A Cognitive Architecture for Ambient Intelligence Systems

Article

Accepted version

Vincenzo Agate, Pierluca Ferraro, Salvatore Gaglio

International Workshop on Artificial Intelligence and Cognition, 6th Edition, Palermo, 2018 July 2-4

A Cognitive Architecture for Ambient Intelligence Systems

Vincenzo Agate, Pierluca Ferraro, and Salvatore Gaglio

University of Palermo, Viale delle Scienze, ed 6. 90128 Palermo, Italy
{vincenzo.agate,pierluca.ferraro,salvatore.gaglio}@unipa.it

Abstract. Nowadays, the use of intelligent systems in homes and workplaces is a well-established reality. Research efforts are moving towards increasingly complex Ambient Intelligence (AmI) systems that exploit a wide variety of sensors, software modules and stand-alone systems. Unfortunately, using more data often comes at a cost, both in energy and computational terms. Finding the right trade-off between energy savings, information costs and accuracy of results is a major challenge, especially when trying to integrate many heterogeneous modules. Our approach fits into this scenario by proposing an ontology-based AmI system with a cognitive architecture, able to perceive the state of the surrounding environment, to reason on the current situation and act accordingly to modify the state of the environment based on the user's preferences.

Keywords: Ambient Intelligence · Cognitive Architecture · Self-modeling.

1 Introduction

Many Ambient Intelligence (AmI) systems are characterized by the use of a wide variety of devices that provide information to them at a cost, in energy and computational terms, which may or may not be appropriate to the importance of the information itself. The abundance of information almost always leads to an increase in the accuracy of reasoning, but also to the creation of systems that are too complex and inefficient in terms of energy and computational effort.

Today, these systems not only use physical sensors to collect data, but increasingly also use entire autonomous software subsystems that can be considered as sophisticated virtual sensors. The use of these modules enriches the available context information, in order to better perceive the heterogeneous and dynamic environmental conditions of the surrounding environment. For example, an app installed on a smartphone that recognizes user activities could be considered as a software sensor by the AmI system. Such high-level information can be combined with other data collected by hardware and software sensors, and used to adapt the environmental conditions to those desired by the user [3]. This will allow the inference system to reason not only with low level information, but also with higher level concepts, achieving better results with less computational effort. Moreover, the same input data, processed by different software modules, can produce completely different results, whose combination generates new knowledge. Additionally, the same raw data acquired by the sensors can lead to information of different quality, based on the maximum cost that the system is willing to incur and, therefore, the quantity and complexity of the software modules that can be used.

This raises another problem related to the convenience of obtaining additional information, while also considering the costs of obtaining it. The level of dynamism and uncertainty of the surrounding environment could in fact influence the choices of the system on how to acquire information. Therefore, in uncertain contexts, where the user interacts continuously with the environment and the AmI system is not sure about what is happening, it could be encouraged to acquire further knowledge to decrease its uncertainty and become more self-confident, even at the expense of incurring higher costs. Conversely, if the system is reasonably sure of what is happening, it may not be worthwhile to acquire further data, as this would not significantly change its level of uncertainty but would still lead it to spend more.

In this work we propose an AmI system which takes inspiration from some typical characteristics of humans, such as self-awareness, in order to adapt the environmental conditions to the specific needs of users, by exploiting its dynamically updated knowledge about the external world and its own subsystems.

In recent years, many research efforts in the field of AmI have focused on the development of autonomous systems, able to reason about their own structure as well as the external world [1]. Similar to [13], our system will maintain and constantly update a model of the environment and one of itself, so as to allow its self-configuration and self-management. One of the main characteristics of our system is its capability to maintain a cognitive representation of its own perceiving process, learning how to improve its perception in order to understand the surrounding environment [19, 8]. Creating autonomous systems has often been a challenge for researchers, especially as regards the use of cognition to achieve minimal autonomous behaviour [20]. In this regard, many artificial intelligence techniques have been exploited over time, such as genetic algorithms [17], reinforcement learning techniques (e.g., Q-learning [10]), and Bayesian networks [11].

The remainder of this paper is organized as follows. Section 2 introduces the multi-layered architecture of the proposed system, providing a high-level description of its modules. Section 3 describes the self-reasoning capabilities of our system, which are used for the dynamic reconfiguration of its subsystems. Finally, Section 4 presents our conclusions.

2 Proposed Architecture

In this paper, we propose an AmI system with a cognitive architecture, able to perceive the surrounding environment, to reason on the current context and to act directly to modify the environment in an appropriate way [6]. We aim to design and develop an AmI system that is self-configuring and self-managing, taking inspiration from some of the peculiar characteristics of humans, such as self-awareness, so as to make the architecture more resilient and context-aware.

