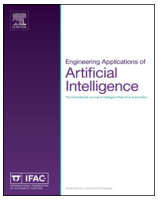
Engineering Applications of Artificial Intelligence 45 (2015) 269–280

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



A cognitive robotic ecology approach to self-configuring and evolving AAL systems

Mauro Dragone [a,1], Giuseppe Amato [b], Davide Bacciu [c], Stefano Chessa [c], Sonya Coleman [d], Maurizio Di Rocco [e], Claudio Gallicchio [c], Claudio Gennaro [b], Hector Lozano [f], Liam Maguire [d], Martin McGinnity [d], Alessio Micheli [c], Gregory M.P. O'Hare [g,*], Arantxa Renteria [f], Alessandro Saffiotti [e], Claudio Vairo [b], Philip Vance [d]

[a] *Trinity College Dublin, Dublin, Ireland*
[b] *ISTI-CNR, Pisa, Italy*
[c] *Università di Pisa, Pisa, Italy*
[d] *University of Ulster, Derry, UK*
[e] *Örebro Universitet, Örebro, Sweden*
[f] *Tecnalia, Bilbao, Spain*
[g] *University College Dublin, Ireland*

ARTICLE INFO

Article history:
Received 26 December 2014
Received in revised form
22 April 2015
Accepted 1 July 2015
Available online 30 July 2015

Keywords:
Robotic ecology
Ambient assisted living
Cognitive robotics
Machine learning
Planning

ABSTRACT

Robotic ecologies are systems made out of several robotic devices, including mobile robots, wireless sensors and effectors embedded in everyday environments, where they cooperate to achieve complex tasks. This paper demonstrates how endowing robotic ecologies with information processing algorithms such as perception, learning, planning, and novelty detection can make these systems able to deliver modular, flexible, manageable and dependable Ambient Assisted Living (AAL) solutions. Specifically, we show how the integrated and self-organising cognitive solutions implemented within the EU project RUBICON (Robotic UBIquitous Cognitive Network) can reduce the need of costly pre-programming and maintenance of robotic ecologies. We illustrate how these solutions can be harnessed to (i) deliver a range of assistive services by coordinating the sensing & acting capabilities of heterogeneous devices, (ii) adapt and tune the overall behaviour of the ecology to the preferences and behaviour of its inhabitants, and also (iii) deal with novel events, due to the occurrence of new user's activities and changing user's habits.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Current smart home solutions utilize sensors and microprocessors pervasively deployed throughout the home to collect information which is then exploited to perform useful tasks, as monitoring the daily activity, safety, health and security of the house occupants.

One of the key drivers of this type of solutions is the need to assist the ageing population, and address the associated social impact due to the increase on the cost of formal health care and the shortage of caregivers by delivering Ambient Assisted Living (AAL) services.

Notwithstanding its potential benefits to the user life style, such as allowing elderly persons to live independently in their own homes and postponing or perhaps even avoiding a potential move to a residential care facility, the technology is far from reaching widespread adoption by consumers. Commercial products are mostly limited to basic monitoring services and suffer from a low user's acceptance rate due to their poor reliability and still prohibitive costs, which often include installation, certification, testing, customization charges and monthly monitoring fees. The perceived value and trust in this type of solution can be seriously undermined and curbed by constant monitoring and by ill-advised or poorly-

[*] Corresponding author.

E-mail addresses: dragonem@tcd.ie (M. Dragone), giuseppe.amato@isti.cnr.it (G. Amato), bacciu@di.unipi.it (D. Bacciu), ste@di.unipi.it (S. Chessa), sa.coleman@ulster.ac.uk (S. Coleman), maurizio.di-rocco@aass.oru.se (M.D. Rocco), gallicch@di.unipi.it (C. Gallicchio), claudio.gennaro@isti.cnr.it (C. Gennaro), hector.lozano@tecnalia.com (H. Lozano), lp.maguire@ulster.ac.uk (L. Maguire), tm.mcginnity@ulster.ac.uk (M. McGinnity), micheli@di.unipi.it (A. Micheli), gregory.ohare@ucd.ie (G.M.P. O'Hare), arantxa.renteria@tecnalia.com (A. Renteria), asaffio@aass.oru.se (A. Saffiotti), claudio.vairo@isti.cnr.it (C. Vairo), p.vance@ulster.ac.uk (P. Vance).

[1] Work done while the first author was at University College Dublin, Ireland. He is currently affiliated with the Trinity College Dublin, Ireland.

http://dx.doi.org/10.1016/j.engappai.2015.07.004
0952-1976/© 2015 Elsevier Ltd. All rights reserved.

timed interventions, especially when they are based on pre-programmed models of user behaviour and/or by relying solely on knowledge that was imbued in the system at design time.

A key source of the complexity associated with effective smart environments in general is the inherent heterogeneity of the end-user population, their housing arrangements, their individual situations and requirements. On one hand, complete off-the-shelf systems that successfully addresses such disparate requirements are unlikely to materialise. On the other hand, customized solutions for individual circumstances do not have the ability to respond to the users' in-situ response or smoothly adapt to changing contexts and evolving users' needs, habits and preferences.

We argue that maximum gain will be attained only by factoring the need for evolution and adaptivity into the design and implementation of smart environments. These should not only be robust and easily maintainable, but must also be able to self-configure and even evolve to meet the requirements of individuals as their needs and circumstances change. Adaptation should be supported both in the way these systems adapt to circumstances in the physical world they inhabit, and in the modular way they combine existing approaches and use available components and computational resources.

The traditional devices found in a smart environment are being increasingly augmented by the rapid introduction of robots in homes. The service robot market, including domestic robots, is predicted to experience a ten-fold growth in the next 20 years (http://www.ifr.org/service-robots/statistics, 2014). Evidence of their role in domestic settings range from autonomous but specialized products, such as the Roomba vacuum cleaner (http://www.irobot.ie/Roomba,); to telepresence robotic healthcare solutions (e.g. Coradeschi et al., 2013); and to the networked robots being validated in the AAL domain (http://www.robot-era.eu/robotera,).

The view presented in this paper is that the increasing addition of networked devices, cheap robotic hardware, wireless sensors, actuators and other smart artefacts to our environment provides a powerful impetus for building assistive smart environments through a cognitive robotic approach. Specifically, Robotic Ecologies (Saffiotti and Broxvall, 2005) can provide complex services with their ability to acquire and apply knowledge about their environment, exploit the flexibility given by robots' mobility, and coordinate actions and information exchange between heterogeneous, and simple devices. Recent advancements developed within the EU project RUBICON (Robotic UBIquitous COgnitive Network http://www.fp7rubicon.eu, 2014; Amato et al., 2012) make these system able to pro-actively assist the users in their activities by learning to adapt to changes in the environment and in users' habits.

The remainder of this paper is organized in the following manner: Section 2 discusses the benefits of a robotic ecology design. It also reviews relevant related work and outlines the challenges associated to build integrated systems and re-use many of the machine learning and artificial intelligence techniques already proposed in the literature. Section 3 outlines the architecture of the integrated system developed in RUBICON, before examining in detail each of its components. Section 4 illustrates how the interplay between these components can be harnessed to deliver a family of assistive services, while allowing to adapt to dynamic situations and also to newly discovered habits and user activities. To this end, we provide an exemplification of the cognitive ecology operation within a real-world case study. The case study is documented with supplemental material, attached to this paper, in the form of a video that captures all the key phases of our experiments.

2. Cognitive robotic ecologies

In Saffiotti et al. (2008), Saffiotti and Broxvall discuss the implications of their PEIS-Ecology instantiation of the Robotic Ecology approach from an ecological point of view, by conceiving the interaction between each device and its environment in terms of mutuality and reciprocity. The devices in the ecology achieve complex tasks by performing several steps in a coordinated fashion while also exchanging sensor data and other useful information in the process.

Especially when combined with basic human–robot interaction (HRI) capabilities, even simple domestic robots may be used to support a number of useful and modular assistive services in this way. Consider these simple examples. A robotic ecology can monitor the strength of the radio signals emitted by wireless sensors pervasively embedded in the environment in order to localize both the robots and the users. By analysing these signals, a robot can inform the elderly person who lives alone that she has forgotten to switch off the stove, or that the main door is not locked when she goes to sleep. A robotic ecology with multiple robots may employ a vacuum cleaning robot to help the user clean the apartment, while another robot may help the user to manage and monitor her health conditions. For example, the robot may remind the user to take her medicine, when the sensors in the pill dispenser and those located in the kitchen show that the user had not taken her prescribed pills after her meal. The robot may also prompt and motivate the user to maintain a healthy and active life-style, for instance, by reminding her to keep rehydrated, perform her daily exercise routine, and control her heart rate afterwards.

Noticeably, the embodied nature of its devices is what makes confronting an ecological view in a robotic ecology characteristically different from what is usually done in pure software systems. Most importantly, there is no sensing and interaction with the physical world in the domains usually addressed with orchestration and dynamic composition of web-services. In addition, self-configuration and self-adaptation are not the main focus of other approaches related to Robotic Ecologies. In particular, works in Cyber-Physical Systems (CPS) put more emphasis on formal models and large scale deployments, while works in Internet of Things (IoT) systems put more emphasis on knowledge services than on action services.

Building smart domestic spaces with a Robotic Ecology design reduces application complexity and costs, and enhances the individual values of the devices involved. However, robotic ecologies are not dissimilar to other smart environments in their basic requirements for assessing the state and the needs of their users, and deciding how to best assist them.

