

Integrating local services and applications into external user home environments

Helder Pinto and Rui José

University of Minho, Information Systems Department
Campus de Azurém, 4800-058 Guimarães, Portugal
{helder,rui}@dsi.uminho.pt

Abstract. Users moving between different ubiquitous computing environments should be able to take a full profit of the resources existent there, if their home environment (e.g. a mobile portal) could seamlessly integrate with local infrastructure. Pursuing such goal, this paper focuses on the aspects of integrating local applications and services into an external user home environment, with particular emphasis on the interoperability and performance issues. Based on the concept of Value ADDED Environment (VADE), as an administrative and physical domain, where the locally available facilities can be combined with external user home environments, the paper presents a middleware solution supporting the integration of local services and applications available in VADE environments into an external portal.

1 Introduction

Research in ubiquitous computing has resulted in prototype ubiquitous computing environments that are often characterized by a strong integration between the computational infrastructure and the physical environment. The main advantage of this strong coupling with the physical environment is essentially the possibility to support functionality that is very specific to that particular environment. The main limitation, however, is that such level of integration can only be reasonably achievable for systems aimed at small and specific physical infrastructures, meaning that, in the course of a day, a user will have to interact with a multitude of independent ubiquitous environments, one for each visited location. This limitation is even more important due to the lack of an administrative infrastructure that could be used across domains to support key features, such as authentication, user profiling or billing.

Mobile network operators (MNO) are also launching services, usually by means of a mobile portal, that use the position of mobile terminals to provide users with information that is associated with their particular location. These location-based services are available everywhere and have a strong administrative framework, but they require from the server to provide information for any of the potential locations of the client, which implies the ability to consolidate all location-based contents into large warehouses, posing operational and data integrity issues that are costly to overcome [1]. Furthermore, these applications are

not able to explore resources that may be available in the physical environment. Without the assistance of a local infrastructure, these applications do not have the potential for reaching the same level of richness and context-awareness as the ubiquitous computing applications created for a specific physical infrastructure.

This research aims to combine the two previous approaches, and is specifically targeted at the issues of interoperability and seamless integration between a user home environment (e.g. a MNO portal) and the ubiquitous computing environments corresponding to the physical locations visited by the user. Our approach introduces the concept of Value ADded Environment (VADE) as an administrative and physical domain, e.g. a shopping center or an airport, where the locally available ubiquitous computing facilities can be combined with the external home environment of visiting users. The overall scenario is that when entering a VADE, mobile users are provided with functionality that corresponds to the dynamic combination of predefined user preferences, currently active applications, current user context and locally available services and applications. While this research copes with multiple challenges, this paper focuses only on the integration of local resources into an external environment. The paper describes the main issues implied in the integration between local and external environments, namely interoperability and performance, using the VADE system as a representative example of the integration challenges. The paper argues for the need for an integration middleware dealing with the identified issues, and describes the solution we developed for the VADE system. For a more complete description of the VADE system and other aspects covered by this research, please refer to [2].

The next section describes the VADE concept, its application models, and the system architecture. Section 3 introduces the issues of dynamically integrating local applications and services into the user home environment. Section 4 explains how we dealt with those integration issues by the use of a VADE-aware integration middleware.

2 Value Added Environment

The VADE concept is here presented as an example of integration between local and external environments. The idea behind this concept is that when a user enters a ubiquitous computing environment that also happens to be a VADE system, the functionality available to that user should be the combination of global and local functionality. In order to clarify these relationships, we have grouped the overall system functionality into three separate domains, more specifically: User, VADE, and External domains. The User domain corresponds to the physical and logical elements carried by a person that is within the physical boundaries of a VADE. The VADE domain corresponds to elements constituting the local infrastructure and directly supporting the VADE functionality (e.g., services, applications, and all sorts of devices – sensors, input/output, etc.). Finally, the External domain corresponds to all the other elements that interact with the VADE but are not part of the VADE, or user domains. Our main assump-

tion about the External domain is the existence of a structured framework for customer relationship that is able to support key horizontal services, such as personal context management, user profiles, and a personal information space, possibly in the form of a mobile portal.

