The SEArch Smart Environments Architecture

Juan Carlos Augusto[1]
Department Computer Science
Middlesex University
London, UK
J.Augusto@mdx.ac.uk

José Ginés Giménez Manuel[1]
Department Computer Science
Middlesex University
London, UK
J.Gimenezmanuel@mdx.ac.uk

Chimezie Leonard Oguego[1]
Department Computer Science
Middlesex University
London, UK
CO527@live.mdx.ac.uk

Mario Quinde[1,2]
Department Computer Science
Middlesex University
London, UK
MQ093@live.mdx.ac.uk

S M Murad Ali[1]
Department Computer Science
Middlesex University
London, UK
SA2305@live.mdx.ac.uk

Carl James-Reynolds[1]
Department Computer Science
Middlesex University
London, UK
C.James-Reynolds@mdx.ac.uk

***Abstract*—we report on a Smart Environment Architecture (SEArch) which has been developed to support innovative Ambient Assisted Living services. We explain SEArch at a conceptual level and also how it has been linked to a sensing environment. We compare SEArch to other similar systems reported in the technical literature. We illustrate how the system works using a practical automation scenario.**

Keywords—Smart environments, Ambient Assisted Living, sensing, actuation.

I. INTRODUCTION

Sensing technology has become one important enabler and stimulating source of innovation in ICT, Computer Science and technology with societal impact [1,2]. 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 are 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 practice 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 of Middlesex University. We illustrate how its components work cooperatively to provide services in environments such as Smart Homes.

II. 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 [3] provided a detailed Ubiquitous System architecture (UbiqtOS/Romvets) through various levels of description. 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. Cook et al. [4] MavHome system architecture 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. [5] 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 [6] presented a system to assist users of a multi-purpose smart meeting room. The system was approached as

[1] All authors are members of the Research Group on Development of Intelligent Environments.

[2] Also member of: Dept. de Ingeniería Industrial y de Sistemas, Universidad de Piura, Perú. Mario.Quinde@udep.pe

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. Cook et al. [7] CASAS system architecture 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.

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, and U= User-centred. 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 number or a higher emphasis in that aspect have a tick.

TABLE 1. SYSTEM ARCHITECTURE COMPARISON

System	M	O	R	L	H	U
Saif [3]	√		√			
Cook et al. [4]	√		√	√		
Fernández-Montes et al. [5]	√	√	√	√		
Heider [6]	√	√	√		√	√
Cook et al. [7]	√		√	√		
SEARch			√	√	√	√

There are other publications of ideal abstract architectures (see for example, Badica et al. [8] and Belaidouni et al. [9]), 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 [10] and Lewis et al. [11] where 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 [12,13,14].

III. 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. At the same time the practical challenges we address surpass what is considered in the market, hence the challenges lead to science with societal impact. 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 reasoning, learning, system-user interaction and user personalization.

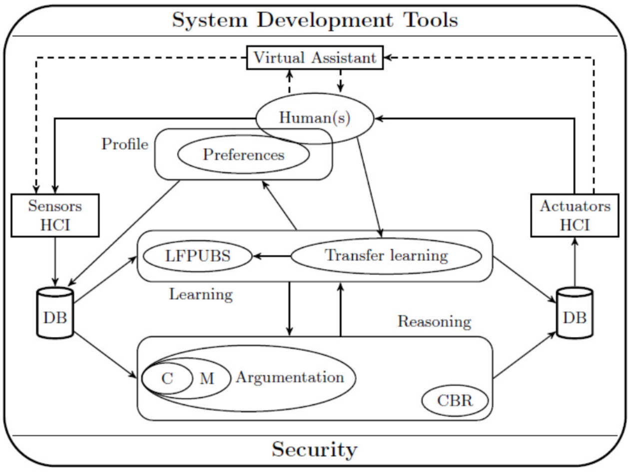

FIG. 1. SEARCH SYSTEM ARCHITECTURE DIAGRAM

IV. SEARCH AT WORK

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 infrastructure. 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. The wireless sensor system is mostly a network of Zwave-compatible commercial sensors (except for the pressure pad assembled in the lab). The Sensor Control box is a Vera Plus box (Figure 3). All sensing is non-intrusive and does not require a sophisticated interaction from the users.

