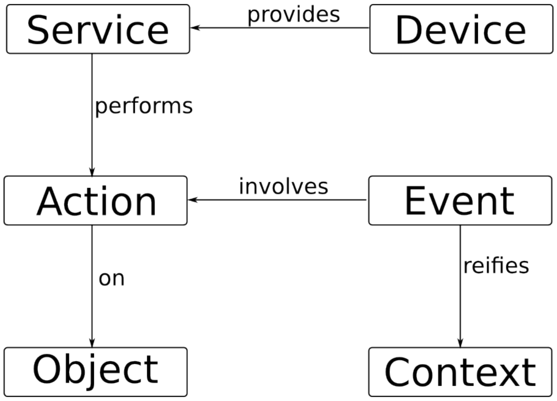


Figure 1: Entities and relationships for the semantic model

The simplicity of the model in comparison to certain other semantic models, such as those listed in the previous work section, is evident. This simplicity is the result of reducing Ambient Intelligence to those concepts that cannot be avoided, in other words, those of which the quintessence of Ambient Intelligence is composed. This semantic model can be used to model the domain knowledge, independently of the application context. Moreover, this simplicity eases the implementation of the semantic model in those technologies involved in devising systems for Ambient Intelligence. The following definitions state the foundation of the proposed semantic model:

Definition 1. A Context is a set C composed of statements which, when used together, describe knowledge about the world. There may be multiple contexts describing each of the different views of the world. The meaning or truth value of a statement is a function of the context in which it is being considered.

The function $meaning : S, C \rightarrow M$, where S is the set of statements describing the world, C is the set of possible contexts, and M the set of possible meanings, $meaning(s, c)$ therefore returns the meaning or truth value of the statement s in the context c. This can be formally stated as:

$$\forall c_i \in C \forall s_i \in S : m_i = meaning(s_i, c_i) \iff s_i \subseteq c_i \tag{1}$$

The meaning or truth value of a given statement depends on the contexts in which it has been declared.

Definition 3. An Action is the set A of individual actions that have been described from the perspective of their relation to the primary reason that rationalizes them. The function $AG : A \rightarrow G$, such that A is the set of possible actions, G is the set of possible agents, and the function AG returns the agent performing the given action. Furthermore, the function $PR : A, G \rightarrow E$ is the primary reason for an agent performing an action in seeking the effects of the event caused. Finally, the function $PA : A \rightarrow O$, such that O is the set of possible objects, and the function returns the agent that performs the action upon the given object.

$$\exists g \in G \exists a \in A \exists o \in O : (AG(a) \wedge PR(g, a)) \iff PA(a, o) \tag{2}$$

Therefore, an action is performed upon an object, if and only if there exists an agent with a primary reason to perform the action.

Definition 3. An Event is the set E of individual occurrences that cause changes in the world. The criteria followed by the Davidsonian doctrine on individuation of events argues for the equality of events when the same effects occur. The Davidsonian view is here adapted to internalize the multiple contexts approach. In this paper it is therefore considered that two events are equivalent when the same effects are caused by different actions. The effects of events are captured in the *after context*, while the preconditions for an event to take place are described by the *before context*. The functions $BC : E \rightarrow C$ and $AC : E \rightarrow C$, such that $BC(e)$ and $AC(e)$ respectively return the statements that compose the before and after context of a given event. Furthermore, the function $effect : A, O \rightarrow S$, such that S represents the set of statements that describe the world after the event took place.

$$\forall e \in E : (BC(e) \cup effect(a, o)) \rightarrow AC(e) \tag{3}$$

Given the events e_1 and e_2, it is said that e_1 is equivalent to e_2 when they have equivalent *after contexts* or when they cause the same effects:

$$\exists e_1, e_2 \in E : e_1 = e_2 \iff AC(e_1) \subseteq AC(e_2) \tag{4}$$

Definition 4. A Service is the set S of individual services provided by a device which performs a set of actions upon an object or a set of objects. The function $PD : S \rightarrow D$, such that D is the set of available devices, and the function returns the device or devices providing a given service.

$$\exists s \in S \exists d \in D \exists a \in A \exists o \in O : (PA(a, o) \wedge PD(s)) \rightarrow AG(a) = d \tag{5}$$

The definition of service therefore implies that the agent of an action provided by a service is a device.

Definition 5. A Device is the set D of electronic devices deployed in the environment which a set of services, such that:

$$\exists s \in S \exists d \in D : PD(s) = d \tag{6}$$

Definition 6. An Object is the set O of possible environmental objects upon which actions are performed. The function $OA : A \rightarrow O$ returns the set of possible objects that can receive a given action.

$$\exists o \in O \exists a \in A \exists e \in E : OA(a) \wedge PA(a, o) \rightarrow e \tag{7}$$

The occurrence of an event e implies the existence of an object o upon which the action a is performed.

4. Implementation details

The great potential of the proposed semantic model is founded on its fulfillment of the inter-module connectivity and communication requirements of a framework for Ambient Intelligence. This section therefore describes the implementation details[3] of a framework for Ambient Intelligence so as to demonstrates how the proposed architecture successfully tackles automatic service composition by means of a shared semantic model of actions and events. The aim of the experiment is to show the impact that automatic service composition has on the achievement of autonomous, self-managed, and pro-active systems which have for so long been demanded by Ambient Intelligence. Figure 2 depicts a system overview from the perspective of the modules involved in the architecture. The following subsections analyze those modules, paying particular attention to the implications of the semantic model and the different strategies followed for its implementation.

4.1. Scenario description

The scenario in Figure 3 depicts an Ambient Intelligence environment, in which devices are devoted to monitoring the security conditions of the building, in order to foresee or prevent accidents or subversive actions from taking place, and to diagnose security faults when they occur and self-heal them. These devices are comprised of presence, sound, light, and proximity sensors, video cameras, microphones, and Bluetooth devices, among those which are most relevant. The distributed architecture upon which these devices are deployed retrieves information from them and uses it to provide services such as object tracking, face recognition, geographical location, etc.

This comprehensive scenario of devices and services provides an appropriate context in which to test the suitability of the responses generated by the system, and a batch

[3] A complete description of the implementation code is available at:
http://arco.esi.uclm.es/ mariaj.santofimia/

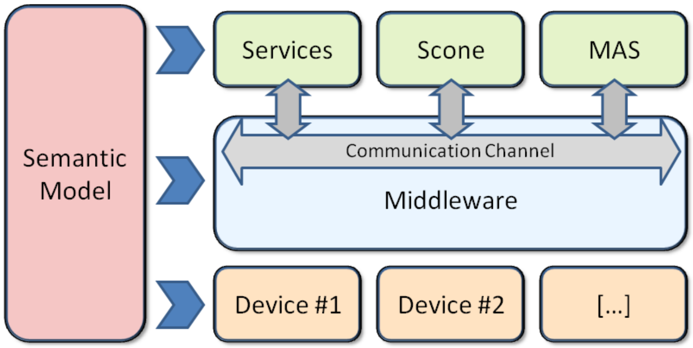


Figure 2: System architecture overview

of different events and situations, generated by a synthetic data generator[4], is employed to this end.

