To appear in *Applied Artificial Intelligence*
Vol. 00, No. 00, Month 20XX, 1–25

A Smart Environments Architecture (SEArch)

J. Augusto[a,b]* J. Giménez-Manuel [a,b] M. Quinde [a,b,c] Ch. Oguego[a,b] M. Ali[a,b] C. James-Reynolds[a,b]

[a] *Research Group on Development of Intelligent Environments*
[b] *Department of Computer Science, Middlesex University, London, UK*
[c] *Departamento de Ingeniería Industrial y de Sistemas, Universidad de Piura, Piura, Perú*

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

We report on a system architecture, SEArch, and its associated methods and tools we have been developing, testing and extending for several years through a number of innovation processes in the field of Smart Environments.

We have developed these infrastructure in a bottom up fashion directed by the needs of the different projects as opposed to an ideal one which projects have to conform to. In this sense is practical and although necessarily incomplete, it has significant versatility and reasonable efficiency. Projects developed using this architecture have been funded by different companies and funding bodies in Europe.

The different components of the architecture are explained through the software supporting those aspects of the system and through the functionality they exhibit in different practical scenarios, extracted from some of the projects implemented with SEArch.

Keywords: Smart environments, sensing, actuation, human behaviour, Ambient Assisted Living, human-centric computing.

1. Introduction

Sensing technology has become one important enabler and stimulating source of innovation in ICT, Computer Science and technology with societal impact Augusto and Aghajan (2019); Augusto et al. (2013). Academia and industry have recently produced a huge number of systems based on the concept of “smart technologies”, suggesting the capability to gather precise contextual information through sensing, supports more effective decision-making.

Much of these developments are exploratory as some of these technologies are still recent and not as reliable as desirable. Many of these environments are engineering in nature, systems and methods are being developed bottom up and there is a lack of methodologies and other community resources which act as standards or at least as guides of good practice. There are also few shared resources, beyond some datasets, and therefore most teams are forced to “reinvent the wheel”.

Systems in this area are typically a combination of subsystems: Artificial Intelligence, Computer Networks, Human-computer Interface, and Mobile Computing. This exacerbates the difficulties of sharing and reusing as each system can be conceived and developed in so many different ways. Hence the creation of methodologies, guidelines, and good practices that we can share with interested communities is valuable. Part of our effort has been directed towards connecting tools which correspond to the common critical tasks required in systems of this area. Trying to make it practically effective, sometimes sacrificing expressiveness in favour of system features which can help adoption such as: response time, user friendliness, and overall cost.

Here we present a Smart Environments Architecture currently in use at the Smart Spaces lab

[*] Corresponding author. Email: j.augusto@mdx.ac.uk

of Middlesex University. We illustrate how its components work cooperatively to provide services in environments such as Smart Homes.

2. Related Work

Some system architectures have been used and reported in the literature, usually with a specific focus on the technical progress they were reporting. Here we are considering comprehensive system level architectures so this area will favour higher level architectures reported in the technical literature. Our typical target will be Smart House or Smart Office systems which use several sensors and interfaces and are supposed to provide services for more than one user. The literature of this area is very prolific and there are many systems reported which we cannot possibly fully cover here. We focus on publications offering a wider and higher level view of system architectures for context-awareness systems. Typical terms used in the literature to refer to them are "Smart Environments", "Intelligent Environments", "Pervasive Systems", "Ubiquitous Systems", and "Internet of Things Systems". We also have a strong interest in so called "User-centred Systems" which not necessarily have to be sensing supported and/or context-aware able, however given the availability of sensing supported technology pervading our daily life, user-centred systems based on sensing and context-awareness are becoming increasingly intertwined.

The summary of the selected systems which report system architectures of our interest are presented in historical order.

Saif provided a detailed Ubiquitous System architecture (UbiqtOS/Romvets) through various levels of description Saif (2002). The various levels include Middleware, context-awareness handling, reasoning based on production system driven agents. The system is focused at quite a low level of services, and is very representative of systems at that time. The MavHome system architecture Cook et al. (2003) was organized in four main tiers: Physical, Communication, Information, and Decision. Reasoning is made through a multi-agent system. Learning was linked to the multi-agent system. Fernández-Montes et al. (2009) structured their system based on the Perception-Reasoning-Acting cycle arising from the multi-agents area. Within the Perception section they encapsulated tasks related to data acquisition, there is an explicit mention of ontology and a number of middleware related tasks. The Reasoning section also includes a Data Mining sub-module. Heider presented Heider (Heider) a system to assist users of a multi-purpose smart meeting room. The system was approached as a goal oriented problem solver, supplemented with a range of interesting features including a variety of interfaces, middleware and ontologies to support reasoning by planning. Although the system is not strong on learning they do emphasize on the concept of a personal environment for each system user. The CASAS system architecture Cook et al. (2013) has three main tiers: Physical, Middleware and Applications, the later containing sub-modules as activity recognition and activity discovery were reasoning and learning are used. One very significant European project was universAAL Ferro et al. (2015), which was mostly focused on providing support for AAL systems and tried to tackle one of the main area bottlenecks: interoperability amongst the myriad of different market solutions from different vendors. As such universAAL was very beneficial in many ways, with a significant and ambitious architecture including good services management complemented with conext handling. The main difference we see with our architecture is that in SEArch we place more emphasis on higher levels of the architecture (reasoning, learning, user preferences, logical inconsistency handling) and very little so far on interoperability. In this sense both SEArch and universAAL are complementary. Another comprehensive modern development is FIWARE Fiware (2019), which to a large extent has the same areas of strength than universAAL with more support at lower levels on the middleware interfacing various IoT devices with the network and with context formation, there are clear statements that context in Fiware is largely dominated and defined by IoT data whilst in SEArch we place more emphasis on a user driven notion of context so not all IoT data is part of a context, only if it can contribute to a context

System	M	O	R	L	H	U	S
Saif (2002)	√		√				
MavHome - Cook et al. (2003)	√		√	√			
Fernández-Montes et al. (2009)	√	√	√	√			
Heider (Heider)	√	√	√		√	√	
CASAS - Cook et al. (2013)	√		√	√			
universAAL - Ferro et al. (2015)	√	√			√		√
FIWARE - Fiware (2019)	√				√		√
SEArch			√	√	√	√	√

Table 1. System Architectures Comparison

previously identified as important to a service explicitly requested by the user. Again in SEARch we put more emphasis on higher level reasoning and learning (beyond context-awareness), whilst in Fiware these are largely left to the developers of future applications as they need. From all those systems above there is a more recent trend, to make available substantial building components as Open Source to stimulate growth and community uptake. Amongst these universAAL, FIWARE and SEArch share more of their methods and tools.

Table 1 gives a summary of the systems mentioned above, where the following abbreviations were used: M=Middleware, O=Ontology, R=Reasoning, L=Learning, H=human-computer interaction options, U= User-centred, and S=Open Source. These dimensions and the resulting scoring assigned to each of them in the table are difficult to measure given that is difficult to quantify. All systems have some sort of human-computer interaction option, however only those with a higher emphasis in that aspect have a tick, an absence of a tick in a cell does not necessarily indicates complete absence, so the table reflects more clearly the strengths and differences of each approach.

There are other publications of ideal abstract architectures (see for example, Badica et al. (2013) and Belaidouni et al. (2016)), however they do not report on a live implemented system so we did not include them in the table above and the overall comparison. Other systems like Piyare (2013) and Lewis et al. (2008) were implemented mostly focused on sensors with a Middleware glue, although they report working prototypes they do not fit in the wider analysis we reflect in our table. One salient feature in comparing SEArch with other related options is that we do not yet include middleware and ontology. This has to do with strategic decisions made in the past on which were considered the most fundamental components and where to start building the system from. The initial obvious priority was the acquisition of affordable standard sensing and standard network communication. The first higher priority software components were reasoning and learning modules. We considered if we did not have good ways for the system to make sensible decisions about interesting practical situations then the system will not succeed in the long run. The next level of priority was with interfaces providing meaningful interaction between human(s) and system. These interfaces provide the flow of information from-to environment and system components.

The current step we are focusing on is to organize and standardize data, information and knowledge flow. Other teams started from this point, imagining what services the system will need in the future and starting by providing first a middleware and ontology support for them. These are strategic decisions over which probably there is no definitive answer and it may also be argued that eventually both can lead to good systems with enough time and resources. So we are not arguing here our approach is better than anyone else, we are only reporting on the approach we followed. The user-centred dimension is a significant one for us. We have methodologies and tools which guide design, development and validation throughout all stages of the creative process. Examples of that support have been reported in Jones et al. (2015); Evans et al. (2014); Augusto et al. (2017).

3. System Architecture Explanation

Our system evolved in a succession of bottom up developments and top down system reviews. Its components were created out of genuine practical needs driven by practical challenges we faced on the way. The evolution of the system has been informed by a variety of projects we have taken part of at national and international level, most of which had business and industrial involvement. One common feature in all this projects was the need to support humans with well-being priorities Martin et al. (2013) through technology which is usually referred to as Ambient Assisted Living, AAL, (Li et al. (2015); Murabet et al. (2018)). Although the assistance was supported by data gathered at home environments, some of these projects used the information acquired by the system to support citizens outside the home (Covaci et al. (2015); Kramer et al. (2015)).

At the same time the practical challenges we addressed surpassed what is being considered in the market. Whilst business is also engaged on innovation, it is a sector with different goals. priorities and resource constraints. This led to technical innovation focusing on affordable 'gadgets' to be the latest Christmas sell hit. These business led innovations are necessary steps of technological evolution and in different degrees, beneficial to society. However there are problems and issues which are often sidelined because they require higher investment in terms of human resources dedication and have longer term less obvious return on investment. This is how in our area of specialism there is an abundance of isolated systems however there is a lack of methods and tools. Although we cannot promise to solve all of these long term obstacles for development, we address some of them in a persistent way. As a result in this article we share the system underpinning our development, some of which have strategies associated, some of which have tools to support them. We do not claim these are the ultimate and definitive solution, rather we think more discussion on these topics is needed and academia is the natural place where discussions at this level can happen, this article is offering one such discussion and debate opportunity. In general there is an absence of publications which share the methods and tools used. There seems to be a significant number of systems developed 'from scratch', in some sort of continuous 'reinventing of the wheel' exercise. Although the constructivist approach of developing a system in itself is positive and a good training exercise, it does not help the scaling up, it does not help our technical community to develop better specialized tools, as a result significant energy is used to develop innovation with old inadequate methods and tools or almost with no methods at all. Sometimes good things come from random attempts, however, usually it is an inefficient waste of precious energy and time.

Traditionally home automation started as a building engineering activity more formally known as Domotics Asadullah and Raza (2016). On one hand there is the issue of isolated unconnected technology, so an alarm pendant is useful for a person who may be at risk at home, however connecting it to other home infrastructure can provide additional improvements (for example, when the location sensors can understand in which part of the house the person is). In some cases there could be some connected technology which does not integrate to other set of connected technology. Interoperability has been one of the main barriers hindering progress Furfari et al. (2011); Asad and Taj (2018). Hence one important principle we have kept throughout our project is the adoption of a single type of technology, in our case all technology is wireless and all use the same communication protocol (this is explained in greater detail in a later section).

Figure 1 shows an overall picture of the system architecture. There are ways of capturing data on the right hand side of the figure and ways to convey information back to the system on the left hand side, these involve sensing and interfaces. Various databases are used as repositories within the system. The most important working components are the different resources for contextualization, reasoning, learning, system-user interaction and user personalization. The following sections will explain the main components and their interdependencies.

