A MULTIAGENT SYSTEM SUPPORTING SITUATION AWARE INTERACTION WITH A SMART ENVIRONMENT

Davide Cavone, Berardina De Carolis, Stefano Ferilli and Nicole Novielli
Dipartimento di Informatica, Universita' di Bari, Bari, Italy

Keywords: Smart environment, Multi Agent System, User modeling, Pervasive interaction.

Abstract: Adapting the behavior of a smart environment means to tailor its functioning to both context situation and users' needs and preferences. In this paper we propose an agent-based approach for controlling the behavior of a Smart Environment that, based on the recognized situation and user goal, selects a suitable workflow for combining services of the environment. We use the metaphor of a butler agent that employs user and context modeling to support proactive adaptation of the interaction with the environment. The interaction is adapted to every specific situation the user is in thanks to a class of agents called Interactor Agents.

1 INTRODUCTION

Users of a smart environment often have contextual needs depending on the situation they are in. In order to satisfy them, it is important to adopt an approach for personalizing service fruition according to the user situation. Moreover it is important to make the interaction with services easy and natural for the user. To this aim we propose an approach based on software agents able to provide Smart Services, i.e. integrated, interoperable and personalized services accessible through several interfaces available on various devices present in the environment in the perspective of pervasive computing. We adopt the metaphor of a butler in grand houses, who can be seen as a household affairs manager with duties of a personal assistant, able to organize the housestaff in order to meet the expectations of the house inhabitants. Starting from the results of a previous project (De Carolis et al., 2006), we have developed a MAS in which the butler agent recognizes the situation of the user in order to infer his possible goals. The recognized goals are then used to select the most suitable workflow among a set of available candidates (Yau and Liu, 2006). Such a selection is made by semantically matching the goals, the current situation features and the effects expected by the execution of the workflow. Once a workflow has been selected, its actions are executed by the effector agents. Then the system guides interactively the user in finding, filtering and composing services (Kim, 2004), exploiting a semi-automatic approach: the user may change the execution of the selected workflow by substituting, deleting, or undoing the effects of some services. Moreover, the butler is able to learn about situational user preferences but it should leave to its "owner" the last word on critical decisions (Falcone and Castelfranchi, 2001). To this aim, the butler agent must be able to interpret the user's feedback appropriately, using it to revise: (i) the knowledge about the user, with respect to his preferences and goals in a given situation, and (ii) the workflow or the services invoked in it (Cavone et al., 2011).

One of the most important features of our architecture is the presence of a class of interactor agents that are responsible for implementing several kinds of interfaces according to contextual factors and to the user preferences.

2 THE PROPOSED MAS

We propose an agent-based system that supports the user in daily routines but also in handling exceptional situations that may occur. As a main task, the butler must perceive the situation of the house and coordinate the housestaff. To this aim we have designed the following classes of agents:

- **Sensor Agents (SA):** they are used for providing information about context parameters and features (e.g., temperature, light level, humidity, etc.) at a higher abstraction level than sensor data.

Cavone D., De Carolis B., Ferilli S. and Novielli N..
A MULTIAGENT SYSTEM SUPPORTING SITUATION AWARE INTERACTION WITH A SMART ENVIRONMENT.
DOI: 10.5220/0003802800670072
In *Proceedings of the 2nd International Conference on Pervasive Embedded Computing and Communication Systems* (PECCS-2012), pages 67-72
ISBN: 978-989-8565-00-6

- **Butler Agent (BA):** its behavior is based on a combination of intelligent reasoning, machine learning, service-oriented computing and semantic Web technologies for flexibly coordinating and adaptively providing smart services in dynamically changing contexts;
- **Effector Agents (EA):** each appliance and device is controlled by an EA that reasons on the opportunity of performing an action instead of another in the current context.
- **Interactor Agents (IA):** they are in charge of handling interaction with the user;
- **Housekeeper Agent (HA):** it acts as a facilitator among the agents.

All agents are endowed with two main behaviors, reasoning and learning, whose implementation depends on the specific kind of agent.

