Developing a Pervasive System for a Mobile Environment

M. Howard WILLIAMS[1], Nick K. TAYLOR[1], Ioanna ROUSSAKI[2], Patrick ROBERTSON[3], Babak FARSHCHIAN[4], Kevin DOOLIN[5]

[1]School of Math & Comp Sci, Heriot-Watt University, Riccarton, Edinburgh EH14 4AS, UK
Tel: +44 131 4513430, Fax: +44 131 4513327, Email: {M.H.Williams, N.K.Taylor}@hw.ac.uk
[2]School of Elec & Comp Eng, NTUA, 9 Heroon Polytechneiou St, 157 73, Athens, Greece
Tel: +30 210 772 2422, Fax: +30 210 772 2530, Email: nanario@telecom.ntua.gr
[3]Institute for Comms & Navigation, DLR, PO Box 1116, 82230 Wessling, Germany
Tel: +49 8153 282808, Fax: +49 8153 281442, Email: Patrick.Robertson@dlr.de
[4]Telenor R&D, Otto Nielsens vei 12, NO-7004 Trondheim, Norway
Tel: +47 9928 6869., Fax: +47 9621 4263., Email: Babak.Farshchian@telenor.com
[5]Telecomms Software & Systems Group, Waterford Inst. of Technology, Waterford, Ireland
Tel: +353 51 302935, Fax: +353 51 302901, Email: kdoolin@tssg.org

Abstract: The problem of personalised context aware service selection and composition is an important research area that is addressed within the pervasive service platform being developed by the Daidalos project. This paper briefly outlines the scenarios used and the overall platform architecture that underpin this development. It then describes the approach used to select service components and compose them to produce a composite service that satisfies an individual user's needs and takes account of changing context. It also discusses the interaction between personalisation and context management,. These ideas have been extensively tested and demonstrated with the scenarios. In the second phase of Daidalos, the developments will be generalised and subjected to wider testing.

1. Introduction

The environment for mobile users is steadily increasing in complexity. This is due in part to the rapidly expanding range of different services available, the increasing number of heterogeneous communication networks and the developments in sensor and device technologies leading to potentially large numbers of sensors and devices in the user's environment [1]. The goal of pervasive computing is to provide an intelligent environment to enable the user to manage this situation with minimal intervention [2, 3]. This includes dealing with services and resources in a dynamic fashion that best meets the user's needs and preferences [4].

Adaptability and personalisation are two key aspects of a pervasive computing system. In the case of adaptability, the system needs to adapt its functionality and behaviour depending on the context of the user and the resources that are available at any instant. In particular, as the user moves around, a pervasive system should track the changing context of the user, including the devices, services and networks available as well as their properties (e.g. Quality of Service) and adapt its behaviour when necessary. The role of personalisation is crucial in this respect as the system needs to keep track of a user's personal preferences and their dependence on context [5]. This is essential in order to adapt the behaviour of the system to meet the needs of the user with minimal user intervention.

These two aspects are particularly important for telecommunication services in a wireless pervasive computing system and, together with managing the context of the user and ensuring security and privacy, provide the key elements on which the Daidalos pervasive system is based. The FP6 project Daidalos [6] aims to integrate a range of heterogeneous networks in a seamless way and develop a pervasive service platform on top of this to provide the user with personalised and context aware dynamic behaviour with minimal user intervention. This paper is concerned with some of the issues and approaches used to achieve adaptability and personalisation in this pervasive system.

This paper outlines briefly the scenarios that have motivated the need for such a system and driven its development. Section 3 describes the architecture that has been derived from the scenarios and which has been implemented and demonstrated. Section 4 focuses on the specific problem of personalised context-aware service selection and composition and describes briefly the approach used to select service components and compose them to produce a composite service that satisfies an individual user's needs and takes account of changing context. Section 5 concerns the infrastructure established to handle the management of context information in the pervasive service platform. This infrastructure is utilised by the personalisation module and their close interdependency add substantially to the user-friendliness of the pervasive service provision. Section 6 describes the problems of integration while section 7 covers the business models.