First of all, machine learning techniques for human activity recognition (HAR) can be harnessed to merge and interpret the information gathered by multiple noisy sensors (Tapia et al., 2004; Ye et al., 2012; Alam et al., 2012; Roggen et al., 2011; Puteh et al., 2013) in order to determine the state of the user, and possibly predict their behaviour. For instance, by exploiting the stream of sensor data gathered from pressure sensors placed under the cushions of the sofa, and a networked bracelet equipped with a 3-axis accelerometer, a HAR system may be trained to recognise when the user is exercising and also distinguish that event from the one in which the user is relaxing on the sofa. A number of initiatives have successfully exploited exploratory data analysis techniques to discover users' behavioural patterns (Cook and Krishnan, 2014). The same systems can be used to detect anomalies in users' behaviour, such as deviation from daily routines which may be symptoms of problems worthy of notification to carers or to the users directly (Jakkula and Cook, 2011).

Secondly, pre-programmed domain knowledge, in terms of service rules stating relations that a human domain expert has identified between sensor data and/or inferred context, and robot services, may be used at this point to know when to provide contextualized assistance to the users (Fratini et al., 2008). In the active life-style scenario above, these techniques may be used to define a rule such as *remind the user to measure her heart rate five minutes from the end of her daily exercise routine*. Dedicated user interfaces can also be exploited to involve the final user in the definition of personalised service rules (Saunders et al., 2013). Alternatively, Q-learning or other adaptation policies based on utility maximization can be exploited to automatically learn useful automation patterns (Tapia et al., 2004; Rashidi, 2009; Aztiria et al., 2013). The user can be involved also in this adaptation process, for instance, with user interfaces enabling them to manually activate or rate their automation preferences and the ones automatically selected by the system (Rashidi, 2009).

Finally, plan-based control approaches (Saffiotti and Broxvall, 2005; Lundh et al., 2007) can be used to decide which robotic devices and/or software components must be activated, what they need to achieve, and what information they should exchange in the process in order to provide context-aware assistance to the user. One of the key advantages of using planning techniques with robotic ecologies is the possibility of using alternative means to accomplish application goals when multiple courses of action are available, to increase the robustness of the overall system. For instance, a robot may be able to localise itself through the use of its on-board laser sensor, but also by detecting RFID tags placed under the floor, if the more precise laser-based localization fails for some reason (Bacciu et al., 2014a).

Instantiating the right combination of the approaches discussed above is not a straightforward decision. Each of them can suffer from a number of limitations and there are difficulties also in the way they are usually combined, especially if they need to be applied in a robotic ecology context.

First of all, the majority of HAR solutions relies exclusively on supervised information available previously to system deployment, which demands time consuming annotation of users' activities to build up labelled training examples. Besides the cost and the technical problems associated with data annotation, such an approach assumes that only activities for which training data was available before system deployment will be recognized in the future. This makes it impractical to account for novel users' activities as well as major changes in users' habits.

Secondly, the majority of activity discovery techniques are poorly integrated with HAR solutions. Rather, they take information collected by sensors as a starting point and then discover frequent patterns by analysing the history of sensor data that can be associated with different actions carried out by the user (Jakkula and Cook, 2011; Rashidi et al., 2011a, 2011b; Aztiria et al., 2012; Lotfi et al., 2012; Aztiria et al., 2013; Cook et al., 2013a). Besides the missed opportunity for building integrated, efficient and modular solutions, it is difficult for these systems to account for the richness of heterogeneous sensor sources to which HAR solutions are now accustomed. Consequently, it is common for these systems to be limited to the offline analysis of only binary sensor data. Only a few past attempts in the same domain have demonstrated the value of combining activity detection with pattern discovery—although with solutions not designed with a robotic ecology in mind. In particular, (Cook et al., 2013a, 2013b) have shown how unsupervised discovery can shed light on users' behavioural patterns by discovering patterns in the data that do not belong to a predefined class. Segmenting the data into learnable classes has been proved to boost the performance of activity recognition algorithms, while also reducing their reliance on costly samples for supervised learning. However, only the activity recognition algorithm in those examples is able to run online, while the pattern recognition algorithm is run offline, and only on binary sensor data. Other solutions have built a hierarchical model of user's behaviour, using supervised methods to learn low-level activities (e.g. eating, walking, sitting, standing), in combination with topic-based, unsupervised discovery of more complex activities (e.g. having lunch, commuting, office work, (Huynh et al., 2008; Niebles et al., 2008). The solution proposed by Wan et al. (2014) considers that the home occupants maintain a relatively regular routine. Their method performs online event segmentation by extracting different threshold parameters from past sensor data. Contrary to our approach, they do so by relying on the assumption that there will be no significant changes to the user's daily routine.

Thirdly, existing plan-based and adaptive control solutions (Tang and Parker, 2005; Saffiotti and Broxvall, 2005; Vig and Adams, 2006; Lundh et al., 2008) are ill suited to be used in robotic ecologies operating in real settings, where they will be subjected to multiple, dynamically changing and possibly conflicting goals (e.g. recharge batteries versus exploration, cleaning the kitchen versus reminding the user to measure her heart rate). In addition, since they will typically find multiple options to satisfy each goal, robotic ecologies need the ability to evaluate different strategies in terms of resource utilization and time-frames. Moreover, past examples used in AAL applications, such as (Saffiotti and Broxvall, 2005) have strictly relied on pre-defined models of the robots, of their users, of the environment, and of its associated dynamics. These models can be used to find strategies to coordinate the participants of the ecology and to react to perceived situations, but they lack the ability to pro-actively and smoothly adapt to evolving contexts. These limitations make such systems still difficult to deploy in real world applications, as they must be tailored to the specific environment and application. Relying on the same solutions to support the operations of robot ecologies in real settings would quickly become ineffective, unmanageable and prohibitively costly.

Finally, user-interfaces give the application designer and, to a limited degree, the final user(s) some control on personalization and service customisation features. However, such an approach puts an additional burden on the developers to model each possible situation, and on the final users to customize and adapt each service to their specific needs. We argue that such solutions are only feasible if counter-balanced by systems with a degree of autonomy and able to use their past experience to refine but also to extend their capabilities.

3. The RUBICON architecture

RUBICON adopts a modular design, which builds and integrates a number of different solutions, to deliver robotic ecologies with the ability to self-adapt to their environment and to their user.

Such an approach stands upon a modular software suite made out of a series of interacting layers, as depicted in Fig. 1. Each layer is designed to (i) attend to one of the pivotal roles in the system; (ii) exhibit a degree of internal self-organization in order to suit gradually evolving settings; (iii) cooperate via well-defined interfaces with the other layers, so that the system as a whole can more easily adapt to new situations.

Sitting at the heart of the RUBICON system is a Communication Layer (Amato et al., 2012), dealing with how data and functionalities are shared between system components, including distributed software components, robots and heterogeneous wireless

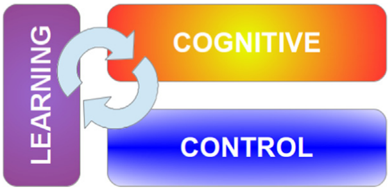


Fig. 1. High-level RUBICON architecture illustrating its modular integration between its cognitive capabilities.

sensor and actuator networks. The Communication Layer re-uses and combines existing wireless sensor networks (Amato et al., 2010) and peer-to-peer communication middleware (Broxvall et al., 2007). This allows system developers to frame the robotic ecology as a multiagent system in which each device is modelled as an autonomous agent with sensing and acting capabilities (Dragone et al., 2012).

The key factors to enable goal-oriented and adaptive behaviours in the ecology are to (i) extract meaning from noisy and imprecise sensed data, (ii) to learn what service to pursue, and (iii) how to pursue them, from experience, rather than by relying on pre-defined goal and plan selection strategies.

The first of these challenges is met by the Learning Layer (Bacciu et al., 2012), a distributed and adaptable learning infrastructure that can be used to support a variety of learning requirements of a robotic ecology. The Learning Layer processes streams of data gathered by the sensors in the ecology to classify events and delivers predictions about the state of the ecology and of its users. The Learning Layer can be trained, for instance, to forecast the exact location of the user by examining the history of the RSSI measured between a wearable device worn by the user and anchor motes embedded in the environment, or to provide timely and predictive information on the activities being performed by the user, such as recognizing that the user is cooking by analysing the signal and temporal pattern received from sensors installed in the kitchen, such as switches triggered upon opening and closing the cupboards and refrigerator. Finally, the Learning Layer allows the incremental acquisition and deployment of new computational learning tasks, in order to adapt to new environmental and/or user conditions.

The second challenge is the responsibility of the Cognitive Layer (Ray et al., 2012; Leng et al., 2013). A distinct feature of the RUBICON's Cognitive Layer is that it does not analyse sensor data directly, but it reasons over the output of the Learning Layer. These capabilities are exploited to learn to predict the need for activating appliances and/or robotic services to suit the preferences and the requirements of each user. In addition, by examining the output of the Learning Layer over time, the Cognitive Layer is capable of online novelty detection to identify novel situations, for example, corresponding to previously user's activities. The system's response to the detection of these new situations is to trigger and feed the incremental learning mechanism of the Learning Layer with teaching information related to their occurrence. In this manner, those situations will be directly recognized by the Learning Layer, by exploiting its distributed learning mechanisms, and thus also included in further reasoning in the Cognitive Layer.

The final challenge is met by the Control Layer (Di, Rocco 2013), a plan-based executive component providing high level control over the nodes within the ecology. The Control Layer finds, realizes and monitors collective and adaptive strategies to satisfy the goals set by the Cognitive Layer. It is also capable of self-organisation in the way it uses available resources (e.g. robots, actuators, sensors) in the process.