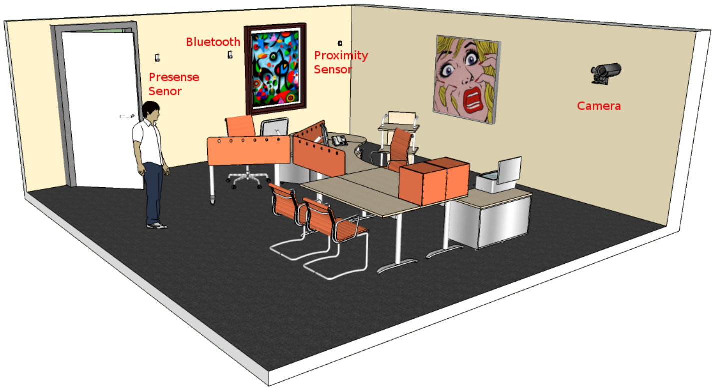


Figure 3: Scenario representation

4.2. The middleware framework

The middleware layer is one of the key elements of this architecture, essentially because it supports the connectivity of services and devices running on different platforms, or using different network protocols, even when different programming lan-

[4] A Synthetic Data Generator is available at: http://ailab.eecs.wsu.edu/casas/datasets.html

guages are involved. The middleware technology used to implement DOBS, the framework upon which this approach rests on is ZeroC ICE[5]. Please refer to [53] for further details on the implementation details of DOBS and a detailed list of the services offered.

ZeroC Ice is an object-oriented and CORBA-like middleware technology that provides the means (tools, API, libraries) to easily build object-oriented client-server applications. The ICE technology also supports the implementation of event-oriented architectures by means of an efficient publish/subscribe service, known as IceStorm.

Certain ICE concepts need to be settled before undertaking the service architecture description. For example, an ICE object is no more than a conceptual abstraction that responds to client requests. An ICE object has one or more interfaces, in which an interface is understood as the set of named operations provided by the object. Moreover, a single ICE object can be instantiated in one or several servers. Each ICE object has a unique object identity which is used to differentiate objects from each other. And finally, the proxy concept is used from the client side to contact a specific object. In order to invoke an operation on an ICE object, the client invokes that operation on the proxy, whose address is already known by the client. From the client's point of view, the proxy apparently adopts the role of a local object that serves the invocation. However, in reality the ICE run time, which runs in the background, is in charge of locating the ICE object referred to by the proxy, activating it, transmitting the arguments, waiting for the operation to be completed and returning the operation results, if any. Bear in mind that the indirect proxy abstraction and the use of the same naming policy for interfaces suppose the cornerstone for automatic service composition. These two features basically standardize the method instantiation strategy. There is no need to know much about how to instantiate a method in a given service, since the indirect proxy and the implemented naming policy suffice to the perform this task.

With regard to the naming policy, every service in the system implements the `Service` interface. As a direct consequence of implementing this interface, every service in the architecture provides a common set of operations, such as the `performsAction` method, among others. Independently of the specific functionality provided by each service, there is a set of methods that are common to all services, since they all share a common naming policy.

```
module SemanticModel {
  dictionary <string, Object> Properties;

  struct Event {
    Properties theProperties;
  };

   interface EventMonitor{
     void report(Event e);
   };
```

[5] http://www.zeroc.com/

```

  interface Device{
    void deviceName(string name);
    void setProperties(Properties spec);
    Properties getProperties();
  };

  sequence<Event>  EventSeq;

  interface Service {
   EventSeq performAction(string idAction, Object thg);
   };
};
```

The semantic model formalized in the previous section is adopted by the middleware framework by mapping concepts from the semantic model to implementational aspects of the service oriented architecture. The $Service$ concept from the model adopts the shape of the `Service` interface. The $Action$ concept of the model finds its equivalent in the `Action` interface implemented by all the methods that can be instantiated in the available services. Moreover, the semantic model notion of $Object$ is mapped to the `Thing` interface. With regard to the functions formalized in the semantic model, the $PA(a, o)$ is implemented in the middleware layer by the `performAction(Action act, Thing thg)` method of the `Service` interface. The `agentOf(Action act)` method is the middleware implementation of the semantic model function $AO(a)$. Finally, the semantic model notion of $Event$ is mapped to the `Event` struct.

It should be noted that the notion of $Context$ does not have an equivalent in the middleware layer since the notion of context is irrelevant at this level . Moreover, there are some other methods and interfaces that are not part of the semantic model although they are implemented in the middleware framework. For example the `setAction` and `getAction` methods are not part of the semantic model and their presence in the middleware layer responds to implementational needs.

4.3. The Multi-Agent System

The Multi-Agent System (MAS) works as a link between the Ambient Intelligence environment and the other elements of which the Ambient Intelligence framework is composed. The MAS is basically in charge of adopting the plan as outlined by the planning algorithm, and undertaking it. The interaction with the other architectural elements is, once again, supported in the semantic model. At the MAS level, the semantic model is implemented by means of an OWL ontology. The Agent Communication Language (ACL) messages, exchanged among agents, therefore contain classes of the ontology, which are simultaneously concepts of the semantic model.

Figure 4 depicts the OWL ontology mapping for the proposed semantic model. As occurred in the previous subsection with regard to the service architecture implementation, there are some ontology classes that do not have an equivalent concept in the semantic model. The explanation for this is supported by the need to abstract implementational issues from the cognitive aspects of the model. For example, the taxonomy

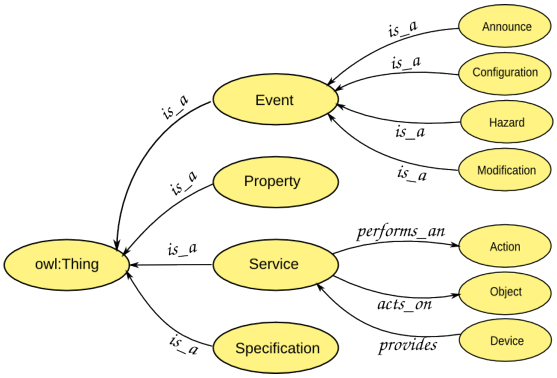


Figure 4: Semantic Model Ontology for Ambient Intelligence

of event types is intended to adapt the MAS response to the specific type of event detected. Nevertheless, this aspect must remain specific to the MAS, since neither the middleware nor the planner need to know about the differences between event types.

