Intelligent Environments 2018
I. Chatzigiannakis et al. (Eds.)

doi:10.3233/978-1-61499-874-7-364

A Highly Flexible Architecture for Intelligent Workplaces Enabling Easy Customization of Environments

Óscar ORTIZ[a,1], Iván PAU[a] and Mario VEGA-BARBAS[b]
[a] *Department of Telematic Engineering and Electronic, Universidad Politécnica de Madrid, ETSIS de Telecomunicación - Campus Sur, 28031 Madrid, Spain*
[b] *Institute of Environmental Medicine, Karolinska Institutet, 171 65 Stockholm, Sweden*

Abstract. Intelligent environments are a natural tool for the creation of intelligent workplaces. However, in order to be used in this way, it is necessary to face the challenge posed by the variability of the elements included in the space and the need to adjust the smart elements to the preferences of the users. This paper presents a service-based architecture that enables the creation and management of intelligent workplaces based on the concept of intelligent environment. To this end, a series of functional restrictions are presented, which simplify the design and implementation of the services to be deployed, and information models, to formalize both the space and the preferences of the users based on the concept of activity. The architecture and models presented facilitate the adaptation of a solution developed by a supplier to any type of intelligent environment in an efficient and scalable way.

Keywords. User Context Management, Microservices, Smart Space, Service Adaptation, Intelligent Workplace

1. Introduction

The inclusion of intelligent objects, Internet of Things (IoT) and advances in Ambient Intelligence (AmI) facilitate the development of new pervasive sensitive services for their inclusion in the user's vital spaces, transforming them into intelligent environments [1]. In this way, it is possible to develop services that can accompany the users in their daily life, supporting them in the performance of daily tasks, and using these intelligent environments as a means of deployment [2]. This approach allows to abstract users from the underlying technology to each service which is included in the environment, translating the capabilities of the intelligent environment into comprehensible concepts such as activities, actions or tasks. In this way it is possible to address important challenges in the development of intelligent environments such as the acceptance and adoption of the offered solution [3], which go beyond the traditional ones such as technology, security or privacy.

[1] Óscar Ortiz, Corresponding author, Universidad Politécnica de Madrid, Campus Sur Universidad Politécnica de Madrid, Crtra de Valencia, km 7, 28031 Madrid, Spain; E-mail: oscar.ortiz@upm.es.

Thus, the capabilities of the intelligent environment are defined through the set of services deployed in it [4]. However, although this approach allows to speak a language closer to the user, the correct functioning of each service is conditioned by the techology available at each moment and, above all, by the user's profile that interacts with the environment, that is, the context of use [5].

A clear example where this customization is very important are the workspaces [7]. A clear idea of the capabilities and needs of the user greatly facilitates the adaptation of the services deployed in the environment to ensure not only that the user can carry out the tasks that must be performed, but also accompany the user to perform them in a healthy way, transforming this work scenario into a healthy intelligent workspace.

This article presents a software architecture that allows the reconfiguration of the intelligent environment through the management and redeployment (if necessary) of pervasive sensitive services when context variations are detected. The concept of context includes both the use of intelligent objects when interacting with actors in the intelligent space and the profile of said actors.

2. System Overview

The system that describes the problem raised in this work is detailed in Figure 1. As can be seen, each intelligent environment is determined by the set of available solutions. These solutions are built by combining small services [8,9] that are proposed by a provider and will depend both on the technology available in the environment and the context of the environment, i.e., the available smart objects and how the user decides to use them.

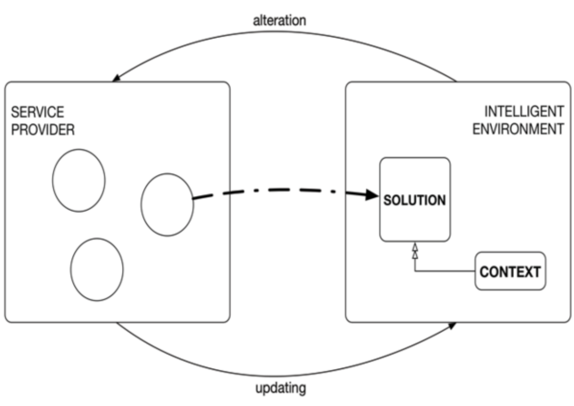

Figure 1. System overview.