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 [15]) and the learning is made through a combination of LFPUBS (Aztiria et al [16]) and an LFPUBS2MReasoner translation (Aranbarri-Zinkunegi [17]). 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 [18]).

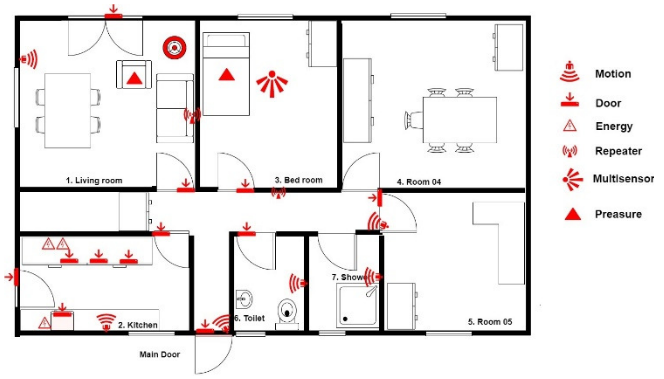

FIG. 2. SMART SPACES LAB LAYOUT

FIG. 3. ZWAVE SENSING EQUIPMENT AND VERA BOX (TOP RIGHT).

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.

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.

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 cause 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);

system_time_mill	BedRoomLigh	BedRoomMov	BigPadIdl	PollutionAlert	getup	KitchenMovement	KitchenLight	CorridorLight	FrontDoorMoveme	ToiletLight	ToiletMoveme	EntranceDoor
1548837586630	0	0	0	0	0	0	0	0	0	0	0	0
1548837587630	0	0	0	0	0	0	0	0	0	0	0	0
1548837588630	0	1	0	0	0	0	0	0	0	0	0	0
1548837589630	1	1	1	1	1	0	0	0	0	0	0	0
1548837590630	1	0	1	1	1	0	0	0	0	0	0	0
1548837591630	1	0	1	1	1	0	0	0	0	0	0	0

Fig. 4. Sample of internal states evolution through time. Rows show the getting up from bed process reflected and internal states triggered.

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 5) 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. [18]).

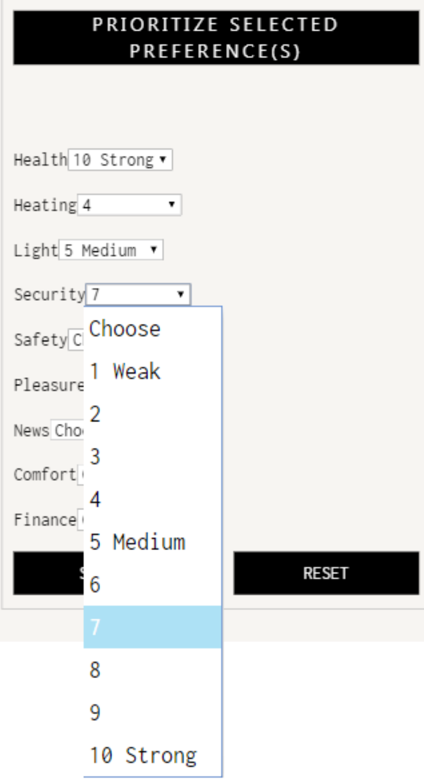

FIG. 5. PREFERENCES SELECTION INTERFACE

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 6.

