Butler-ising HomeManager**: A Pervasive Multi-Agent System for Home Intelligence**

Enrico Denti and Roberta Calegari
Dipartimento di Informatica - Scienza e Ingegneria (DISI), Alma Mater Studiorum—Università di Bologna, Bologna, Italy

Keywords: Home Management, Ambient Intelligence, Pervasive Computing, Energy Saving, Smart Homes, Smart Living, Coordination Infrastructures, Multi-Agent Systems, Gamification.

Abstract: Home Manager is an agent-based application for the control of an intelligent home, where the house is seen as an intelligent environment made of independent devices that participate to an agent society. The society is governed by a coordination infrastructure aimed at satisfying the users goals and preferences (lighting, temperature, etc.) while achieving the global house policies and objectives (e.g. energy saving) in a highly-configurable way. In the existing prototype, designed mostly to prove the feasibility and effectiveness of the above approach, the testbed house was kept intentionally simple, with a limited number of rooms, user types, control devices and policies, and the infrastructure implementation lacked some features.

The recent, widespread adoption of smart mobile devices (smartphones, tablets) enabling mobile connectivity has dramatically changed the reference scenario: users now expect at least to be able to monitor, and possibly control, their home devices in mobility, and in fact all major vendors now offer some app for this purpose. Yet, this is just the basic step: exploiting the *situated* connectivity enabled by GPS and the other geo-localisation techniques embedded in today's smartphones, novel pervasive scenarios can be devised that could not even be imagined in the past years. This aspect is developed in the Butlers architecture, which provides a general framework and reference model for intelligent home management where the smart home is managed by an intelligent butler and interacts with its inhabitants taking into account their habits, behavior, location, preferences and any other sort of information to anticipate their needs and support their goals.

In this context, this paper presents the novel "Butler-ised" Home Manager, that evolves the previous system in the Butlers perspective: the new prototype not only supports the remote control of the house appliances via an Android app, but exploits the user position, tracked via geo-localisation, to anticipate the user's needs in a simple, yet significant, scenario – namely, autonomously switching the house oven on when discovering that the user has just bought a take-away pizza in his/her way back home.

1 INTRODUCTION

Technology evolution is making the appliances that populate our homes smarter and connected: it is common to find TV sets, air conditioners, washing machines, refrigerators, etc. networked and often remote-controlled, typically via an app for Android or iOS smartphones so that integration frameworks are being proposed (Google, 2014; Apple, 2014).

These trends are enlarging the application perspectives of "traditional" home automation in the consumer market, merging aspects from Ambient Intelligence (AmI) (Ducatel et al., 2001; Ayala et al., 2013), smart environments (Bartolini et al., 2012), energy monitoring (Innova, 2012; World, 2013) and domotics (Ché et al., 2010; Menon et al., 2013) to focus onto people's needs from different viewpoints (Chong and Mastrogiovanni, 2011).

In Ambient Intelligence, the emphasis is on user-friendliness, user-empowerment, and support for human interaction: the home environment is "smart" in that it can handle some aspects (e.g. lighting, heating) based on the user's preferences, so as to improve the quality of life. This is why it is often found in "assisted living" applications, whose aim is to support specific user categories (elderly, disabled, etc).

Recent works, like (Denti, 2014), emphasise that there is an extra value in considering all these aspects together: novel, intriguing scenarios are devised that could not even be imagined in the past years. In such new scenarios, the smart home can interact with its inhabitants not only to monitor and remote-control the home appliances, but also to take into account the users' habits, behavior, location, preferences and any other sort of information to *anticipate their needs* and overall support their goals. The basic idea is that knowing the user's habits and observing his/her daily behavior via mobile devices – in particular his/her

Denti E. and Calegari R..
Butler-ising HomeManager - A Pervasive Multi-Agent System for Home Intelligence.
DOI: 10.5220/0005284002490256
In *Proceedings of the International Conference on Agents and Artificial Intelligence* (ICAART-2015), pages 249-256
ISBN: 978-989-758-073-4

location, thanks to the *situated* connectivity enabled by GPS and the other geo-localisation techniques – can enable an intelligent system (the home *butler*) to take autonomous decisions and possibly anticipate the users' needs, managing the home devices on the user's behalf. In (Denti, 2014), these aspects are put in context with other research results from intelligent agents, multi-agent systems (mainly in the area of power generation and consumption (Tolbert et al., 2001), power restoration (Nagata et al., 2000), load management in power grid systems (Zhang et al., 2011)), and coordination technologies (Papadopoulos and Arbab, 1998; Omicini and Papadopoulos, 2001; Busi et al., 2001), on the one side, and "entertainment" aspects, in the *gamification* (Gamification Community, 2013) perspective, on the other – the latter being more and more recognised as a key factor between success and failure in technology acceptance from the consumers' viewpoint.