These layers are briefly discussed in the following sections, while reports on both theoretical details and empirical evaluations of each layer can be downloaded from http://www.fp7rubicon.eu (2014).

3.1. Distributed adaptive memory for sensor timeseries

The Learning Layer realizes a distributed, adaptable and robust learning infrastructure for the RUBICON ecology that is specifically tailored to deliver short-term predictions based on the temporal history of its input signals. These include information gathered from the sensors embedded in the environment as well as from off-the-shelf software components running within the ecology. It does so through a distributed neural computation comprising

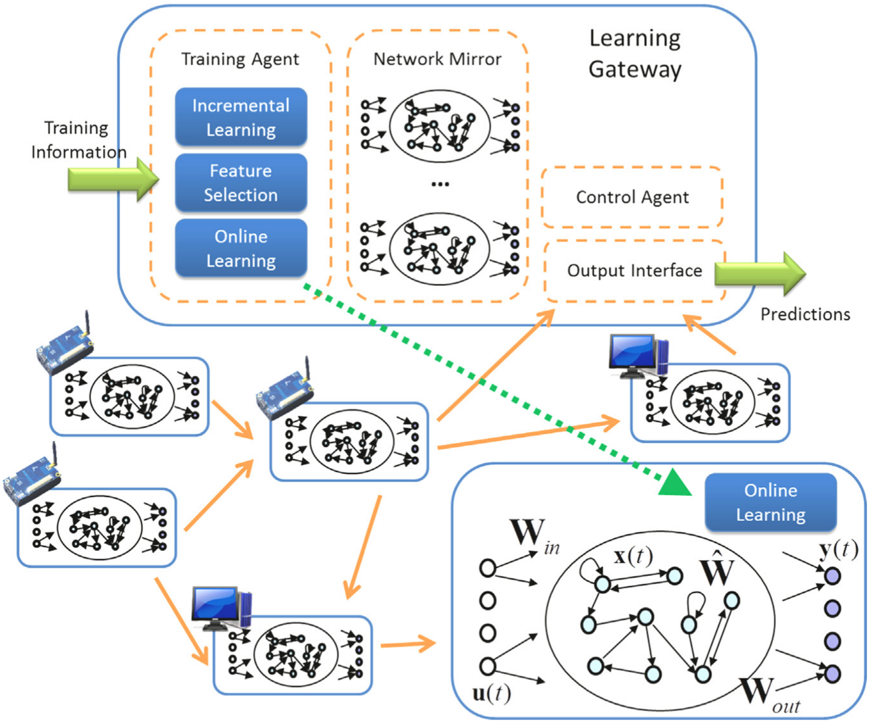


Fig. 2. High-level view of the learning layer distributed learning system: blue boxes with solid contours denote different devices (of various computational capabilities); orange boxes with dashed contours identify relevant software agents composing the learning system. A special device, called the Learning Gateway, hosts the bulkier learning mechanisms, such as feature selection and incremental learning, as well as a mirror copy of the distributed learning modules. Online learning functionalities are instead distributed on the single host devices. The parameterization of the Echo State Network model is shown on a zoomed view of an ecology node. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

independent learning modules deployed on the ecology nodes that interact and cooperate through synaptic communication channels provided by the Communication Layer.

Fig. 2 provides an high-level view of the learning system that highlights how the learning modules are distributed across a variety of ecology participants. Online learning functionalities are distributed on each device in the ecology, while the bulkier learning mechanisms, as well as a mirror copy of the distributed learning modules, are deployed on a Learning Gateway.

The learning modules are the key learning machinery underlying the Learning Layer and they have been designed based on two cornerstones, namely: (i) the need to deal with time-dependent and noisy data typically observed in AAL solutions, and (ii) the heterogeneity of the computational resources of the ecology devices, which demands learning models with limited computational requirements. Based on such considerations, the learning modules have been realized by Echo State Networks (ESNs) (Jaeger and Haas, 2004), a recurrent neural model from the Reservoir Computing paradigm, characterized by a good trade-off between computational efficiency and ability to deal with dynamic systems and noisy data.

ESNs are used to implement non-linear dynamical systems, being composed of a recurrent non-linear part, called *reservoir* and of a non-recurrent linear part, called *readout*. From a dynamical system point of view, by updating a network state at each time step of computation, the reservoir provides the model with a non-linear dynamical memory of the past input history. The extreme efficiency of training of the ESN approach, with respect to standard recurrent neural networks, stems from the fact that only the readout component (i.e. $\mathbf{W}_{in}$ and $\mathbf{W}_{out}$ in Fig. 2) is trained, while the reservoir part (i.e. $\hat{\mathbf{W}}$ in Fig. 1) is left untrained after being initialized under the constraint of the Echo State Property (Jaeger and Haas, 2004; Gallicchio and Micheli, 2011). The Learning Layer provides a variety of learning mechanisms that allow to continuously adapt the knowledge captured by the distributed memory. A Training Agent component (deployed on the Learning Gateway) manages the training information received by the Learning Layer and takes responsibility for running the more computationally demanding processes. These include (i) an *Incremental Learning* mechanism, which allows to incrementally acquire new learning tasks (i.e. new predictions to be performed) or to re-train from scratch the existing ones by exploiting the mirrored copy of the learning models in the Network Mirror, and (ii) a set of *Feature Selection* mechanisms to allow to automatically filter out those input sources that are either redundant or provide irrelevant information. Finally, a lightweight *Online Learning* mechanism is implemented on all the ecology nodes that allow to refine the predictions of the embedded learning modules based on instantaneous supervised or reinforcement teaching signals.

3.2. Cognitive layer

There are three core modules in the Cognitive Layer; namely (i) a *cognitive memory module*, (ii) a *cognitive reasoning module* and (iii) a *cognitive decisions module*. The cognitive memory module is responsible for holding current and historical states of the RUBICON ecology as perceived and processed by the Learning, Control and Cognitive layers. This is implemented through a MySQL database and allows interaction between the layers via PEIS middleware. The two primary components of the Cognitive Layer (Ray et al., 2012; Leng et al., 2013), which utilise this information, are the reasoning module, based on a self-organising fuzzy neural network (SOFNN), and the decision module, based on a Type-2 fuzzy neural network (FNN).

Reasoning Module: The SOFNN (Leng et al., 2004) is a hybrid network which has both reasoning and learning capabilities that can be used to model and forecast complex nonlinear systems. It is a five-layered network consisting of an input layer, an Ellipsoidal Basis Function (EBF) layer, a normalised layer, a weighted layer, and an output layer. Adding and pruning strategies are utilised to enable the self-organising capability to produce a fuzzy neural network with a concise and flexible structure. This capability ensures that the SOFNN structure is continually adapting to the input data to decrease errors during learning such that the cognitive layer maintains good performance in dynamic AAL applications. The SOFNN is also ensures model compactness as its structure and parameters are determined from the available data. The number of inputs to the network is normally fixed although the architecture employed in this work includes the capability to adapt its input structure based on variations of the dimension of the input data.

Decision Making Module: The decision module integrates the status outputs generated by the reasoning module to generate decision signals that are then interpreted as goals of the RUBICON ecology. To accommodate noise and uncertainty of the generated status outputs, the decision module needs to show a high level of robustness. To this end, an SOFNN based on Type-1 fuzzy sets is developed, which provides a degree of robustness to noise due to the inherent approximate reasoning capability of the Type-1 fuzzy sets. However, it has been demonstrated that an additional level of uncertainty can be mitigated by exploiting Type-2 fuzzy sets (Zadeh, 1975) which describe the membership functions as a fuzzy set rather than as crisp numbers. Incorporating a Type-2 approach in a fuzzy neural network enhances its ability to handle uncertainty which is important in the AAL settings where the robotic ecology operates.

The Type-2 FNN is created by combining the SOFNN and a Type-2 fuzzy learning system; the strategy for development exploits the following steps:

1. A trained SOFNN structure, which is a Type-1 fuzzy structure, is obtained.
2. This Type-1 fuzzy structure is then used to initialise the parameters of a Type-2 fuzzy neural network structure.
3. The initialised Type-2 structure is trained off-line using gradient descent and Kalman filtering algorithms.
4. The final output of the system can be generated after information has passed through type-reduction and defuzzification. This results in a decision module (illustrated in Fig. 3) that responds well to high levels of noise, thus adding robustness to the cognitive architecture.

Finally, the decision process is supplemented with a secondary system which supports the cognitive role in the form of novelty detection. Novelty detection can be defined as the process of identifying interesting new stimuli that are different from anything known before (Chandola et al., 2012; Marsland, 2003). In this sense, novelty detection can be seen as a form of selective learning which treats any experience which falls outside of those seen during training as novel. A number of novelty detection methods have been proposed in the literature, mainly focussing on detecting anomalies and outliers, i.e. identifying patterns that do not conform to expected behaviour (Chandola et al., 2012; Marsland, 2003; Markou and Singh, 2003; Hodge and Austin, 2004). Typically for these problems there are substantial data about the normal classes but very little data displaying the novel features that should be detected. Hence, it is essential to learn a model of what is normal and then attempt to detect deviations from this model. Within an AAL application, novelty detection serves to compile a list of an inhabitants activities in terms of sensor readings emanating from the Learning Layer during a training period. This set becomes what is deemed normal. Significant

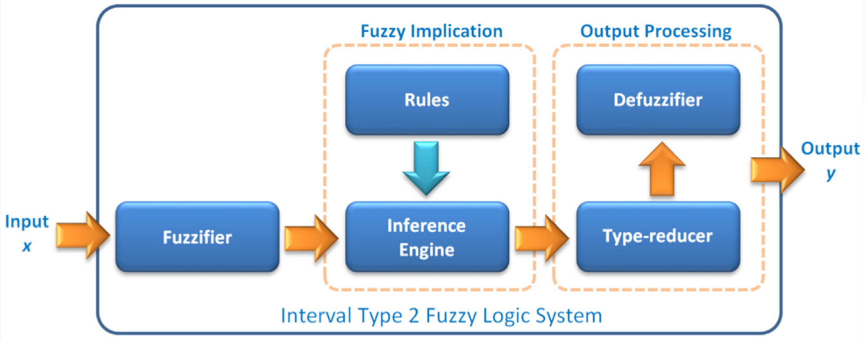


Fig. 3. High-level view of the cognitive layer architecture, illustrating the combination between the Type-1 and Type-2 fuzzy reasoning components.