The reasoning behavior interprets the input and processes it according to its specific role. Due to the complexity of most real-world environments this requires powerful kinds of reasoning and knowledge management, such as deduction, abduction and abstraction (Esposito et al., 2006). We use a logic language to express all the items included in the knowledge base of our agents. In particular, the need to handle relationships among several entities and possible situations calls for the first-order logic setting. An advantage of this setting is that the knowledge handled and/or learned by the system can be understood and checked by humans.

The learning behavior, on the other hand, is used by an agent to refine and improve its future performance through Inductive Logic Programming (ILP) (Muggleton, 1991), which is particularly powerful, and suitable for the specific needs of adaptivity posed by the present application. In fact, ILP allows an incremental approach to learning new information which is mandatory in our case, because the continuous availability of new data and the evolving environment cannot be effectively tackled by static models, but require continuous adaptation and refinement of the available knowledge. An incremental ILP system that is able to exploit different kinds of inference strategies and hence fits the above requirements, is described in (Esposito et al., 2006) also abstraction and abduction theories can be learned automatically (Ferilli et al. 2005).

Regardless of the specific role played by an agent, its behaviors strictly cooperate in the same way. Reasoning uses the agent's knowledge to perform inferences that determine how the agent achieves its objectives. Learning exploits possible feedback on the agent's decisions to improve that knowledge, making the agent adaptive to the specific user needs and to their evolution in time.

These agents coordinate themselves as follows: cyclically, or as an answer to a user action, the butler runs its reasoning model about the user. According to the situation provided by the appropriate SAs, the butler infers and ranks which are the possible user goals and needs. Then, the butler selects the workflow associated with a specific goal by matching semantically the goal with all the Input, Output, Pre-Condition and Effect (IOPE) descriptions (Martin, 2007) of the workflows stored in a workflow repository. Once the most appropriate workflow has been selected and activated, it is necessary to select the services/actions to be invoked among those available in the environment. This process is performed through semantic matchmaking, as well. Therefore, each workflow is planned by initially describing its execution flow and, when needed, the IOPE features of all Web services included in the process. Then, the matchmaker module is responsible of performing the semantic match between the workflow predefined requests and the available semantic Web services, which are listed in a Semantic Web Services Register (SWSR) according to the IOPE standard representation (Meyer, 2007). Hence, the workflow services are invoked dynamically, matching the user's needs in the most effective way (see Section 3.1 for more details). As regards predicates of Web Services, both simple and complex Web Services will be implemented according to the standard Web Ontology Language for Services (OWL-S). We will sometimes refer to the OWL-S ontology as a *language* for describing services that enables automatic service discovery, invocation, composition and execution monitoring. In particular, the composition of complex services from atomic services is based on their pre-conditions and post-conditions.

A detailed description of the agents behaviors and the workflow composition was presented in (Cavone et al., 2011).

3 AGENT CLASSES

This section describes the different agent classes, showing examples that illustrate how they work. The following scenario will be assumed:

It's evening and Jim, a 73 y.o. man, is at home alone. He has a cold and fever. He is a bit bored since he cannot go downtown and drink something with his friend, like he does every evening. Jim is sitting on the bench in his living room in *front of the TV, holding his smartphone. The living room is equipped with sensors, which can catch sound/noise in the air, time, temperature, status of the window (open/close) and of the radio and TV (on/off), and the current activity of the user, and with effectors, acting and controlling windows, radio and TV and also the execution of digital services that may be visualized on communication devices, as for instance the TV. The smartphone allows Jim to remotely control the TV. Moreover, the house is equipped with several interaction devices through which the SHE communicates with the user by implementing different interaction metaphors. Examples of such devices, employed for controlling the house appliances, are the touch screens on the fridge or on mirrors, the smartphone that the users usually brings with him and a socially intelligent robot that is able to move around in the home and to engage natural language dialogue with its inhabitants.*