Smart environments tend to undergo frequent context alterations and, in many cases, are directed by users of space. In these cases, the system must adapt in two different ways:

- Adaptation of the set of services that make up the solution according to the context. As indicated previously, the solution consists of a set of services that are coordinated to provide the requested functionalities. Since several of the

services interact directly with the objects of the space, a change of context must imply a change in the number and type of services deployed. This adaptation of the number of services can be done locally, within the scope of the intelligent environment, or remotely in the service provider. In the latter case, a process of service deployment must be produced;
- Coordination of services to meet the objectives. In addition to choosing the appropriate set of services to be deployed it is also important to define the coordination criteria between them. Service may sue the other service capabilities to carry out its mission. This coordination process should be defined from the moment in which an adaptation of the services is made to changes in context.

In the case of the adaptation of intelligent workspaces to the characteristics of the users, both the process of adaptation and the coordination of services can be simplified to offer a simple but functional solution. The set of services to be deployed will depend solely on the furniture or devices configurable in the space. The coordination of the services can be based on the preferences of the user of the space. These preferences define the mode of operation of each service without requiring them to exchange information between them.

So the functioning of the system would be the following. In the workspace a process of discovery of furniture and devices with capacity to be managed by services is carried out. Based on the information obtained after this process, the services are deployed in the service manager of the intelligent space or an adaptation is made if it is applicable. In general, each configurable element will have a service that will be responsible for its management.

On the other hand, the user transmits his profile to the space. In this profile preferences are defined in terms of the work situation in which the user is located. The services, in addition to being able to configure the space devices, must know when they should act by changing the configuration of the environment; they must understand the existing information in the user profile and finally translate it to the specific configuration of the device they serve.

This operation, although it does not seem sophisticated, poses some challenges that must be resolved in the design phase:

- How are user preferences managed?
- How do services know they must make a configuration change?

The next sections show both the high-level information models that support the coordination between services to adapt the environment to user preferences and an architecture that allows their implementation in a scalable manner. In this work, the process of deploying services has not been taken into account, assuming that these are already deployed.

3. Information Model

The fundamental element of coordination between services is the concept of activity. It is assumed that the user of an intelligent workspace is always carrying out at least some activity. From the concept of activity, two information elements are defined:

- Service-Activity Link model (SAL). Represents the activities in which the services are involved.

- User's Preferences for Activities model (UPA). User preferences stored in your profile and classified according to the activity.