Home Manager (Molesini et al., 2009) is a prototype agent-based application for the control of an intelligent home, where the house is seen as an intelligent environment made of independent devices that participate to an agent society: the system aims to manage the overall energy consumption while supporting the user's living and activities inside the house. The agent society is governed by the TuCSoN coordination infrastructure (Omicini and Zambonelli, 1999; Ricci et al., 2002; Omicini and Rimassa, 2004; Mariani and Omicini, 2014; TuCSoN, 2008) and tries to satisfy the user's goals and preferences as concerns the lighting and room temperature, while achieving the global house policies and objectives. Since the above prototype was designed mostly to prove the feasibility and effectiveness of the approach, the testbed house was kept intentionally simple, with a limited number of rooms and user categories, and a small set of home appliances and policies; moreover, the implementation was rooted on TuCSoN 1.4, where some relevant features were not fully available.

In the Butlers (Denti, 2014) perspective, Home Manager can be seen as an early implementation of a small subset of the full architecture – in particular, with no support for mobility and pervasive aspects.

Accordingly, this paper presents the new "Butlerised" Home Manager—the evolution of the previous system in the Butlers perspective: the new prototype not only supports the remote control of the house appliances via an Android app, but exploits the user position, tracked via geo-localisation, to anticipate the user's needs in a simple, yet significant, scenario – namely, autonomously switching the house oven on when discovering that the user has just bought a take-away pizza in his/her way back home.

So, after shortly summarising Home Manager (Section 2), the Butlers vision (Section 3), and the TuCSoN infrastructure (Section 4), we present the new Home Manager system (Section 5), from the main requirements to the implementation, with the related discussion (Section 6). Related work and conclusions are reported in Sections 7 and 8, respectively.

2 HOME MANAGER

Home Manager (Molesini et al., 2009) is a prototype application for the control of an intelligent home, designed as a multi-agent system via the SODA methodology (SODA, 2008) and implemented on top of the TuCSoN coordination infrastructure (TuCSoN, 2008).

The system considers a house with independent devices (air conditioners, lights, etc.), each equipped with an agent to participate to the agent society. The coordination infrastructure, programmable via *tuple centres*, embeds the coordination laws required both to mediate among the different user's preferences and to pursue the overall system goals – in this case, to manage (limit) the overall energy consumption.

The system tries to satisfy the user preferences in terms of room lighting and temperature, unless higher-order energy constraint are violated; if multiple users with different preferences are in the same room, it also mediates among them by applying some suitable global policy. It features a rather complete role-based model, and a (pc-based) graphical user interface to configure and use the system with no need to operate directly on the underlying infrastructure.

Coherently with its proof-of-concept nature, the testbed was kept intentionally simple: no real mobility aspects were included (users were only considered inside the house), no remote control capabilities were supported, and the user's profile was limited. Also, there was no idea of anticipating any user need or desire – because "anticipating" an action requires to be informed of what is occurring, and where.

More recently, the Home Manager system has been re-interpreted, given its goals and features, in the Butlers perspective (briefly summarised in the next section), positioning the prototype in the Butlers conceptual reference layers (reported, with the author's permission, in Figure 1 for the reader's convenience).

3 THE BUTLERS VISION

The Butlers architecture (Denti, 2014) defines a framework with seven conceptual layers, which relate the availability of physical devices and enabling