2.1 Application models

VADE-aware global applications. A VADE-aware global application is an external application that, when the user enters a VADE, is able to enhance its functionality by taking advantage of the locally available services. A global application is not aimed at any VADE in particular and may be used regularly outside a VADE. We assume that these applications are context-aware, and also that, as users move, they are continuously searching for additional location-based resources. Entering a VADE is thus a major opportunity for finding new, probably much more specific, resources and context information that will allow them to improve their relationship with the physical environment of the user, and thus the value of the provided functionality. For example, it is simple for a global map application to present a detailed map of the area where a shopping center is located, but for the people inside the shopping what would be real value would be to have access to a detailed plan of the shopping. The main challenge is to be able to discover, recognize and use the potentially very vast range of services that may be found at different environments.

VADE Local Applications. A VADE local application provides functionality that is specifically associated with a particular VADE. When a user enters a VADE, these applications should seamlessly integrate with the users mobile information space. i.e. the user should not need to look for them, configure them or study their functionality. The main advantage of this application model over the VADE-aware model is that, since these are local applications, they can be tailored for the particular environment for which they were produced. The main challenge for this type of application is how to manage their integration into the user external environment.

2.2 System architecture

Figure 1 depicts the VADE system architecture. The main components of the VADE architecture are a browser (User domain), a personal portal and a personal context manager (External domain), and a set of applications and services in the Vade domain (some of which are not stated in this paper for the sake of conciseness).

Each mobile user has her/his own Context Manager, which gathers and aggregates context information (e.g. location, activity, etc.) from several context sources. The Personal Portal relies on the Context Manager to obtain user context information, possibly influencing the content produced by each portal application (also called portlet). The context information may contain a notification of the user entrance in some VADE domain. This notification contains a reference to the VADE's internet domain name, which is a basis to access the VADE

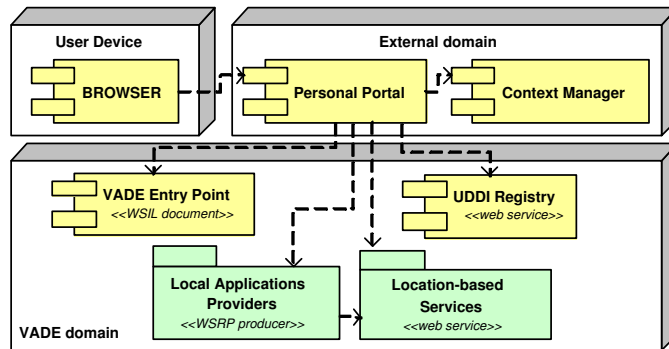


Fig. 1. General architecture of the VADE system

Entry Point at a well-known URL. The VADE Entry Point is a Web Services Inspection Language (WSIL) [3] document providing references for the VADE core services. One of the VADE core services is a UDDI registry that is used by the Personal Portal and its VADE-aware global applications to discover local resources (applications and services respectively). VADE Local applications correspond to Web Services for Remote Portlets (WSRP) [4], a standard allowing for the integration of remote portlets into a portal. Location-based Services are also Web Services that implement a well-known interface used by VADE-aware global applications.

3 Integrating local resources into external environments

This section describes the main problems of integrating local ubiquitous infrastructures with external environments: interoperability and performance. These issues are analyzed in the scenario considered in our work, where essentially two forms of integration between local and external environments take place: VADE location-based services are used by VADE-aware global applications running on external portals; and VADE local applications are integrated into external personal portals. We believe the issues raised by this scenario represent the general cases of integration between local and external environments, and thus our work should be viewed as a model of how to deal with these specific issues.

3.1 Interoperability

Interoperability is an unavoidable challenge for the integration between heterogeneous environments. However, external user home environments should not be faced with several different integration mechanisms for each local environment the user moves in. A homogeneous ground for local resource discovery and access should exist between all local environments, so that interoperability with external environments is possible.

For example, VADE-aware global applications are continuously searching for local resources in order to provide finer-grained information to mobile users. The need for interoperability essentially comes from service discovery and access. VADE-aware global applications must rely on widely accepted protocols for service discovery and access. Furthermore, local services interfaces should be well-known, so that VADE-aware global applications developers can assume some level of homogeneity in the different services their applications interact with.

The discovery of local VADE applications requires the same homogeneity than service discovery. Moreover, local VADE applications must be clearly distinguished from other types of applications, requiring conformance to a given application format. Integrating local VADE applications functionality and user-interface into an external environment also requires agreed protocols.

