Autonomic context management in interoperable pervasive platforms

Philippe Lalanda[1], German Vega[1], Felix Maximilian Roth[2], Christian Becker[2]

[1]Grenoble Alpes University
621 Avenue Centrale, St Martin d'hères
38000 Grenoble, France
firstname.name@univ-grenoble-alpes.fr

[2]University of Mannheim
Schloss
68131 Mannheim, Germany
firstname.name@uni-mannheim.de

***Abstract*— Pervasive computing envisions environments where computers are blended into our environment to provide services. This vision is today very popular and a number of pervasive platforms are already used today in smart homes or smart plants. These platforms however are based on different middleware and networks. Interoperability has recently emerged as a key issue. Interoperability is a difficult problem which requires interactions between platforms that were not designed to do so. It demands to be able to communicate but also to share contextual information on demand. We here present an architecture to federate platforms and a solution to autonomically manage context at the platform level. This is demonstrated in the smart home domain and, more precisely, with the iCasa/iPOJO and Base pervasive platforms.**

Keywords—* *interoperability; autonomic management; smart home.

I. Context

Pervasive computing envisions environments where computers are blended into everyday objects in order to provide added-value services to people. This vision is set to become reality and is actually raising huge expectations from consumers and service providers alike. As a result, many smart devices and open platforms are now proposed and installed in our homes and workplaces. However, because of a lack of standards and the absence of cooperation between providers, the deployment of services in our living places is today rather anarchic. This results in highly heterogeneous and sometimes incoherent installations, leading to major software-related issues.

Our purpose is to define architectures and supporting techniques enabling interoperability in pervasive environments. In our vision, an application developed on a given platform could use services provided by another one and vice versa. This goal is challenging. Amongst other things, it requires platforms to build and maintain knowledge about themselves, about the environment, and about the capabilities of other platforms. It also demands abilities to conduct interactions with others in order to build shared contextual information and to find out if remote services can be actually used. Finally, it calls for a development framework including these facilities and facilitating programming activity.

We have designed a service-based architecture promoting interoperability in pervasive environments. This architecture is founded on XWARE, a customizable framework enabling communication between heterogeneous services run on different pervasive platforms [1]. XWARE is based on a modular design in order to facilitate the customization towards a specific domain and the integration of further middleware platforms by composition of components. Thus, for the framework configuration, the developer/administrator can choose between different components for service-oriented features, e.g., advertisement, lookup, matching, and invocation. The framework can be executed in the pervasive infrastructure, on an intermediary node on the network for instance, or at-the-edge on mobile devices. Several XWARE instances can be distributed on different nodes in order to support various middleware platforms. The current version of XWARE supports a number of platforms, including iCasa/iPOJO [2], BASE [3], and UPnP (see upnp.org). This is illustrated hereafter by figure 1.

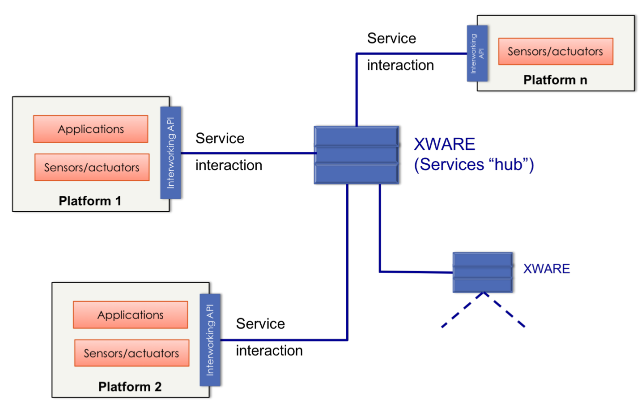

Fig1. Architecture for pervasive platforms interoperability.

This architecture allows applications run on a given platform to use remote services. This is however hard to manage from the application developers' perspective, especially if there are a large number of available services (which is rapidly the case). Indeed, the application code can get very complex if the application is in charge of finding the best services at any moment.

II. Proposed Approach

To cope with this issue, we have extended our pervasive platform, iCasa, with a service-oriented context (see figure 2). This platform builds upon the Apache service-oriented component model, iPOJO. Context appears as a dynamic set of services. Depending on available data sources and application needs, different services are published in and withdrawn from the platform service registry. They are then opportunistically used by the pervasive applications, coded in iPOJO. The context module tracks any contextual modifications and sends an event to alert consumer applications. This feature for interoperability has to be achieved in an autonomic way [4,5]. A major constraint in pervasive environments is that there is usually no local administrator. Most management operations must be undergone by the system itself.