The implementation of the goal-driven MAS has been accomplished here by means of JADEX[6], with some modifications to allow it to run upon the middleware layer. The JADEX framework supports the development of BDI (belief, desire, and intention) agents. Beliefs are those properties that an agent considers to be true, goals are those properties that an agent desires to be true, and finally plans are those actions that lead an agents to a desired goal. These basic instances define what is known as the agent's mental state.

The agent's beliefs in combination with contextual information (held in the Scone knowledge base) are what lead the agent's behavior towards the goals that the agent desires to achieve or maintain. Interaction among agents, the knowledge base, services and devices are grounded on the fact that all of them share the same semantic model.

For example, consider the aforementioned security surveillance context, and the occurrence of an event consisting of a presence sensor activation announcement coming from a restricted area. The following mental state is held by the agent supervising the context:

Believe(a, b): Agent *a* beliefs that an intrusion event b has taken place.
Goal(a, g): Agent *a* desires to halt the intruder.
Plan(a, p); Agent *a* resorts to a set of actions so as to halt the intruder.

However, how is this presence sensor event translated into an intrusive situation? How does the agent know which set of actions to perform when trying to halt an intruder if this plan depends on the place in which the event took place, and the resources available at the exact intruder location? These questions pose two of the main chal-

[6] Java based and FIPA compliant agent environment.

lenges facing Ambient Intelligence, namely, context understanding and autonomous and automatic behavior generation. Providing a solution to these two challenges is one of the main assets of the architecture proposed here.

The main activity of the so called Manager agent is the supervision of the events occurring in the context. To this end, the agent is subscribed to those communication channels in which the events are notified by the sensor devices. In this particular situation, the presence channel is where the sensor publishes the notification and from which the Manager agent has received the notification. Since the notification comes from that specific channel, the agent labels it as an unauthorized presence event, and notifies the Perceptor agent of this occurrence. Whenever the Perceptor agent believes that an unauthorized presence event has taken place, one of the goals that it triggers is intended to achieve intruder identification.

The following code, extracted from the goal description section of the Perception agent, shows how the intruder_identification goal is dispatched whenever an eventType occurrence is known by the agent or, in other words, when a fact of the eventType class is asserted to the agent beliefs.

```
<!-- O2. Intruder identification -->
<achievegoal name="intruder_identification">
    <parameter name="unauthorisedPresence" class="Event">
        <bindingoptions>$beliefbase.eventTypes</
            bindingoptions>
    </parameter>
    <unique/>
    <!-- Create a new goal when new unauthorisedPresence
    event has been notified from the presence sensor. -->
</achievegoal>
```

The intruder_identification goal requires a plan in order for it to be achieved. There are several ways in which to accomplish an intruder identification, one of which is by performing a biometric identification (fingerprints, iris, face recognition, etc.).

```
<!-- 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>
```

Plans in JADEX are traditionally static procedural recipes coded in Java. Constraining a plan to a static set of actions prevents the architecture from achieving the versatility and dynamism demanded by Ambient Intelligence. The agent's plans therefore resort to a planning strategy to identify the course of actions that best fulfil the desired goals. Note the abstract character of the goal that allows the planner the responsibility to specify the type of biometric identification that has to be carried out. As listed in the code shown below, the plan request specifies very general constraints, and it is simply engaged in accomplishing an `identification` action upon a `biometric feature` in order to obtain a `person identity` result.

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

The result of the planning algorithm, stated as a set of quaternary elements of the form `<proxy, action, thing, result>`, is sent to the `Processor agent` which simply executes the `action`, served by the `proxy`, upon the `thing` in order to obtain a specific `result`.

The set of quaternary elements of which the plan is composed provides the MAS with the information required to automatically undertake the plan. Note that the agent plan has been composed in an ad-hoc manner, considering the availability of services and devices. Once again, it is important to highlight that the MAS capability to undertake on-the-fly generated plans is a direct consequence of using a common naming strategy for interfaces.

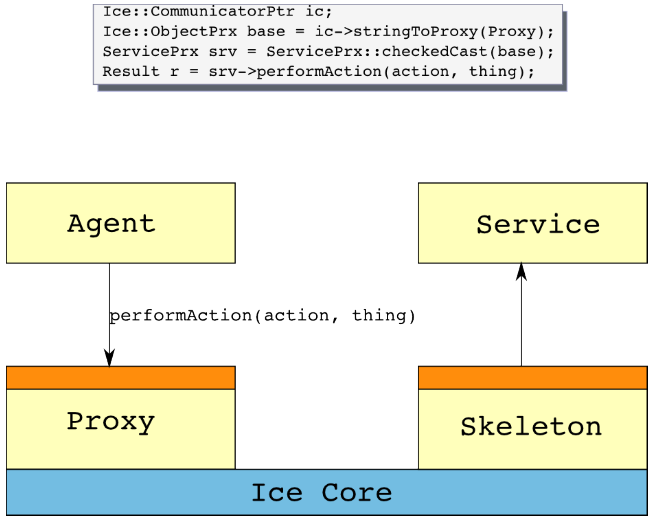
In order to carry out the proposed plan, the MAS simply invokes the `performsAction` operation on the service identified by the given `proxy` so as to perform the `action` upon the specified `thing`. Note how all this information is extracted from the quaternary set returned by the planner. Figure 5 depicts the logic schema for the method invocation.

4.4. The planner

Making the most of service versatility enables Ambient Intelligence systems to respond to whatever the needs are by adapting available services and devices to the desired functionality. Indeed, in this context, arising needs are treated as a desire to perform actions upon objects. By making this assumption and adapting a Hierarchical Task Networks (HTN) approach to consider actions as tasks, the task of satisfying arising needs can be automatically accomplished by means of an HTN-like planner.

The actions that can be performed by the system, at a specific location and time, are determined by the devices and services available at that location and time. Those actions that cannot be performed, owing to the lack of services that provide such functionality, are named here as non-feasible actions. Whenever the system demands the execution of a non-feasible action, the planner comes into play.

As is listed below, the `Planning` algorithm starts with an empty plan, the Π plan, to be filled with the list of actions provided by services. This course of actions is intended to emulate the non-feasible action demanded. The course of actions is



```
Ice::CommunicatorPtr ic;
Ice::ObjectPrx base = ic->stringToProxy(Proxy);
ServicePrx srv = ServicePrx::checkedCast(base);
Result r = srv->performAction(action, thing);
```

Figure 5: Logic schemata for remote method invocation

provided as a set of actions performed upon objects, A and O respectively, and the results R of accomplishing these actions.