3.2 Performance

Integrating external environments with local resources implies multiple, often time-consuming interactions (e.g. discovering and interacting with local resources). Continuous changes in the user context originate correspondent interactions with the local environment in order to keep contents up to date with the context. Furthermore, the external home environment may be physically distant from the local infrastructure, which implies a cost on the time required for the interactions. Using physically near resources (from the user's viewpoint) should not result in visibly worse performance, otherwise users may think out of a proximity paradox.

VADE-aware global applications are particularly affected by the performance issue. Since the moment a user enters a VADE domain, several time-consuming steps are undertaken until the information is presented by each application to a portal user: a) obtaining the user context from the Context Manager; b) accessing the VADE Entry Point to obtain the URL of the local registry; c) querying the local registry for a service type available at the user location; d) interacting with the service to extract relevant information; and e) integrating it into the information to be presented to the user. Furthermore, the last two steps are repeated every time the user physical location inside the VADE domain changes. If we consider that dozens of VADE-aware global applications may be constantly performing those operations, multiplied by hundreds of users moving inside a VADE domain, performance becomes a first-class issue. Therefore, some mechanism must be implemented to alleviate VADE-aware global applications from all that burden.

The integration of local VADE applications into an external environment is also time-consuming. When a user enters a VADE domain, local applications are discovered and integrated into the running external environment. These operations should not have an impact on the user experience, i.e., they should occur as invisibly as possible, so that the user does not have to hang on waiting for the integration operations to conclude in order to proceed with the experience.

Likewise, when leaving the VADE domain, the removal of local applications from the external environment should not annoy the user.

3.3 Solving the integration issues

From this analysis, we conclude that a solution for the integration between local and external environments must guarantee interoperability and an acceptable level of performance at all levels of the integration. The solution we propose to deal with interoperability issues is based on widely accepted standards: all components of different domains interact with each other over HTTP and SOAP; local applications and services are discovered using UDDI; finally, local VADE applications are implemented as WSRP portlets, a standard allowing for remote access to user-facing portal components.

Our solution for managing the performance issue consists of an integration middleware coupled to the external user home environment. Such a middleware intermediates the interaction with the local environment, acting on behalf of the external environment components, and thus saving a considerable number of time-consuming operations. In the next section, we describe the integration middleware solution we developed in the context of the VADE system.

4 Personal VADE-aware Middleware

The Personal Vade-Aware Middleware (PerVAM) is a component of the external environment whose main functionality is to mediate interactions with the VADE system. The PerVAM middleware provides mechanisms improving the performance of local applications and services integration into external portals. The PerVAM functionality is to: 1) manage the integration of local applications into the portal every time the VADE domain entrance/exit is detected; 2) act on behalf of VADE-aware global applications in the process of discovering and interacting with VADE core and location-based services, each time the user location changes; and 3) intermediate interaction between global applications and Context Manager.

The PerVAM middleware is especially useful for dealing with the time-consuming operations performed by VADE-aware global applications (see Sect. 3.2) each time a local service is needed. Redundant operations, like VADE core services discovery, are performed by PerVAM only once, just after a user enters the VADE domain. This relieves applications from repeating unnecessarily the same operation. Other operations are only needed when the Context Manager notifies context change. Obtaining user context, discovering local services, or interacting with local services are examples of such operations. Without PerVAM, most of these operations would be performed by VADE-aware global applications, each time the user loads the portal page in her/his browser, even if the context had not changed. This results in significant and unnecessary overhead, specially if a portal page is composed of multiple VADE-aware global applications. The PerVAM middleware relieves applications from those time-consuming

operations, by pre-fetching, in the background, context and application-specific information, each time the Context Manager notifies a user context change (when in the VADE domain). This information is further directly accessed (at the user profile) by applications only when they are invoked by the user, drastically improving their performance.

4.1 Architecture and functionality

The PerVAM architecture (see Fig. 2) is composed by the PerVAM servlet, the portal database, the VADE-aware global applications surrogates, and the PerVAM API. The PerVAM servlet is responsible for managing user context changes and its influence on the integration of local resources. The portal database stores the profile for each registered user, containing portal- and VADE-related information. Each VADE-aware global application surrogate pre-fetches, in the background, location-based information for the application it represents. Finally, the PerVAM API provides a higher-level interface for both the portal and the VADE-aware global applications to use PerVAM functionality, such as access to application-specific content, access to user context, local service discovery, or mediation of interactions with WSRP producers.