2. Scenarios

A complex pervasive system offers countless possibilities regarding its physical scope and deployment, configuration, user-interaction, service provider integration, and end-device support. In order to balance the need for a suitably flexible system (i.e. a generic platform) against those of having sufficiently concrete, implementable, and realistic applications, a scenario driven design approach has been adopted.

At an early stage two important scenario domains were identified for the Daidalos architecture. The first, the "Mobile University", is based on the need to adapt European academic institutions to the needs of mobile citizens of today, and illustrates possibilities of a service platform architecture with strong network capabilities in a university environment. The second, "Automobile Mobility", addresses the challenge of integrating a car's networking, navigation, communications, and I/O resources into an overall environment. This scenario covers a combination of business, leisure and other day-to-day activities, and includes the continuity between activities at home or in the office and those in the car. More recently, a single motivating scenario ("Nidaros") that captures most of the functionality of the current platform was adopted. The following is a brief excerpt from this scenario:

Bart is a sales representative for company X. The scenario opens with him sitting at home watching a newscast on his monitor screen when his boss phones him. The system recognizes that a call from his boss should have priority and that Bart's preferred device for this is his PDA. As a result the system directs the VoIP call to his PDA and automatically puts the newscast on hold. His boss asks him to go to the airport to meet a client, Rosalyn.

While continuing to talk to his boss, Bart leaves the house and goes to his car. When he starts the car, the voice session with his boss is automatically switched seamlessly from his PDA to the car system. And he drives off while still talking. When the voice call is terminated, the newscast that he was watching at home is resumed in his in-car multi-media system – playing as audio only since he is driving.

As Bart approaches the airport, the display automatically switches to the Airport Information System. This system shows him specific information about Rosalyn's flight. As the flight is on time, Bart leaves his car to go to the Arrivals Hall. Meanwhile, Rosalyn disembarks from her flight and turns on her PDA. Rosalyn's location is monitored by Bart's BuddyFinder application (which has been authorized by Rosalyn to access her location).

3. Architecture

From the scenarios an architecture has been derived [6], which provides the full range of functionality required not only to meet the needs of the official scenarios but also those of other scenarios and demonstrations.

However, the overall objective of Daidalos is wider than simply developing a pervasive system. The main aim is to develop and demonstrate an open architecture, which will combine diverse complementary network technologies beyond 3G in a transparent and seamless manner and on top of this provide pervasive access to services. As a result of this more ambitious goal, the overall architecture is based on two separate platforms: the Pervasive Service Platform (PSP) and the Service Provisioning Platform (SPP).

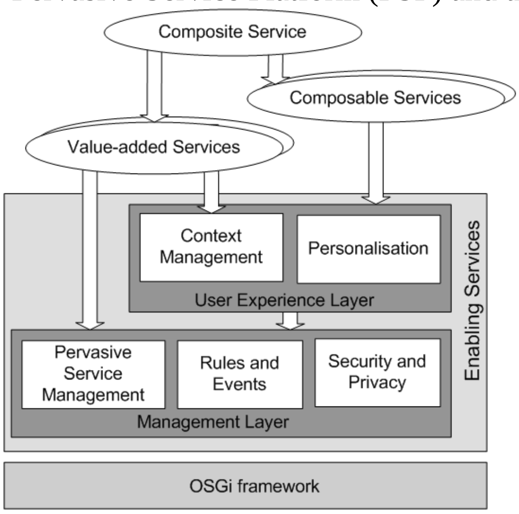

Figure 1: The Pervasive Service Platform

The PSP is at the upper level, i.e. closest to the user. It cooperates with the underlying SPPs to achieve its main task: the provision of services to the user in a pervasive environment. The SPPs are responsible for the lower level support needed to provide end-to-end service delivery across a range of different networks. In particular, their purpose is to provide full telecommunication support for real-time and non-real-time session management. This includes establishing, managing and terminating sessions in a multi-provider federated network. They also interact with components of the PSP in brokering the QoS (Quality of Service), A4C (Authentication, Authorisation, Accounting, Auditing and Charging) and other enabling services on behalf of the user. In doing so they must take account of personalisation of the enabling services .

In the first phase of Daidalos the architecture of the PSP was based on five main software components (see Figure 1), namely:

(1) Pervasive Service Manager - responsible for the discovery, selection and (re-) composition of services in a way that protects the user from the complexity of the underlying networks, devices and software.