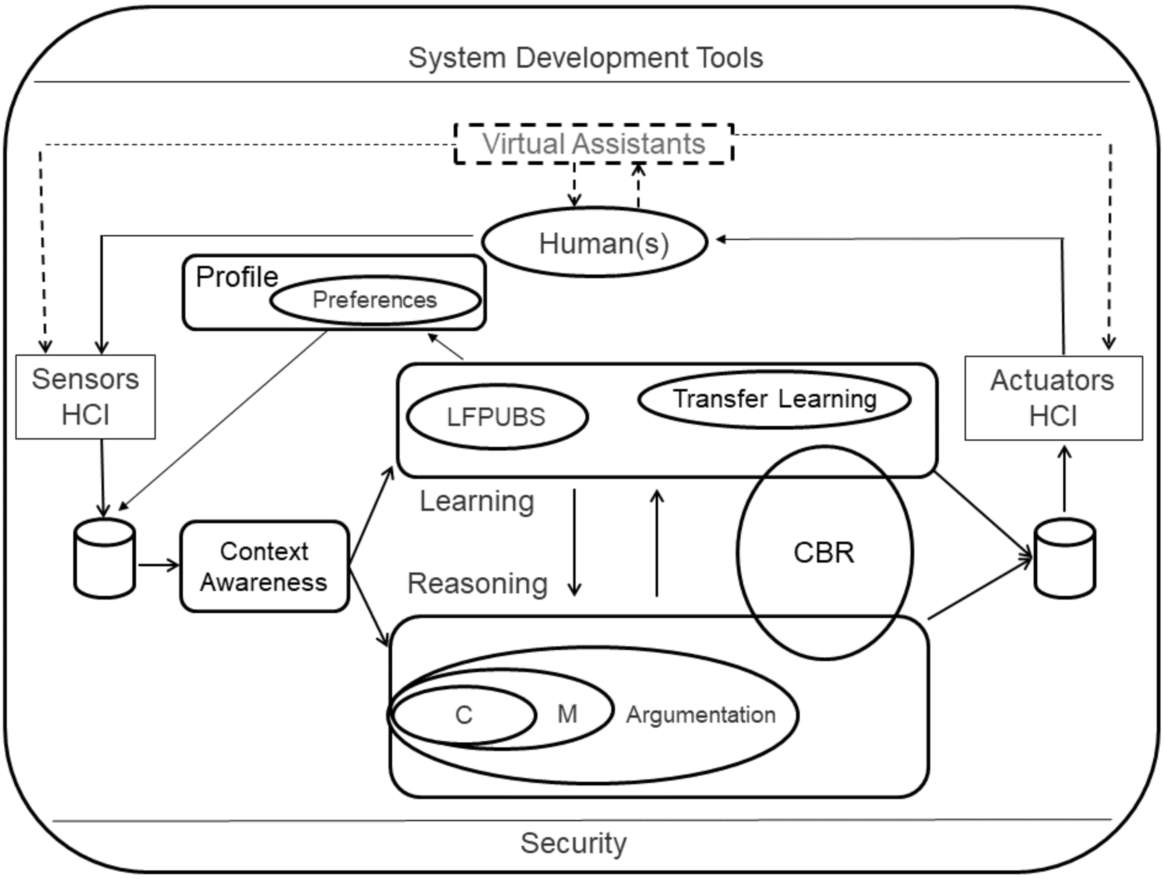


Figure 1. SEArch System Architecture Diagram

3.1. *Interfacing with the World*

Any intended application of our system is supposed to be in contact with some physical world and be deployed there to primarily for benefit of humans. Here we adopt on the definitions and assumptions associated with Intelligent Environments as made in Augusto et al. (2013), that is we assume the existence of some reasonably well delimited part of the physical world, enriched with sensing and actuation capabilities, as well as other IT equipment, including a range of interfaces. The system is expected to consume data which reflects what happens in that physical world, using sensing and interfaces which depend on the application domain, process that data and then convey information or actuate in the environment. It could be the system takes information from one physical environment and actuates in a complete different environment, however we will assume here systems which actuate in the same environment they observe and monitor, for example, a Smart Home.

Our Smart Spaces Lab consists of a self-contained building in Hendon Campus (North London). Figure 2 below shows the lab layout and its sensing/actuation infrastructure. By "sensor" we understand a device which *can transform a physical dimension into a digital signal*, for example the presence of light into a digit 1, and by "actuator" one which *can transform a digital signal into a physical dimension*, for example a digit 1 command into a light being turned on.

Most of the building is dedicated to the Ambient Assisted Living space which replicates a normal house with exception of two rooms on the right hand side of the picture which are multi-purpose.

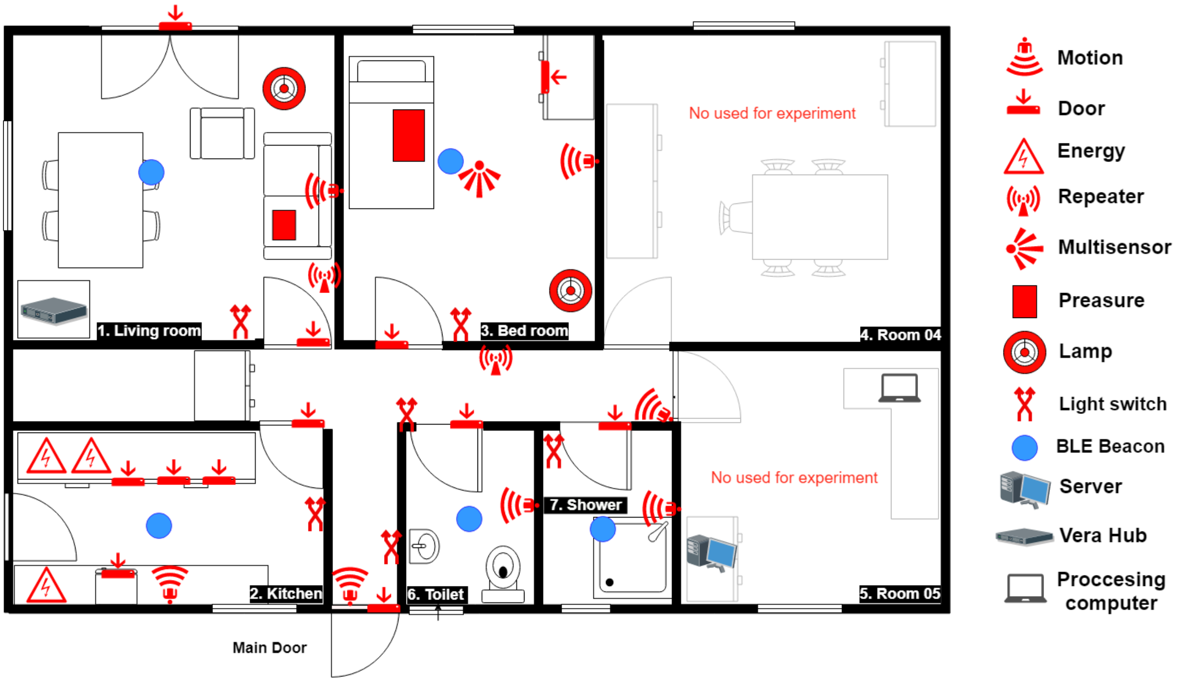


Figure 2. Smart Spaces Lab Layout

The wireless sensor system is mostly a network of Zwave-compatible commercial sensors (except for the pressure pad which was converted from wired to ZWave wireless in the lab). The Sensor Control box is a Vera Plus box which is shown at the centre of Figure 3. Also in that figure, at the top right hand side, a big pad with a pressure sensor is shown. Then clockwise, in the middle there is a multisensor which can sense movement, luminosity, temperature and humidity. On the lower right corner there is a window/door opening sensor which we attach not only to room windows and doors but also to the fridge door and to kitchen covert doors and wardrobe doors to understand and assess health indicators in fundamental daily life activities. Next, clockwise there is a picture of a sensor which detects movement (Passive Infra Red, P.I.R.). The next three are both sensors and actuators. At the bottom centre left one which can detect whether a device plugged in to it is being used or not as well as turn on/off the device. In the lower left corner there is a sensor-actuator which can tell whether the lamp bulb is on or off as well as turn it on-off through a command. The picture on the upper left shows a touch light switch which again can be used to turn on-off room ceiling lights in the traditional way by the user with fingers or automated by the system sending the appropriate command.

All sensing is non-intrusive in the sense it does not capture sound or real images from the occupants of the house. All sensing is passive, not requiring sophisticated interaction from the users and it also does not interfere with the life style of the individual.

There is only one exception to the Zwave sensing, that is a set of BLEs (Bluetooth Low Energy) boxes from RAD which are used to individualize environment occupants and better personalize services (more on this in section 4).

Some of those devices, can produce very simple binary data. For example, a Passive Infra Red sensor exposed to a glow of energy, within certain range of perception and strong enough to reach a specific threshold of sensitivity, will trigger the sensor producing a simple bit ‘1’ being transmitted from the sensor to a central processing unit the sensor has been paired with. Then it is up to other system components to give that ‘1’ stored in a database adequate interpretation. A typical meaning

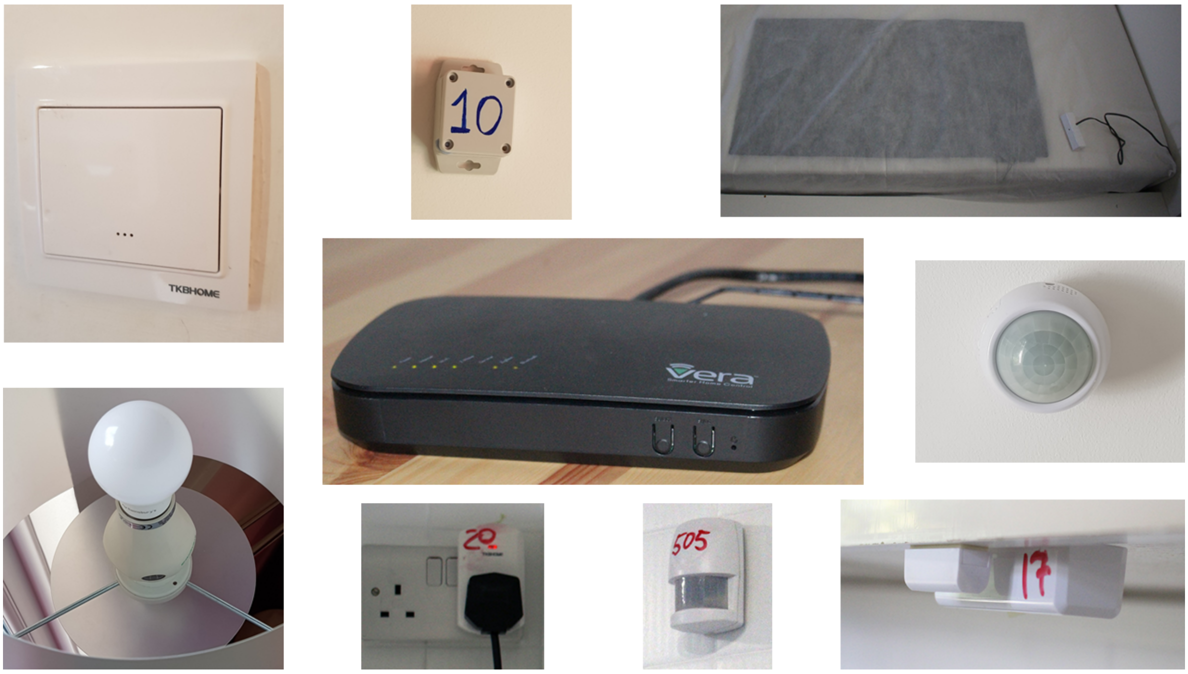


Figure 3. Sensing Equipment and Vera Box (top right)

is that of movement detected, depends on the context what produced that movement, typically in they type of environments we consider that will be a human.