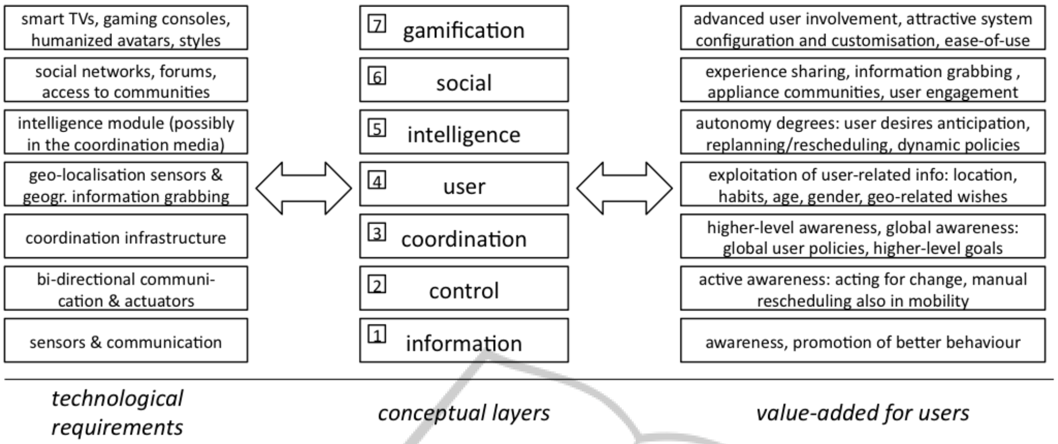


Figure 1: Butlers multi-layer reference architecture.

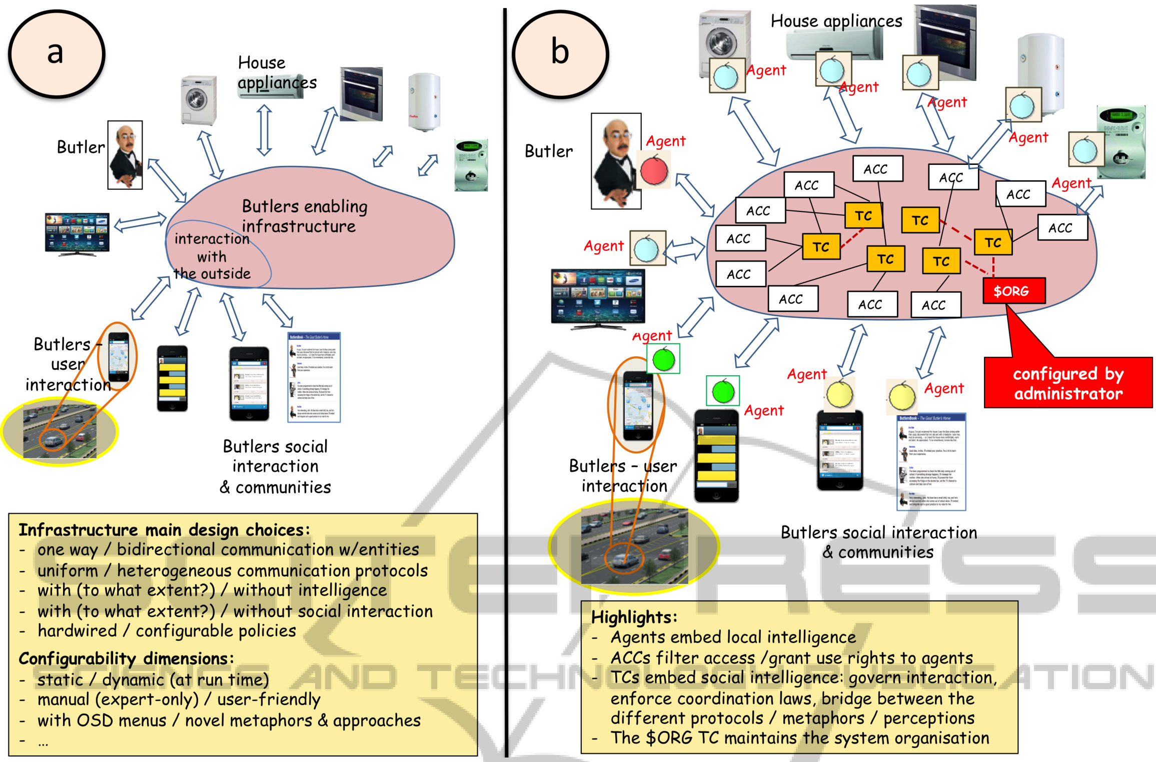
technologies with the set of features that a home management system can expectedly provide, on the one hand, and with the corresponding value-added for users, on the other (Figure 1). The bottom layers concern mainly enabling technologies (mono or bi-directional communication-enabled sensors, meters, actuators, etc.), while the middle layers are mainly infrastructural / middleware layers (aimed at providing coordination and geographical information services); the top layers, instead, are not necessarily to be taken in the sequence, for they focus on specific aspects like intelligence, social aspects, and *gamification* (the latter seen as a key success factor to promote technology acceptance in the mass market). The resulting conceptual map can be used both to locate a given system based on its feature – for instance, most of today's remote-controllable appliances, accessible via Android or iOS apps, are clearly located at level 2, while the Home Manager prototype above can be easily located at level 3, with minor aspects from level 5 – and, conversely, to identify the unexplored market niches – that is, possibly-interesting systems that are not currently available, suggesting their development.