(2) Personalisation - uses information on the user's current context and his/her preferences to provide personalisation at various points in the provision of user services.

(3) Context Manager - manages the context information relating to the user's current situation, including information from sensors in the user's vicinity.

(4) Rule & Event Manager - monitors events that may occur when some aspect of the context changes or when they are activated by rules and also manages the rules that drive the overall control of the system.

(5) Security and Privacy Manager - responsible for overall security and privacy by ensuring that the user's identity and aspects of personal data are not revealed to unauthorised parties.

The first three of these cooperate together to provide the adaptability and personalisation required of a pervasive system in a mobile environment. The next one is used as the "glue" connecting these components and driving the system while the final component aims to protect the user against problems arising from breaches of security and privacy. Value-added services are not part of the platform specification, but offer additional functionality to the end user. They may include hardware services such as printers or displays, or software services such as web or multimedia services. Composite services use the functionality offered by the components of the PSP and are seamlessly composed with value-added services and other composable services in order to satisfy user requests.

4. Personalised Context-Aware Service Selection and Composition

One very important aspect of a pervasive system is to provide an adaptive approach to service provision that takes account of the user's current context and personalises the service offering accordingly. Thus when the user requests a service, a pervasive system should locate the most appropriate resources that will meet the user's needs and compose these to create a service session for the user. In particular, such a system needs to take account of user preferences relating to the QoS on a networking and service level, the cost of services, preferred suppliers, etc., to select appropriate service components and compose them into a tailored service that meets the requirements of the user in the current context.

This process is handled by the Pervasive Service Manager (PSM). It creates customised services from existing services and resources through a process of discovery, personalised selection and integration of those services in a planned order and, finally, execution of the result to satisfy the original request [7, 8].

The process starts with a request from the user for a specific service. This is passed to the PSM to perform Initial Service Composition. In order to achieve this, the process is broken down into four steps:

(1) Service Discovery. When the PSM receives a request from a user, the Composition Manager breaks it down into a number of separate component service requests. These are passed to the Service Discovery module. The latter follows a conventional approach. However, when it receives a request for a service, there will in general be more than one candidate service, which is able to fulfil the specific task required, and hence this component returns a list of possible services that could be used to satisfy the request.

(2) Service Selection. It is unrealistic to expect the user to select the most appropriate of these components at this stage. Thus, when the list of possible component services is returned by the Service Discovery module, control is passed to the Personalisation component to perform the selection. The latter fetches the user preferences from the Context Manager and executes these, consulting the current context of the user wherever required. This filters the list of discovered services and selects the service that will best meet the user's specific goal given the user's preferences in the current context.

User preferences capture the particular requirements or preferences that need to be taken into account and which will, in general, depend on the context of the user. These include the cost of a service, the QoS, speed, preferred supplier, etc. For example, if a user is at work, QoS may be the sole criterion to be used whereas if the user is at home he/she may want the best QoS within certain cost constraints.

(3) Personalisation of Component Services. Once the most appropriate services have been selected, these individual services must be personalised. This is achieved by passing an appropriate set of attributes to each service as parameters. These parameters, referred to as Personalisable Parameters, provide an opportunity for individual users to tailor particular features of a service to their choice. A service that permits personalisation must provide appropriate interfaces, which clearly identify the personalisable parameters of the service. Again user preferences determine the parameter values used based on the current context.

(4) Service Composition. Finally, the set of component services need to be combined together to give a complete service offering. This is achieved through the use of a 'Service Model', which defines the set of service types needed to make up the composed service. Some of these may be compulsory for the functioning of the composed service while others are optional. Once again Personalisation has a role to play in adapting the order of component services, where necessary, to satisfy user preferences in the current user context.

This approach provides an initial service composition. Similar approaches have been used by others for web service composition [9] and for automatic configuration of applications [10] as well as for pervasive services [11]. However, one of the important aims in Daidalos is to provide dynamic service re-composition, which goes beyond the work done by others thus far. The idea here is that once a service has been composed in response to a user request, it will be monitored throughout its execution to ensure that any changes that may occur do not substantially affect any of the constituent services.