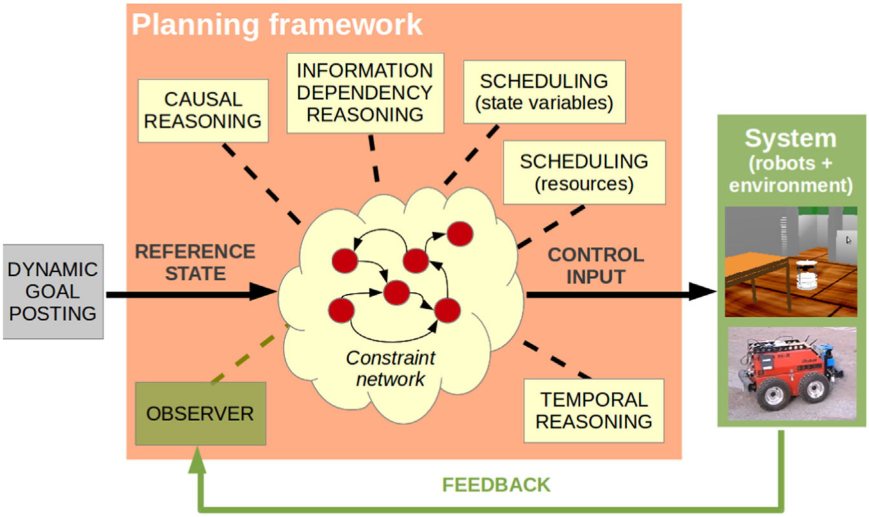


Fig. 4. Schematic view of the configuration planner. The observed state and the current goals are continuously injected in the shared constraint network. Several solvers (yellow boxes) complete this network with a plan that connects the current state to the goals. (From Di, Rocco 2013 with permission.) (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

deviations from what is normal are considered novel and cause the system to act on this new information by presenting the Learning Layer with teaching information related to a possible new activity. Further details on this secondary support system, including how its memory is managed is presented in (Vance et al., 2014).

3.3. Configuration planning

At the heart of the Control Layer lays a *configuration planner*: a planner system that generates fine-grained plans which specify the causal, temporal, resource and information dependencies between the different components of the robots and devices in the robotic ecology. For example, a configuration plan may indicate that robot-1 must navigate to the kitchen while taking location input from the camera-based localization system, and after the kitchen automatic door has been opened. The type of configuration planning used in RUBICON was initially developed in the context of PEIS Ecologies (Lundh et al., 2008), but similar approaches were proposed, e.g., by Tang and Parker (2005) and by Vig and Adams (2006).

The configuration planner is a constraint based planner (see Di, Rocco 2013) for a more detailed technical description. It is grounded on the notion of state variable, which models elements of the domain whose state in time is represented by a symbol. State variables represent parts of the real world that are relevant for the planners decision processes. These include the actuation and sensing capabilities of the devices in the RUBICON ecology, as well as physical features in the environment. For instance, a state variable can represent the actions of a given robot, whose meaningful states might be “navigating”, “grasping” and “idle”. Another state variable can represent the state of a given light which can be “on”, “off” or “broken”. Goals are also represented through specific values of state variables. The possible evolution of state variables in time are bound by temporal constraints, e.g., stating that navigation must occur *while* the light is on, or that *after* navigation is completed the robot will be located at the kitchen. We represent temporal constraints by an extension of Allens Interval Algebra (Allen, 1984).

State variables and constraints are maintained in a constraint network. The configuration planning process manipulates this network by incrementally adding variables and constraints, until the network contains a feasible plan that connects the initial state to the goals. The resulting constraint network represents one or more temporal evolutions of the state variables that guarantee the achievement of the goals under nominal conditions. The configuration planner is itself composed of several *solvers* which all manipulate the same shared constraint network. Each solver takes into account a specific type of constraint, e.g., causal, topological, information, temporal, or resource constraints. The solvers are orchestrated by a meta-CSP approach (Fratini et al., 2008). The configuration planner operates in a closed loop at a cycle of about 1 Hz. At each cycle, the constraint network in the planner is updated by an observer to account for the current state of the

RUBICON ecology as well as goals newly posted by the Cognitive Layer, and the solvers are re-invoked to update the plan if needed. The conceptual structure of the configuration planner is shown in Fig. 4.

4. Powering AAL solutions with RUBICON

The solutions outlined in the previous section have been evaluated both in isolation and as part of integrated systems (Di Rocco 2013; Palumbo et al., 2013; Bacciu et al., 2014a, 2014b). In our previous publications we have discussed the performances of the multi-layered RUBICON approach in a smart home environment. In such a context, we have described how our system can identify and react to user needs, activities and preferences, learning to automatically perform robotic services in relation to the activities of the user, e.g. cleaning the floor after having recognized that the user has had her meal.

We begin here from an initial system configuration in which the Learning Layer has been trained to provide timely and predictive information on users' location, i.e. in the hall, in the living room, in the kitchen, in the bedroom or outside, and on whether the user is performing a range of daily-life activities, i.e. eating, setting table, cleaning, washing dishes, preparing food, exercising, relaxing, sleeping or entering the house. Such a system is trained under factory conditions using a purposely collected HAR dataset, comprising a large set of sensor data streams and ground-truth information gathered in a smart-home environment. Although out of the scopes of this paper, a detailed analysis of the overall training phase and of predictive performance achieved on the considered HAR tasks (reaching test accuracy in the range of [0.84, 1.00] for the different indoor locations and baseline daily-life activities) is contained in our reports (see http://www.fp7rubicon.eu, 2014).

In here we focus on the operations of the integrated systems and, more specifically, on its ability to evolve from its initial configuration. Noticeably, while the initial training phase of the RUBICON relies on the provisioning of supervised information to the Learning Layer, a robotic ecology can use this as a starting point, while it adapts to its new environment and to its user.

Fig. 5 illustrates an extract of the outputs of the Learning Layer, when the user has carried out her daily exercise routine and then relaxed on the sofa. At this point, the user has also measured her heart rate by using a bluetooth-enabled pulsometer. We use a magnetic sensor in order to detect whenever the user takes the pulsometer from its box. The outputs in Fig. 5 correspond to relevant user activities to be recognized, where a value approaching +1 denotes maximum confidence on the activity being performed, while values approaching 0 denote increasing confidence on the activity not being performed.

Over time, the Cognitive Layer successfully learns to use the outputs of the Learning Layer to predict the occurrence of the event raised by the opening of the box with the pulsometer. Fig. 5 also shows the Cognitive Layer's response to the expected state of the ecology in the situation where the user was relaxing after her exercise but forgotten to measure her heart rate. Based on the analysis of past instances, the Cognitive Layer signals to the system when it is time for the user to do that.

The signal received from the Cognitive Layer is interpreted by the Control Layer as a request to have one of the robots in the household approach and remind the user to measure her heart rate. In this example, the resulting plan is straightforward. Since the robot was previously idle, it is tasked to approach and prompt the user. Fig. 6 shows a sequence of frames (extracted from the video attached to this paper) showing the events just described.

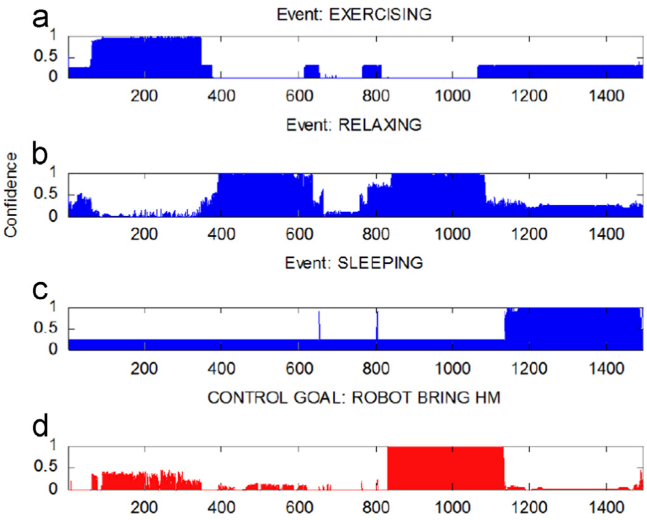


Fig. 5. Example output of the learning layer (a–c) while the user performs her daily exercise routine, and output of the cognitive layer (d), after it has learnt to request the delivery of the heart monitoring (HM) equipment while the user relaxes after her exercise.