Algorithm 1 Planning(Π, A, O, R)

```
$\pi = (A, O, R)$
**if** $A$ is non-feasible **then**
    get all the actions $A = (a_1, a_2, ..., a_n)$ that have the same result $A$
    **while** $a_i$ is non-feasible **do**
        delete $a_i$ from $A$
    **end while**
    **while** only doable actions $a_i$ does not have an equivalent target object **do**
        list all the objects $Objects = (o_1, o_2, ..., o_n)$ of action $a_i$
        check if those $o_i$ are equivalent to or can be $O$
    **end while**
    Recursively call $\pi = Planning(a_i, o_i, resultOf\ a_i)$
**end if**
Add $\pi$ to $\Pi$
Return $\Pi$
```

4.5. The Scone Knowledge Base

The last of the architectural modules analyzed here is the Scone Knowledge Base[7] (Scone KB). The use of Scone is founded on the need for common-sense knowledge

[7] The Scone Knowledge Base, http://www.cs.cmu.edu/ sef/scone

modelling and reasoning capabilities, particularly when that knowledge refers to actions and events. As with the previously described modules, the semantic model has also been mapped into Scone. In fact, the *Context* concept is one of the features of Scone that makes it so suitable for reasoning about actions and events. Please refer to [54] for a description of the multiple context insights.

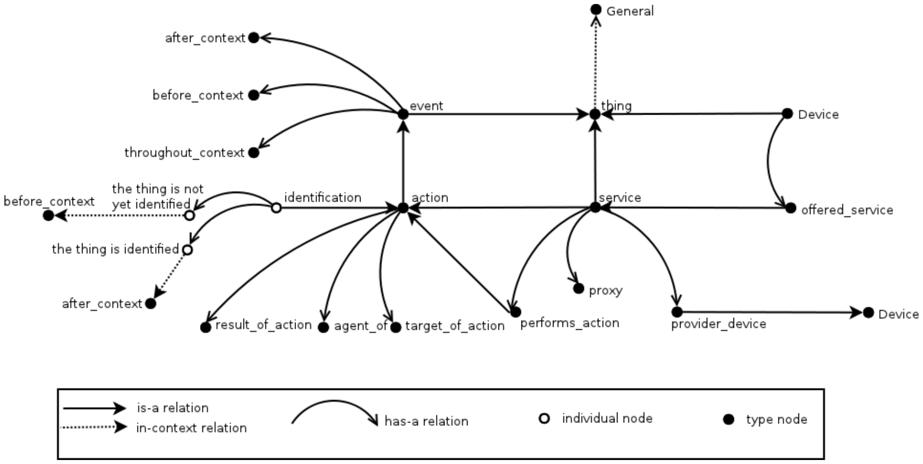


Figure 6: Semantic Model in Scone

Moreover, not only contexts are relevant for the modeling of actions and events, but also the services that provide them, the agents that bring them up, or the generated outputs. Figure 6 depicts how the semantic model has been mapped into the Scone KB. Note how the semantic model concepts and relationships are implemented, respectively, as nodes and links in Scone. This semantic model has been used as a foundation for the coding of a dictionary of actions and events[8]. The following lines, using the Scone syntax, show an example of this encoded dictionary for the `capture` event:

```
(new-event-type {capture} '({event})
        :roles
        ((:indv {captionSource} {thing})
         (:indv {captionObject} {thing})
         (:indv {captionTarget} {data}))

   :throughout
   ((new-statement {captionObject} {is noticed in} {captionSource}))
   :before
   ((new-not-statement {captionObject} {is recorded in}
          {captionTarget}))
   :after
```

[8] A description of the considered scenario (building, devices, and general knowledge) and some of the most relevant actions and events, along with a detailed justification of the key issues for common-sense satisfaction can be found at: http://arco.esi.uclm.es/ mariaj.santofimia/

```
  ((new-statement {captionObject} {is recorded in}
        {captionTarget})))
```

For example, the above lines describe (from a common-sense perspective) what the `capture` event represents in terms of relevant elements and states of the world involved (*before*, *throughout*, and *after contexts*). Event roles symbolize those domain elements that characterize the world states. For example, the `captionSource` role is played by the thing being captured. When referring to the `capturingImage` action, the `captionSource` role is specified in the light photons captured by a photographic camera. The after context for the `capturingImage` action descibes a state of the world in which, after taking place, that action results in a new state in which there is an image file, picturing the instant photo frame captured by the camera.

The planning algorithm, based on the actions and events dictionary and the domain knowledge held in the Scone KB, resorts to the inference capabilities of Scone so as to devise the course of actions which, given a desired state of the world, lead to its consecution.

The following lines show Scone's strenghts with regard to inferring and deducing the knowledge that seems so obvious to people, but is so difficultly handled by computers. The Scone type and property hierarchy KB and its implementation of the marker-passing inference strategy, provide the means to enhance planning with common-sense knowledge and reasoning capabilities, resembling the process in which people make decisions.

For example, when attempting to figure out the `identity` of an intruder by performing the `identification` of a `biometric feature`, the first step consists of determining the existance of a service that is capable of providing such a functionality. At first glance, one might easily conclude that this is too generic a task to be provided by a service, and Scone is no exception. When asked about the existance of such a service, Scone answers that there is no type or individual node whose `performs-action` role is the `identification` event. In other words, the `identification` event is not directly provided by any of the available services:

```
CL-USER> (x-is-the-y-of-what? {identification}  {performs-action})
{identification} is not known to play the {performs-action} role of anything.
NIL
```

At this stage, a sensible approach is to seek those events or actions that cause the same effects as those caused by the `identification` event:

```
CL-USER> (list-events-causing-x (new-statement {biometric feature} (car (list-parents(car (list-after {identification})))) {person identity}))
({recognition} {faceRecognition}
 {identityIdentificationAccess} {identityIdentification})
```

The Scone answer to this query is a set of actions and events that produce the same effects as the `identification` event. However, not all of them are equally useful, and those directly provided by available services are preferred to those that cannot be served by available services. In order to figure this aspect out, Scone is again queried about the existance of services performing the given actions. As is listed below, the `recognition` action is not performed by any of the available services, while the `faceRecognition` is indeed provided by the cited service:

```
CL-USER>  (x-is-the-y-of-what?  {faceRecognition} {performs-action})
{SimpleRecognizer:default -p 12000}
```

Note that the `performs-action` property (the so called role) symbolizes the action or set of actions capable of being undertaken by individual nodes of the `service` type node.