Other sensors can provide more complex output, for example a temperature sensor will provide a real number. Other ways of acquiring information are far more sophisticated providing several simultaneous dimensions per unit of time, for example a microphone capturing sound or a camera capturing video.

Examples of input data mentioned above were mostly passive ways of collection, so that they can be deployed in an environment and humans may not even have to pay attention or interact with them explicitly for the acquisition of data to take place. Humans however can also pro-actively act to provide data to the system, this typically can happen through an interface when making a choice, when answering a question posed by the system in a screen or when giving an order to the system, verbally or through a gesture.

The system interaction with the system Here by sensing we refer to *a device which transform a digital dimension into a physical dimension*. Examples of this could be transforming a tap in an option to trigger the turning on or off of a device, a decision made by the system creating a sound or a message which is visible or audible by a user.

One important thing to notice is that rarely only one input is very clear and powerful in meaning. This can happen in a system which has been designed to have a few pre-specified options which trigger complex concepts, although such systems are usually very dedicated and not very flexible. For systems which are conceived to observe and consider a wider range of scenarios, decision-making is more complex: it takes a bigger combination of data, more potential interpretations are possible and there are various possible courses of actions. An example of the first type could be touching specific areas of a touch screen device to selecting the purchase of a product through an app. There are a few predetermined options with quite well defined semantics which if performed in a specific sequence will cause an object shipped (perhaps even created) to a specific other location of the planet. An example of the later type of systems is a smart home where the system may be expected to provide comfort to a human using a specific area of the building, useful parameters the system can consider may be ambient temperature, location of the human, position of the body, activity being developed, perhaps even mood, which may support decisions on whether to activate/deactivate temperature regulating devices such as heaters or air conditioners, altering luminosity levels, starting/stopping/retuning entertaining devices, ordering specific food, suggesting activities, etc.

Another aspect noticeable at the top of figure 1 is the possibility to consider Virtual Assistants which offer options for interaction with users (see for example, Nurgaliyev et al. (2017) and Ospan et al. (2018)). These are linked to the overall architecture as it is an optional feature, although in practice is common. This is sometimes a contentious issue in the success of a system. We give high importance to systems which interact with users in the most natural way possible (Weiser (1991), Augusto et al. (2013)). How 'natural' and 'friendly' these interfaces are can make a big difference on the success of system adoption. Interfaces and virtual assistants which favour interaction options not requiring specialization will eliminate barriers for humans. For example 'touch and select' options in screens or use of voice, will be more likely to benefit a wider range of users. This also has an ethical side to this approach on reducing technological discrimination Jones et al. (2015).

3.2. *Context-Awareness*

Context-awareness is an essential aspect of Intelligent Environments. These are environments reactive to specific circumstances in expected ways. consistently with our user-centred approach we differentiate our approach from other more system and data oriented approaches Dey and Abowd (1999) by emphasizing the humans which are supposed to benefit with the system:

Context: "the information which is directly relevant to characterise a situation of interest to the stakeholders of a system".

Context-awareness: "the ability of a system to use contextual information in order to tailor its services so that they are more useful to the stakeholders because they directly relate to their preferences and needs".

Our Context-aware System Architecture is shown in Figure 4. At the top the relevant contexts are determined based on user feedback, based on that the collectors are decided, and once the collectors retrieve information they are used to decide which contexts can be said have been detected, this process sometimes require some intra-context inferences, and sometimes contexts can also be aggregated to form higher level contexts and do inter-contexts reasoning. Finally once meaningful contexts have been detected they are related to the rest of the knowledge management part of the system (more general reasoning and learning) to produce changes in the environment which are fed back to the users.

Contexts are an important part of our system development as we know they really decide how responsive is the system to what the main beneficiaries care about and expect from the system, the behaviours which make users have the feeling the system is intelligently looking after their interests and improving their life experience. So we use contexts from a practical sense to guide our system testing and validation, see for example our COATI methodology Augusto et al. (2019).

3.3. *Reasoning*

Knowledge Representation and Reasoning have been for long established areas of Artificial Intelligence Russell and Norvig (2003). Powerful strategies have been explored, developed and applied in the last decades. Intelligent Environments carry in their name the expectation of being imbued with certain intellectual capabilities. In our case we are not necessarily addressing to solve core automated reasoning problems or long standing issues within the area. Mostly, we aim to provide a systems with a reasonable level of reasoning capabilities enough to deliver specific services at the

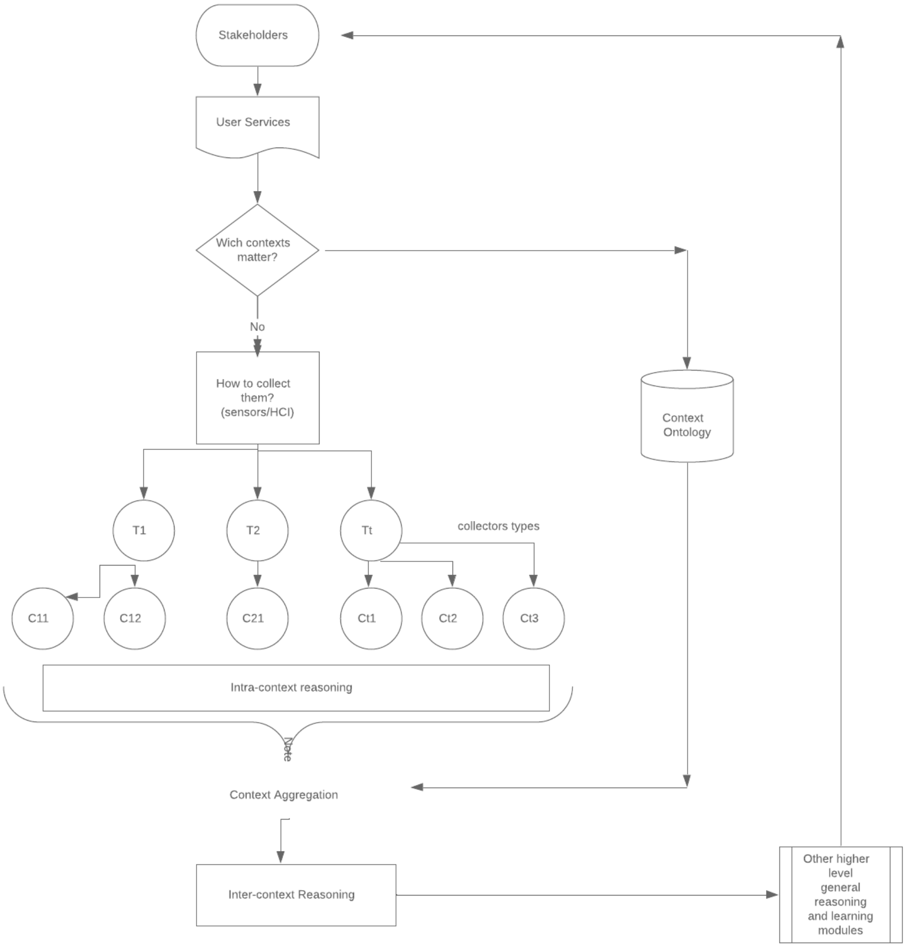


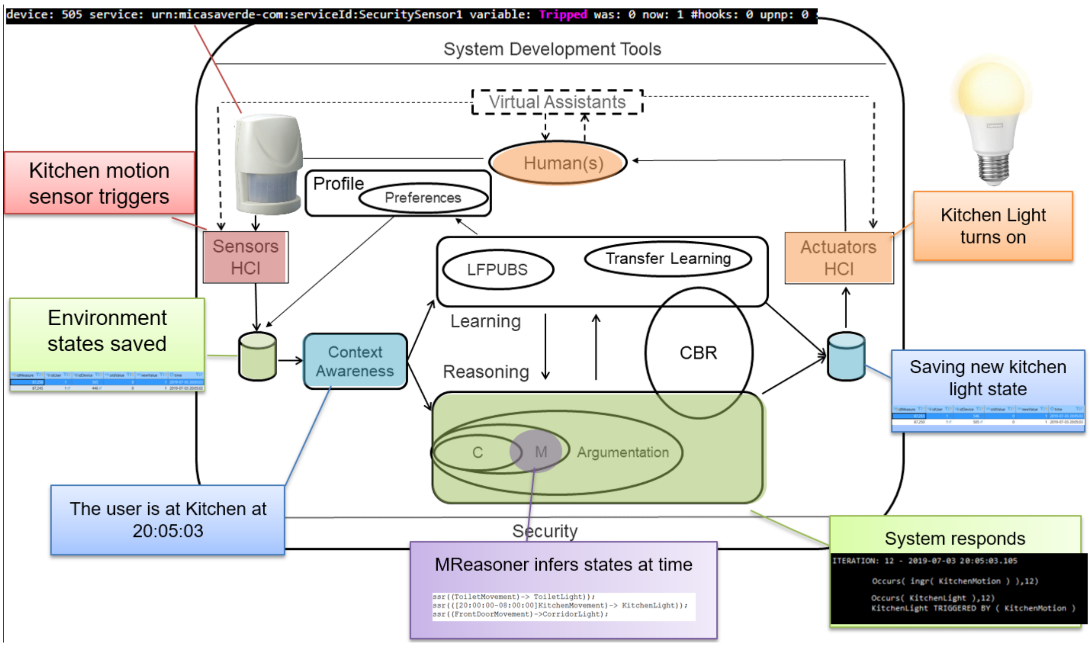
Figure 4. Context-awareness Architecture

same time it is ligthweight to be able to perform efficiently in real-time with constrained resources.

The architecture shows a number of integrated reasoning mechanisms, including M, a rule-based system which can reason with timed events Alegre et al. (2014). This consists basically of a deterministic system based on rules which are supplemented with temporal constraints stating when they may be considered for triggering. This temporal conditioners can be relative to the current system processing time or absolute as in a calendar type. This is enough to reason about states which have been holding for a while as well as to trigger services which depend on specific times or days of the week. As the M reasoner can only handle consistent and monotonic datasets, it cannot deal with conflicts and contradictions. Hence an extension in the form of an argumentation system has been developed Oguego et al. (2018). Another interesting aspect of the argumentation system is that when a decision has been taken it provides an explanation based on the winning argument. There is also a Case-based reasoner Ospan et al. (2018) which can handles the evolution of certain cases providing some level of adaption and personalization.

We do not state these are the only reasoning components or that all teams should use them, rather that these satisfied needs and solved challenges we encountered when developed various of our system and we believe they can be useful to other teams too. However, each team can plug here the reasoning modules they find most suitable for their problems and challenges. Also, as it can be seen in later sections where we provide examples of the system inner working based on practical examples, we tend to favour declarative or rule-based systems which facilitate understanding of system behaviour, explainability, traceability, testing and validation.