4.1. *Driving system adaptation*

The events described in the previous section stand as a template for a family of useful and robust AAL services. Similar plans can be activated whenever there is a mismatch between the output of the Cognitive Layer and one of the events that are raised by an action of the user. The system can be programmed to prompt the user when something out of the ordinary is taking (or some user's routine is not taking) place and may require her attention and/or further instructions, thus ultimately helping the system to gather feedback and gradually adapt its behaviour to the needs of its users.

In the remainder of this section, we show how a similar process can be exploited to account for un-modelled user behaviors and activities in the AAL application discussed above. Consider, for instance, what would happen in the heart-monitoring example if our user injures herself and is instructed by her doctor to perform a rehabilitation activity in place of her usual exercise routine, in order to accelerate her recovery. The resulting activation of sensor readings will now be new to the Learning Layer. The user's rehabilitation routine is considerably different from her previous exercise, shown in Fig. 6, as the user's movements are different (causing different readings from the accelerometer on her wrist) and the user performs her rehabilitation routine in a different position (e.g. sitting on the sofa instead of standing in front of the TV). This will result in events being presented to the Cognitive Layer with a lower confidence than is usually expected. Consequently, the system might stop to help the user just when the user needs it the most!

Our final aim is to take advantage of the layered architecture described in Section 3 to determine when a change in user routine becomes incorporated into the reasoning system. To this end, the *Incremental Learning* mechanisms described in Section 3.1 empower the ecology with a dynamic memory formation process that can be exploited to learn to account for new user habits, activities and situations. This can have the effect to further relieving the ecology from the need of human supervision, allowing it to self-cater the training information needed to acquire knowledge on the novel situations. In our reports (e.g. in Bacciu et al., 2014a) we have described how the incremental learning functionalities in the Learning Layer can be exploited by the Control Layer to learn un-modelled context-dependent plan-selection policies.

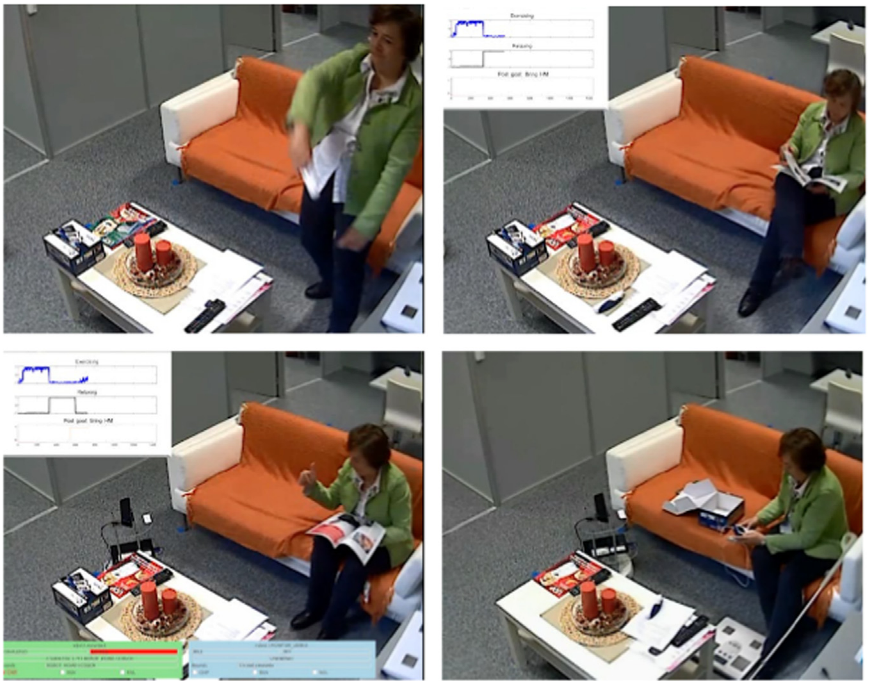


Fig. 6. Extract from the video attached as supplemental material. From top left, clockwise: user exercising; user relaxing; robot approaching and reminding the user to measure her heart rate; user using the pulsometer after she fetched and opened the pulsometer's box from the table.

To further validate these mechanisms in our example AAL scenario, we have implemented a novelty detection feature inspired by cognitive models of habituation of neural responses and synaptic decay. This is implemented within the Cognitive Layer through a three tiered system. The first tier searches all active incoming events for synchronous activity groups and assigns them to the *suspected event tier* if they are recorded at lower confidences than expected. Over time, new groups are promoted to the *potential event tier*, and, eventually, recognized as *new events*. The progression over these three stages is represented by the dashed lines in Fig. 7, where it is superimposed to the output of the Learning Layer in one live episode in which the user was carrying out her rehabilitation exercise.

Upon recognition of a new event, the Cognitive Layer notifies the system with the timings of all the instances in which such an event occurred in the past.

At this point, the Control Layer invokes the Learning Layer's mechanisms to, respectively:

1. initialize a new HAR learning task corresponding to the newly detected user activity;
2. notify an approximated indication of the sensory sources that should be considered as candidate inputs for the new learning task (these are simply the union of the sensor sources that were used by the Learning Layer to detect the events grouped by the first tier of the novelty detection, i.e. the user relaxing and user exercising events in our example);
3. train the Learning Layer by exploiting the information on new event timings and associated sensory sources to assemble appropriate training datasets. Noticeably, through its built-in feature selection mechanisms, the Learning Layer can refine the preliminary superset of sensor sources by extracting a subset of non-redundant and highly-predictive inputs relevant for learning the new task.

After this re-training, the Learning Layer will produce a new output event that will present with high confidence whenever the user will engage in the new activity. As shown in Fig. 8, the Rehabilitation exercise has now been added to the detectable range of events processed via the Learning Layer and as a result of this, the Cognitive Layer has also added it as one of its inputs.

Noticeably, at this point the Control Layer may also be instructed to ask the user what she is doing. This may help the system to give an actual label to the new activity (Rehabilitation) but it is irrelevant to how that event will be treated in the future. The system can now use the new event in its cognitive reasoning, which will ultimately lead to the restoration of its ability to assist the user.

4.2. Dynamic planning

The final experiment we describe was designed to show the operation of the system now able to recognize and process the new event, together with its ability to react to contingencies and reason about dynamic situations.

Specifically, the experiment shows the ability of the configuration planner presented in Section 3.3 to consider multiple goals, as well as its ability to reason about resources and times.

In this experiment, the robotic ecology employs two robots, each equipped with a vacuum cleaner and with a container where the user can place little objects for the robot to carry, such as a box with her pills or with the pulsometer. We have also added two additional constraints to the planning domain: The first constraint states that only the robot carrying the box with the pulsometer should go close to the user when the Control Layer is instructed to do so by the Cognitive Layer. In a real application, such a constraint could be supported by placing an RFID tag on the box and by employing an RFID reader on the robot to inform it when it is carrying the box in its tray. The second constraint is used to enable the two robots to coordinate their movements in the apartment. Specifically, since there is not space for two robots to negotiate at the same time the narrow space between the sofa and the wall, the area connecting the livingroom to the kitchen is modelled by the planner as a single resource that cannot be accessed concurrently by two robots.

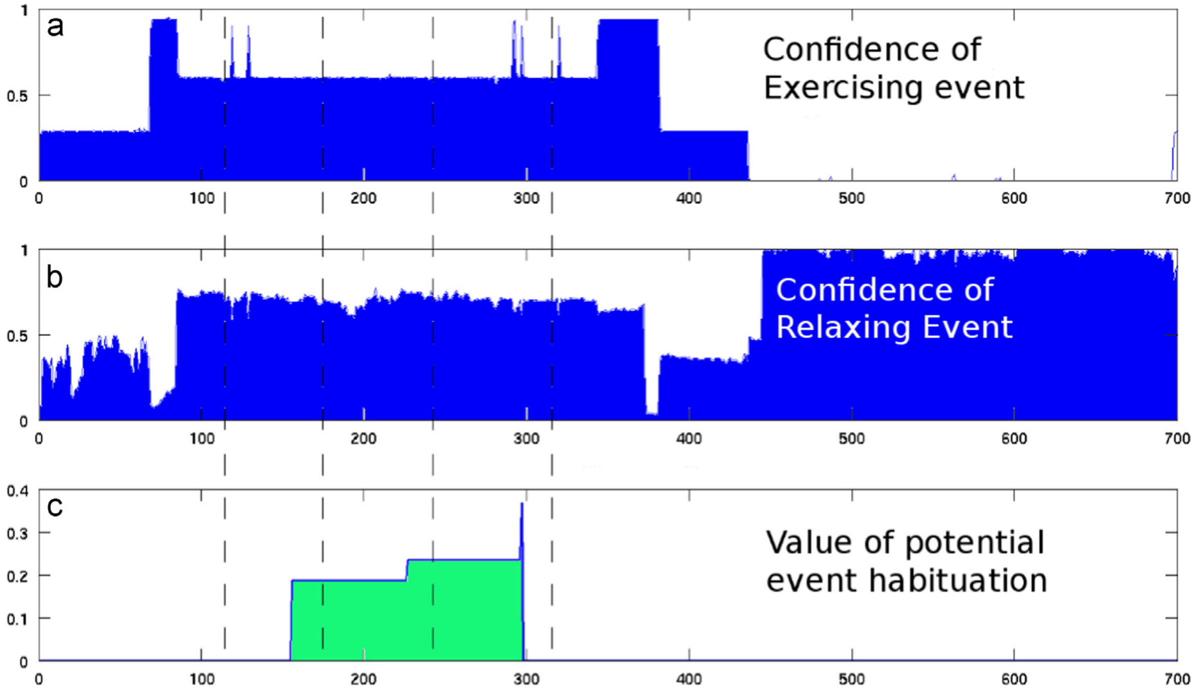


Fig. 7. (a) Exercising and (b) Relaxing events pertaining to the rehabilitation exercise with the associated (c) habituation values of the potential event leading to its discovery.