When queried about the existence of an individual `service` performing the action of `faceRecognition` Scone answers that the individual, with proxy property `SimpleRecognizer:default -p 12000`, is capable of performing an equivalent `identification` event. The proxy property is also a role or property of the `service` node. This is used to hold the remote location address from which actions can be called to be executed.

In order to match the request, not only must the *after contexts* be equivalent, but also the items upon which actions are performed. Therefore, it is also necessary to check that those items supporting the equivalent actions or events are equivalent. In other words, the following steps consist of checking that the `faceRecognition` action can be performed upon a `biometric feature` as stated in the initial requirements:

```
CL-USER> (list-all-x-of-y {object-of}  {faceRecognition})
({events:face})

CL-USER> (can-x-be-a-y? {face} {person identity})
T
```

`Face` is the item upon which the `faceRecognition` action is performed. It is an obvious fact to people that a `face` is also a `biometric feature`, and this is confirmed by Scone when queried. Since the `face` object works as an input to the `faceRecognition` action, the following step consists of devising how to obtain or satisfy the action requirements:

```
CL-USER> (x-is-the-y-of-what? {faceRecognition} {performs-action})
{SimpleRecognizer:default -p 12000}

CL-USER> (list-events-preceding {faceRecognition})
({detectingFace})
```

If the `detectingFace` action is required so as to permit the `faceRecognition` action to take place, Scone should once again be queried about the inputs or requirements for the `detectingFace` action, and should also verify whether any of them is compliant with the `face` object. The following lines show how to implement such an interaction with Scone:

```
CL-USER> (list-all-x-of-y {object-of} {detectingFace})
({captureResult of recordingImage}
{A-role of is picture of})
CL-USER> (can-x-be-a-y? {captureResult of recordingImage} {face})
T
```

The interpretation of the above results concludes that the `detectingFace` action has to be performed either upon the result of a recording image device or a picture file. However, apart from the required input, the `detectingFace` action might also demand some other requirements to be undertaken. Scone is therefore queried about this matter:

```
CL-USER> (x-is-the-y-of-what? {detectingFace} {performs-action})
{SimpleDetector:default -p 11000}
CL-USER> (list-events-preceding {detectingFace})
({capturingFace} {performs-action} {recordingImage}
{recordingVideo})
CL-USER> (list-all-x-of-y {object-of} {capturingFace})
({captionTarget of capturingBiometricFeature}
{B-role of is recorded in}
{captureResult of recordingVideo)}
{captureResult of recordingImage}
{captureResult of detectingLight}
{captureResult of detectingPresence}
{A-role of is picture of (0-1290)})
```

Scone concludes that in order to fulfill the requirements demanded by the `detectingFace` action, the following could be undertaken: `capturingFace` `performs-action` `recordingImage` `recordingVideo`.

```
CL-USER>  (x-is-the-y-of-what? {capturingFace} {performs-action})
{videoCamera1Service}

CL-USER> (b-wire (car (list-after {capturingFace})))
{imageFile}
```

Steps are repeted using different actions until a point is reached at which the action does not require any inputs, and can therefore be directly accomplished. When this point is reached, Scone is asked about the result of the action. As can be observed in the above lines, the output of the `capturingFace` is an image file, from which a face can be detected in order to perform a face recognition action so as to resolve the intruder identification.

The planning algorithm proposed in this works is intended to automate the generation of the queries presented above. By starting from a ternary query composed of the action, the object or item that receives the action, and the expected result, the planning algorithm is able to attain the course of actions that provides a similar functionality. To summarize, the result provided by the planner for the example analyzed here generates the following course of actions:

```
((capturingImage, thing, imageFile),
 (detectingFace, imageFile, imageFile),
 (faceRecognition, face, person identity),
 (identification, biometric feature, person identity))
```

5. Experimental validation

The design of an evaluation methodology is a crucial issue by which to assess the end user's degree of satisfaction, 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 which are highly subjective, are involved. The methodology proposed here is outlined in Figure 7.

According to Figure 7, the process of evaluation for each event that matches the MAS goals generates a vector of attributes, which are tightly related to the services generated and their characteristics. This vector is then evaluated to obtain the fitness (i.e. the goodness) of the system's response by means of a rule-based grading approach, which is generated using human expertise and the end user expectations. It should be noted that the overall goal is to identify intrusions as quick as possible, notify the security staff providing useful information and reduce human intervention as much as possible in the whole process. Finally, the outcome of the evaluation process provides the grading of the system's response and generates statistics and time series for a more in-depth analysis.

The evaluation process described here can be implemented to be executed on-line in real-time, or off-line from the data captured. 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 (such as Arena, or Matlab/Simulink).

The result of the evaluation process may also be very useful for other purposes such as condition monitoring. To this end, fitness variations can alert us 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 system response 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, such as the level of human intervention and the resources involved.