Figure 5 shows how different parts of the architecture are used when sensors are triggered, in this case due to movement in a room, and this in turn gets associated with contexts, modelled in rules, which trigger associated actions, in this case turning on the room lights.


```
device: 505 service: urn:micasaverde-com:serviceId:SecuritySensor1 variable: Tripped was: 0 now: 1 #hooks: 0 upnp: 0
```

```
ssr((ToiletMovement)-> ToiletLight));
ssr(([20:00:00-08:00:00]KitchenMovement)-> KitchenLight));
ssr((FrontDoorMovement)->CorridorLight);
```
```
ITERATION: 12 - 2019-07-03 20:05:03.105

        Occurs( ingr( KitchenMotion ) ),12)

        Occurs( KitchenLight ),12)
        KitchenLight TRIGGERED BY ( KitchenMotion )
```

Figure 5. SEArch System Architecture Being Exercised

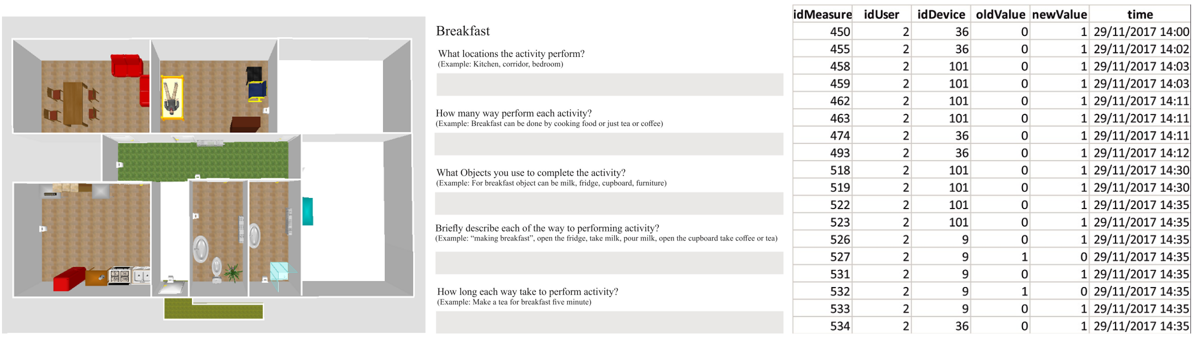
3.4. *Learning*

The capability of a system to learn from experience in a similar way as humans do and improve from experience has been also a relevant area since the early times of AI Mitchell (1997). Associated concepts with different emphasis on different aspects of the challenge are Machine Learning, Knowledge Discovery, Data Mining, and Deep Learning just to mention a few. By 'learning' here we will refer to mechanisms that distil new knowledge from the long history of the system, part of that new knowledge could be new factual atoms however more important discoveries involve relations between two knowledge components or the system discovering the need to adopt new behavioural rules. Again we do not rule out any tool or learning approach so different teams can plug here the module that represent the best strategy for a given problem. We have usually favoured methodologies which are declarative and more transparent on what they do, what they achieved and how they achieved it because of the benefits that brings to the understanding and development processes by the developing team.

Our learning is mostly focused on discovering habits from users so that we can adapt the system in a sort of high level personalization. For example, if the system realizes all days from Monday to Friday the user gets up more or less at the same time, say 7:45-8:00, go to the bathroom, stays there a few minutes, go to the kitchen, prepares breakfast, and leaves home, then some of the switch on/off activities can be automated.

To achieve the recognition of habits and their transfer to the system as knowledge which can be used in a practical way we have mostly relied on LFPUBS Aztiria et al. (2013), a system which transforms datasets into Event-Condition-Action rules. Although the rule-style output of LFPUBS system is quite close in style and syntax to the input consumed by reasoning modules such as our M reasoner, it still not directly usable and also it is not optimized for automation so a transformation is performed to bridge the two Aranbarri-Zinkunegi (2017).

Still there are other practical issues to be tackled to enhance alignment between system and human. One such obstacle is the initial personalization of a system to a specific user. Although some systems claim they can learn from the user, some of this aspects may be hard to learn, the learning may be inaccurate, and above all it requires a data sample of certain length to be statistically significant. Part of our methodology addresses this with human intervention to help the house to start with some meaningful data from the very beginning, addressing the “cold start” problem Bobadilla et al. (2012). Trough a combination of tools and processes we interview users, model, simulate, generate data and create rules (Figure 6), to give the system help to start with some knowledge of what a specific user lifestyles looks like Ali et al. (2019).



Breakfast

What locations the activity perform?
(Example: Kitchen, corridor, bedroom)

How many way perform each activity?
(Example: Breakfast can be done by cooking food or just tea or coffee)

What Objects you use to complete the activity?
(Example: For breakfast object can be milk, fridge, cupboard, furniture)

Briefly describe each of the way to performing activity?
(Example: “making breakfast”, open the fridge, take milk, pour milk, open the cupboard take coffee or tea)

How long each way take to perform activity?
(Example: Make a tea for breakfast five minute)

idMeasure	idUser	idDevice	oldValue	newValue	time
450	2	36	0	1	29/11/2017 14:00
455	2	36	0	1	29/11/2017 14:02
458	2	101	0	1	29/11/2017 14:03
459	2	101	0	1	29/11/2017 14:03
462	2	101	0	1	29/11/2017 14:11
463	2	101	0	1	29/11/2017 14:11
474	2	36	0	1	29/11/2017 14:11
493	2	36	0	1	29/11/2017 14:12
518	2	101	0	1	29/11/2017 14:30
519	2	101	0	1	29/11/2017 14:30
522	2	101	0	1	29/11/2017 14:35
523	2	101	0	1	29/11/2017 14:35
526	2	9	0	1	29/11/2017 14:35
527	2	9	1	0	29/11/2017 14:35
531	2	9	0	1	29/11/2017 14:35
532	2	9	1	0	29/11/2017 14:35
533	2	9	0	1	29/11/2017 14:35
534	2	36	0	1	29/11/2017 14:35

Figure 6. Co-design with Users of House Automation

3.5. *User-centric approach*

SEArch is mainly a system perspective and as such it shows how data flows through the system and how is processed. However, we emphasize that systems we develop with the SEArch template are meant to be for humans and the technological part of the system should be in the background and as little noticeable as possible Weiser (1991)). As such we put significant effort on following user-centred approaches rather that the more common system-centred approaches where sometimes well intentioned engineers ’build systems because they can, rather than because they should’ so there is an abundance of systems but low large scale uptake. Part of our user-centred approach is that it is embedded in the methods and tools we use, we try to design and develop systems with humans at the centre. The overall development strategy we use is the “User-centric Intelligent Environments process” Augusto et al. (2017).

Most of our user-centred approaches are actually embedded in the methods and tools we use to develop systems, however there is an echo of this concern in the upper part of the architecture where it can be explicitly seen a module that has been designed to store and process user preferences (see for example, Oguego et al. (2018, 2019)). This resource allows a way for the main users to tell the systems which preferences they have and it offers engineers a way to better align the system with the users expectations and this should lead to better user satisfaction with the system and a more positive perception of the importance and usefulness of the system.

Other elements of the user-centric approach have to do more with ethical principles where we place issues such as security, privacy and other aspects such as ensuring benefits for the user, preventing harm, transparency, data protection, autonomy, equality, and being sensitive to different users preferences. we encapsulated that into the “eFriends” methodology which goes beyond design and tracks that those desirable features listed above are mapped into practical aspects of the system through requirements Evans et al. (2014), implemented, tested and validated Jones et al. (2015).

4. Differentiating Amongst Multiple Users

The problem of being able to more precisely discern where a specific individual is in a specific environment has been a hard problem to solve in a satisfactory way. On one hand it is a system feature which can unlock many benefits because knowing who is where with precision not only helps accurate understanding of the context but also the next desirable step of being to deliver more specialized services which are better aligned to a specific user preferences. On another hand it is a design minefield quickly linked to issues of privacy and associated with words which have negative connotations like 'monitoring' and 'surveillance'. We consider there are situations where the benefits exceed the problems and also it depends very much on the individual perception of the people oriented. For some users they will not be concerned about the house knowing where they are at each time for as long as that is used positively and kept private. Also, in environments where there is more than an individual, because they live permanently there or because of visiting people, some useful services are not feasible unless the system is capable to know which ones are supposed to be provided with which services.

Different approaches have been tried and tested for several years. Initially RFID triggered a wave of identifying objects in industry, still used today, which was subsequently expanded to trying to identify humans passing through doors or being at specific places of a room Singh et al. (2014); Zhi and Singh (Zhi and Singh). Other widely available technologies such as PIRs and body-worn sensing have also been tried supplemented with sophisticated algorithms relating guessing individuals based on their actions and itineraries in an environment Crandall and Cook (2011); Roy et al. (2013). Location services based on WiFi fingerprinting Mok and Retscher (2007) as well as on Image Processing Bakar et al. (2016) have also been explored. All these approaches come with different level of advantages and disadvantages in the consideration of cost-benefit, privacy, security, etc. For example RFID requires the person wearing a very specialized equipment, PIR and worn sensors are also taxing on the user, WiFi fingerprinting requires significant initial mapping and regular recalibrations and video cameras have a huge privacy stigma attached plus higher costs. Benmansour et al. provides a survey Benmansour et al. (2015) of some of these earlier attempts.

Our system design to address multi-occupancy has been strongly influenced first by the seminal contribution from Palumbo et al. Palumbo et al. (2015) and then Sora et al. Sora and Augusto (2018) who designed and tested ways to personalize services based on the use of BLE (Bluetooth Low-energy) sensing. This approach is non-intrusive, reasonably accurate and cost-effective. The left hand side of Figure 7 shows one of BLE sensor. Each of these boxes can connect with other bluetooth enabled devices. One such device is a modern mobile phone. So an app evolved from the above cited work which can be deployed in a mobile phone and provides distance estimation based on signal strength. Of course this makes the system dependent on the user carrying the mobile phone, the system identifies person location with mobile phone location, which is a weakeness, from comparing various pros and cons from different approaches mentioned above considering reliability, cost and naturalness, we favoured the user of BLE. Sometimes a person placed in a spot which is roughly equidistant from two boxes in different rooms will cause the location algorithm to diagnose alternatively these two locations. Further algorithm processing can reduce these issues. Another way we follow which has complementary benefits is to place two LBE in each room increasing reliability. The right hand side of Figure 7 show how they are placed together in one room. Blue dots in Figure 2 show rooms where this infrastructure has been deployed in our lab.

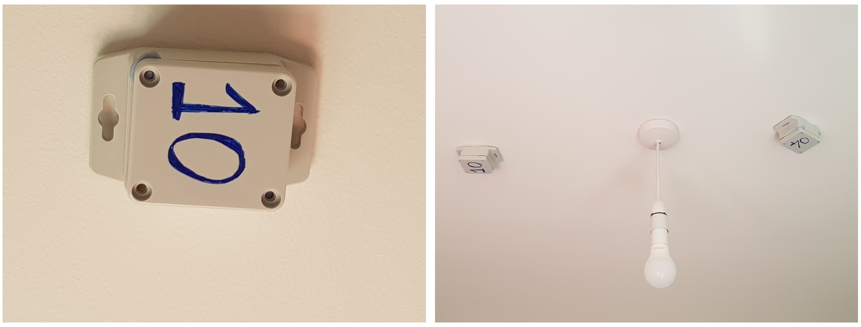


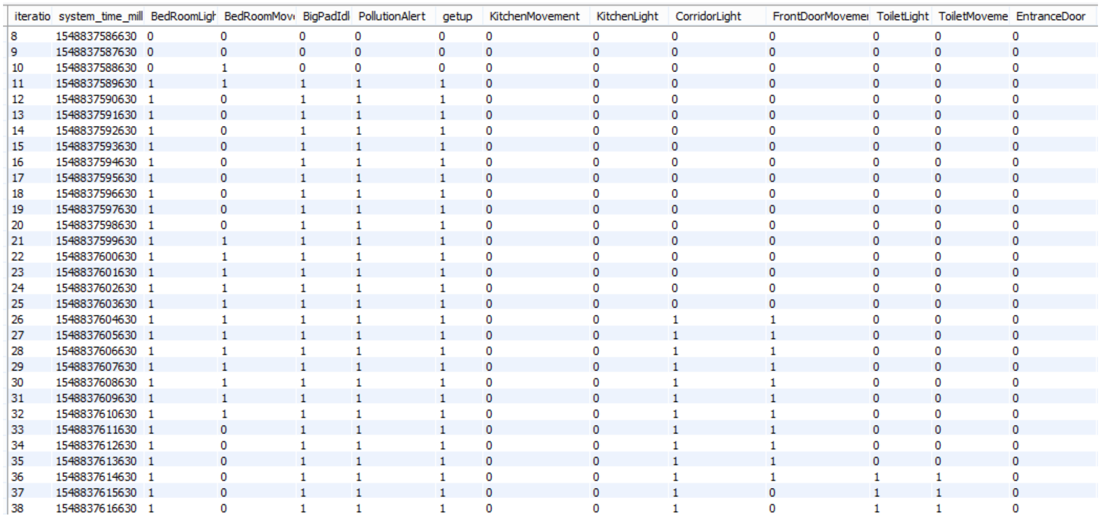
Figure 7. A BLE (left) and a set up in a room ceiling (right).