This approach suggests new, intriguing scenarios. While a complete discussion is outside the scope of this paper, what is relevant here is that a smart home could interact with its inhabitants not only to monitor (level 1) and remote-control (level 2) the home appliances, but - provided that a suitable coordination infrastructure is available (level 3) - also to take into account the users' habits, behavior, location, and preferences (level 4) to reason on the overall situation (level 5) so as to possibly anticipate the user's needs. In principle, the social networks (level 6) and the gamification perspective (level 7) could also be exploited as a further source of information; however, these aspects are also outside the scope of this paper and will not be further considered in the following.

The resulting architecture, where the envisioned system is represented as an intelligent "home director" – the *butler* –, is shown in Figure 2(a): at this stage, the architecture is still technology-neutral—no specific coordination technology is selected, nor is the architecture necessarily agent-based. However, if an agent-based approach is adopted and the TuCSoN coordination technology is selected, the architecture can be refined as in Figure 2(b), where the TuCSoN coordination artifacts (*tuple centres*, TCs, and *agent coordination contexts*, ACCs) are explicitly highlighted. This process is described in full in (Denti, 2014): here, we just provide the essential details in Section 5 after summarising the basics of the TuCSoN model and infrastructure.

4 TuCSoN IN A NUTSHELL

TuCSoN (Tuple Centres Spread over the Network) (TuCSoN, 2008) is a coordination model and infrastructure based on a programmable coordination medium, the *tuple centre*, which is a tuple space enhanced with the notion of behaviour specification. Since the specification language, ReSpecT (Casadei and Omicini, 2009; Casadei and Omicini, 2010), is Turing-equivalent and both time-situated and space-situated, so as to perceive the environment as needed, any coordination-related computation can potentially be expressed, including those that need to perceive/act on the surrounding environment. As a result, the coordination tasks can be charged on top of the coordination media, where they conceptually belong, rather than onto the agents' shoulders. Agents coordinate by accessing and consuming tuples – ordered sets of data chunks – to/from the tuple centres, by means of the three basic *read*, *in* and *out* primitives.



Infrastructure main design choices:
- one way / bidirectional communication w/entities
- uniform / heterogeneous communication protocols
- with (to what extent?) / without intelligence
- with (to what extent?) / without social interaction
- hardwired / configurable policies

Configurability dimensions:
- static / dynamic (at run time)
- manual (expert-only) / user-friendly
- with OSD menus / novel metaphors & approaches
- …

Highlights:
- Agents embed local intelligence
- ACCs filter access /grant use rights to agents
- TCs embed social intelligence: govern interaction, enforce coordination laws, bridge between the different protocols / metaphors / perceptions
- The $ORG TC maintains the system organisation

Figure 2: The Butlers architecture in general (a) and in the TuCSoN-based concretisation (b).

The *Agent Coordination Context* (ACC) is the basic abstraction for modeling the space of interaction and agents communication, as well as the perception of the environment in which it is located. Roughly speaking, it can be seen as the conceptual boundary between the agent and the infrastructure: its aim is to bind and govern the interaction between the agent and the infrastructure, based on a customisable set of rules that define what the agent is allowed/denied to do. Any agent entering a TuCSoN-governed system must first ask for (negotiate) a suitable ACC, which is released with the rules granting the agent the appropriate access rights; in their turn, the ACC configuration is controlled by higher-level policies, set by the system administrator in the configuration phase.

So, properly-configured ACCs and tuple centres together can be exploited also to design and enforce the security aspects required in a distributed system, encapsulating authorisation and access control policies with the desired granularity.

5 BUTLER-ISING THE SYSTEM

Our goal is to extend Home Manager towards the Butlers layer 4 and 5, exploiting the user's *location* – tracked in real time thanks to the GPS and the other geo-localisation techniques embedded in modern smartphones – to enable an intelligent *reasoner* agent to take some autonomous decisions (for instance, adjusting the air conditioner temperature), possibly even anticipating some user's needs, managing the related devices on the user's behalf (for instance, deducing the opportunity to switch on the oven, or post-pone the washing machine, etc.).