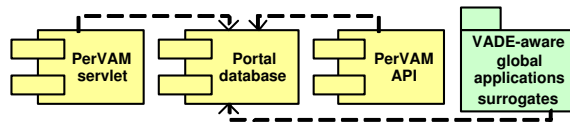


Fig. 2. PerVAM architecture

When the PerVAM servlet is initialized (or whenever a new user registers to the portal), it subscribes portal users for context update notifications in its respective Context Manager. When the Context Manager sends a notification to the PerVAM servlet, PerVAM asks the Context Manager for the corresponding user's context and checks if the user has entered/left a VADE domain, or if the location of a user that previously entered the VADE has changed.

If the user has entered a VADE, PerVAM performs all the necessary integration steps (see Fig. 3). This happens only once for each VADE the users moves in. It starts by discovering VADE core services through the VADE Entry Point. Among other core service URLs, the URL of the local UDDI registry is stored in the user profile for further reuse. PerVAM then searches for local applications, by querying the UDDI registry for WSRP producers. The local WSRP producer is then queried to obtain the references to the portlets it offers. PerVAM finally integrates each WSRP portlet reference in the user's portal. If the user has left a VADE domain, PerVAM simply erases references to previously integrated WSRP portlets.

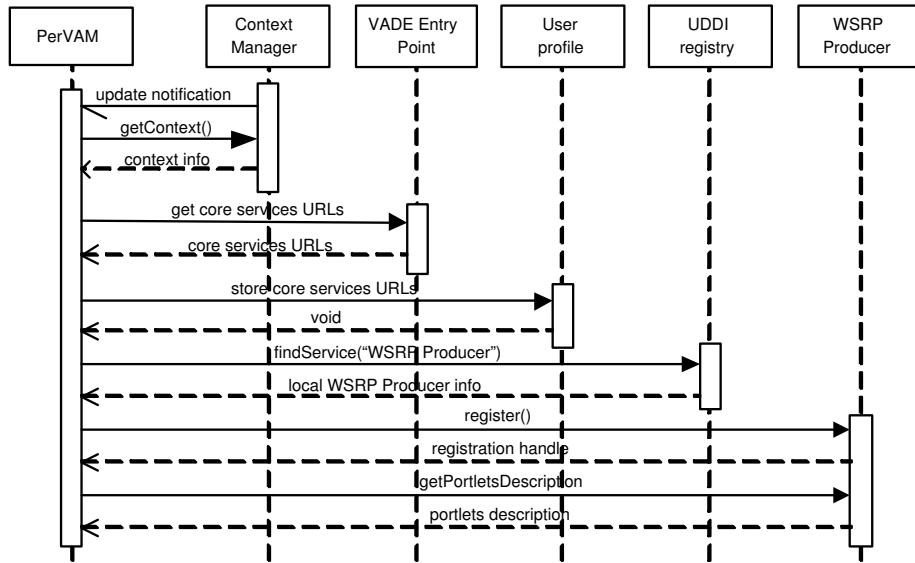


Fig. 3. Sequence diagram for the integration of local VADE applications

Each time the user location is updated¹, PerVAM notifies every VADE-aware global application surrogate (see Fig. 4). Each surrogate then queries the UDDI registry for needed services associated to the new user location. The surrogates then interact with those services and store the collected information in the user profile. Finally, each time a user portal page is displayed, VADE-aware global applications ask PerVAM to extract from the user profile the latest information their respective surrogates collected from location-based services.

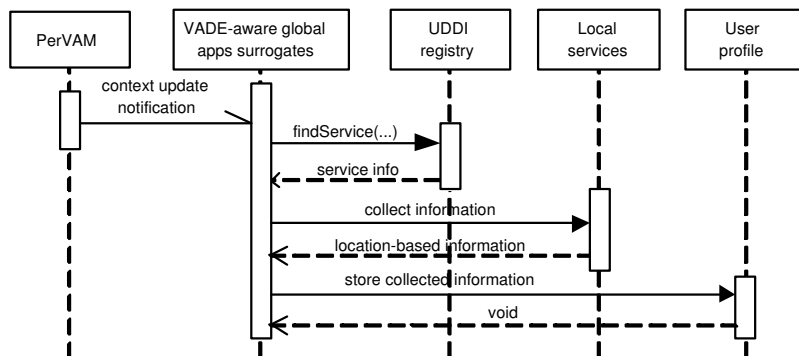


Fig. 4. Sequence diagram for the location-based information pre-fetching

¹ entrance in a VADE domain is regarded as a location update

Global VADE-aware applications implement location-based information presentation, while their surrogates gather the required information from each VADE. Each surrogate's functionality depends on the application it represents, both parties agreeing on some protocol to represent the collected information. Surrogates do not implement any other location-based functionality than those related to VADE environments. Information whose scope is outside a VADE is gathered by applications themselves.