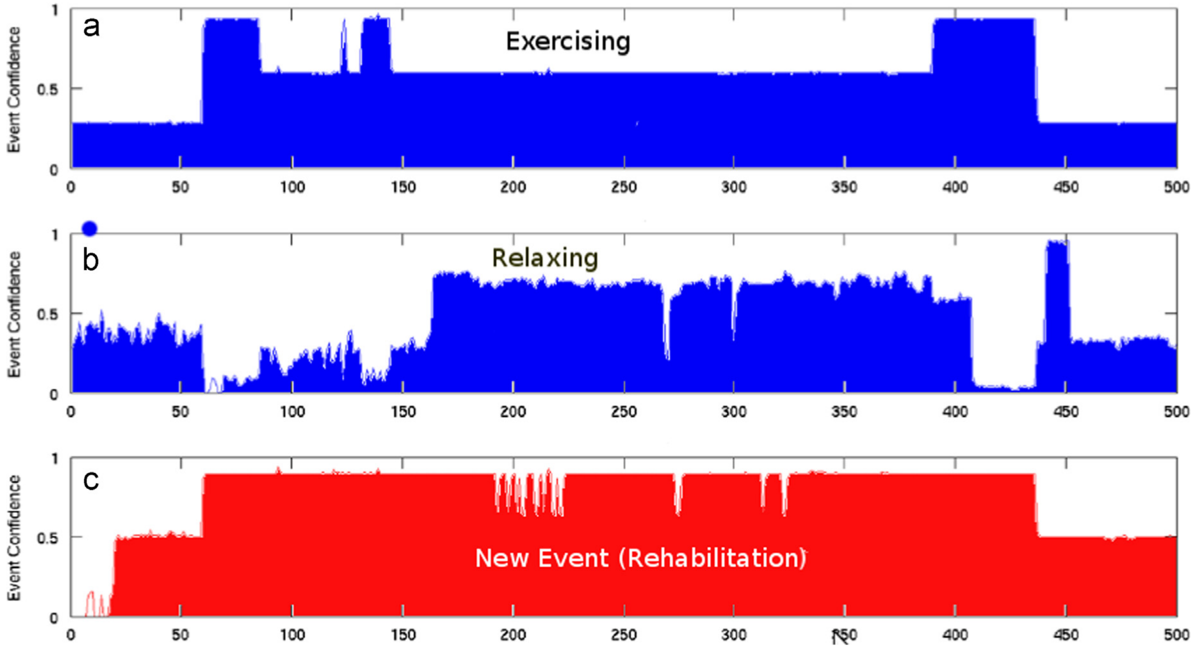


Fig. 8. Example output of the learning layer while the user performs a sequence of rehabilitation and relaxation activities. The diagrams shows the outputs for *known* events (exercising and relaxing) prior to the novelty detection, and the *novelty-detected* rehabilitation event after the learning layer has adapted.

Fig. 9 shows a sequence of pictures (extracted from the last phase of the video) starting from a situation in which the first robot is cleaning the kitchen while the user is relaxing after her rehabilitation. Note the difference between the user's rehabilitation routine and her previous exercise, shown in the first scenes of the same video.

After the user stops her rehabilitation routine, the Cognitive Layer instructs the Control Layer to bring the box with the pulsometer to the user.

Since the robot cleaning the kitchen is also currently carrying the box with the pulsometer when the Cognitive Layer raises the heart monitor goal, the Control Layer's planner has to find a way to (i) use that specific robot to prompt the user as soon as possible (and in the process bringing the pulsometer to the user, who is currently indisposed), while (ii) also completing the cleaning task (which is still far from completion at the moment the heart monitoring goal is issued).

Fig. 10 and the accompanying video show how the planner solves this problem by re-allocating the cleaning task, i.e. asking the second robot (previously idle) to go to clean the kitchen, while the first robot, now relieved of its cleaning duty, is asked to move to the living room and approach the user. However, rather than instructing both robots at the same time, the planner successfully orchestrates their movements to resolve their use of the single resource representing the narrow passage. Firstly, the planner instructs the first robot to dispense the

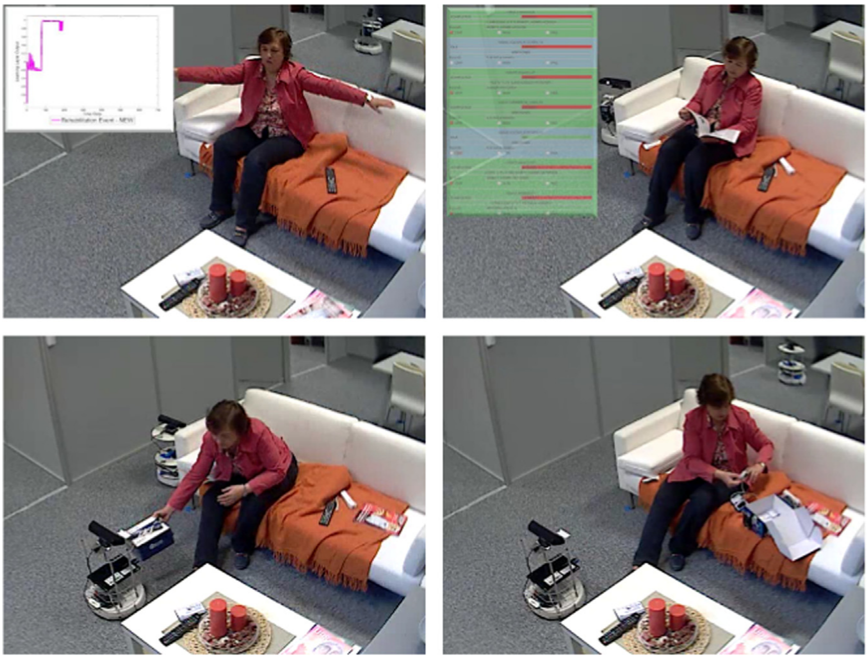


Fig. 9. Extract from the video attached as supplemental material, showing the task re-allocation between the two robots after the system has recognized the need to bring the pulsometer to the user. From top left, clockwise: robot 1 cleaning the kitchen (in the top right corner); robot 1 moving toward the living-room; robot-2 moving to the kitchen while the user collects the pulsometer from robot-1; robot-2 taking over the cleaning duty in the kitchen while the user uses the pulsometer.

heart monitoring tool, since its task has a tighter deadline. Secondly, the planner instructs the second robot to go to the kitchen and take over the cleaning task. A graphical representation of the activities of the planner are superimposed to the final scene of the video.

5. Discussion

Thanks to an adaptive, open and general purpose learning infrastructure and its innovative combination with online novelty detection functionalities, a RUBICON system can more easily adapt to evolving situations and achieve useful services that are not restricted to only those situations that are envisioned by their designer. Successful goals and novel events are absorbed into the cognitive model which shows an ability to evolve and grow to accommodate and account for new situations.

Such a process allows the system to be driven by using easily identifiable rules, while delegating, over time, symbolic reasoning to data-driven inference for the purpose of increasing flexibility, robustness and adaptation.

This is a clear improvement on past solutions, which demanded for all context and goal rules to be specified a priori.

To the best of our knowledge, the integration approach discussed in this paper constitutes a novel way to design adaptive robotic ecologies by extending plan-based control mechanisms with a combination of machine learning methods for context recognition, and cognitive reasoning for goal deliberation and novelty detection.

A useful characteristic of such an integrated solution is its clear distinction between (i) *events*, i,e, the result of the activity and context recognition process performed by the Learning Layer on the basis of raw sensor data, (ii) *goals*, i.e. the result of the cognitive reasoning performed by the Cognitive Layer on the basis of the events originated by the Learning Layer, and (ii) *plans* describing the configuration and coordination strategies activated and supervised by the Control Layer to achieve the goals set by the Cognitive Layer.

Our implementation is an instance of such a modular design, whereas each component may be replaced with possible alternative implementations covering similar roles.

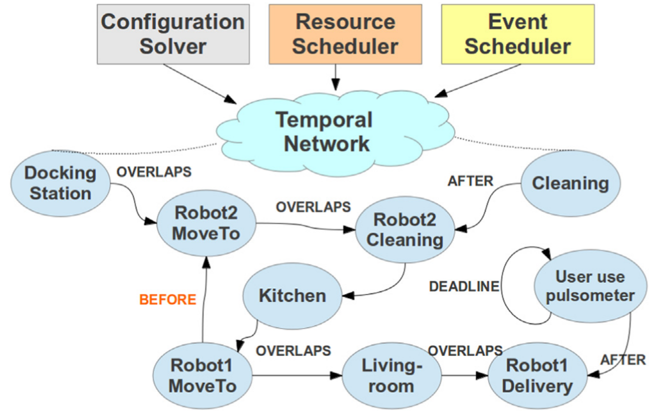


Fig. 10. The final plan, which is refined by the scheduler to make sure that the robots will not have to traverse the narrow space between the wall and the wall at the same time (they both consume a resource of capacity one).

While it has been tested in a limited scenario, the two-layer interaction between the Learning and Cognitive layers implemented in RUBICON has shown a promising direction toward which to advance the results of past attempts that have integrated activity detection with pattern discovery functionalities in the same application domain. Compared to those methods, which were discussed in Section 2, our system has the ability to account for heterogeneous and noisy sensor data and can leverage a rich selection of both binary and non-binary sensors while performing online novelty detection and reasoning on context and automation patterns.