More precisely, we mean *i)* to add the support for mobility, enabling users to operate on the home appliances remotely via a suitable Android application; *ii)* to geo-localise the user in real time, reifying this information into the Home Manager system as a suitable tuple; *iii)* to exploit this information to detect relevant user patterns – for instance, whether he/she is buying a take-away pizza when coming home – and consequently anticipate his/her needs proactively – in this case, to switch on the oven at 150 °C so as to warm the pizza (possibly rescheduling other energy-consuming tasks if necessary).

Figure 3 highlights the architectural changes. On the left side (a), the TuCSoN-based architecture of the general Butlers system depicted in Figure 2 (b) is downsized to the limited set of features supported by the original Home Manager prototype (Section 2). With respect to the full system, the selected subsystem considers only lighting, air-conditioning and heating, has no intelligent Butler recognisable as

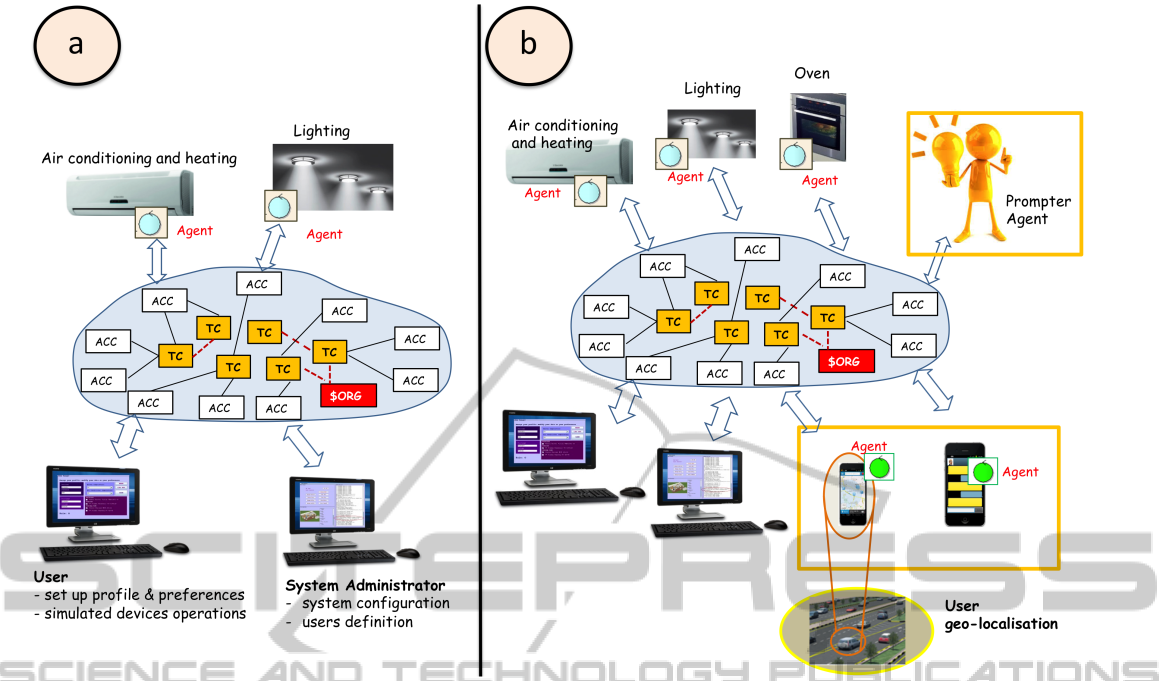


Figure 3: Home Manager architecture: current (a) vs Butlers-oriented (b).

Hi, grace,what do you want to do?

You have changed your air conditioner

Current status: On
Set temperature: 23°C

Figure 4: Home Manager Remote Control Screen.

such, user localisation is limited to the interior of the house and is simulated via GUI, and no social/gamification aspects are supported. Instead, the new "butler-ised" prototype (b) aims to introduce an Android app to achieve the goals *(i)* and *(ii)* above, and a "prompter" (reasoner) agent, in charge of providing suggestions for autonomous actions (issue *(iii)*).

As for issue *(i)*, the app allows the user, after authentication, to remote-control the home appliances: Figure 4 shows the case of the air conditioner.

As for issue *(ii)*, the user position is reified as a tuple of the form:

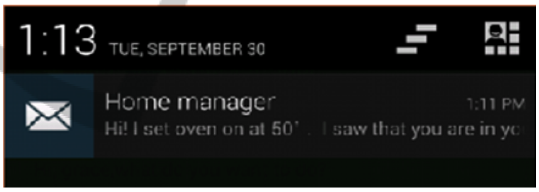
`geo_position(user, lat, lon)`

Such tuples are stored in a specific tuple centre, which is programmed to generate a time-stamped version of the same tuple, like the following:

`geo_position(user, lat, lon, time)`

used to record the user position history.