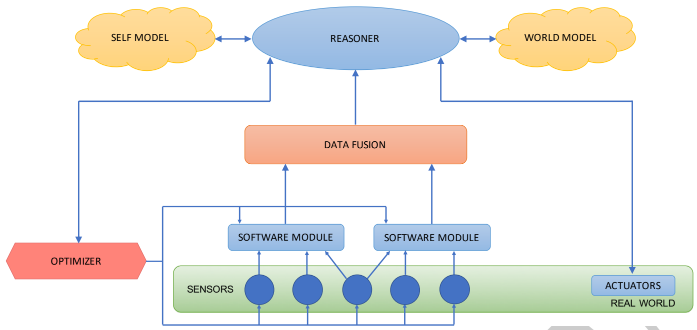

Fig. 1. Proposed architecture.

Our system is able to fully exploit the continuously updated information contained within two ontologies (*self-model* and *world-model*), which form its knowledge base. In this way, the system can monitor the status of all its software sub-modules [9], dynamically adapting its behavior to the current context, in order to achieve an optimal trade-off between performance and costs.

The system relies on a distributed sensory infrastructure composed of heterogeneous devices to gather raw data. Such data are processed through several software modules that employ artificial intelligence and machine learning techniques to aggregate them and to infer high-level information. Such knowledge is then exploited to deeply understand the system status and that of its surroundings. Such understanding process is exploited to adapt the system behavior to the current context, by dynamically choosing the subset of sensors and software modules to exploit in the inference process. Such a selection allows the system to activate only the components strictly necessary to achieve its goals, rather than using all data sources. As demonstrated in [4], exploiting all sensory and contextual information available can be detrimental, both in terms of costs and in terms of accuracy of reasoning, since unnecessary information would only add noise to the data.

This approach aims to reduce the cognitive load of the system, allowing it to focus only on the most important aspects at any given time. Moreover, this allows reducing the energy consumption, while maintaining a high degree of accuracy in its reasoning.

Figure 1 shows the multi-layered architecture of the system. The lowest level is composed of sensors and actuators, which are the system's connection points with the outside world. Such heterogeneous level includes, for example, Wireless Sensor Networks [12] and motion sensors [2, 15]. Raw data collected by sensors are sent to dedicated software modules, which use specific data fusion and machine learning techniques to process them. Several software modules can be used in parallel, in cascade or in competition with each other, depending on the situation. If, for example, there are several modules that infer the same concept, the system selects the best combination of these to ensure the optimal trade-off between accuracy and cost, as will be shown in Section 3. Each software module, however complex, can be considered as a virtual sensor, which infers high-level concepts. The output of such heterogeneous software sensors is merged by a probabilistic multisensor *data fusion* module, which produces a *belief* of the system on what is happening in the surrounding environment.

The *reasoner* evaluates the quality of the inference process perfomed by the *data fusion* module, adopting opportune quantitative quality indexes (i.e., the uncertainty of reasoning and the cost of the inference while using the current sensors and software modules). Such evaluation takes into account the knowledge contained in the ontologies (e.g., the cost function associated with each component, as will be described in Section 3). The resulting quality evaluation is then sent to the *self-optimizer*. The *self-optimizer* estimates the contribution of each sensor and software module to the current inference process and, therefore, the importance of each component, as will be described in Section 3. Using a multi-objective optimization technique, the optimizer identifies the best trade-off between accuracy and costs, and decides which sensors and software modules should be used to ensure optimal performance.

Finally, the *reasoner* uses information coming from the *data fusion* module, along with knowledge regarding the state of the surrounding environment contained in the *world model* ontology, to plan the best sequence of actions to satisfy the user's needs, based on his preferences, according to the principles of AmI. As a result, a series of messages are sent to actuators, which will effectively change the state of the environment.

3 Cognitive Self-Reasoning

The *data fusion* system is based on a Dynamic Bayesian Network (DBN), which is a particular type of Bayesian Network that takes into account past observations, and it is thus suitable for reasoning on dynamic problems [16].

A DBN is composed of slices: each slice represents the state of a phenomenon in a particular instant. A slice can be composed of an arbitrary number of hidden nodes (the objective of the inference, such as the activity performed by the user) and evidence nodes (e.g., sensory readings) [7].

As mentioned in the previous section, software modules are considered as sophisticated virtual sensors. Their outputs are treated as evidence nodes of the DBN, and are used to infer a higher level concept, such as user activity.