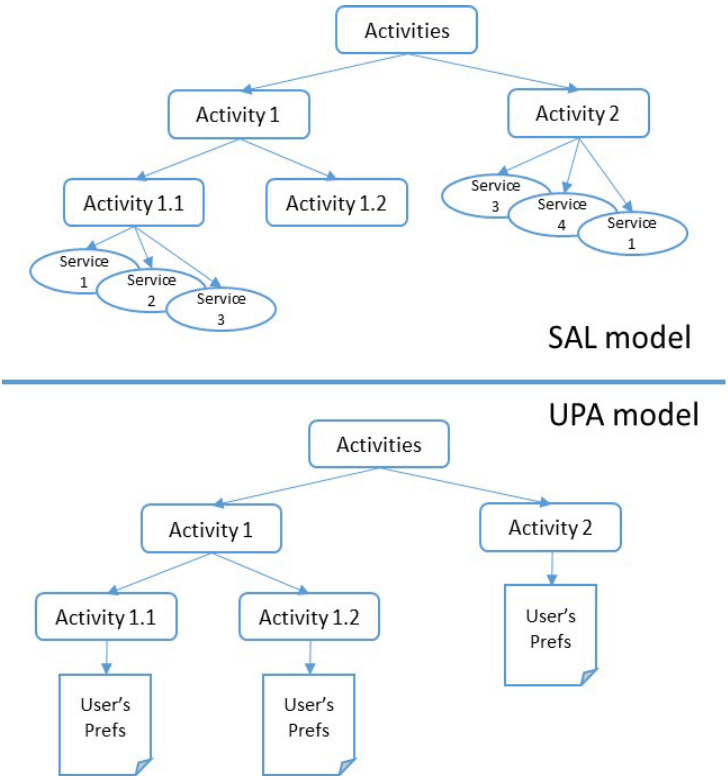

Figure 2. Information model overview.

As can be seen in the previous figure, services are labelled according to the activity in which they have to act (they can be in various activities). When a change of activity occurs, the system must notify it to all the services of the activity that ends, and to the services of the activity that begins.

The services that have been notified that the activity begins must consult the user's preferences for that activity. The information of the preferences must be made through an ontology. This ontology reflects concepts similar to furniture or devices managed by the services to facilitate the correspondence process. Therefore, the services must know the ontology in which the profile is represented and, in addition, be aware of which entity corresponds to the element they manage.

The user can alter the conditions of the workspace in two ways:

- Changing your profile, which implies that the services adjust the elements they control in another way.
- Adding new elements to the context that, associated with a control service, must again be adapted according to the information contained in the profile.

In this version has not been considered the problem of parallel activities and how to resolve their possible configuration conflicts.

4. Architecture Description

Once the information models that facilitate the described adaptation process have been defined, it is necessary to define an architecture that allows integrating the necessary functionalities so that the different services can carry out their functions in a coordinated manner.

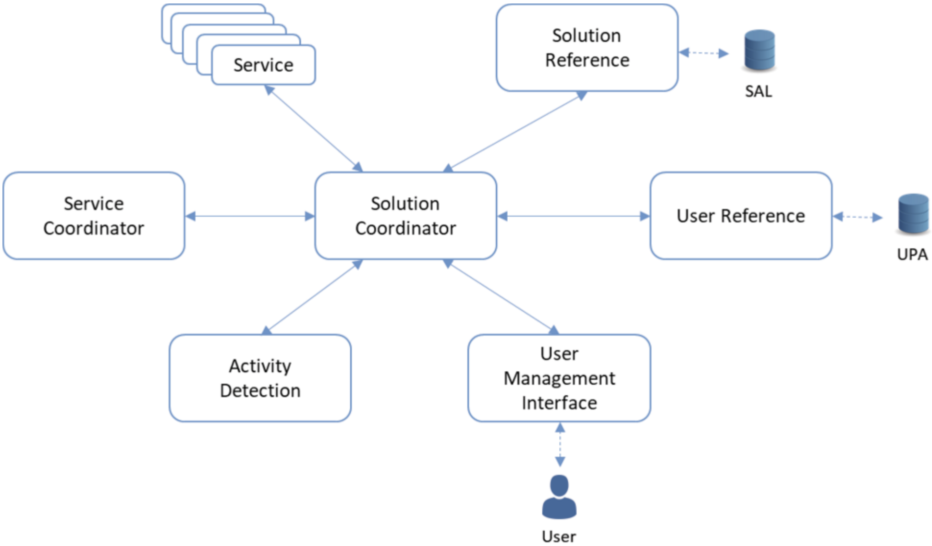

Figure 3. Architecture overview.

In the previous figure, a high-level representation of the proposed architecture is shown. The architecture is based on an intermediate element called "Solution Coordinator" that facilitates the interconnection of all the elements. In the architecture only the necessary elements are shown to carry out the functionalities described in this article. However, in a real environment it is necessary to perform context monitoring tasks, deployment management, service version management, communications interface with service provider, etc. All these elements can be integrated through the "Solution Coordinator".