This information is then exploited, together with any other relevant information available in the user



1:13 TUE, SEPTEMBER 30

Home manager 1:11 PM

Hi! I set oven on at 50°. I saw that you are in yo

Figure 5: Notification of the prompter agent activity.

preferences, profile, etc., by the prompter agent to formulate its suggestions (issue *(iii)*): the consequent actions are autonomously performed, notifying the user via a text message on his/her phone (Figure 5). Of course, the user can change/cancel the action if, for any reason, he/she does not like it.

6 DISCUSSION

With respect to the original Home Manager described in (Molesini et al., 2009), the "Butler-ised" version overcomes the pc-based simulated control, in favor of an actual mobile support: users can interact with the system dynamically, in mobility, and are free to change their mind and programs any time and anywhere. On the other hand, the smart home is still simulated, and no real device is actually connected.

The value-added of this prototype, therefore, lies mainly in its step towards the Butlers vision: by adding the ability to reason over the user position, it shows – albeit in a very simplified, preliminary form

– the potential applications and the innovative pervasive scenarios envisioned in (Denti, 2014).

Clearly, the current implementation is just a proof of concept: a lot of work remains to be done to actually support the Butlers upper layers, both on the Home Manager system itself, and on the interface with actual (non-simulated) appliances. For instance, just to mention some aspects, the user authentication is currently un-ciphered, the reasoning of the prompter agent is barely trivial, no user profile information is actually exploited to customise the suggestions, etc.—not to mention the social and gamification aspects, which are totally missing at this stage. At the same time, the choice of the TuCSoN infrastructure as the underlying mechanism confirms to be a winning point: its ability to bridge between the different agent perceptions, particularly in the handling of the `geo_position` tuples, turned out to be essential to add the new features smoothly, and with virtually no impact on the existing system.

Widening the view, the aspect of *situatedness*, intended as the strict coupling with the environment, is emphasised as more and more crucial in today's complex computational systems (Weyns et al., 2007; Omicini and Mariani, 2013).

Accordingly, one further dimension, definitely worth exploring, is the specific support introduced in the latest TuCSoN edition (Mariani and Omicini, 2014), which explicitly considers environment *probes* and *transducers* as native metaphors. Moving from the basic consideration that the user's daily activities are unpredictable, but mostly depend on the environment being in some proper "enabler state" (and, conversely, impact on / are affected by the environment), the Home Manager/Butlers scenario could be matched on the TuCSoN meta-model by looking at environment changes as generated by probes and mediated by transducers, thus enabling a uniform representation of environmental properties despite the appliances' heterogeneity. Physically, agents would then be deployed to personal devices (like smartphones or desktop PCs, depending on their role and function), as in our new prototype, and probes to home appliances (as assumed by the Butlers' architecture), while ACCs, tuple centres and transducers could be put either on the user smartphone or remotely (the desktop PC), based on design considerations.

7 RELATED WORK

Given the broad scope of this paper, the relevant literature spreads onto several different research areas—from smart homes, domotic and home automation to energy monitoring and saving, ambient intelligence, assisted living and healthcare, and of course multi-agent systems, coordination models and infrastructures.

Due to space restrictions, the references that have already be discussed above will not be reported again.

Domotic and home automation systems have always aimed at automating the user interactions with home appliances, in several ways (Ché et al., 2010; Menon et al., 2013). Assisted living applications also aim to support the user, though from another perspective and context: in (Coronato and Pietro, 2010), for instance, a pervasive application is presented that aims to assist people everywhere and at any time.

In the smart home context, all the major players are introducing smarter appliances, architecture and applications. For instance Google in 2014 proposed 'The Works With Nest' approach, whose ecosystem aims to integrate heterogeneous apps and services from different vendors in an unique framework (Google, 2014), while Apple defined the Home Kit software platform aiming to integrate third party components and tasks with voice recognition capabilities for their activation (Apple, 2014). More generally speaking, appliance monitoring ranges from energy meters such as (Innova, 2012; World, 2013) to high-end air conditioners, fridges, etc. which today are often provided with Android or iOS apps for remote monitoring and control purposes.