3.1 Sensor Agents

Sensor Agents are in charge of controlling a set of sensors that are suitably placed in the environment for providing information about context parameters and features (e.g. temperature, light level, humidity, etc.), such as meters to sense physical and chemical parameters, microphones and cameras to catch what happens, indicators of the status of various kinds of electric and/or mechanical devices. The values gathered by the physical sensors are sent in real-time to the reasoning behavior of the associated SA, which uses abstraction to strip off details that are known but useless for the specific current tasks and objectives. For instance, the SA providing information about temperature will abstract the centigrade value into a higher level representation such as “warm”, “cold”, and so on. This abstraction process may be done according to the observed specific user’s needs and preferences (e.g. the same temperature might be cold for a user but acceptable for another). For instance, let us denote the fact that the user Y is cold in a given situation X with cold(X,Y). This fact can be derived from the specific temperature using a rule of the form:

```
cold(X,Y) :- temperature(X,T), T<18, user(Y), present(X,Y), jim(Y).
```

(it is cold for user Jim if he is present in a situation in which the temperature is lower than 18 degrees). In turn, the above rule can be directly provided by an expert (or by the user himself), or can be learned (and possibly later refined) directly from observation of user interaction (Ferilli et al., 2005).

3.2 The Butler Agent

The Butler Agent recognizes user goals starting from percepts received by SAs and composes a smart service corresponding to a workflow that integrates elementary services according to the particular situation.

The reasoning of this agents mainly involves deduction, to draw explicit information that is hidden in the data, and abduction, to be able to sensibly proceed even in situations in which part of the data are missing or otherwise unknown. However, in some cases, it may also use abstraction, which is performed at a higher level than in SAs.

Each observation of a specific situation can be formalized using a conjunctive logic formula under the Closed World Assumption (what is not explicitly stated is assumed to be false), described as a snapshot at a given time. A model, on the other hand, consists of a set of Horn clauses whose heads describe the target concepts and whose bodies describe the pre-conditions for those targets to be detected. For instance, the following model might be available:

```
improveHealth(X) :- present(X,Y), user(Y), has_fever(Y).
improveHealth(X):-
    present(X,Y), user(Y), has_headache(Y), cold(X,Y).
improveHealth(X) :- present(X,Y), user(Y), has_flu(Y).
improveMind(X) :- present(X,Y), user(Y), sad(Y).
improveMind(X) :- present(X,Y), user(Y), bored(Y).
```

On the other hand, a sample observation might be:

```
morning(t0),closedWindow(t0),present(t0,j), jim(j), user(j), temperature(t0,14), has_flu(j), bored(j).
```

Reasoning infers that Jim is cold: cold(t0,j) . Being all the preconditions of the first and fourth rules in the model satisfied by this situation for X = t0 and Y = jim, the user goals improveHealth and improveMind are recognized for Jim at time t0, which may cause activation of suitable workflows aimed at attaining those results.

The BA reasons not only on goals but also on workflows. Indeed, once a goal is triggered, it selects the appropriate workflow by performing a semantic matchmaking between the semantic IOPE description of the user's high-level goal and the semantic profiles of all the workflows available in the knowledge base of the system (W3C, 2004). As a result, this process will produce from zero to *n* workflows that are semantically consistent with the goal, ranked in order of semantic similarity with the goal.

For instance, as shown in Figure 1, the semantic matchmaking process leads to two different workflows associated, respectively, to the two high-level goals improveHealth and improveMind previously recognized. The semantic matchmaking process starts from these goals and leads to the desired workflow.

