

Research Challenges in Mobile and Context-Aware Service Development

Julien Pauty, Davy Preuveneers, Peter Rigole, Yolande Berbers

Department of Computer Science, K. U. Leuven
B-3001 Leuven
Belgium
Email: `firstname.lastname@cs.kuleuven.be`

Abstract

In this paper we present the major research challenges in mobile and context-aware service development. We present our contribution to several research challenges and we discuss challenges that remain open. Our research is carried out through two projects: CoDAMoS (Context-Driven Adaptation of Mobile Services) and CROSLOCIS (Creation of Smart Local City Services). We contribute to the following five challenges: (1) context-aware adaptation of mobile services; (2) service relocation; (3) service description and discovery; (4) security in service architectures; (5) management of context data.

1 Introduction

With the need for more flexibility and adaptability the IT industry is shifting from making products to providing services. A service typically involves two participants: the service provider and the service consumer. A service is provided upon the request of the consumer.

The advantages of the service approach include:

- Services can be requested by an external entity, provided that the service interface is public. For example, Amazon [Ama] and Google [Goo] already provide services to external entities.
- Service composition: several services can be composed by an entity to provide an enhanced service to other entities.

A software architecture which relies on the service approach is commonly called a service-oriented architecture (SOA).

The service industry is currently facing several challenges [SM05], such as:

- How to develop new services? This challenge includes the traditional problems of software engineering. More specific to the service industry, it also includes :
 1. **Service description:** how to describe the service from the service developer point of view and from the service consumer point of view?
 2. **Service discovery:** how to discover new services, matching the user's needs.
 3. **Service monitoring:** how to monitor service behavior, in order to ensure that it matches service specification?
 4. **Service composition:** how to compose a new service on the basis of several existing services in order to create a new service?
- How to provide services that leave the user in control? This challenge addresses the compromise between automatic and user controlled service provision.

- How can different kinds of services interoperate with each other and on different kind of networks?

In this paper, we focus on mobile and context-aware services. In their simplest form, mobile services are traditional services delivered via mobile devices, such as mobile phones or PDA's. Mobile services can also be specifically tailored to the needs of mobile users. A context-aware mobile service is adapted to the current situation of the user. The goal of a context-aware service is to support the user by providing him with the right service at the right moment. If the user context changes, the context-aware service should self-adapt or be adapted to the new context. A context-aware service is autonomous and tries to support the user without too much interaction with a computing device.

The creation of mobile and context-aware services raises new research challenges, including:

- **Service adaptation:** mobile and context-aware services must be provided on several kinds of devices, ranging from wall displays to mobile phones. These devices have different resources, such as screen size or memory, so services must be adapted to the available resources on the running host.
- **Context-awareness:** the services must be aware of the current context of the user and self-adapt to context changes.

In this paper, we present the major research challenges of mobile and context-aware service development. We present how we address part of these research challenges in our current research projects and we discuss the remaining open challenges. Our research projects are mostly dedicated to the creation of methodologies, middlewares and infrastructures to develop and execute mobile and context-aware services.

We are coordinating the CoDAMoS (Context-Driven Adaptation of Mobile Services) project and are a research partner in the CROSLOCIS project. In CoDAMoS, we work on innovative and generic software methodologies and techniques to support the context-driven adaptation of mobile services. The developed system enables any service to detect changes in the user context and to dynamically adapt the services to this new context. In CoDAMoS we address three challenges: (1) context-aware adaptation of services; (2) service relocation; (3) service description and discovery.

The aim of CROSLOCIS (Creation of smart local city service) is to design and develop a service framework. This framework will enable the framework users to easily develop local services for smart cities, especially context-aware services. In CROSLOCIS we will address several research challenges:

- a service and architecture that spans the telecom and IT world;
- a set of relevant enabling services to ease the creation of end-users services;
- a set of tools to enable the framework users to easily create services;

In CROSLOCIS, we will be responsible for two major challenges: (1) security infrastructure in the service architecture; (2) management of context data in a privacy respecting way.

Section 2 presents the CoDAMoS projects and the associated research challenges. We continue in section 3 with the presentation of the CROSLOCIS project. Before concluding, we present in section 4 two related research projects.