In the multi-agent systems area, literature mainly concerns on energy-management (mostly about energy balancing and control, from the viewpoint of the energy provider than of the final consumer): very little exists on MAS applied to the management of home-related aspects. In (Tolbert et al., 2001), a scalable MAS controls the energy resources for reliability and efficiency purposes in power generation and consumption, while in (Nagata et al., 2000) the MAS handles the power restoration after faults. In (Zhang et al., 2011) the MAS combines local intelligence with coordination issues for load management in power grid systems, integrating the advantages of centralized and decentralized architectures. (Conte and Scaradozzi, 2007) is perhaps the most interesting example of a MAS used for home automation: the aim is to schedule the operation of different appliances according to the user defined priorities, keeping the global electric load below a given power threshold, to prevent a shut-down.

8 CONCLUSIONS

In this paper we presented the new Home Manager system, extended in the Butlers perspective.

Despite its many limitations and strong simplifications, due to its proof-of-concept nature, the "Butlerised" prototype makes one first, yet fundamental, step towards the Butlers upper layers: by reifying the user position via geo-localisation and adding the ability to reason over such information, enabling the anticipation of the user's needs, it highlights the potential of the innovative pervasive scenarios envisioned in (Denti, 2014), and shows their feasibility in principle.

For this reason, this result is more a starting point than the end of the story: as discussed above, a lot of work remains to be done, and many extensions, in several different directions, are worth considering.

For instance, if the above-cited frameworks by the major players become eventually popular, an integration with our approach could be studied, although their goals refer mainly to the Butlers lower layers (2, 3 and possibly 4) while we mean to incrementally enable advanced features from the Butlers upper layers.

At the same time, serious difficulties have to be expected when moving on, especially towards the upper Butlers layers: apart from technical difficulties in enabling the interaction among so many heterogeneous entities, the effectiveness and the actual scalability of the approach are still to be proved, and so are the robustness and reliability of the TuCSoN infrastructure in such a challenging scenario. Proper knowledge representation is also likely to be a critical issue when the application scenario goes beyond the toy example presented in this paper. Moreover, in the perspective, proper design metaphors and user metaphors will be needed to deal with the complexity of layers 6 and 7, with their challenging goals of supporting experience sharing, advanced user involvement, etc.

So, our approach will be to proceed stepwise: the first planned step is to go beyond the single, predefined take-away pizzeria considered in this example, adding the notion of user's favorites and possibly importing them from existing social networks (Facebook, Google+, and others), thus adding some (minimal) layer 6 feature. The next step will be to integrate the system with Google Maps (at least partially), so as to grab the user's check-in location (take-away pizzeria, restaurant, chemist's, etc) from that source, removing the current requirement that it is known a-priori. Apart from the better usability, this step would strongly enhance the prompter agent role, while constituting a major testbed both for the Butlers architecture and the Home Manager system design.

ACKNOWLEDGEMENTS

Authors would like to thank Dipl. Eng. Ilaria Bertoletti for her contribution to this project and her work to the new prototype and the Android app, and Dr. Eng. Leo Di Carlo for his work to the extension of basic the prototype.

REFERENCES

Apple(2014). Home kit https://developer.apple.com/homekit/

Ayala, I., Amor, M., and Fuentes, L. (2013). Self-configuring agents for ambient assisted living applications. *Personal and Ubiquitous Computing*, 17(6):1159–1169.

Bartolini, S., Milosevic, B., D'Elia, A., Farella, E., Benini, L., and Cinotti, T. (2012). Reconfigurable natural interaction in smart environments: approach and prototype implementation. *Personal and Ubiquitous Computing*, 16(7):943–956.

Busi, N., Ciancarini, P., Gorrieri, R., and Zavattaro, G. (2001). *Coordination models: a guided tour*. Spinger.

Casadei, M. and Omicini, A. (2009). Situated tuple centres in ReSpecT. In Shin, S. Y., Ossowski, S., Menezes, R., and Viroli, M., editors, *24th Annual ACM Symposium on Applied Computing (SAC 2009)*, volume III, pages 1361–1368, Honolulu, Hawai'i, USA. ACM.

Casadei, M. and Omicini, A. (2010). Programming agent-environment interaction for mas situatedness in respect. *The Knowledge Engineering Review*. Regular Submission.

Ché, N., Pardons, N., Vanrompay, Y., Preuveneers, D., and Berbers, Y. (2010). An intelligent domotics system to automate user actions. In *Ambient Intelligence and Future Trends-International Symposium on Ambient Intelligence (ISAmI 2010)*, volume 72 of *Advances in Intelligent and Soft Computing*, pages 201–204. Springer Berlin Heidelberg.