However, the goal of our work is not the classification of all users' behavioural patterns, but only those whose prompt detection can facilitate the robotic ecology in assisting its users. Indeed, as (Wan et al., 2014) also notes, seeking to recognize every possible activity can be overly complex, and may also be perceived by home occupants as unacceptably compromising to their privacy. The only activities of interest are those that the occupant undertakes which undermine their capability to function independently. The integration approach discussed in this paper shows a way to do just that, by starting from a system trained to recognize simple events, and by letting it learn, over time, to recognize and adapt to evolving situations. In this concern, the Incremental Learning mechanisms implemented in the Learning Layer (see Section 3.1) provide the RUBICON system with the ability to deal with environmental changes by allowing re-training on existing computational tasks, whenever a significant environmental change is identified by the Cognitive Layer.

Noticeably, the effect of this process is similar to what is achieved by approaches addressing the concept drift problem (Widmer and Kubat, 1996) in activity classification, as in (Kurz and Ferscha, 2010). Concept drift addresses slow changes in the mapping between sensor signals and activity classes in the feature space. Usually, this is achieved by monitoring drift over time, and by triggering either (i) a re-calibration of the learning system (e.g., by adjusting the parameters of the classifier), or (ii) a complete re-training (by taking advantage of many repetitive occurrences of context in daily life). The solution implemented with the interaction between the Cognitive and the Learning layers described in Section 4.1 is an example of the latter. However, in our architecture the same approach is used both to account for changes in known activities corresponding to already trained learning module, and also for discovering new activities for which the learning system had not received supervised training information.

In addition, activity classification is a by-product of our approach: new events are internalized and detected by a distributed learning network. They may not exactly match with recognizable users' activities, but the outcome is a system that can successfully adapt to new user habits and changing user needs.

Supporting heterogeneous and distributed systems and varying computational constraints has been a cross-cutting concern addressed in each of the solutions described in this paper. This was an important requirement, which is not met by the centralized solutions traditionally used in smart environments, as robotic ecologies may contain devices such as computers with large processing and bandwidth capacities, as well as much simpler devices such as micro-controller-based actuators and sensor nodes, and even devices with no (customizable) computational capability at all, such as Radio Frequency Identifications (RFIDs). Sensor data in a cognitive robotic ecology is processed as much as possible locally on computational constrained and robotic devices. Information (e.g. goals) is extracted and exploited by the higher layers. These are equipped with bigger computational capabilities, which they exploit to deliberate and supervise the execution of complex tasks.

Our approach may inspire similar advancements in similar technical domains. In particular, the Robotic Ecology vision shares many similarities with the one pursued within the CSP and IoT communities: the ideal aim in all those fronts is that arbitrary combinations of devices should be able to be deployed in unstructured environments, such as those exemplified in a typical household, and therefore, efficiently cooperate to the achievement of complex tasks. A widespread solution in IoT system is to deploy high-level, rule-based reasoning modules on cloud-based infrastructures. However, systems that need to close the loop between analysis and control may find it useful to implement a similar approach to the one implemented in RUBICON, and combine high-level, cloud-based system intelligence with data processing and learning functionalities deployed on each device.

6. Conclusions and future work

This paper has highlighted the challenges in endowing robotic ecologies with cognitive capabilities, and has illustrated the general principles underlying the construction of self-adaptive robotic ecologies.

We have shown how adaptive, and proactive system behaviour can be obtained by exploiting: (i) a learning infrastructure to extract meaning from noisy, imprecise and heterogeneous sensed data, (ii) cognitive reasoning capabilities to learn what service goals to pursue, and (iii) advanced planning solutions that are able to reason upon different objectives, resources, and dynamic situations. Furthermore, this paper has shown how the dynamic acquisition and modelling of new events can also be implemented as an interlayer cooperative process.

All our solutions have been purposefully designed to be as open, flexible and extensible as possible. Key components of our software suite can already be downloaded from our project website (http://www.fp7rubicon.eu, 2014).

These components can be applied, either in the fully integrated form presented in this paper, or by using only selected combinations, as we have shown in (Bacciu et al., 2014a). This has the potential to drive the adaptation of many existing smart environment scenarios, and also to inform the development of similar integrated approaches.

However, more work is needed to improve the usability and manageability of this type of integrated solutions. Building actual AAL applications requires many researchers to program each part of the system using domain-specific languages and research-oriented software and tools, before systems prototypes can be evaluated. In addition, setting up each system still requires a degree of manual configuration, for instance, to identify narrow passages in the environment that must be considered as a single resource by the configuration planner. The issue of whether this configuration could also be automated, for instance, as part of autonomous exploration and mapping of the environment, is still an open research question. Its resolution would be an important step toward allowing fully autonomous systems able to adapt to the settings where they are installed.

Moreover, further work needs to be directed towards improving both the efficiency of these solutions and the user's experience when dealing with them, in order to enable us to carry out long-term evaluations with real users in their own homes. To this end, the interplay between the different components composing a cognitive robotic ecology provides flexible mechanisms, which can be used to define, at run-time, new learning requirements, and can adapt to changes in the settings of the ecology or to newly discovered situations and users' activities. These mechanisms may be used by both internal ecology components – as in our examples – as well as by external components, such as user interfaces. These may be used to provide feedback to the learning system, to drive its smooth adaptation to user preferences, but also to verify and possibly accelerate its ability to adapt to novel situations. The research question in this case is how to find ways for the human to meaningfully interact with an opaque system that sometime may incorporate concepts that do not correspond to recognizable activities. Future work should explore scenarios in which users and cognitive robotic ecologies collaborate and learn to exploit their different capabilities to their mutual benefit.

7. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.engappai.2015.07.004.

Acknowledgements

The work described in this paper was partially funded by the European project RUBICON (Robotic UBIquitous COgnitive Network). Grant agreement no.: 269914.

References

Alam, M., Reaz, M., Ali, M.Mohd., 2012. An inhabitant activity prediction algorithm for smart homes. IEEE Trans. Syst. Man Cybern. Part A: Syst. Hum. 42 (4), 985–990.

Allen, J.F., 1984. Towards a general theory of action and time. Artif. Intell. 23 (July (2)), 123–154.

Amato, G., Chessa, S., Vairo, C., 2010. Mad-wise: a distributed stream management system for wireless sensor networks. Softw. Pract. Exp. 40 (5), 431–451. http://dx.doi.org/10.1002/spe.965.

Amato, G., Broxvall, M., Chessa, S., Dragone, M., Gennaro, C., Vairo, C., 2012. When wireless sensor networks meet robots. In: Proceedings of The Seventh International Conference on Systems and Networks Communications (ICSNC 2012). ThinkMind (TM) Digital Library, November 18–23, 2012, pp. 35–40.

Amato, G., Broxvall, M., Chessa, S., Dragone, M., Gennaro, C., López, R., Maguire, L., McGinnity, T.M., Micheli, A., Renteria, A., O'Hare, G., Pecora, F., 2012. Robotic ubiquitous cognitive network. Ambient Intelligence—Software and Applications, ser. Advances in Intelligent and Software Computing, 153. Springer, Berlin, Heidelberg, pp. 191–195.

Aztiria, A., Augusto, J.C., Basagoiti, R., Izaguirre, A., Cook, D.J., 2012. Discovering frequent user-environment interactions in intelligent environments. Pers. Ubiquitous Comput. 16 (1), 91–103 http://dblp.uni-trier.de/db/journals/puc/puc16.html.

Aztiria, A., Augusto, J.C., Basagoiti, R., Izaguirre, A., Cook, D.J., 2013. Learning frequent behaviors of the users in intelligent environments. IEEE T. Syst. Man Cybern. 43 (6), 1265–1278. http://dx.doi.org/10.1109/TSMC.2013.2252892.

Bacciu, D., Chessa, S., Gallicchio, C., Lenzi, A., Micheli, A., Pelagatti, S., 2012. A general purpose distributed learning model for robotic ecologies. In: Proceedings of the 10th International IFAC Symposium on Robot Control (SYROCO'12), 10, September 5–7, 2012, pp. 435–440.

Bacciu, D., Gallicchio, C., Micheli, A., Di Rocco, M., Saffiotti, A., 2014. Learning context-aware mobile robot navigation in home environments. In: The 5th International Conference on Information, Intelligence, Systems and Applications, IISA 2014, July 2014, pp. 57–62.

Bacciu, D., Barsocchi, P., Chessa, S., Gallicchio, C., Micheli, A., 2014b. An experimental characterization of reservoir computing in ambient assisted living applications. Neural Comput. Appl. 24 (6), 1451–1464.

Broxvall, M., Seo, B., Kwon, W., 2007. The PEIS kernel: a middleware for ubiquitous robotics. In: Proceedings of the IROS-07 Workshop on Ubiquitous Robotic Space Design and Applications, San Diego, California, 2007.

Chandola, V., Banerjee, A., Kumar, V., 2012. Anomaly detection for discrete sequences: a survey. IEEE Trans. Knowl. Data Eng. 24 (5), 823–839.

Cook, D.J., Krishnan, N., 2014. Mining the home environment. J. Intell. Inf. Syst., 1–17.

Cook, D.J., Crandall, A.S., Thomas, B.L., Krishnan, N.C., 2013a. Casas: a smart home in a box. Computer 46, 7.

Cook, D.J., Krishnan, N.C., Rashidi, P., 2013b. Activity discovery and activity recognition: a new partnership. IEEE T. Cybern. 43 (3), 820–828.