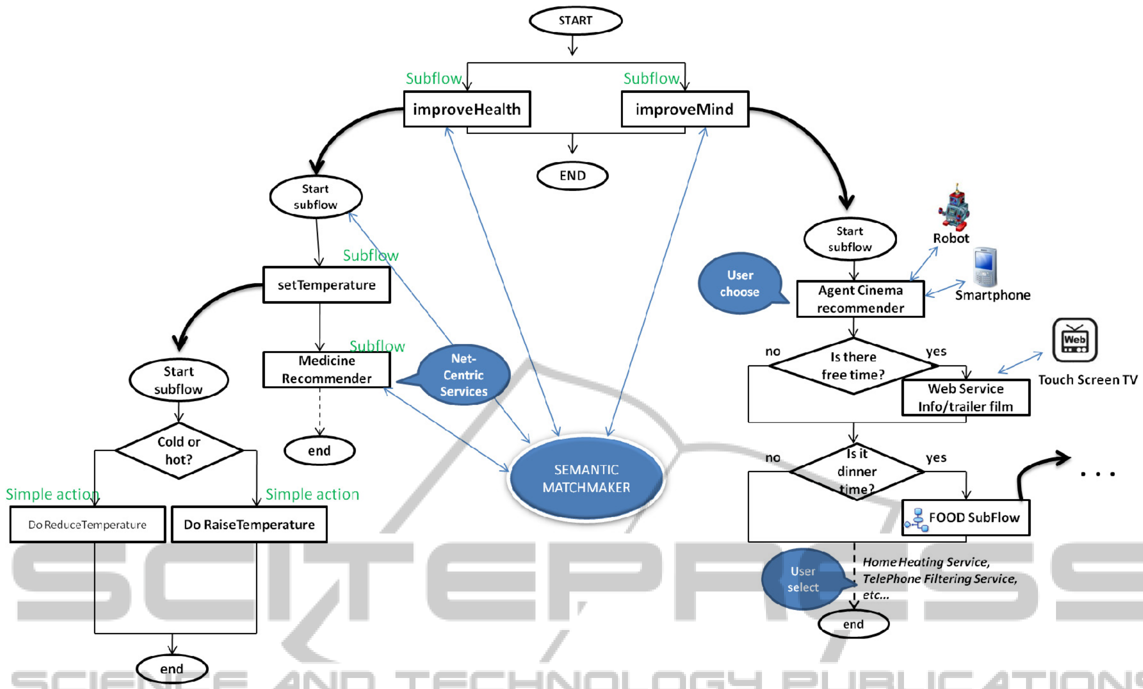

Figure 1: An example of a Smart Service Workflow composed by the Butler Agent.

The semantic matchmaking can also be used within a workflow, to find both the most appropriate subflows and services. In the simplest case, indeed, the best workflow may consist of a sequence of actions. Hence, the behavior implementation allows to deal with complex workflows consisting of a flow of actions and other sub-goals corresponding to subflows, which are again processed according to the matchmaking phase described above. In our example, the main workflow includes two goals that need to be executed by selecting two different subflows corresponding, respectively, to each goal: improveHealth and improveMind. These subflows include both simple actions, that can be directly executed, and subflows that need to be satisfied, such as setTemperature. In this case, the BA will process the information collected by the temperature sensors in order to understand whether to raise or reduce the environment temperature.

This hierarchical matchmaking process stops when the resulting workflow is composed of simple goals that can be directly satisfied by invoking a net-centric service or through simple actions performed on the effectors. In both cases, the BA asks to the HA which EAs can satisfy each planned action and sends a specific request to the EA in charge for handling actions regarding changes of a particular parameter (e.g., temperature, light, etc.). In particular, when the goal satisfied by a workflow (or by part of it) regards a communicative action, its execution is delegated to the IAs. In this specific case, the HA returns to the BA the list of agents that are responsible for implementing the interaction with the user through different modalities (e.g. on a touch screen, on the smartphone or by using the social robot present in the smart environment).

3.3 The Effector Agents

Effector agents are in charge of taking appropriate decisions about actions to be executed on the environment appliances and devices in order to fulfil simple goals determined in the workflow by the BA.

To best satisfy the user needs, these agents reason about different possible solutions to attain the same goal. For instance, if the goal is reducing the temperature, the EA in charge of temperature control may decide whether turning on air conditioning or opening the window; additionally, it decides how to control those devices (in the former case, which fan speed to select; in the latter case, how widely the window must be opened). If the goal is reminding Jim to take medicines and this can be done through a Web service accessible on TV, the EA invokes it.