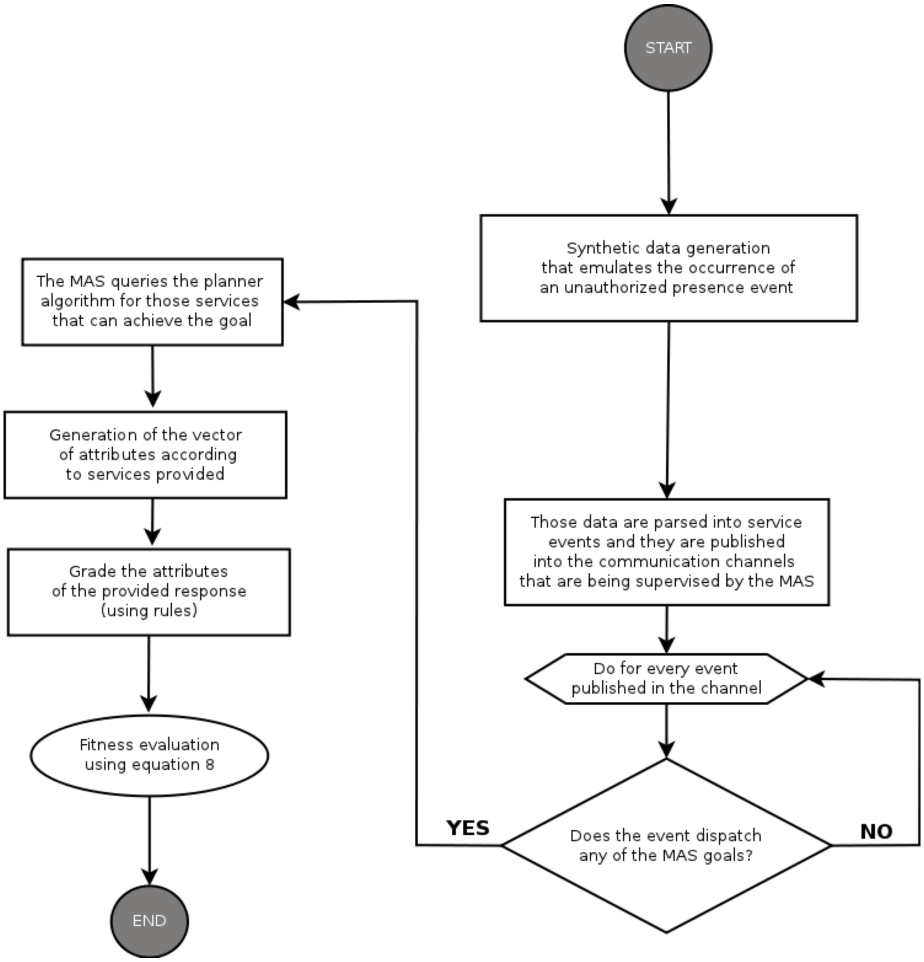

START
Synthetic data generation
that emulates the occurrence of
an unauthorized presence event
Those data are parsed into service
events and they are published
into the communication channels
that are being supervised by the MAS
Do for every event
published in the channel
Does the event dispatch
any of the MAS goals?
YES
NO
The MAS queries the planner
algorithm for those services
that can achieve the goal
Generation of the vector
of attributes according
to services provided
Grade the attributes
of the provided response
(using rules)
Fitness evaluation
using equation 8
END

Figure 7: A Evaluation Process

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 vector of attributes. Depending on the type of event, there are certain user expectations which must 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 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 and the rate of success of certain services.

The grades given, based on those rules, must be weighted, according to the significance of each graded feature and added to obtain overall fitness value. Since composite services reduce human intervention and provide more valuable information, the attributes related to them will have higher weights because their contribution to the goodness of the response is higher.

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, only one type of event is considered: human presence in the room. Regarding the services provided, those which are most basic 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 these according to the common-sense reasoning 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 system response and the services provided. Examples of attributes for this particular case are the availability and output of the presence sensor, or the output of the face recognition application and the delay when providing this output. The fitness evaluation is then performed by grading each individual attribute according to a set of rules. Finally, the overall fitness of the system response is obtained by multiplying the grades of each attribute by a weight and adding them all, as in Equation 8:

$$F = \frac{\sum_{n=1}^{N} w_n \cdot g_n}{\sum_{n=1}^{N} w_n} .100 \qquad\qquad g_n \epsilon [0,1] \tag{8}$$

where F is the fitness, N is the total of number of attributes, and w_n and g_n are the weight and grade of the nth attribute respectively. The value of F is normalized in the range 0 to 100. The weight of each concept is established according to its contribution and how significant it is when assessing the goodness of the response. Attributes related to composite services have a higher weight than basic ones since they better fulfil user needs and will reduce human intervention when responding to events. Hence, the goodness of a response will be enhanced by composite services and the resulting fitness value will be higher.

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 case 1 only basic services are provided; in case 2 basic and composite services are provided, and the face recognition application has a success rate of 10%; in case 3 basic and composite services are also provided, but the face recognition application has a success rate of 90%. A simulation with 1000 events has been run for each case. Figure 8 shows the results obtained for the three different cases in the last 100 events logged.

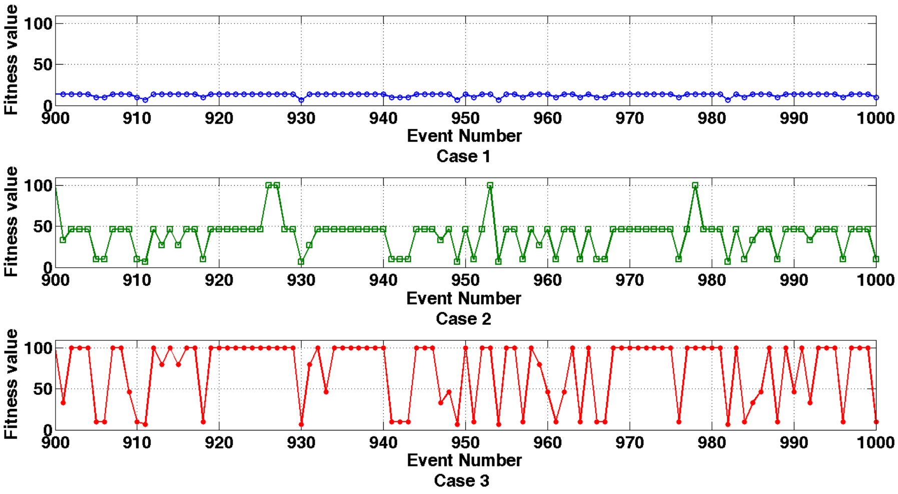


Figure 8: Simulation results

Table 1: Simulation results

Case	Mean of F	Std dev of F	% of Comp Services
1	12.55	1.937	0
2	42.31	20.33	75.4
3	77.21	35.93	75.4

Table 1 shows the mean value and standard deviation of the fitness for each case. It also shows the percentage of events in which composite services are provided.

It can be concluded from the observed results that case 3 obtains the highest fitness in most of the cases, since it involves composite services that better fulfil the user needs and the success rate of the face recognition application is higher. The suitability of the evaluation methodology proposed here has been proved to serve as a mean to rate the goodness of system responses and assist in the validation of the proposed solution.

6. Conclusions

The work presented in this paper provides a semantic model based solution with which to the address the self-sufficiency issue demanded by systems for Ambient Intelligence by means of an automatic service composition approach. The proposed semantic model is adopted by the different elements of which the architectural solution is composed, thus allowing composite services to be automatically devised, composed and executed.

This endeavor has been mainly motivated by the need to encompass user-centered and device-centered approaches into a context-centered view, less restrictive and more coherent with the Ambient Intelligence paradigm. The revision of the state of the art of frameworks for Ambient Intelligence brings into light the rigidness and the incapability to address novel situation that characterize such systems. The analysis of those systems evidence that these weaknesses are rooted at the independence with which the different framework layers are being addressed. In this regard, the proposal of a simple enough semantic model is the approach advocated here as the most suitable mean to make the system work as a whole.

Additionally, the proposed semantic model needs to be grounded in a philosophical theory of actions, in such a way that actions and events should be at the core of the formal model advocated here. In this sense, the capability to provide responses to unexpected situations can be split it into two sub-capabilities. On the one hand it depends on the capability to understand ongoing situations and on the other hand it is also based in the capability to elaborate a system responses to the noticed situations. Understanding situations and implementing responses are two activities that cannot be decoupled from the notion of actions and events. Ultimately, the fact that the majority of the semantic model found in the literature overlook both concepts, is recalled here as an additional reason motivating the need for yet another semantic model.