There are various reasons why the applicability of a service to the composition might change. For example, the service may no longer be available (if the user moves out of range of a WLAN) or a service with a higher preference becomes available (if the preferred service was initially not available but subsequently becomes available). In particular, the QoS available on a service may change, especially as the user moves around or the traffic on a network varies. Other attributes of the service such as cost may change (e.g. cost charges based on time), which may violate the preferences of the user. If any of these changes do occur, the Personalisation module will check whether the composition still meets the user's preferences in the new context, and if not the PSM will be triggered to re-compose the service.

However, the general approach envisaged for dynamic service re-composition is complex and potentially time-consuming. As a first step in this direction a lightweight mechanism is needed that can change the configuration of a composed service very rapidly without any obvious delay to the running services. Such is the case when it becomes necessary to change networks. For example, suppose that one is talking on the hands-free phone while driving in one's car. If the QoS for the network one is using starts to fall, and there is another network available that is able to provide a better QoS, ideally the user would like the system to switch networks to provide the best QoS and to do so in a seamless way so that he/she is not aware that anything has happened.

In this case one does not want to change the composed service apart from changing one of the resources used by it. In view of the speed required to avoid any loss of information, this needs to be treated efficiently. Here the SPP needs to take action directly rather than the PSP becoming involved. User preferences still need to be taken into account, but these can be complex and evaluation can delay the process of changing network when the QoS falls. For instance, the user may have different preferences depending on his/her location, the time or day of week, etc. To facilitate the process of switching networks as rapidly as possible, Personalisation determines the user preferences with regard to the network and stores a copy of them in the underlying layer for access by the SPP. This copy is updated by Personalisation whenever the related user preferences are changed. The SPP monitors QoS and takes the decision to change network based on user preferences when required.

5. Pervasive Context Management

The current context-aware applications are usually built in an ad-hoc manner and lack the generality and standards to be useful in pervasive environments [12]. This section is concerned with context awareness in pervasive computing environments that is established by the Context Management system (CM) of the Daidalos Pervasive Service Platform [6].

The main functionality offered by the implemented CM consists of the following:

- Context representation and modelling based on location-centric data models and ontologies [13].
- Collection of context information from various context sources (e.g. sensors, users, services/applications, network, …)
- Notification of the interested parties about specific context updates (CM event handling mechanism)
- Context maintenance and access support
- Distributed context management and federation, so that context data are exchanged between CMs and replicated whenever necessary in an efficient manner [14]

In the second phase of Daidalos the following aspects will also be addressed:

- Intelligent context retrieval and filtering including location-based, entity-based and free-text-based query handling [15]
- Context inference based on Bayesian techniques [16]
- Sophisticated context consistency control aiming to minimise the overall communication and outdated context retrieval costs [12]
- Representation and maintenance of context history
- Integration of quality of context concepts in the context semantics
- Security of context data [14]

As already stated, Personalisation often interacts with the CM for various reasons. It stores as context entities the user preferences in the context repository. It also stores new preference rules deduced from the user's behaviour that are triggered by the CM whenever the value of a relevant context information changes. Finally, it retrieves context information relevant to users in order to select services and configure service parameters.

6. Integration

One of the serious challenges of this project and one which sets it apart from some other pervasive system developments is the fact that it encompasses the full range of software from a very low level (such as network control) to a very high level (such as user preferences, reference services and applications). This led to the division of functionality between two separate platforms and to the subsequent major task of integrating all of this to create a single operational system. The project chose to adopt OSGi for hosting the PSP components on each of the deployment nodes (e.g. CarPC, homePC, PDAs, servers, etc). Therefore, an OSGi container was instantiated on each node and the appropriate Daidalos PSP services were deployed, configured and started. Communication between the containers was performed using SOAP over HTTP and the lower layer interfaces provided by the SPS on the same node were addressed via java RMI. The exemplary third party services which were developed to demonstrate the overall system and the user experience comprised server components and some simple applications with appropriate user interfaces. These applications were also deployed in the OSGi containers and they used the Daidalos API to interface with the PSP enabling services described above.