4.2 Prototype

The prototype we have been developing is composed of a VADE system serving the university campus, two external portals, and a number of mobile devices (laptops, PDAs, and mobile phones). The VADE system offers a set of local services and applications, and different types of contextualization mechanisms. The external portals include some VADE-aware global applications supported by their own PerVAM middleware.

Portal development was based on the Jetspeed open-source platform [5]. For location-based services we used Axis [6], and for WSRP portlets development and production we used WSRP4J [7].

5 Related work

The MIDAS project [8] explores the proactive, dynamic functional extension of mobile devices, by means of a local infrastructure, and using the aspect-oriented programming paradigm. The supporting scenario proposes that when mobile devices enter a new computing environment, the environment provides the necessary extensions so that the device can operate in that location at that moment. The scenario assumes that mobile devices must carry a platform that is able to dynamically acquire, apply, and discard extensions as needed. While MIDAS sets the initiative of extending functionality to the local infrastructure, our approach confers the active behavior to the external environment. Besides that, MIDAS only allows for extending a specific functionality with local characteristics. Our approach goes beyond that, making possible to add new, different functionality, augmenting the functional scope of the external environment.

The second perspective relating our work to others is the pre-fetching mechanism. Pre-fetching has been proposed as a valuable strategy to increase performance in Web usage. The performance increase can be accomplished by pre-fetching hyperlinks [9]. This technique corresponds to pre-fetching web documents referred by hyperlinks in another web document. The decision about which documents to pre-fetch derives mainly from the evaluated access probability of each hyperlinked document. The approach we implemented in the PerVAM middleware for augmenting the performance of VADE-aware global applications is very similar to pre-fetching in the WWW. However, location-based information access probability is not considered in PerVAM. As future work, we could consider this topic and also take into account the relevance of updating

location-based information in specific situations. For example, a VADE-aware global application providing map information should not require from its surrogate to search for a local map service each time the user location changes, as the user may be moving in a region that may be well represented by a single map.

6 Conclusions

This paper has described part of an ongoing work in which the concept of a value-added environment (VADE) is explored as a mechanism for enabling the combination of functionality between a local ubiquitous computing infrastructure and external user home environments. We focused on the integration of local services and applications into external environments, with particular emphasis on the interoperability and performance issues. The paper presents the PerVAM middleware, a solution to support the integration of local VADE applications and services into an external portal. Based on Web Services standards, and thus promoting interoperability, PerVAM also improves integration performance, by means of a location-based pre-fetching mechanism.

7 Acknowledgements

This work was carried out as part of the VADE project supported by the Portuguese Foundation for Science and Technology (Ref. POSI/CHS/38475/2001).

References

1. OGC: In support of an open location services (OpenLS) testbed. Technical report, Open GIS Consortium (2000)
2. José, R., Pinto, H., Meneses, F., Boas, N.V., Rodrigues, H., Moreira, A.: System support for integrated ubiquitous computing environments. In: System Support for Ubiquitous Computing Workshop (UbiSys) at the 5th Annual Conference on Ubiquitous Computing (UbiComp 2003), Seattle, United States (2003)
3. IBM: Web services inspection language (2001) <http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html>.
4. OASIS: Web services for remote portlets (2003) <http://www.oasis-open.org/committees/wsrp>.
5. Apache: Jakarta Jetspeed (2003) <http://jakarta.apache.org/jetspeed>.
6. Apache: Webservices Axis (2003) <http://ws.apache.org/axis>.
7. Apache: Webservices WSRP4J (2003) <http://ws.apache.org/wsrp4j>.
8. Popovici, A., Frei, A., Alonso, G.: A proactive middleware platform for mobile computing. In: 4th ACM/IFIP/USENIX International Middleware Conference (Middleware 2003), Rio de Janeiro, Brazil (2003)
9. Duchamp, D.: Prefetching hyperlinks. In: 2nd Usenix Symposium on Internet Technologies and Systems, Berkeley, United States (1999)