At any given moment, the DBN output is a probability distribution that represents the belief of the system about the phenomenon under observation. In particular, we can define the belief on a specific system state x_t, at time slice t, as $Bel(x_t) = P(x_t|\boldsymbol{E}_{1:t})$, where $\boldsymbol{E}_{1:t}$ is the set of sensory readings collected from the beginning up to time slice t. Such equation seems to be dependent on all the observations made from the beginning, and therefore not practical to use. However, as demonstrated in [5], the same equation can be expressed by the following recursive formula:

$$Bel(x_t) = \eta \cdot \prod_{e_t^i} P(e_t^i|x_t) \cdot \sum_{x_{t-1}} P(x_t|x_{t-1}) \cdot Bel(x_{t-1}), \tag{1}$$

where η is a normalization constant and e_t^i represents the reading of sensor i at time t. The *reasoner* extracts two features from the output of the *data fusion* module, namely the uncertainty of the reasoning and the cost of the inference.

The uncertainty is calculated using a metric based on Shannon's entropy [18], with the following formula:

$$U = -\sum_{x_t} Bel(x_t) \log_2(Bel(x_t)). \tag{2}$$

The higher this value, the more uncertain the system is about what is happening in the outside world. By using the software modules and sensory data most appropriate for the current situation, it will be possible to decrease it, thus making the system more self-confident and more aware of the surrounding environment. Since the sensors and software modules used by the system are heterogeneous, each will be characterized by a different cost function, specified within the ontology of the *self-model*. For example, the cost of a software module could be the computational effort, while the cost of a sensor may be related to its energy consumption. The total cost of a configuration is the sum of the costs of the active components at the current time.

Generally, reducing costs and uncertainty are conflicting goals, because decreasing one means raising the other. When cost or uncertainty exceed certain thresholds, the *reasoner* relies on the *self-optimizer* to find the best trade-off between these two conflicting goals. The *self-optimizer* selects which software modules and sensors to use based on their importance for the current inference. To this end, we propose a metric to estimate the information gain obtained using a particular component in the current context. This metric is based on the Kullback-Leibler (KL) divergence [14], also known as relative entropy. The KL divergence measures the information that is lost when using an approximate distribution G instead of the true distribution F, according to the following formula:

$$D_{KL}(F\|G) = \sum_{i} F(x_t) \log\left(\frac{F(x_t)}{G(x_t)}\right). \tag{3}$$

The information gain of an active component of the AmI system can be calculated as the KL divergence between the belief that takes into account the output returned by the component and the belief that ignores it.

Calculating the information gain of a component that is not active at the current time is more difficult, since its exact evaluation would involve an estimate that takes into account all the possible values obtainable as output of the component itself. To reduce the computational cost of the *self-optimizer*, we have therefore decided to approximate this metric using a heuristic, which takes into account the history of the values of information gain of each component, updating them only when that component was active. As demonstrated in [5], this heuristic is a very effective estimate of the proposed metric and can be calculated with a low computational cost. The information gain values calculated by the *self-optimizer* are eventually returned to the *reasoner*, which updates the *self-model* ontology appropriately. The *self-optimizer* will then be able to select the physical sensors and software modules best suited to each situation by using a multi-objective optimization criterion. As an example of high-level inference that could be carried out by the *reasoner*, using data from low-level modules and the *self-optimizer*, we consider the application scenario of a smart home, in which the AmI system has to recognize the daily activities carried out by two users. Recognizing activities in a multi-user scenario is very challenging, and our system can select the best components to use in different situations.

For example, suppose that the system infer that the two users are in different rooms, exploiting a software module for presence detection (e.g., one user may be in the bedroom, while another may be in the kitchen preparing breakfast). In this case, the only additional difficulty, compared to the single user scenario, is to figure out which user is in which room. To do this, the system could use the Bluetooth module on users' smartphones, and locate them using beacons pervasively deployed throughout the smart home, which provide an estimate of the position of users with a room-level granularity.

Instead, if both users are in the same room (e.g., in the living room), exploiting Bluetooth beacons may no longer be sufficient. In this case, the uncertainty of the system would increase, and the *reasoner* may decide to use additional software modules to understand what activity is carried out by each user. For example, if the two activities were *watching TV* and *exercising with a treadmill*, the software module with the greatest information gain could be one that uses the accelerometer and gyroscope in users' smartphones, allowing the system to easily discriminate between a stationary activity and a physical one.

4 Conclusions

In this paper, we have presented the cognitive architecture of an AmI system, which is characterized by its self-reasoning and self-optimizing capabilities. These features allow the system to accurately perceive the state of the surrounding environment, adapting to the context and actively modifying the state of the environment based on the users preference. Our system uses the knowledge contained within two ontologies to model the state of the world and to be aware of its own hardware and software structure.