5. SEArch at Work

The real-time automation is based (mostly) on a rule based language with capabilities to check for conditions fulfilling certain temporal properties based on Alegre et al. (2014) and the learning is made through a combination of LFPUBS Aztiria et al. (2013) and an LFPUBS2MReasoner translation Aranbarri-Zinkunegi (2017). Preferences are managed through a system which uses user profiles mixed with the Mreasoner rules to create arguments pro-against the different possibilities open to the house Oguego et al. (2018).

Scenario description: This scenario captures part of the morning routine of a person with asthma. It starts with the person waking up, then going to the toilet, then to the kitchen, and finally leaving home. The system is sensitive to the user needs and checks the air pollution status after waking up as it has been indicated this is an important asthma trigger for this user. The systems turns on lights as needed and then turns them off when the user left home, figure 8.

Data collection: Sensors send wirelessly events to Vera Box. A log of those events are stored in a DB reflecting state changes. The DB is scanned in a continuous loop and rules in the system specification are checked for satisfaction at the current iteration (this can include states extending for a period of time). Rules triggered can change internal states or order wireless actuations through the Vera Box.



iteratio	system_time_mill	BedRoomLigh	BedRoomMov	BigPadIdl	PollutionAlert	getup	KitchenMovement	KitchenLight	CorridorLight	FrontDoorMoveme	ToiletLight	ToiletMoveme	EntranceDoor
8	1548837586630	0	0	0	0	0	0	0	0	0	0	0	0
9	1548837587630	0	0	0	0	0	0	0	0	0	0	0	0
10	1548837588630	0	1	0	0	0	0	0	0	0	0	0	0
11	1548837589630	1	1	1	1	1	0	0	0	0	0	0	0
12	1548837590630	1	0	1	1	1	0	0	0	0	0	0	0
13	1548837591630	1	0	1	1	1	0	0	0	0	0	0	0
14	1548837592630	1	0	1	1	1	0	0	0	0	0	0	0
15	1548837593630	1	0	1	1	1	0	0	0	0	0	0	0
16	1548837594630	1	0	1	1	1	0	0	0	0	0	0	0
17	1548837595630	1	0	1	1	1	0	0	0	0	0	0	0
18	1548837596630	1	0	1	1	1	0	0	0	0	0	0	0
19	1548837597630	1	0	1	1	1	0	0	0	0	0	0	0
20	1548837598630	1	0	1	1	1	0	0	0	0	0	0	0
21	1548837599630	1	1	1	1	1	0	0	0	0	0	0	0
22	1548837600630	1	1	1	1	1	0	0	0	0	0	0	0
23	1548837601630	1	1	1	1	1	0	0	0	0	0	0	0
24	1548837602630	1	1	1	1	1	0	0	0	0	0	0	0
25	1548837603630	1	1	1	1	1	0	0	0	0	0	0	0
26	1548837604630	1	1	1	1	1	0	0	1	1	0	0	0
27	1548837605630	1	1	1	1	1	0	0	1	1	0	0	0
28	1548837606630	1	1	1	1	1	0	0	1	1	0	0	0
29	1548837607630	1	1	1	1	1	0	0	1	1	0	0	0
30	1548837608630	1	1	1	1	1	0	0	1	1	0	0	0
31	1548837609630	1	1	1	1	1	0	0	1	1	0	0	0
32	1548837610630	1	1	1	1	1	0	0	1	1	0	0	0
33	1548837611630	1	0	1	1	1	0	0	1	1	0	0	0
34	1548837612630	1	0	1	1	1	0	0	1	1	0	0	0
35	1548837613630	1	0	1	1	1	0	0	1	1	0	0	0
36	1548837614630	1	0	1	1	1	0	0	1	1	1	1	0
37	1548837615630	1	0	1	1	1	0	0	1	0	1	1	0
38	1548837616630	1	0	1	1	1	0	0	1	0	1	1	0

Figure 8. Sample of internal states evolution through time. Rows show the getting up from bed process reflected and internal states triggered.

MReasoner specification rules: states(BedRoomLight, BedRoomMovement, KitchenLight, KitchenMovement, CorridorLight, FrontDoorMovement, ToiletLight, ToiletMovement, BigPadIdle, PollutionAlert, getup);

```
is(KitchenMovement); is(#KitchenMovement);
is(FrontDoorMovement); is(#FrontDoorMovement);
is(ToiletMovement); is(#ToiletMovement);
is(BedRoomMovement); is(#BedRoomMovement);
is(BigPadIdle); is(#BigPadIdle);
holdsAt(#getup, 0);
holdsAt(#BedRoomLight, 0);
holdsAt(#KitchenLight, 0);
holdsAt(#CorridorLight, 0);
holdsAt(#ToiletLight, 0);
holdsAt(#PollutionAlert, 0);
ssr((BigPadIdle ^ BedRoomMovement)-> getup);
ssr((getup)->BedRoomLight);
ssr((getup)->PollutionAlert);
ssr((ToiletMovement) -> ToiletLight);
ssr((KitchenMovement) -> KitchenLight);
ssr((FrontDoorMovement) -> CorridorLight);
ssr(([-][60s.]#ToiletMovement ^
<->[61s.]ToiletMovement) -> #ToiletLight);
ssr(([-][60s.]#KitchenMovement ^
<->[61s.]KitchenMovement) -> #KitchenLight);
ssr(([-][60s.]#FrontDoorMovement ^
<->[61s.]FrontDoorMovement) -> #CorridorLight);
ssr(([-][60s.]#BedRoomMovement ^
<->[61s.]BedRoomMovement) -> #BedRoomLight);
```

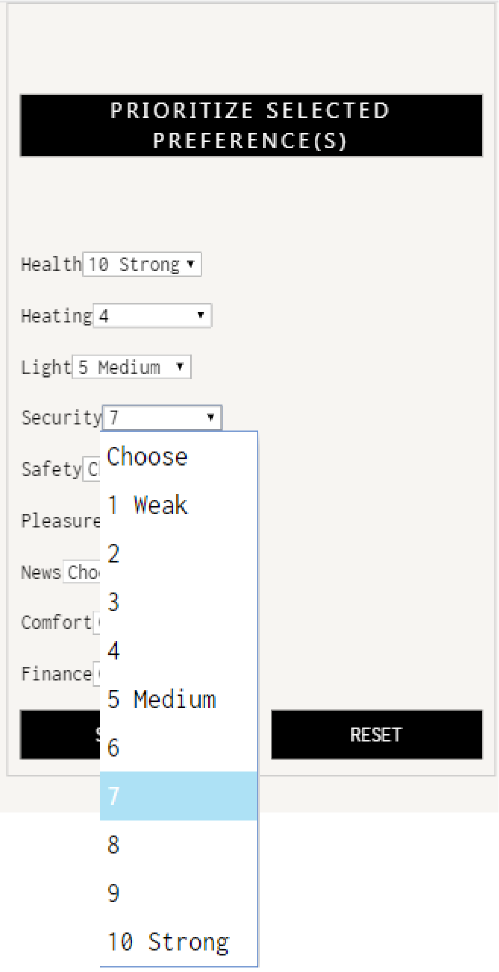
The States line lists states tracked by the system, all relevant ones have to be declared. The is(...) lines tell the system which states are independent. Dependent states are the outcome of rules, whilst independent states represent sensors triggered (e.g., FrontDoorMovement is related to the PIR sensor near the front door which is triggered when the person gets out of the bedroom and into the corridor) or human actions. The next section initializes dependent states, here assuming them false. Then ssr lines specify state updates and actuation rules. They first indicate what the system should do when the user gets up. Then other rules turn lights on as the user visits rooms. Finally a set of rules identify rooms which have been visited but not used for at least a minute and turn their lights off. This causes for example that in leaving home for work all lights remaining on will be soon turned off. Learning patterns: LFPUBS learns habits as rules including temporal contexts (e.g. days of the week and times during those days):

```
ssr((weekDayBetween(monday-friday)) -> day_context) ;
ssr((#weekDayBetween(monday-friday))->#day_context );
ssr((clockBetween(08:01:00-00:00:00)) -> time_context );
ssr((#clockBetween(08:17:00-00:00:00)) -> #time_context);
```

These contexts are used to constrain the triggering of rules to the contexts where that actuation is meaningful according to the experience gathered by the system, for example:

```
ssr(([-][05s.] #BedRoomDoor ^ actionMap_time_context ^ actionMap_day_context )
               -> Pattern_1);
```

User preferences representation: Although not evident in our simplified system specification above SEArch includes an interface (Figure 9) to associate levels of priority with certain preferences which can then be linked to rules and affect the actuation of the system as preferences of a person change or when comparing the preferences of several users (see Oguego et al. (2018).



PRIORITIZE SELECTED PREFERENCE(S)

Health 10 Strong

Heating 4

Light 5 Medium

Security 7

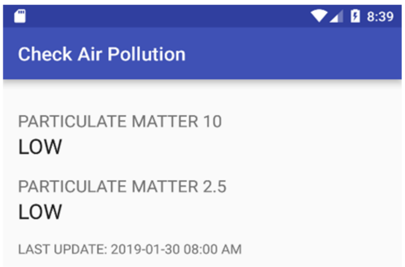
Figure 9. Preferences Selection Interface

In our scenario, high preferences for health associated with the user combined with system rules lead to the user being presented with relevant information (for example, air quality report when getting up). One rule in the system triggered a report on Air Quality the user can check when getting up as seen in Figure 10.

The resulting house actuation when the above scenario is exercised as well as other scenarios recorded in the Smart Home can be seen here: `https://mdx.figshare.com/account/home#/projects`

6. Methods and Tools

The SEArch system architecture template can act as a guide on which components can be considered useful for developing applications in a given area. however on its own they do not account



Check Air Pollution

PARTICULATE MATTER 10
LOW

PARTICULATE MATTER 2.5
LOW

LAST UPDATE: 2019-01-30 08:00 AM

Figure 10. Air Quality Report for User with Asthma

for the whole complex process of innovation. The reasoning, learning, personalization modules as well as interfaces and technological infrastructure are important however, higher level strategies are needed to guide teams on applying more effectively their tools.

Our team has been developing strategies and developer teams supporting tools which help them to focus on the highest impact concepts and aspects of systems in these area:

U-CIEDP One overarching methodology, which is user-centred and emphasize stakeholder's engagement, is the User-centred Intelligent Environments Development Process, reported in Augusto et al. (2017). This is an iterative method which is highly flexible and in the extremes ways of using it could resemble traditional ‘waterfall’ or ‘agile‘ approaches,

eFRIENDS This method helps identified highly ethical concerns and to materialize them into real product services Jones et al. (2015),

R4IE Requirements for Intelligent Environments guide developers into gathering requirements which are focused on contexts and guide the developers to transform these into context-awareness features in the real system Evans et al. (2014),

CoMo the Context Modeller helps modelling and translating automatically into C-SPARQL context-aware rules Kramer et al. (2015),

MIRIE the Method for Increased Reliability of Intelligent Environments is a guideline and examples on how to use pre-existing formal methods Augusto and Hornos (2013), and,

CoaTi the Context-aware Testing and Validation approach guides developers into identifying key aspects to test and also when failed validations detected it helps trace back the potential origin of the fault Augusto et al. (2019).