Coradeschi, S., Cesta, A., Cortellessa, G., Coraci, L., Gonzalez, J., Karlsson, L., Furfari, F., Loutfi, A., Orlandini, A., Palumbo, F., Pecora, F., Von Rump, S., Štimec, A., Ullberg, J., Östlund B., 2013. Giraffplus: combining social interaction and long term monitoring for promoting independent living. In: Proceedings of Human System Interaction (HSI).

Di Rocco, M., Pecora, F., Kumar, P., Saffiotti, A., 2013. Configuration planning with multiple dynamic goals. In: Proceedings of AAAI Spring Symposium on Designing Intelligent Robots, 2013.

Dragone, M., Abdel-Naby, S., Swords, D.,O'Hare, G., Broxvall, M., 2012. A programming framework for multi-agent coordination of robotic ecologies. In: ProMAS, 2012, pp. 72–89.

Fratini, S., Pecora, F., Cesta, A., 2008. Unifying planning and scheduling as timelines in a component-based perspective. Arch. Control Sci. 18 (2), 231–271.

Gallicchio, C., Micheli, A., 2011. Architectural and Markovian factors of echo state networks. Neural Netw. 24 (5), 440–456.

Hodge, V., Austin, J., 2004. A survey of outlier detection methodologies. Artif. Intell. Rev. 22 (2), 85–126. http://dx.doi.org/10.1007/s10462-004-4304-y.

Huynh, T., Fritz, M., Schiele, B., 2008. Discovery of activity patterns using topic models, Int. Conf. Ubiquitous Comput., 10–19, 2008.

International federation robotics, service robot statistics, world robotics 2014 service robots, December. Available online at: ⟨http://www.ifr.org/service-robots/statistics⟩.

Jaeger, H., Haas, H., 2004. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 304 (5667), 78–80.

Jakkula, V.R., Cook, D.J., 2011. Detecting anomalous sensor events in smart home data for enhancing the living experience. In: Artificial Intelligence and Smarter Living, ser. AAAI Workshops, vol. WS-11-07. AAAI, 2011. Available online at ⟨http://dblp.uni-trier.de/db/conf/aaai/aisl2011.html⟩.

Kurz, M., Ferscha, A., 2010. Towards a framework for opportunistic activity and context recognition. In: 12th ACM International Conference on Ubiquitous Computing (Ubicomp 2010), Workshop on Context.

Leng, G., Prasad, G., McGinnity, T.M., 2004. An on-line algorithm for creating self-organizing fuzzy neural networks. Neural Netw. 17 (December (10)), 1477-–1493. http://dx.doi.org/10.1016/j.neunet.2004.07.009.

Leng, G., Ray, A.K., McGinnity, T.M., Coleman, S., Maguire, L., 2013. Online sliding window based self-organising fuzzy neural network for cognitive reasoning. In: COGNITIVE 2013, The Fifth International Conference on Advanced Cognitive Technologies and Applications, Valencia, Spain: IARIA, May 2013, pp. 114–119.

Lotfi, A., Langensiepen, C., Mahmoud, S.M., Akhlaghinia, M., 2012. Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. J. Ambient Intell. Humaniz. Comput. 3 (3), 205–218.

Lundh, R., Karlsson, L., Saffiotti, A., 2007. Plan-based configuration of an ecology of robots. In: 2007 IEEE International Conference on Robotics and Automation, IEEE, 2007, pp. 64–70.

Lundh, R., Karlsson, L., Saffiotti, A., 2008. Autonomous functional configuration of a network robot system. Robot. Auton. Syst. 56 (October (10)), 819–830.

Markou, M., Singh, S., 2003. Novelty detection: a review — part 2: Neural network based approaches. Signal Process. 83 (December (12)), 2499–2521. http://dx.doi.org/10.1016/j.sigpro.2003.07.019.

Marsland, S., 2003. Novelty detection in learning systems. Neural Comput. Surv. 3, 157–195.

Niebles, J.C., Wang, H., Fei-Fei, L., 2008. Unsupervised learning of human action categories using spatialtemporal words. Int. J. Comput. Vis. 79 (3), 299–318.

Palumbo, F., Barsocchi, P., Gallicchio, C., Chessa, S., Micheli, A., 2013. Multisensor data fusion for activity recognition based on reservoir computing. In: Botía, J., Alvarez-Garcia, J., Fujinami, K., Barsocchi, P., Riedel, T. (Eds.), Evaluating AAL Systems Through Competitive Benchmarking, ser. Communications in Computer and Information Science, 386. Springer, Berlin, Heidelberg, pp. 24–35.

Puteh, S., Lotfi, A., Langensiepen, C.S., 2013. Activities recognition in intelligent office environment. In: BotÃa, J.A., Charitos, D. (Eds.), Workshop Proceedings of the 9th International Conference on Intelligent Environments (IE ‘13), held in Athens, Greece, ser. Ambient Intelligence and Smart Environments, 17. IOS press, pp. 393–402.

Rashidi, P., Cook, D., Holder, L.B., Schmitter-Edgecombe, M., 2011a. Discovering activities to recognize and track in a smart environment. IEEE Trans. Knowl. Data Eng. 23 (4), 527–539.

Rashidi, P., Cook, D., Holder, L., Schmitter-Edgecombe, M., 2011b. Discovering activities to recognize and track in a smart environment. IEEE Trans. Knowl. Data Eng. 23 (4), 527–539.

Rashidi, P., 2009. The resident in the loop: adapting the smart home to the user. IEEE Trans. Syst. Man Cybern. J. 39 (5), 949–959.

Ray, A.K., Leng, G., McGinnity, T.M., Coleman, S., Maguire, L., 2012. Development of cognitive capabilities for smart home using a self-organizing fuzzy neural network. In: 10th IFAC Symposium on Robot Control, 10, Dubrovnik, Croatia, 2012, pp. 447–454.

Robotic ubiquitous cognitive network (rubicon) website. 2014. Available online at ⟨http://www.fp7rubicon.eu⟩.

Robot-era, implementation and integration of advanced robotic systems and intellgent environments in real scenarios for the ageing population, December. Available online at ⟨http://www.robot-era.eu/robotera⟩ (accessed July 2015).

Roomba robot vacuum cleaner, December. Available online at ⟨http://www.irobot.com⟩ (accessed July 2015).

Roggen, D., Calatroni, A., Fãrster, K., Trãster, G., Lukowicz, P., Bannach, D., Ferscha, A., Kurz, M., Hãlzl, G., Sagha, H., Bayati, H., del R. Millãn, J., Chavarriaga, R., 2011. Activity recognition in opportunistic sensor environments. In: Procedia Computer Science, 7, 0, 173-174, 2011, Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11). Available online ⟨http://www.sciencedirect.com/science/article/pii/S1877050911005631⟩.

Saffiotti, A., Broxvall, M., 2005. PEIS ecologies: ambient intelligence meets autonomous robotics. In: Proceedings of the 2005 joint conference on smart objects and ambient intelligence: innovative context-aware services: usages and technologies. ACM, 2005, pp. 277–281.

Saffiotti, A., Broxvall, M., 2008. Affordances in an ecology of physically embedded intelligent systems. In: Rome, E., Hertzberg, J., Dorffner, G. (Eds.), Towards Affordance-Based Robot Control, ser. Lecture Notes in Computer Science, 4760. Springer, Berlin, Heidelberg, pp. 106–121.

Saunders, J., Burke, N., Koay, K.L & Dautenhahn, K., 2013, 'A user friendly robot architecture for re-ablement and co-learning in a sensorised home'. in P Encarnacao, L Azevedo, G J Gelderblom, A Newell & N-E Mathiassen (eds), Assistive Technology: From Research to Practice. vol. 33, Assistive Technology Research Series, vol. 33, IOS Press, Amsterdam, pp. 49-58, AAATE 2013, Villamoura, Portugal, 19-22 September, http://dx.doi.org/10.3233/978-1-61499-304-9-49.

Tang, F., Parker, L.E., 2005. Asymtre: automated synthesis of multi-robot task solutions through software reconfiguration. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, ICRA 2005, April 18–22, 2005, Barcelona, Spain, 2005, pp. 1501–1508. Available online at http://dx.doi.org/10.1109/ROBOT.2005.1570327.

Tapia, E., Intille, S., Larson, K., 2004. Activity recognition in the home using simple and ubiquitous sensors. Pervasive Comput., 158–175.

Vance, P., Das, G., McGinnity, T., Coleman, S., Maguire, L., 2014. Novelty detection in user behavioural models within ambient assisted living applications: an experimental evaluation. In: IEEE International Conference on Robotics and Biometrics (ROBIO), 2014.

Vig, L., Adams, J.A., 2006. Multi-robot coalition formation. IEEE Trans. Robot. 22 (4), 637–649.

Wan, J., O'Grady, M.J., O'Hare, G.M., 2014. Dynamic sensor event segmentation for real-time activity recognition in a smart home context. Pers. Ubiquitous Comput., 1–15.

Widmer, G., Kubat, M., 1996. Learning in the presence of concept drift and hidden contexts. Mach. Learn. 23 (1), 69–101. http://dx.doi.org/10.1023/A:1018046501280.

Ye, J., Dobson, S., McKeever, S., 2012. Situation identification techniques in pervasive computing: a review. Pervasive Mob. Comput. 8 (1), 36–66. http://dx.doi.org/10.1016/j.pmcj.2011.01.004.

Zadeh, L.A., 1975. The concept of a linguistic variable and its application to approximate reasoning – i, 1975, pp. 199–249.