7. Business Models and Impact

The majority of scenarios for pervasive computing emphasize the seamless availability of services and resources in active spaces. In its current form this seamless availability is limited to nomadic situations where users are in trusted pre-configured active spaces. In order to support wider availability and global usage of active spaces we need clear business models and, in particular, clearly defined value networks.

As a part of our research, we have identified a set of functions, which we call *enabling services*, which seem to be common to most pervasive computing involving active spaces. A common denominator for these enabling services is that they support *global pervasiveness*, i.e. pervasiveness that expands across space and time and administrative/operational domains. We assume these enabling services to be present globally and to be operated in a trustworthy manner by specialized actors.

Our view is that most of the functionality of the enabling services will not be provided by a single operator but by multiple operators. One of the major challenges, both technically and from a business perspective, will be to federate the functionality provided by different operators, who in most cases do not know each other. Some examples of where federation will be useful for global pervasive computing include:

- Discovery and registry: Although each user can have a personal discovery component, e.g. on her smart device, most of the time this component has to communicate with other discoverers operated by other operators.
- Identity management: Most existing identity management architectures are based on multiple identity operators and federation among them. Federation will be necessary for exchange of private data among users or between users and resources etc.
- Personalisation: Individually personalised services will amount to no more than customisation unless the personal preferences which a user expresses whilst using one service/application can be used to intelligently predict their preferences whilst using similar services/applications.
- Context management: To provide better value to users, context operators will have to federate their own context information with that provided by other context operators.
- Learning: In addition to the benefits to personalisation and context which the federation of information supply will offer, the sharing of data which federation facilitates is essential to the task of learning. E.g. To predict what a user is likely to want to do at what time and in what context and with which service preferences will require a substantial degree of co-operation and trust between data providers.
- Session management: A pervasive session will include resources and services from different interaction spaces and different administrative domains. We can expect that some providers will have their own session operators for their interaction spaces. In this case, the user's session operator will need to federate and co-operate with other session operators.

These examples demonstrate that users and operators of enabling services for global pervasive computing will have to take into account a dynamic business environment where each operator must cope with the existence of different operators at different levels.

The main impact of our work will be to allow these business propositions to actually become implementable in a real-world, operator driven environment. The main economic benefit from the technology covered in this paper is the ability to provide context aware personalised services, thus tailoring services to users' needs and giving users what they want from their services, thereby generating greater service uptake. This will lead to higher revenues for service providers and network operators. A number of partners have clear development roadmaps for exploiting commercially ideas emerging from Daidalos.

8. Conclusions

The importance of automatic service selection and composition in pervasive environments is well established. In Daidalos we have focused on the use of dynamic user preferences that depend on context to support this process. This is important, especially for mobile users, whose choice of services may often depend on context. We have also taken the ideas a step further by extending them to dynamic service re-composition. In particular, we have considered the problem of changing networks used by a service when the changing context (e.g. falling QoS) causes the current choice to no longer meet user's preferences.

The approach used to achieve this has been described briefly (more details can be found in [17]). The concepts have been extensively modelled, implemented and tested with a small number of scenarios. However, one concern with the current choice is the issue of scalability and how this functionality can be achieved with large populations of users.

The first phase of the project is now finished and the second phase of three years has begun. The architecture developed thus far provides a useful starting point for the second phase in which work will focus on generalizing dynamic service selection and composition.

The functions of service re-composition, personalization and context management are very closely linked …

Finally one of the major challenges has been that of integration, not simply within the PSP but with the SPP and the underlying devices and networks. This is an issue which is being addressed in the second phase by adopting a more top-down approach in the design of the overall architecture.

Acknowledgements

The work described here is based on results of the project DAIDALOS, funded under the European Community's FP6 Programme. Apart from this, the European Commission has no responsibility for the content of this paper. It should be noted that the information in this paper is provided as is and no guarantee or warranty is given that it is fit for any particular purpose. The user thereof uses the information at his sole risk and liability. The authors wish to thank all colleagues in the Daidalos project developing the pervasive system.

References

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265(3), pp. 94-104, 1991.

[2] J. Sun, "Mobile ad hoc networking: an essential technology for pervasive computing", *Int. Conf on Info-tech & Info-net*, Beijing, China, pp. 316-321, 2001.