It cannot be obviated that actions and events are key concepts for Ambient Intelligence domain knowledge, and this should be reflected in the semantic model. On the base of such a semantic model, in which actions differs from events, this work proposes a comprehensive solution to self-sufficiency and autonomy requirements grounded on automatic service composition.

Automatic service composition is a complex task that requires from the use of standards in order to unify the way how services are discovered, instantiated, or modified. Additionally, it is also necessary to established some mapping mechanisms capable of translating high-level actions into service instantiation without requiring from human assistance. To this end, an approach based on common-sense planning is described. The planing strategy requires an extensive knowledge based system in which common-sense reasoning can be accomplished and knowledge concerning contexts and every day life can be held.

Combining a Belief, Desire, and Intention approach with the Scone system sets the basis for implementing an action planning, capable of solving the problem of automating the service composition task. The use of a middleware layer sets an abstraction layer between the heterogeneous services and the system supervising the environment. Service instantiation and supervision is achieved by simply supervising the communication channels in which information is published and from which services receive invocations.

7. Acknowledgements

This research was supported by the Spanish Ministry of Science and Innovation under the project DAMA (TEC2008-06553/TEC) and by Spanish Technological and Industrial Development Center -Ministry of Industry, under the project “ENERGOS: Technologies for the automated and intelligent management future power grid” (CEN-20091048).

[1] C. Ramos, J. C. Augusto, D. Shapiro, Ambient Intelligence—the Next Step for Arlosrtificial Intelligence, IEEE Intelligent Systems 23 (2) (2008) 15–18, ISSN 1541-1672.

[2] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, J. C. Burgelman, ISTAG: Scenarios for Ambient Intelligence in 2010, Tech. Rep., ISTAG, 2001.

[3] E. Kim, J. Choi, A semantic interoperable context infrastructure using web services, in: ICCSA'07: Proceedings of the 2007 international conference on Computational science and Its applications, Springer-Verlag, Berlin, Heidelberg, ISBN 3-540-74475-4, 978-3-540-74475-7, 839–848, 2007.

[4] T. Gu, H. K. Pung, D. Q. Zhang, A service-oriented middleware for building context-aware services, J. Netw. Comput. Appl. 28 (1) (2005) 1–18, ISSN 1084-8045.

[5] R. H. Campbell, A Middleware for Context-Aware Agents in Ubiquitous Computing Environments, in: in ACM/IFIP/USENIX International Middleware Conference, Rio de, 16–20, 2003.

[6] J. Cao, J. Wang, S. Zhang, M. Li, A dynamically reconfigurable system based on workflow and service agents, Eng. Appl. Artif. Intell. 17 (7) (2004) 771–782, ISSN 0952-1976.

[7] H. Chen, An intelligent broker architecture for pervasive context-aware systems, Ph.D. thesis, University of Maryland, Baltimore County, 2004.

[8] R. Reichle, M. Wagner, M. U. Khan, K. Geihs, J. Lorenzo, M. Valla, C. Fra, N. Paspallis, G. A. Papadopoulos, A comprehensive context modeling framework for pervasive computing systems, in: DAIS'08: Proceedings of the 8th IFIP WG 6.1 international conference on Distributed applications and interoperable systems, Springer-Verlag, Berlin, Heidelberg, ISBN 3-540-68639-8, 978-3-540-68639-2, 281–295, 2008.

[9] W. T. Niu, J. Kay, PERSONAF: framework for personalised ontological reasoning in pervasive computing, User Modeling and User-Adapted Interaction 20 (1) (2010) 1–40, ISSN 0924-1868.

[10] I. Roussaki, I. Papaioannou, D. Tsesmetzis, J. Kantorovitch, J. Kalaoja, R. Poortinga, Ontology Based Service Modelling for Composability in Smart Home Environments, in: M. Mühlhäuser, A. Ferscha, E. Aitenbichler (Eds.), Constructing Ambient Intelligence, vol. 11 of *Communications in Computer and Information Science*, Springer Berlin Heidelberg, ISBN 978-3-540-85379-4, 411–420, 2008.

[11] R. Wilensky, Planning and Understanding, Addison-Wesley, Reading, MA, 1983.

[12] D. B. Lenat, R. V. Guha, K. Pittman, D. Pratt, M. Shepherd, CYC: toward programs with common sense, Commun. ACM 33 (8) (1990) 30–49, ISSN 0001-0782.

[13] E. T. Mueller, Commonsense Reasoning, Morgan Kaufmann, ISBN 0-12-369388-8, 2006.

[14] D. J. Cook, J. C. Augusto, V. R. Jakkula, Ambient intelligence: Technologies, applications, and opportunities, Pervasive and Mobile Computing 5 (4) (2009) 277–298, ISSN 15741192.

[15] F. Amigoni, N. Gatti, C. Pinciroli, M. Roveri, What planner for Ambient Intelligence applications?, IEEE Transactions on Systems, Man, and Cybernetics, Part A 35 (1) (2005) 7–21.

[16] K. Erol, J. Hendler, D. S. Nau, HTN planning: Complexity and expressivity, in: In AAAI-94, 1994.

[17] N. Gatti, F. Amigoni, M. Rolando, Multiagent Technology Solutions for Planning in Ambient Intelligence, in: WI-IAT '08: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, Washington, DC, USA, ISBN 978-0-7695-3496-1, 286–289, 2008.

[18] A. W. Stroupe, T. R. Balch, Value-based action selection for observation with robot teams using probabilistic techniques, Robotics and Autonomous Systems 50 (2-3) (2005) 85–97.

[19] K. Jolly, R. S. Kumar, R. Vijayakumar, Intelligent task planning and action selection of a mobile robot in a multi-agent system through a fuzzy neural network approach, Engineering Applications of Artificial Intelligence 23 (6) (2010) 923–933, ISSN 0952-1976.

[20] D. Kang, H.-J. Choi, Robot Task Planning for Mixed-Initiative Human Robot Interaction in Home Service Robot, in: CITWORKSHOPS '08: Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops, IEEE Computer Society, Washington, DC, USA, ISBN 978-0-7695-3242-4, 49–54, 2008.

[21] L. P. Kaelbling, M. L. Littman, A. R. Cassandra, Planning and Acting in Partially Observable Stochastic Domains, Artificial Intelligence 101 (1995) 99–134.