Figure 11 summarize the main methods and tools used in combination with SEArch.

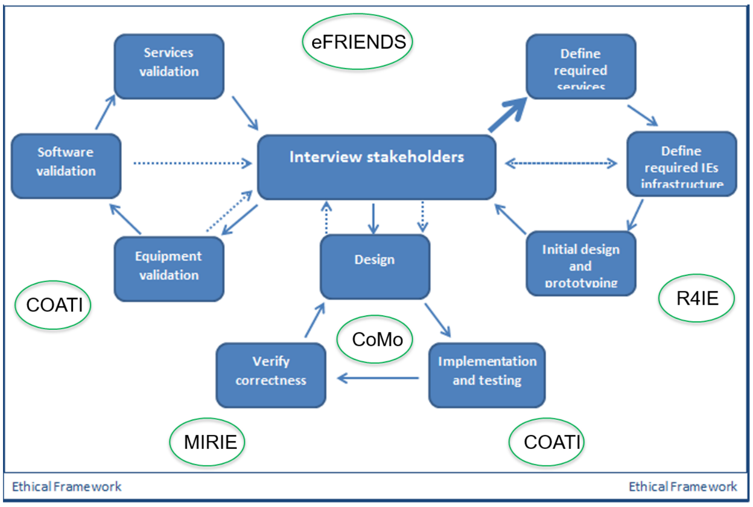


Figure 11. A collection of methods and tools to support Smart Environments development

7. Validation of the Architecture

Here we would like to provide examples of how the architecture use at a higher level and illustrate its role in creating systems which have been developed to be placed and used within society.

7.1. *Using SEArch to create a solution supporting people with dementia*

Activity Recognition plays an important role within Intelligent Environments (IE). Using environment information to understand users activities is crucial to establish an adequate behaviour understanding system adapted a specific context. A line of research within our Smart Spaces Lab explores this topic. It focuses specifically on how to use positively the understanding of the lifestyle of people with dementia (PwD) who is starting to experience difficulties in the so-called Activities of Daily Living (ADLs). Especially important are activities which have an impact in well-being and general health of the individual. Cognitive declines affects people's understanding of time and organization and the progressively loose a sensible alignment with the basic rhythms of daily life. They start shifting meal times, skipping dinner altogether but then getting up during the night to snack. Sleeping time starts to become affected too and they sleep less or at odd times, or they stay in bed without having a proper rest. They try to leave home at 3AM, or they 'wander around', that is walk to different place of the house aimlessly.

Many researches in dementia highlight the importance of a healthier alignment of PwD ADLs in order to avoid distress and healthier lifestyle, Kaufmann and Engel (2016). Also, Kaufmann and Engel (2016) remarks that the time of occurrence of ADLs or their duration, give professionals valuable information about possible user cognitive impairment. Thereby there is a practical interest from a health and well-being perspective to develop a tool that offers a continuous assessment and creates useful personalized interventions addressed to the user. In addition, this ADLs understanding from the system enables secondary benefits such as the possibility to alert caregivers on potentially dangerous situations (e.g., PwD leaving home at a dangerous time).

The development, testing and validation of this Activity Recognition system was performed in out Smart House. The interaction with the users was developed by using reminding methods which were first suggested in previous research addressed PwD concerns Orpwood et al. (2005).

This intelligent environment supports the deployment of a Pilot based on the SEArch architecture as described in previous sections. The main parts of SEArch exercised by this case study are highlighted in figure 12. The sensing environment collects data related to users physical actions

(1) through sensors (for example movement detected in a room, opening cupboard, turning on the kettle, detecting pressure on bed, etc.). The information gathered is analysed by an activity understanding specification which can be processed by the rule-based reasoner (4) based on rules defined by Context-Awareness (2) using the schedules configured in the interface (3). This diagram shows the possibility that a caregiver configures the system, but it could be done by the primary user too. The reasoning system assesses the context in real-time and determines whether the activity is unusual for the current user at this time (5), generating an information similar to an actuator but handled in a different way by the system (6), (say we call it virtual-actuator). These virtual-actuators indicates the system whether it should warn the user (7) or alert a caregiver (8) according with parameters provided in the interface. This project also uses the mobile to interact with users (primary users or caregivers), but the interaction with people with dementia can include different mechanism using actuators such as lights or music.

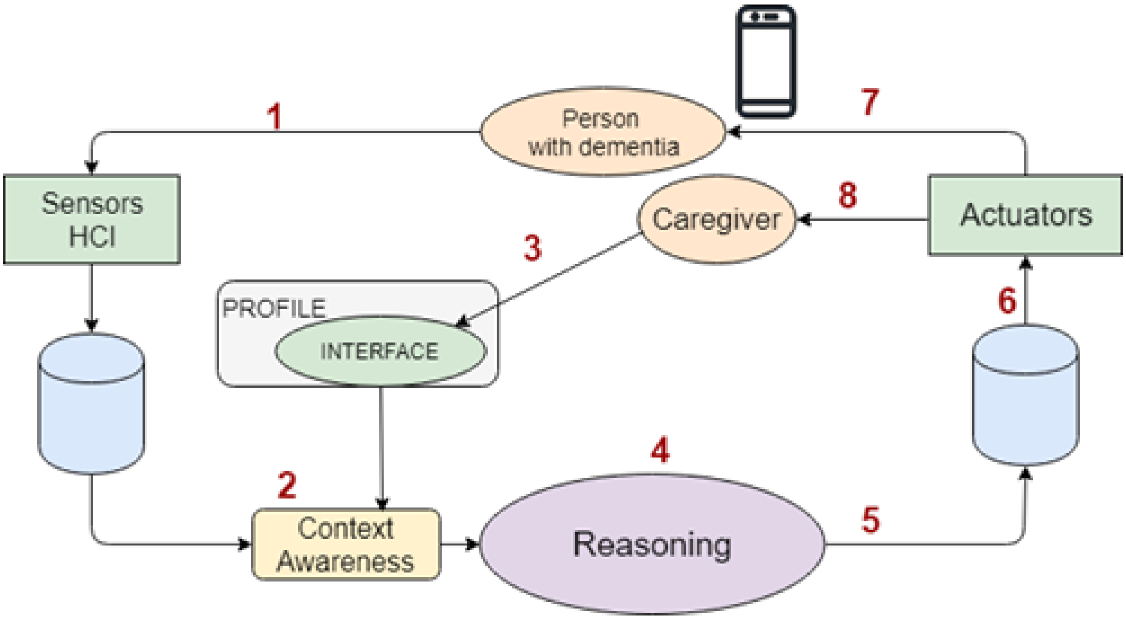


Figure 12. Using SEArch to Understand Human Activities

Recordings have been made and stored [1] in Fighsare to show the system working during the testing process. In addition, we invited University students and colleagues with Environmental Health and Public Health background as well as London Councils representatives to come to the lab and witness the system. We explained the system and invited them to fill a short anonymous survey where they can express their concerns related to the presented solution. In general, there was no questioning about the convenience of the system to enhance PwDs autonomy and support caregivers, see figure 13. However, as figure 14 shows, their main concerns are focused on users personal data misusing by unauthorized people. Other widespread considerations exposed the risk to the user and the professional misunderstandings in evaluated a situation due, both, to a wrong activity detection. This situation can happen when the system does not implements well an activity or when it is not detected a device working wrong in the sensing environment. The first issue depends on validation process in real environments but also it is important a good definition of contexts and their specialized testing Augusto et al. (2019). Current Intelligent environments cope with this inaccuracy from errors control, which is an important issue to deal. Although today the technology

[1] `https://mdx.figshare.com/account/home#/projects`

is becoming progressively affordable for most people, the cost-benefits ratio still is a concern for some respondents. After these validations with

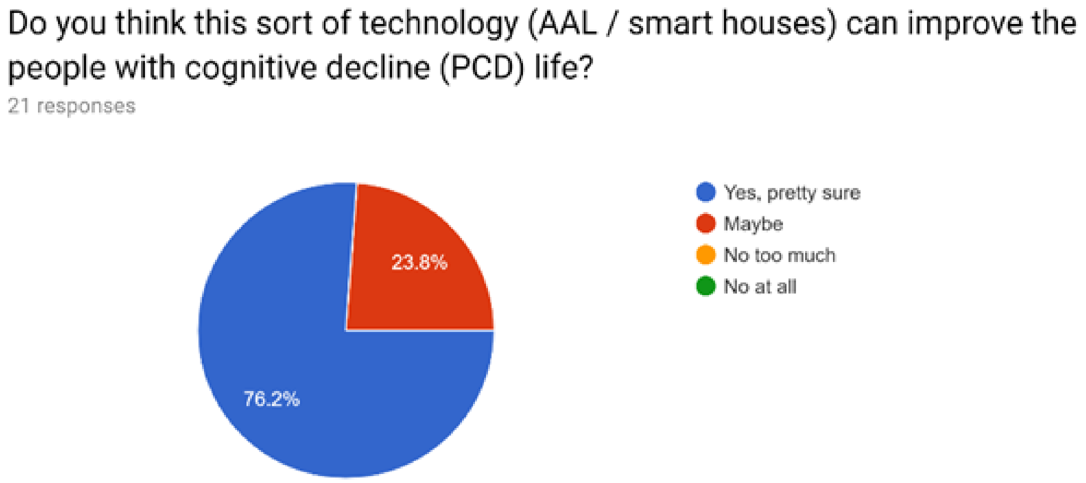


Figure 13. Using SEArch to Understand Human Activities

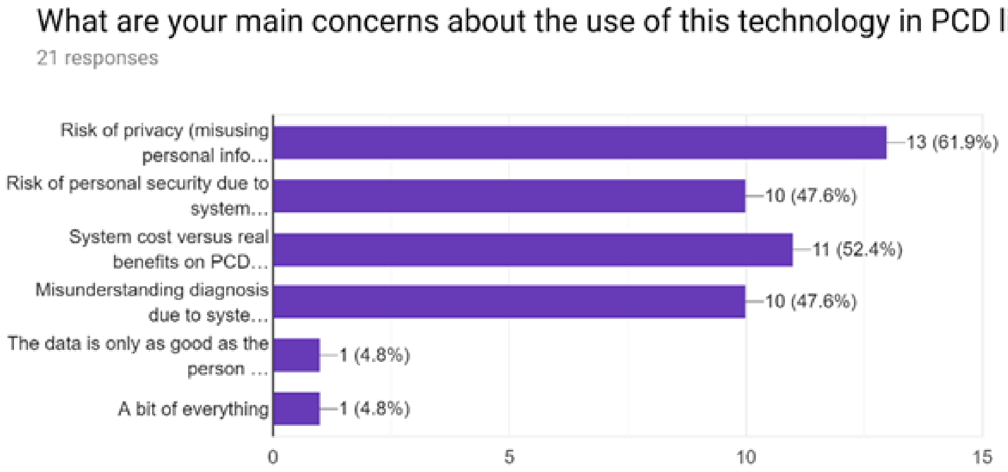


Figure 14. Using SEArch to Understand Human Activities

7.2. *Using SEArch to create a solution supporting personalized asthma management*

In this section, we describe a mobile health solution (mobile app) supporting personalized asthma management whose development was based on SEArch. The solution aims to address the high heterogeneity level of asthma by facilitating the personalization of the self-management approach that is required to deal with this chronic condition. We also explain how the elements of SEArch facilitate the delivery of the main features of the solution that were determined based on the U-CIEDP.