[3] A. Zaslavsky, "Adaptability and interfaces: key to efficient pervasive computing", *NSF Workshop on Context-Aware Mobile Database Management*, Rhodes, Greece, pp. 24-25, 2002.

[4] M. Handte, C. Becker, and K. Rothermel, "Peer-based Automatic Configuration of Pervasive Applications," in *Proc. IEEE Int. Conf on Pervasive Services 2005 (ICPS'05)*, Santorini, pp. 249-260, 2005.

[5] M. H. Williams, I. Roussaki, M. Strimpakou, Y. Yang, L. MacKinnon, R. Dewar, N. Milyaev, C. Pils, and M. Anagnostou, "Context Awareness and Personalisation in the Daidalos Pervasive Environment," in *Proc. IEEE Int. Conf. on Pervasive Services 2005 (ICPS '05)*, Santorini, Greece, pp. 98-107, 2005.

[6] B. Farshchian, J. Zoric, L. Mehrmann, A. Cawsey, H. Williams, P. Robertson, and C. Hauser, "Developing Pervasive Services for Future Telecommunication Networks," *WWW/Internet 2004*, Madrid, Spain, pp. 977-982, 2004.

[7] A.-C. Huang and P. Steenkiste, "Building Self-configuring Services Using Service-Specific Knowledge", *13th IEEE Int. Symposium on High-Performance Distributed Computing*, Honolulu, Hawaii USA, 2004.

[8] R. Hirschfeld and K. Kawamura, "Dynamic Service Adaptation", *4th Int. Workshop on Distributed Auto-adaptive and Reconfigurable Systems*, Tokyo, Japan, 2004.

[9] V. Tosik, B. Pagurek, B. Esfandiari and K. Patel, "Management of compositions of E- and M- business web services with multiple classes of service", *IEEE/IFIP Network Operations and Management Symposium* (MOMS 2002), 8(1), pp. 935-938, 2002.

[10] M. Handte, C. Becker and K. Rothermel, "Peer-based Automatic Configuration of Pervasive Applications", *IEEE Int. Conf on Pervasive Services 2005* (ICPS'05), Santorini, Greece, pp. 249-260, 2005.

[11] D. Chakraborty, Y. Yesha and A. Joshi, "A Distributed Service Composition Protocol for Pervasive Environments", *IEEE Wireless Communications and Networking Conference* (WCNC 2004), 5(1), pp. 2579-2584, 2004.

[12] M. Strimpakou, I. Roussaki, C. Pils, P. Robertson, M. Angermann, and M. Anagnostou, "Context Modelling and Management in Ambient-aware Pervasive Environments", *Int. Workshop on Location & Context-Awareness* (LoCA 2005), LNCS 3479, pp. 2-15, Oberpfaffenhofen, Germany, 2005.

[13] I. Roussaki, M. Strimpakou, C. Pils, N. Kalatzis, M. Anagnostou, "Hybrid context modeling: A location-based scheme using ontologies", *3rd Workshop on Context Modeling and Reasoning* (CoMoRea 2006), Pisa, Italy, 2006.

[14] I. Roussaki, M. Strimpakou, C. Pils, N. Kalatzis, M. Neubauer, C. Hauser, and M. Anagnostou, "Privacy-Aware Modelling and Distribution of Context Information in Pervasive Service Provision", *IEEE Int. Conf. on Pervasive Services* (ICPS'06), Lyon, France, 2006.

[15] C. Pils, I. Roussaki, M. Strimpakou, "Location-based Context Retrieval and Filtering", *2nd International Workshop on Location- and Context-Awareness* (LoCA 2006), Dublin, Ireland, 2006.

[16] M. Angermann, P. Robertson, T. Strang, "Issues and Requirements for Bayesian Approaches in Context Aware Systems", *Int. Workshop on Location & Context-Awareness* (LoCA 2005), LNCS 3479, pp. 235-243, Oberpfaffenhofen, Germany, 2005.

[17] Y. Yang, F. Mahon, M. H. Williams, T. Pfeifer, "Context-aware Dynamic Personalized Service Re-composition in a Pervasive Service Environment", *Proc UIC 06*, China, 2006.