[22] T. Patkos, A. Bikakis, G. Antoniou, M. Papadopouli, D. Plexousakis, Distributed AI for Ambient Intelligence: Issues and Approaches., in: B. Schiele, A. K. Dey, H. Gellersen, B. E. R. de Ruyter, M. Tscheligi, R. Wichert, E. H. L. Aarts, A. P. Buchmann (Eds.), AmI, vol. 4794 of *Lecture Notes in Computer Science*, Springer, ISBN 978-3-540-76651-3, 159–176, 2007.

[23] R. Fagin, J. Y. Halpern, Y. Moses, M. Y. Vardi, Reasoning about Knowledge, MIT Press, 1995.

[24] D. E. Wilkins, M. desJardins, A Call for Knowledge-Based Planning, AI Magazine 22 (1) (2001) 99–115.

[25] M. Minsky, Society of Mind: A Response to Four Reviews, Artif. Intell. 48 (3) (1991) 371–396.

[26] M. Minsky, The emotion machine: from pain to suffering, in: Creativity & Cognition, 7–13, 1999.

[27] D. Lenat, CYC: A Large-Scale Investment in Knowledge Infrastructure, Communications of the ACM 38 (1995) 33–38.

[28] H. Liu, P. Singh, ConceptNet: A Practical Commonsense Reasoning Toolkit, BT Technology Journal 22 (2004) 211–226.

[29] S. E. Fahlman, The Scone Knowledge-Base Project, available online at: `http://www.cs.cmu.edu/ sef/scone/`. Retrieved on February 28th, 2010., 2010.

[30] G. A. Miller, WordNet: A Lexical Database for English, Communications of the ACM 38 (1995) 39–41.

[31] M. Witbrock, D. Baxter, J. Curtis, D. Schneider, R. Kahlert, P. Miraglia, P. Wagner, K. Panton, G. Matthews, A. Vizedom, An Interactive Dialogue System for Knowledge Acquisition in Cyc, in: In Proceedings of the Workshop on Mixed-Initiative Intelligent Systems. IJCAI, 138–145, 2003.

[32] P. Shah, D. Schneider, C. Matuszek, R. C. Kahlert, B. Aldag, D. Baxter, J. Cabral, M. Witbrock, J. Curtis, Automated Population of Cyc: Extracting Information about Named-entities from the Web, in: In Proceedings of the Nineteenth International FLAIRS Conference, AAAI Press, 153–158, 2006.

[33] S. E. Fahlman, Marker-Passing Inference in the Scone Knowledge-Base System, in: First International Conference on Knowledge Science, Engineering and Management (KSEM'06), Springer-Verlag (Lecture Notes in AI), 2006.

[34] S. E. Fahlman, NETL: A System for Representing and Using Real-World Knowledge, MIT Press, Cambridge, MA, 1979.

[35] R. Lewis, J. M. C. Fonseca, Delivery Context Ontology, World Wide Web Consortium, Working Draft WD-dcontology-20080415, 2008.

[36] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, P. Steggles, Towards a Better Understanding of Context and Context-Awareness, in: HUC, 304–307, 1999.

[37] M. M. Richter, The search for knowledge, contexts, and Case-Based Reasoning, Engineering Applications of Artificial Intelligence 22 (1) (2009) 3 – 9, ISSN 0952-1976.

[38] M. P. Villalon, M. Suárez-Figueroa, A. Gómez-Pérez, Reusing Ontology Design Patterns in a Context Ontology Network, in: Proceedings Second Workshop on Ontology Patterns (WOP 2010) co-located at ISWC 2010, vol. 671, Alemania, URL `http://oa.upm.es/5432/`, 2010.

[39] M. Poveda-Villalon, R. Garcia-Castro, Entregable 1.1.3 Formalización e implementación de las ontologías mIO!, Tech. Rep., Ontology Engineering Group. Universidad Politecnica de Madrid, 2009.

[40] R. de Freitas Bulcao Neto, M. da Graca Campos Pimentel, Toward a Domain-Independent Semantic Model for Context-Aware Computing, Web Congress, Latin American 0 (2005) 61–70.

[41] A. Bandara, T. Payne, D. De Roure, N. Gibbins, T. Lewis, A pragmatic approach for the semantic description and matching of pervasive resources, in: GPC'08: Proceedings of the 3rd international conference on Advances in grid and pervasive computing, Springer-Verlag, Berlin, Heidelberg, ISBN 3-540-68081-0, 978-3-540-68081-9, 434–446, 2008.

[42] H. Chen, T. Finin, A. Joshi, An Ontology for Context-Aware Pervasive Computing Environments, Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review 18 (2003) 197–207.

[43] D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, K. De Bosschere, Towards an Extensible Context Ontology for Ambient Intelligence, 2004.

[44] K. Fujii, T. Suda, Component Service Model with Semantics (CoSMoS): A New Component Model for Dynamic Service Composition, in: Proceedings of the 2004 Symposium on Applications and the Internet-Workshops (SAINT 2004 Workshops), SAINT-W '04, IEEE Computer Society, Washington, DC, USA, ISBN 0-7695-2050-2, 348–, 2004.

[45] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervasive and Mobile Computing 6 (2) (2010) 161 – 180, ISSN 1574-1192, context Modelling, Reasoning and Management.

[46] J. F. Allen, Towards a general theory of action and time, Artif. Intell. 23 (1984) 123–154, ISSN 0004-3702.

[47] B. Hommel, J. Musseler, G. Aschersleben, W. Prinz, The theory of event coding (TEC): A framework for perception and action planning, Behavioral and Brain Sciences 24 (2001) 849–878.

[48] K. Bach, Actions Are Not Events, Mind 89 (1980) 114–120.

[49] J. Hyman, Three Fallacies About Action, in: 29th International Wittgenstein Symposium, 2006.

[50] M. P. Georgeff, A Theory of Action for MultiAgent Planning, in: A. H. Bond, L. Gasser (Eds.), Readings in Distributed Artificial Intelligence, Kaufmann, San Mateo, CA, 205–209, 1988.

[51] D. Davidson, Actions, Reasons, and Causes, The Journal of Philosophy 60 (23) (1963) 685–700, ISSN 0022362X.

[52] E. C. Sahin, Episodic Memory Representation in a Knowledge Base, with Application to Event Monitoring and Event Detection, Master's thesis, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, Pennnsylvania, 2008.

[53] F. J. Villanueva, D. Villa, M. J. Santofimia, F. Moya, J. C. Lopez, A framework for advanced home service design and management, IEEE Transactions on Consumer Electronics 55 (2009) 1246–1254.

[54] W. Chen, S. E. Fahlman, Modeling Mental Contexts and Their Interactions, in: AAAI 2008 Fall Symposium on Biologically Inspired Cognitive Architectures, Washington, 2008.