4 THE INTERACTOR AGENTS

Interactor Agents (IAs) satisfy the goal of interacting with the user. Their behavior mainly consists in executing communicative actions through different interaction modalities and with different communicative goals. IAs choose the most suitable interaction metaphor according to the situation and to the user's needs and preferences. There are several communicative goals that IAs may carry out:

Information Seeking: IAs exploit interaction with the user to get hints on how to attain a simple goal and, based on this, possibly learn new preferences of that user with respect to the given context and situation, in order to continuously and dynamically improve adaptation.

Information Providing: this task involves the delivery of information to the user when he requests explanations about the SHE appliances behavior or about the decision to include some specific subflows in the main workflow built by the BA. E.g., referring to the previous scenario, the user may ask the robot to provide more justification for choosing a given medicine. In this case the social robot may intervene explaining to the user why that drug is suited for his problem by showing the advantages of taking it. This task may be seen as a simpler version of the persuasion task described in (De Carolis et al., 2009).

Remind: this task may be implemented through different interaction devices. The BA may choose the more appropriate one according to the user preferences and to the specific situation. For example, if the object of the reminder is to take a medicine, it might be useful to provide a reminder on the smartphone.

4.1 Exploiting Pervasive Interaction

The general interaction behavior of the IA is implemented according to the specific appliances to which the IA instances are associated. The interaction devices are distributed in the environment and are selected according to the specific interaction task to be carried out, the context (e.g. how close is the device with respect to the position of the user in the SHE), and the user preferences.

Let's refer to the scenario described earlier. Jim is sitting in front of the TV and is bored, hence the BA builds a workflow in order to satisfy the ImproveMind goal. Starting from this data and the information describing the situation, the SHE infers that a possible goal of Jim is to WatchTV. This will be obtained as a subflow of ImproveMind through the semantic matchmaking process, as described in the previous section. It selects the workflow, shown in Figure 1, for satisfying this goal according to the context condition.

The BA starts the execution of the workflow and, as a first service, it recommends to Jim a set of movies that could be of interest to him . Jim may accept the proposed service, or refuse it and select another one. This interactive task is delegated by the BA to the IA associated to the smartphone, since it is the device that Jim can access immediately. Let's suppose that Jim accepts and selects the recommended movie. Then, since it is almost dinner time, the smart service recommends his favourite take-away food, sushi. In this way, using the interface on TV, Jim may evaluate the agent proposal. He may accept, refuse or modify the proposed services choosing among several other services that are available in that situation.

Robot: 'Hi Jim, how are you doing?'

Jim: 'I think I have got a flu, I think I need some medicine'

R: 'Do you have fever or is just a cold? I would suggest you take a pill of paracetamol'

J: 'Are you sure? I don't feel like having fever, I rather have some headeache and sore throat'

R: 'Paracetamol is ok for headeache. But if you prefer, you can take a pill of anti-inflammatory, which could solve both problems'

Figure 2: An example of interaction with the social robot.

Another example is provided by the subflow called 'Medicine Recommender' included in the workflow in Figure 1. It may be satisfied differently according to the specific situation of the user. Let's suppose, for example, that the information about the user's disease is not complete, that is the BA knows that the user needs to improve his health because he has got a flu but it is necessary to decide which is the most suitable drug for him to take. In such a situation, it is necessary to further investigate the user's physical state in order to select the most appropriate drug to suggest him. In Figure 2 we provide an example of the interaction with the social intelligent robot present in the house.

5 CONCLUSIONS AND FUTURE WORK

This contribution shows a preliminary work towards the development of a MAS aiming at handling the situation-aware adaptation of a Smart Environment behavior. In this MAS different types of agents cooperate to the adaptation process, which is performed at different levels, starting from the interpretation of sensor data from Sensor Agents, planning services satisfying the recognized user's goals and arriving to the decision on how to act on devices from Effector Agents. Pervasive interaction with the user is implemented through the Interactor Agents' behavior, which adapts the choice of the most appropriate interaction methapor to the context and to the user preferences and needs.