The features of the solution were defined through the interaction with people with asthma and their carers of people with asthma who expressed their main concerns about the issues they went through in their asthma management process. The methodology used to engage them were unstructured interviews and questionnaires that were applied in partnership with Asthma UK through their Centre for Applied Research [2]. This interaction was complemented with a literature survey on context-aware solutions for personalized asthma management that can be found in Quinde et al. (2018). The features were defined with the aim of addressing the following main issues:

- It would be convenient to choose what indicators to monitor and the places to monitor these indicators, considering the specific characteristics of someones asthma.
- The first year after someone is diagnosed with asthma is very difficult because they go through a trial-and-error process until discovering what their triggers are.
- There is a need of involving carers of people with asthma in their management process. This involvement should also be personalized.

SEArch was used to create the solution and, in this case, a higher emphasis was put in the development of its Context Awareness element that is directly linked to the personalization of the asthma management process. Figure 15 shows SEArch adapted for the development of the solution. In order to clarify this, we will show the roles of each SEArch element that were used to create the solution.

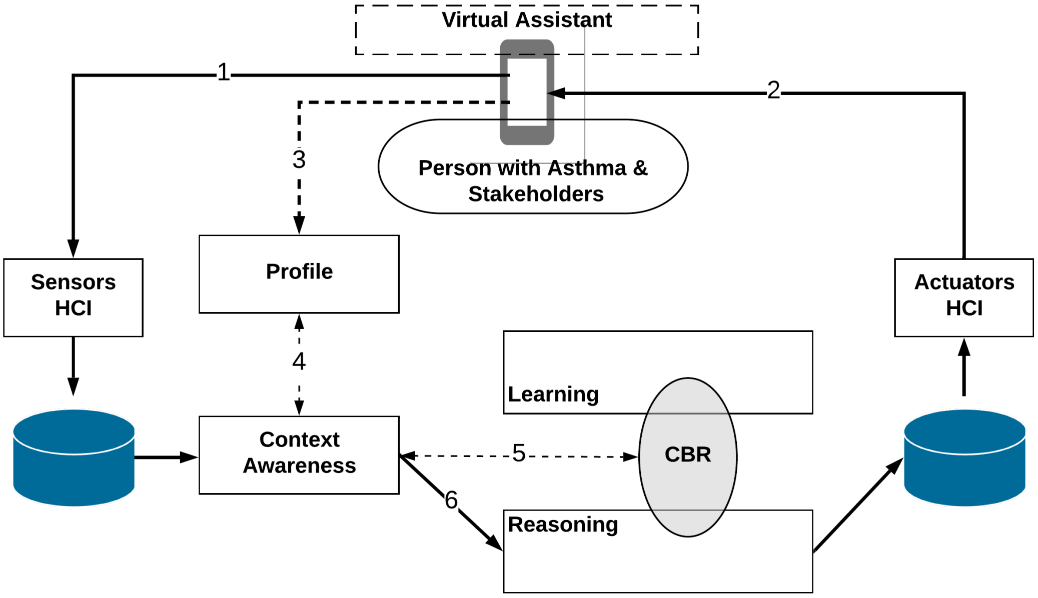


Figure 15. SEArch Being Excercised by the Asthma Management System

The Human element is represented by the people with asthma and their carers. The main interaction between the system and them is established through a smartphone or tablet (1, 2). The Profile element is in charge of storing the characteristics framing the asthma management process of the person with asthma. Users configure their profile through the mobile application (3). This profile

[2] www.aukcar.ac.uk/about/asthma-uk-support

is used by the Context Awareness element (4) in order to customize the system features that are then delivered to the users. The Sensors element includes devices monitoring temperature and humidity in the Smart Home as well as virtual sensors in the form of APIs that allow the collection of outdoor environmental data at the places of interest of the person with asthma. The APIs that are used to collect these data are London Air API [3] from the London Air Quality Network for pollution information, and the World Weather API (APIXU [4]) for weather information. The Actuators element are mostly related to delivering reports, notifications and alerts (2) to the people involved in the asthma management process that is being supported by the system. The Learning and Reasoning elements are implemented as a CBR (5) that supports people with asthma when they not know or poorly know their triggers. The CBR creates cases describing the conditions a person with asthma has been exposed to, and the asthma health status s/he experienced for when they were exposed to those conditions. Thus, the CBR attempts the prediction of his/her asthma health status based on the previous experiences that are stored in the form of cases Quinde et al. (2019). The database of cases of the CBR represents the Learning element as previous experiences, and the algorithm implementing the CBR cycle that is used to attempt the prediction is the Reasoning element. The other relevant reasoning process performed by the system is the one considering the profile configured by the user. This profile is queried by the Context-Awareness element (4) and delivered to the Reasoning element (6) that analyzes the indicators and the places that are of interest for the person with asthma **?**. The positive results of the validation process that included real users were possible to obtain because of the flexibility that SEArch provides when it comes to create user-centric solutions. The user-centric flexibility achieved in this case study is one of the most important outcomes to highlight. More information about the methodology and validation of this project can be found in Quinde et al. (2019),**?**, and Quinde et al. (2019). Moreover, the capacity of SEArch to facilitate the communication with external sources (APIs) is another relevant outcome to highlight. Finally, it is also important to state that the research team is currently working on implementing a Virtual Assistant that is able to support the personalization of the asthma management process. Figure 16 shows the main screen of the Asthma Management app.

The Virtual Assistant G.U.I. of Figure 17 shows the screen with the instructions that were delivered to the users during the conflict resolution project validation of the system.

8. Conclusions

The branch of computing we have considered in this article has been increasingly explored for many years, decades already. Starting with Ubiquitous and Pervasive Computing and reincarnating under different labels with different emphasis on one aspect of infrastructure or another. Most reports in the area have been examples of implemented systems to address specific solutions, these are of course beneficial, however the methodological discussion in these communities have been rather weak. As a result there are very few methods and/or tools the community considers as anything near standards of good practice. We believe the time has come for this to be brought more prominently to the community strategy.

This article provides an overview of the infrastructure, architecture and operational description of the system we have developed through various iterations. We also mentioned associated methods and tools we have to develop in order to complement what was already existing in traditional

[3] www.londonair.org.uk/LondonAir/API/

[4] www.apixu.com/api.aspx

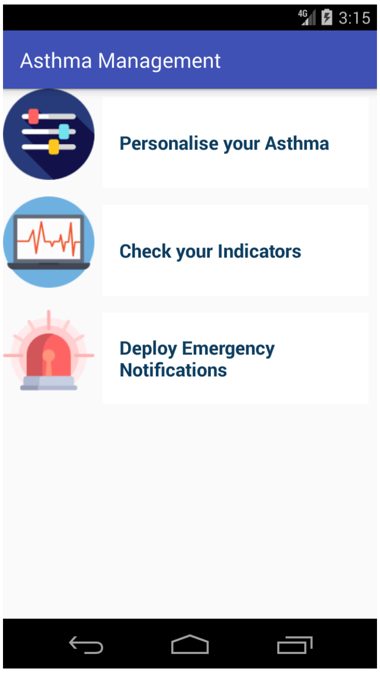


Figure 16. Main Asthma Management menu

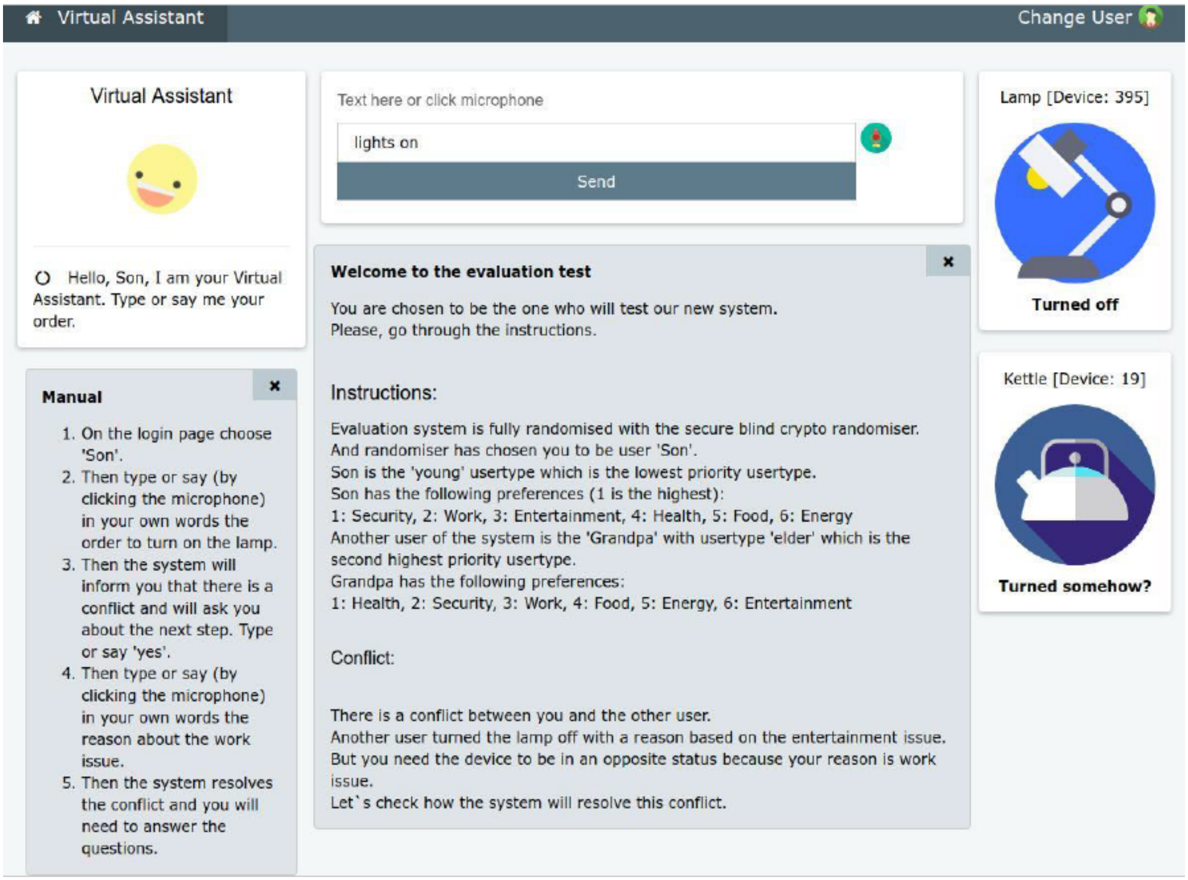
technical literature. We do not suggest all developing teams have to use it, rather it could be considered as an option, as a point of reflection and inspiration for other teams to create their own, and hopefully, to start a long lasting open discussion within our technical communities on how systems of this nature should be created.

One emphasis of our approach is the importance we give to stakeholders through the whole system life cycle. This is another interesting aspect which our community needs to prioritize. There are many systems built but not many popular systems massively consumed. Co-creation has to be embraced more explicitly and more naturally by our community. After all the systems we create are for humans and because each human or group of humans is different, there is no simple system who can satisfy everyone so an understanding of their expectations is only natural.

Another sense in which we think our article differs and complements what our community has produced in terms of architectures before is that we provide a more holistic approach. The architecture considers the input/output of the system to interact with the real world, and internally includes real-time context analysis as well as higher reasoning and learning complemented with user preferences. The architecture is also supported with methods and tools the different stages of scoping, development and validation.

The development of these methods and tools complement industry and business innovation as those tend to be proprietary and also force the use of specific products those companies try to do business with. In our case we adopted mainly one hardware platform for consistency and because it represents the state of the art in the field in terms of cost, security, installation complexity, etc. This infrastructure layer can be replaced with minimal adjustments in the storage of data in databases which is the real source of the algorithms used in our system.

We hope to motivate other teams to publish their tried and tested processes to create intelligent environments, to encourage comparisons, learning and sharing and help increasing the quality of innovation in this industry.



○ Hello, Son, I am your Virtual Assistant. Type or say me your order.