Next, the rest of the elements of the proposed architecture are described:

- *User Management Interface.* It is responsible for obtaining the user's profile and sending it to the module that manages the information related to the user (User Reference). It also takes care of the interaction with the user to establish the activity that is being carried out at that moment or to make changes in the profile. A change of activity will be communicated to the "Service Coordinator" module.
- *Activity Detection.* Module in charge of detecting the activity that the user will perform. Changes of activity will be communicated to the "Service Coordinator" module.
- *Service Coordinator.* Manage activity changes Consult the services involved in the outgoing activity and the incoming activity through the "Solution

Reference" module. After knowing these services will send a message indicating the cessation or start of an activity.

- *User Reference.* It manages all the user information stored in the UPA. Respond to queries from the rest of the modules.
- *Solution Reference.* It manages all the user information stored in the SAL. Respond to queries from the rest of the modules.

5. Conclusions

The presented architecture allows solving the problem of context variability and user preferences when implementing intelligent workspaces. For this, several functional restrictions have been defined that enable the creation of solutions based on simple services with few dependencies with other services. The services are also aligned with the different elements (devices and furniture enabled) of the intelligent space where they are deployed. The coordination of these services, which in general is limited, is carried out through a series of mandatory services in architecture. These services manage information elements that represent both the preferences of the user and the specific type of services deployed. The correspondence between preferences and typology is made based on the concept of activity.

Although the presented architecture supposes a first outline of solution, it has some limitations that will have to be solved in future investigations. To begin with, the article raises general concepts and models of information little detailed. It is necessary to further detail those models specifying the fields that should be included. This is especially important in the case of user preferences. It is necessary to find an information model that is capable of collecting all the aspects of interest for the intelligent workspace.

In addition to the limitations inherent to an initial research proposal, there is one aspect that must be adequately addressed: the existence of more than one person in the work space. Although the architecture could be adapted to this circumstance, it is possible that there are inconsistencies in the preferences of the users that must be resolved through the appropriate strategies.

References

[1] J.C. Augusto, H. Nakashima, *Handbook of Ambient Intellignence and Smart Envirionments,* Ambient Intelligence and Smart Environments: A State of the Art, 2010:3–31.

[2] M. Kuniavsky, *Smart things: ubiquitous computing user experience design*. Elsevier, 2010.

[3] M. Vega-Barbas, I. Pau, F. Seoane, Confidence: Dependencies and their critical role in fostering user acceptance in pervasive applications, In *Wireless Mobile Communication and Healthcare (Mobihealth 2014),* 283-286, IEEE.

[4] M. Vega-Barbas, I. Pau, M.L. Martín-Ruiz, F. Seoane, Adaptive software architecture based on confident HCI for the deployment of sensitive services in Smart Homes. *Sensors*, *15*(4), 2015, 7294-7322.

[5] M. Maguire, Context of use within usability activities. *International Journal of Human-Computer Studies*, *55*(4), 2001, 453-483.

[6] K.P. Lam, V. Srivastava, Living in the intelligent workplace structuring and managing building operation information, 2005. (In Internet at https://oaktrust.library.tamu.edu/bitstream/handle/1969.1/5093/ESL-IC-05-10-02.pdf?sequence=4&isAllowed=y)

[7] N. Alshuqayran, N. Ali, R. Evans, A systematic mapping study in microservice architecture. In *Service-Oriented Computing and Applications (SOCA), 2016 IEEE 9th International Conference on*, 44-51, IEEE.

[8] L. Chen, M.A. Babar, Towards an evidence-based understanding of emergence of architecture through continuous refactoring in agile software development. In *Software Architecture (WICSA), 2014 IEEE/IFIP Conference on*, 195-204, IEEE.

[9] M. Garriga, Towards a Taxonomy of Microservices Architectures. In *International Conference on Software Engineering and Formal Methods, 2017,* 203-218, Springer.