Still, open problems remain and will be the subject of our future work. An open issue regards how to reason on users' reactions to the proposed flow of activities in order to adopt the optimal behavior of the SHE. In fact, when the user undoes or gives a negative feedback to one or more actions of the selected workflow, it is necessary to understand if this is just an exception or if it must affect the reasoning models, e.g. because there is: (i) a change in the situation that has not been detected or taken into account, (ii) a mistake in controlling the effectors to achieve a simple goal, (iii) a mistake in interpreting the user's goals or in selecting or composing the workflow.

Each of the latter cases determines which agent in the MAS has made a wrong decision, and is to be involved in theory refinement. Identification of the specific case should be obtained by an analysis of the user's feedback, and introduces a related issue, that is who is in charge of identifying the problem, gathering the feedback and notifying it to the proper agent that must activate its learning behavior. A candidate for taking care of these activities is the Interactor Agent, because it is equipped with the skills necessary for the execution of communicative goals.

Finally, in the near future we plan to collect more examples of interaction with the system to simulate and evaluate its behavior in all the possible situations that are relevant for our application domain.

REFERENCES

Cavone, D., De Carolis, B., Ferilli, S., Novielli, N., 2011. An Agent-based Approach for Adapting the Behavior of a Smart Home Environment. In *Proceedings of WOA 2011,* http://ceur-ws.org/Vol-741/ID14_CavoneDeCarolisFerilliNovelli.pdf

De Carolis, B., Cozzolongo, G. and Pizzutilo, S., 2006. A Butler Agent for Personalized House Control. *ISMIS*: 157-166.

De Carolis, B. Mazzotta, I. Novielli, N., 2009. NICA: Natural Interaction with a Caring Agent. In *Ambient Intelligence, European Conference, AmI 2009*, Salzburg, Austria, November 18-21, vol. 5859/2009, pages 159-163, ISBN: 978-3-642-05407-5, ISSN: 0302-9743 (Print) 1611-3349 (Online), DOI: 10.1007/978-3-642-05408-2_20, Springer Berlin / Heidelberg, http://www.springerlink.com/content/4636294560388663/ 2009.

Esposito, F., Fanizzi, N., Ferilli, S., Basile, T. M. A. and Di Mauro, N., 2006. Multistrategy Operators for Relational Learning and Their Cooperation. *Fundamenta Informaticae Journal*, 69(4):389-409, IOS Press, Amsterdam.

Falcone, R. and Castelfranchi, C, 2001. Tuning the Collaboration Level with Autonomous Agents: A Principled Theory. *AI*IA*: 212-224.

Ferilli S., Basile T. M. A., Di Mauro N., Esposito F., 2005. On the LearnAbility of Abstraction Theories from Observations for Relational Learning. In: *J. Gama, R. Camacho, G. Gerig, P. Brazdil, A. Jorge, L. Torgo. Machine Learning: ECML 2005*. P. 120-132, Berlin: Springer, ISBN/ISSN: 3-540-29243-8.

Ferilli S., Esposito F., Basile T. M. A., Di Mauro N., 2005. Automatic Induction of Abduction and Abstraction Theories from Observations. In: *S. Kramer, B. Pfahringer. Inductive Logic Programming. p. 103-120,* Berlino: Springer, ISBN/ISSN: 3-540-28177-0.

Kim J. et al., 2004. An Intelligent Assistant for Interactive Workflow Composition. *9th international conference on Intelligent User Interfaces*, pp:125-131, ACM Press.

Meyer, H., 2007. On the Semantics of Service Compositions. *Lecture Notes in Computer Science, Volume 4524/2007*, 31-42.

Muggleton, S. H., 1991. Inductive Logic Programming. *New Generation Computing, 8(4):295-318*.

Martin, D. Burstein, M. McDermott, D., McIlraith, S. Paolucci, M., Sycara, K., McGuinness, D., Sirin, E. and Srinivasan, N., 2007. "Bringing Semantics to Web Services with OWL-S", *World Wide Web Journal,* Vol. 10, No. 3.