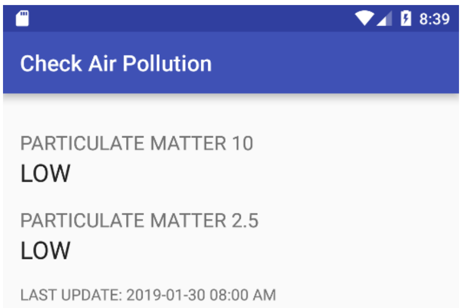

FIG. 6. AIR QUALITY REPORT FOR USER WITH ASTHMA

The resulting house actuation when the above scenario is exercised can be seen here:
https://mdx.figshare.com/s/23a75c103c20aa99f67f

V. CONCLUSIONS

We report on a home automation system which is being developed with emphasis on Ambient Assisted Living services. Our SEArch Smart Environment system architecture emphasizes Learning and Reasoning automation supplemented and guided by User Personalization components as our Research Group prioritizes a user-centred approach. This paper contrasted SEArch with other relevant developments. Our architecture puts more emphasis on higher level components driven by priority services needed. Our current emphasis is on optimizing the system with a view to improve practical uptake and innovation outside academia.

ACKNOWLEDGMENTS

The Smart Environment experimentation area has been strategically supported by two visionary managers: Prof. B. Barn and Prof. M. Loomes. The system has been developed 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

[1] J.C. Augusto and H. Ahajan (2019) Preface to 10th Anniversary Issue, J. of Ambient Intelligence and Smart Environments 11(1). IOS Press.

[2] J. C. Augusto, V. Callaghan, A. Kameas, D. Cook, I. Satoh (2013) Intelligent Environments: a manifesto. Human-centric Computing and Information Sciences, 3:12. Springer. DOI: 10.1186/2192-1962-3-12

[3] U. Saif (2002) Architectures for ubiquitous systems, Technical Report 527, Cambridge Computer Laboratory, UCAM-CL-TR-527.

[4] D. Cook, M. Huber, K. Gopalratnam, G. Youngblood (2003) Learning to control a smart home environment. Proc. IAAI.

[5] A. Fernández-Montes, J. A. Ortega, J. A. Álvarez, L. González (2009) Smart Environment Software Reference Architecture. NCM: 397-403

[6] Th. Heider (2009) Goal-based Interaction with Smart Environments, PhD Thesis, Universität Rostock.

[7] D. Cook, A. Crandall, B. Thomas, and N. Krishnan. CASAS: A smart home in a box. IEEE Computer, 46(6):26-33, 2013.

[8] C. Badica, M. Brezovan, A. Badica (2013) An Overview of Smart Home Environments: Architectures, Technologies and Applications. Proceedings of BCI: 78-85. Thessaloniki, Greece.

[9] S. Belaidouni, M. Miraoui, C. Tadj (2016) Towards an Efficient Smart Space Architecture. CoRR abs/1602.05109

[10] R. Piyare (2013) Internet of Things: Ubiquitous Home Control and Monitoring System using Android based Smart Phone, J. IoT2(1):5-11

[11] G. Lewis, E. Morris, D. Smith,S. Simanta (2008). SMART: Analyzing the Reuse Potential of Legacy Components in a Service-Oriented Architecture Environment. Technical Note CMU/SEI-2008-TN-008.

[12] S. Jones, S. Hara, J. C. Augusto. eFRIEND: an Ethical Framework for Intelligent Environment Development (2015) Ethics and Information Technology (Springer Verlag), Vol 17, issue 12, pp. 11-25, Springer.

[13] C. Evans, L. Brodie, J. C. Augusto (2014) Requirements Engineering for Intelligent Environments. Proc. IE14. pp. 154-161. IEEE Press.

[14] J. Augusto, 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(1), 115-130. Springer.

[15] U. Alegre, J. C. Augusto, A. Aztiria (2014) Temporal Reasoning for Intuitive Specification of Context-Awareness, Proc. International Conference on Intelligent Environments, pp. 234–241. IEEE.

[16] A. Aztiria, J. C. Augusto, R. Basagoiti, A. Izaguirre, and D. J Cook (2013) Learning frequent behaviors of the users in intelligent environments. IEEE Transactions on SMC: Systems 43(6):1265–1278.

[17] M. E. Aranbarri-Zinkunegi (2017) Improving the Pattern Learning System integrating the reasoning system. MS. thesis, Mondragon Univ.

[18] C. L. Oguego, J. C. Augusto, A. Muñoz, M. Springett (2018) Using Argumentation to Manage Users' Preferences. Future Generation Computer Systems 81:235-243. Elsevier.