Manual ×

1. On the login page choose 'Son'.
2. Then type or say (by clicking the microphone) in your own words the order to turn on the lamp.
3. Then the system will inform you that there is a conflict and will ask you about the next step. Type or say 'yes'.
4. Then type or say (by clicking the microphone) in your own words the reason about the work issue.
5. Then the system resolves the conflict and you will need to answer the questions.

Welcome to the evaluation test ×

You are chosen to be the one who will test our new system.
Please, go through the instructions.

Instructions:

Evaluation system is fully randomised with the secure blind crypto randomiser.
And randomiser has chosen you to be user 'Son'.
Son is the 'young' usertype which is the lowest priority usertype.
Son has the following preferences (1 is the highest):
1: Security, 2: Work, 3: Entertainment, 4: Health, 5: Food, 6: Energy
Another user of the system is the 'Grandpa' with usertype 'elder' which is the second highest priority usertype.
Grandpa has the following preferences:
1: Health, 2: Security, 3: Work, 4: Food, 5: Energy, 6: Entertainment

Conflict:

There is a conflict between you and the other user.
Another user turned the lamp off with a reason based on the entertainment issue.
But you need the device to be in an opposite status because your reason is work issue.
Let`s check how the system will resolve this conflict.

Figure 17. Virtual Assistant G.U.I.

Acknowledgements

The system has been developed through many iterations and with contributions from many researchers. Those currently involved are co-authors of this paper. Past contributors are listed here: `http://ie.cs.mdx.ac.uk/some-of-our-contributions/`

References

Alegre, U., J. C. Augusto, and A. Aztiria (2014). Temporal reasoning for intuitive specification of context-awareness. In *Proc. International Conference on Intelligent Environments*, pp. 234241. IEEE.

Ali, S. M. M., J. C. Augusto, and D. Windridge (2019). A survey of user-centred approaches for smart home transfer learning and new user home automation adaptation. *Applied Artificial Intelligence*.

Aranbarri-Zinkunegi, M. E. (2017). Improving the pattern learning system integrating the reasoning system.

Asad, U. and S. Taj (2018). Interoperability in iot based smart home: A review. *Review of Computer Engineering Studies 5*, 50–55.

Asadullah, M. and A. Raza (2016). An overview of home automation systems. In *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, pp. 27–31.

Augusto, J., D. Kramer, U. Alegre, A. Covaci, and A. Santokhee (2017). The user-centred intelligent environments development process as a guide to co-create smart technology for people with special needs. *Universal Access in the Information Society 17*, 115–130.

Augusto, J. C. and H. Aghajan (2019). Preface to 10th anniversary issue. *JAISE 11*(1), 1–3.

Augusto, J. C., V. Callaghan, D. Cook, A. Kameas, and I. Satoh (2013). Intelligent Environments: a manifesto. *Human-centric Computing and Information Sciences 3*, 12.

Augusto, J. C. and M. J. Hornos (2013). Software simulation and verification to increase the reliability of Intelligent Environments. *Advances in Engineering Software 58*, 18–34.

Augusto, J. C., M. J. Quinde, and C. L. Oguego (2019). Context-aware systems testing and validation. In *Proceedings of 10th International Conference Dependable Systems, Services and Technologies, Leeds, United Kingdom, June 5-7*.

Aztiria, A., J. C. Augusto, R. Basagoiti, A. Izaguirre, and D. Cook (2013). Learning Frequent Behaviours of the User in Intelligent Environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems 43*(6), 1265–1278.

Badica, C., M. Brezovan, and A. Badica (2013). An overview of smart home environments: Architectures, technologies and applications. In *Local Proceedings of the Sixth Balkan Conference in Informatics, Thessaloniki, Greece, September 19-21, 2013*, pp. 78.

Bakar, U., H. Ghayvat, S. Hasanm, and S. Mukhopadhyay (2016). Activity and anomaly detection in smart home: A survey. In S. Mukhopadhyay (Ed.), *Next Generation Sensors and Systems. Smart Sensors, Measurement and Instrumentation*, Volume 16. Springer.

Belaidouni, S., M. Miraoui, and C. Tadj (2016). Towards an efficient smart space architecture. *CoRR abs/1602.05109*.

Benmansour, A., A. Bouchachia, and M. Feham (2015, December). Multioccupant activity recognition in pervasive smart home environments. *ACM Comput. Surv. 48*(3).

Bobadilla, J., F. Ortega, A. Hernando, and J. Bernal (2012). A collaborative filtering approach to mitigate the new user cold start problem. *Knowl.-Based Syst. 26*, 225–238.

Cook, D., M. Huber, K. Gopalratnam, and G. Youngblood (2003). Learning to control a smart home environment. In *Proc. IAAI*.

Cook, D. J., A. S. Crandall, B. L. Thomas, and N. C. Krishnan (2013). CASAS: A smart home in a box. *IEEE Computer 46*(7), 62–69.

Covaci, A., D. Kramer, J. C. Augusto, S. Rus, and A. Braun (2015). Assessing real world imagery in virtual environments for people with cognitive disabilities. In *2015 International Conference on Intelligent Environments, IE 2015, Prague, Czech Republic, July 15-17, 2015*, pp. 41–48.

Crandall, A. S. and D. J. Cook (2011). Tracking systems for multiple smart home residents. In *Behaviour Monitoring and Interpretation - BMI - Well-Being*, pp. 65–82.

Dey, A. K. and G. D. Abowd (1999). Towards a Better Understanding of Context and Context-Awareness. *Computing Systems 40*(3), 304–307.

Evans, C., L. Brodie, and J. C. Augusto (2014). Requirements engineering for intelligent environments. In *Proceedings IE14*, pp. 154–161. IEEE.

Fernández-Montes, A., J. A. Ortega, J. A. Álvarez, and L. G. Abril (2009). Smart environment software reference architecture. In *International Conference on Networked Computing and Advanced Information Management, Seoul, Korea, August 25-27, 2009*, pp. 397–403.

Ferro, E., M. Girolami, D. Salvi, C. C. Mayer, J. Gorman, A. Grguric, R. Ram, R. Sadat, K. M. Giannoutakis, and C. Stocklöw (2015). The universaal platform for AAL (ambient assisted living). *J. Intelligent Systems 24*(3), 301–319.

Fiware (2019). `http://fiware.org`.

Furfari, F., M. Tazari, and V. Eisemberg (2011). universaal: an open platform and reference specification for building AAL systems. *ERCIM News 2011*(87).

Heider, T. Goal-based interaction with smart environments.

Jones, S., S. Hara, and J. C. Augusto (2015). efriend: an ethical framework for intelligent environments development. *Ethics Inf. Technol. 17*(1), 11–25.

Kaufmann, G. and A. Engel (2016). Dementia and well-being: A conceptual framework based on tom kitwood s model of needs. *Dementia 15*, 774788.

Kramer, D., A. Covaci, and J. C. Augusto (2015). Developing navigational services for people with down's syndrome. In *2015 International Conference on Intelligent Environments, IE 2015, Prague, Czech Republic, July 15-17, 2015*, pp. 128–131.

Lewis, G., E. Morris, D. Smith, and S. Simanta (2008). Smart: Analyzing the reuse potential of legacy components in a service-oriented architecture environment.

Li, R., B. Lu, and K. D. McDonald-Maier (2015). Cognitive assisted living ambient system: a survey. *Digital Communications and Networks 1*(4), 229–252.

Martin, S., J. C. Augusto, P. McCullagh, W. Carswell, H. Zheng, H. Wang, J. Wallace, and M. Mulvenna

(2013). Participatory research to design a novel telehealth system to support the night-time needs of people with dementia: Nocturnal. *International journal of environmental research and public health 10*(12), 6764–6782.

Mitchell, T. M. (1997). *Machine Learning*. The McGraw-Hill and MIT Press.

Mok, E. and G. Retscher (2007). Location determination using wifi fingerprinting versus wifi trilateration. *Journal of Location Based Services 1*(2), 145–159.

Murabet, A. E., A. Abtoy, A. Touhafi, and A. Tahiri (2018). Ambient assisted living systems models and architectures: A survey of the state of the art. *Journal of King Saud University - Computer and Information Sciences*.

Nurgaliyev, K., D. D. Mauro, N. Khan, and J. C. Augusto (2017). Improved multi-user interaction in a smart environment through a preference-based conflict resolution virtual assistant. In *2017 International Conference on Intelligent Environments, IE 2017, Seoul, Korea (South), August 21-25, 2017*, pp. 100–107.

Oguego, C. L., J. C. Augusto, A. Muñoz, and M. Springett (2018). Using argumentation to manage users' preferences. *Future Generation Comp. Syst. 81*, 235–243.

Oguego, C. L., J. C. Augusto, M. Springett, M. Quinde, and C. J. Reynolds. (2019). Interface for managing users' preferences in ami systems. In *Proceedings of 15th International Conference on Intelligent Environments*.

Orpwood, R., C. Gibbs, T. Adlam, R. Faulkner, and D. Meegahawatte (2005). The design of smart homes for people with dementiauser-interface aspects. *Universal Access in the information society 4*, 156–164.

Ospan, B., N. Khan, J. Augusto, M. Quinde, and K. Nurgaliyev (2018). Context aware virtual assistant with case-based conflict resolution in multi-user smart home environment. In *International Conference on Computing and Network Communications (CoCoNet), Astana, doi: 10.1109/CoCoNet.2018.8476898*, pp. 36–44.

Palumbo, F., P. Barsocchi, S. Chessa, and J. C. Augusto (2015). A stigmergic approach to indoor localization using bluetooth low energy beacons. In *12th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2015, Karlsruhe, Germany, August 25-28, 2015*, pp. 1–6.

Piyare, R. (2013). Internet of things: Ubiquitous home control and monitoring system using android based smart phone. *J. IoT2 1*, 5–11.

Quinde, M., N. Khan, J. Augusto, and A. van Wyk (2019). A human-in-the-loop context-aware system allowing the application of case-based reasoning for asthma management. In D. V. (Ed.), *Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. Healthcare Applications. HCII 2019.*, Volume 11582 of *Lecture Notes in Computer Science*. Springer.

Quinde, M., N. Khan, J. Augusto, A. van Wyk, and J. Stewart (2018). Context-aware solutions for asthma condition management: a survey. *Universal Access in the Information Society*.

Quinde, M., N. Khan, and J. C. Augusto (2019). Case-based reasoning for context-aware solutions supporting personalised asthma management. In *Proceedings of 18th International Conference on Artificial Intelligence and Soft Computing*.

Roy, N., A. Misra, and D. J. Cook (2013). Infrastructure-assisted smartphone-based ADL recognition in multi-inhabitant smart environments. In *2013 IEEE International Conference on Pervasive Computing and Communications, PerCom 2013, San Diego, CA, USA, March 18-22, 2013*, pp. 38–46.

Russell, S. J. and P. Norvig (2003). *Artificial Intelligence: A Modern Approach (Second Edition)*.

Saif, U. (2002). Architectures for ubiquitous systems.

Singh, M., N. Malim, P. Jin, L. Bann, and J. Kit (2014). Rfid-enabled elderly movement tracking system in smart homes. In *Proceedings of Knowledge Management International Conference 2014 (KMICe), 12-15 August 2014, Langkawi,Malaysi*, pp. 72–77.

Sora, D. and J. C. Augusto (2018). Managing multi-user smart environments through BLE based system. In *Intelligent Environments 2018 - Workshop Proceedings of the 14th International Conference on Intelligent Environments, Rome, Italy, 25-28 June 2018*, pp. 234–243.

Weiser, M. (1991). The Computer for the 21st Century. In *Scientific American*, Volume 265, pp. 94–104.

Zhi, M. and M. Singh. Rfid-enabled smart attendance management system. In J. Park (Ed.), *Future Information Technology-II*, pp. 213–231.