Chong, N.-Y. and Mastrogiovanni, F. (2011). *Ambient Intelligence and Smart Environments: Trends and Perspectives*. IGI Global.

Conte, G. and Scaradozzi, D. (2007). Applying mas theory to complex home automation systems. *Modeling and Control of Complex Systems*.

Coronato, A. and Pietro, G. D. (2010). *Pervasive and Smart Technologies for Healthcare: Ubiquitous Methodologies and Tools*. IGI Global.

Denti, E. (2014). Novel pervasive scenarios for home management: the butlers architecture. *SpringerPlus*, 3(52):1–30.

Ducatel, K., Bogdanowicz, M., Scapolo, F. Leijten, J., and Burgelman, J.-C. (2001). Scenarios for ambient intelligence in 2010. final, IPTS – European Commission's Joint Research Centre.

Gamification Community (2013). The gamification community portal. http://www.gamification.org.

Google (2014). Works with nest. http://techcrunch.com/2014 /06/23/google-makes-its-nest-at-the-center-of-the-smart-home/.

Innova (2012). Regolazione automatica e telegestione, domotica. http://www.innova-srl.eu.

Mariani, S. and Omicini, A. (2014). TuCSoN coordination for MAS situatedness: Towards a methodology. In Santoro, C. and Bergenti, F., editors, *WOA 2014 – XV Workshop Nazionale "Dagli Oggetti agli Agenti"*, volume 1260 of *CEUR Workshop Proceedings*, pages 62–71. Sun SITE Central Europe, RWTH Aachen University.

Menon, V., Jayaraman, B., and Govindaraju, V. (2013). Enhancing biometric recognition with spatio-temporal reasoning in smart environments. *Personal and Ubiquitous Computing*, 17(5):987–998.

Molesini, A., Denti, E., and Omicini, A. (2009). An agent-based application for home intelligence. In Filipe, J., Fred, A., and Sharp, B., editors, *International Conference on Agents and Artificial Intelligence (ICAART 2009)*, pages 377–384, Porto, Portugal. INSTICC.

Nagata, T., Watanabe, H., Ohno, M., and Sasaki, H. (2000). A multi-agent approach to power system restoration. *Power System Technology*.

Omicini, A. and Mariani, S. (2013). Coordination for situated MAS: Towards an event-driven architecture. In Moldt, D. and Rölke, H., editors, *International Workshop on Petri Nets and Software Engineering (PNSE'13)*, volume 989 of *CEUR Workshop Proceedings*, pages 17–22. Sun SITE Central Europe, RWTH Aachen University.

Omicini, A. and Papadopoulos, G. A. (2001). Editorial: Why coordination models and languages in AI? *Applied Artificial Intelligence: An International Journal*, 15(1):1–10. Special Issue: Coordination Models and Languages in AI.

Omicini, A. and Rimassa, G. (2004). Towards seamless agent middleware. In *IEEE 13th Inter. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2004)*, pages 417–422. IEEE CS.

Omicini, A. and Zambonelli, F. (1999). Coordination for Internet application development. *Autonomous Agents and Multi-Agent Systems*, 2(3):251–269.

Papadopoulos, G. A. and Arbab, F. (1998). Coordination models and languages. *Advances in Computers*, 46:330–401.

Ricci, A., Omicini, A., and Denti, E. (2002). Virtual enterprises and workflow management as agent coordination issues. *International Journal of Cooperative Information Systems*, 11(3/4):355–379.

SODA (2008). Home page. http://soda.apice.unibo.it/.

Tolbert, L., Qi, H., and Peng, F. (2001). Scalable multi-agent system for real-time electric power management. *IEEE Xplore Digital Library*.

TuCSoN (2008). Home page. http://tucson.apice.unibo.it/.

Weyns, D., Omicini, A., and Odell, J. J. (2007). Environment as a first-class abstraction in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30. Special Issue on Environments for Multi-agent Systems.

World, A. (2013). Artie world. http://www.artie-world.com.

Zhang, M., Ye, D., Bai, Q., Sutanto, D., and Muttaqi, K. (2011). A hybrid multi-agent framework for load management in power grid systems. In Bai, Q. and Fukuta, N., editors, *Advances in Practical Multi-Agent Systems*, volume 325 of *Studies in Computational Intelligence*, pages 129–143. Springer Berlin Heidelberg.