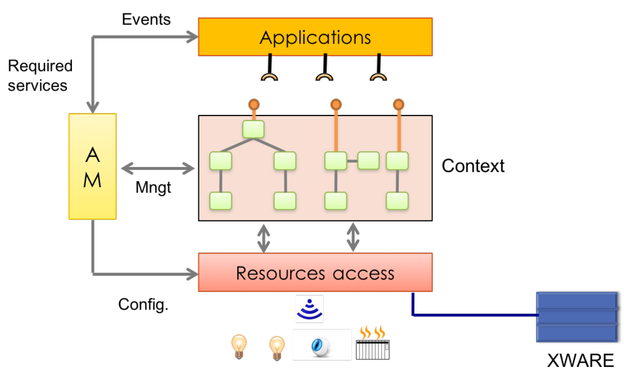

Fig2. Autonomic context management in iCasa.

The purpose of this demonstration is to show how the context is built depending on the evolving needs of applications (in term of required services) and on the services availability. Precisely, if local services are not available, requests are sent to an XWARE instance. Once a service provided by a remote platform is used, events will be sent to iCasa to follow the service life-cycle.

For demonstration purposes, we decided to focus on smart homes and smart buildings, which are today very active and even emblematic domains. Precisely, a smart home can be defined as a house with a computing infrastructure (devices and networks) allowing the execution of services that can be downloaded, installed, removed by users. Services can be found in a variety of domains, including healthcare, energy management, security and comfort. Similarly, smart buildings are equipped with communication and computing capabilities in order to provide services to inhabitants. Here again, those services have their own life-cycle: they can be installed, updated, and retired.

To simulate a smart home (a smart apartment in our case), we use the iCasa suite. It includes a runtime, an Eclipse-based IDE [6] and a smart home simulator that supports the execution of predefined scenarios in order to quickly test pervasive applications. The simulator includes a map presenting a home and its computing infrastructure (devices and states). A map example is provided hereafter in figure 3 (apartment A).

Our iCasa platform is then inserted into a smart building, including several smart apartments sharing services through an XWARE instance. Smart apartments are simulated by heterogeneous pervasive platforms – BASE in this case – run on different computers. Finally, the XWARE instance runs on a specific box installed in the building's common spaces.

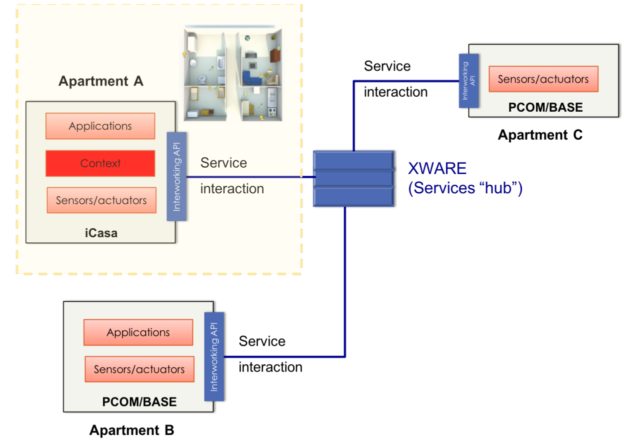

Fig3. Demonstration settings.

We have defined a simple home automation use case, which purpose is to manage HVAC in unoccupied houses. This application uses thermometers, shutters, heaters, and chillers in order to maintain a safe temperature in the house. The purpose of the demonstration is to introduce a number of perturbations and show how the application self-adapts. For instance, if thermometers in the controlled rooms are turned off, the platform uses XWARE instances to find equivalent remote services. In this example, temperatures can be provided by thermometers placed in another apartments, or event outside. the application is aware of the provided qualities of service and adapts to these new conditions. The demonstration is illustrated through a visual tool we have developed to present the context and the applications.

To conclude, IEEE defines interoperability as the ability of two or more systems to exchange information and to use the information that has been exchanged. In this paper, we focus on that later aspect, proposing to extend pervasive platforms with autonomic contexts to ease application development.

References

[1] F.M. Roth, C. Becker, G. Vega, P. Lalanda, "XWARE, a customizable interoperability framework for pervasive computing systems", Pervasive and Mobile Computing, 47, 13-30, 2018.

[2] C. Escoffier, R. S. Hall, and P. Lalanda, "iPOJO: An extensible service oriented component framework," in IEEE SCC 2007, pp. 474–481

[3] C. Becker, G. Schiele, H. Gubbles, K. Rothermel, « BASE A microbroker-based middleware for pervasive computing», Percom 2003.

[4] P. Lalanda, E. Gerber-Gaillard and S. Chollet, Self-Aware Context in Smart Home Pervasive Platforms, IEEE ICAC'16, Columbus, 2017.

[5] P. Lalanda, J. McCann, A. Diaconescu, « Autonomic Computing », Springer Verlag, 2013

[6] P. Lalanda, C. Marin, « A domain-configurable development environment for service-oriented applications», IEEE Software, vol. 24, n°6, 2007.