Moreover, the system is able to reason using concepts at different levels of abstraction simultaneously, merging information from physical sensors and very complex software modules, which are considered to all intents and purposes as virtual sensors. In order to achieve a good trade-off between cost and accuracy, the *self-optimizer* module is able to identify the best configuration of components to use in any situation, based on the system's goals and current context.

References

1. Barandiaran, X., Moreno, A.: On what makes certain dynamical systems cognitive: A minimally cognitive organization program. Adaptive Behavior **14**(2), 171–185 (2006)

2. Cottone, P., Lo Re, G., Maida, G., Morana, M.: Motion sensors for activity recognition in an Ambient-Intelligence scenario. In: Pervasive Computing and Communications Workshops, 2013 IEEE Int. Conf. on. pp. 646–651. IEEE (2013)
3. De Paola, A., Farruggia, A., Gaglio, S., Lo Re, G., Ortolani, M.: Exploiting the human factor in a WSN-based system for Ambient Intelligence. In: Int. Conf. on Complex, Intelligent and Software Intensive Systems. pp. 748–753. IEEE (2009)
4. De Paola, A., Ferraro, P., Gaglio, S., Lo Re, G.: Context-awareness for multi-sensor data fusion in smart environments. In: Adorni, G., Cagnoni, S., Gori, M., Maratea, M. (eds.) AI*IA 2016 Advances in Artificial Intelligence. pp. 377–391 (2016)
5. De Paola, A., Ferraro, P., Gaglio, S., Lo Re, G., Das, S.K.: An adaptive bayesian system for context-aware data fusion in smart environments. IEEE Trans. on Mobile Computing **16**(6), 1502–1515 (2017)
6. De Paola, A., Gaglio, S., Lo Re, G., Ortolani, M.: An Ambient Intelligence architecture for extracting knowledge from distributed sensors. In: Proc. of the 2nd Int. Conf. on Interaction Sciences: Information Technology, Culture and Human. pp. 104–109. ACM (2009)
7. De Paola, A., La Cascia, M., Lo Re, G., Morana, M., Ortolani, M.: User detection through multi-sensor fusion in an AmI scenario. In: 15th Int. Conf. on Information Fusion. pp. 2502–2509. IEEE (2012)
8. De Paola, A., La Cascia, M., Lo Re, G., Morana, M., Ortolani, M.: Mimicking biological mechanisms for sensory information fusion. Biologically Inspired Cognitive Architectures **3**, 27–38 (2013)
9. De Paola, A., Lo Re, G., Morana, M., Ortolani, M.: Smartbuildings: an AmI system for energy efficiency. In: Sustainable Internet and ICT for Sustainability (SustainIT), 2015. pp. 1–7. IEEE (2015)
10. Erbas, M.D., Winfield, A.F., Bull, L.: Embodied imitation-enhanced reinforcement learning in multi-agent systems. Adaptive Behavior **22**(1), 31–50 (2014)
11. Ferreira, J.F., Castelo-Branco, M., Dias, J.: A hierarchical bayesian framework for multimodal active perception. Adaptive Behavior **20**(3), 172–190 (2012)
12. Gatani, L., Lo Re, G., Ortolani, M.: Robust and efficient data gathering for wireless sensor networks. In: System Sciences, 2006. HICSS'06. Proc. of the 39th Annual Hawaii Int. Conf. on. vol. 9, pp. 235a–235a. IEEE (2006)
13. Kawamura, K., Dodd, W., Ratanaswasd, P., Gutierrez, R.: Development of a robot with a sense of self. In: Proc. of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA 2005). pp. 211–217. IEEE (2005)
14. Kullback, S., Leibler, R., et al.: On information and sufficiency. The Annals of Mathematical Statistics **22**(1), 79–86 (1951)
15. Lo Re, G., Morana, M., Ortolani, M.: Improving user experience via motion sensors in an Ambient Intelligence scenario. In: PECCS. pp. 29–34 (2013)
16. Murphy, K.P.: Dynamic Bayesian Networks: representation, inference and learning. Ph.D. thesis, University of California, Berkeley (2002)
17. Ram, A., Boone, G., Arkin, R., Pearce, M.: Using genetic algorithms to learn reactive control parameters for autonomous robotic navigation. Adaptive Behavior **2**(3), 277–305 (1994)
18. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review **5**(1), 3–55 (2001)
19. Vakarelov, O.: The cognitive agent: Overcoming informational limits. Adaptive Behavior **19**(2), 83–100 (2011)
20. Van Duijn, M., Keijzer, F., Franken, D.: Principles of minimal cognition: Casting cognition as sensorimotor coordination. Adaptive Behavior **14**(2), 157–170 (2006)