2 CoDAMoS: context driven adaptation of mobile services

CoDAMoS [Cod] is a strategic basic research project aimed at solving a set of key challenges in the area of Ambient Intelligence (AmI), where personal devices will form an extension of each user's environment, running mobile services adapted to the user and his context.

In CoDAMoS, research is carried out by a research consortium consisting of four partners from different Flemish universities. This consortium is backed up by a user commission consisting of industrial companies, such as Alcatel, Philips and Siemens. As part of the research consortium, the DISTRINET research group is responsible for the creation of a context-aware middleware, which enables the creation of services that self-adapt to context variations.

In CoDAMoS we contribute to the first global challenge (service creation) identified in section 1. We also contribute to the execution support of mobile and context-aware services. We propose a generic methodology and a middleware dedicated to the creation of context-aware mobile services.

In this section we first outline the research challenges addressed by the context-aware middleware via an ambient intelligence scenario. Afterwards, we describe our context-aware middleware which addresses part of these challenges. Finally we discuss some remaining open challenges.

2.1 Example scenario

Mr. Smith is a sales representative of a large company and is attending a business meeting where the economic analysis manager is presenting the results of a recent survey and where the participants are discussing marketing strategies. All attendants can use a display to interact with the shared white board in the conference room. Mr. Smith has to leave the meeting early as he is scheduled to visit a client to address some long-standing maintenance problems. During the trip, he would like to stay in touch with his colleagues and discuss the proposals being presented.

The conference room is able to broadcast meetings for audience members who cannot contribute live to the meeting, but who are able to watch and listen to the presentation remotely. They can interact during the meeting using a text-based communication application to send and respond to questions or to read the text that the presenter enters. Upon detection that Mr. Smith is leaving the conference room, the conferencing client moves from his display in the conference room to his personal wireless handheld device. As the I/O capabilities of his handheld are rather limited, some parts of the conferencing client can temporarily be disabled, replaced or moved to other devices. Parts of the client can also be moved to devices in the vicinity of Mr. Smith because the battery has no more power, a larger display is available...

Additionally to challenge one (service creation), this scenario underlines several research challenges that are more specific to the development of context-aware services:

- **Service adaptation:** the mobile conferencing client should be deployable on different hardware platforms, ranging from desktop systems to mobile handheld devices. This should be possible without any modification or manual reconfiguration of the service by the user.
- **Context-awareness:** the supporting infrastructure should be aware of Mr. Smith leaving the conference room and initiate the relocation of the conference client to preferably his personal handheld device or otherwise to an appropriate device in his immediate vicinity.
- **Resources discovery:** service relocation requires on the fly discovery of devices and resources in an ad-hoc network with the ability to check whether these devices provide the required capabilities, such as specific input-output facilities.
- **Service state capture and transfer:** for seamless relocation, the state of the conferencing client should be preserved while moving from one system to another to ensure that the contact list, the current conversation, the presenter's notes, the slides, etc., are not lost during transfer.

2.2 Context-aware middleware

Our context-aware middleware [PVR+05] includes: a context-awareness layer, a distribution module and Draco, which is our component oriented middleware (see Figure 1). Draco enables the service developer to create services based on components. The distribution module enables transparent relocation of the service or part of the service on a distant host. The context-awareness layer enables the service to detect changes in the context of the user and to initiate service relocation on nearby devices. The context-awareness layer can also dynamically replace components with lighter versions to save resources.

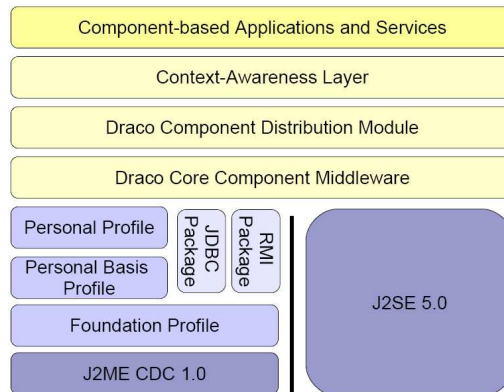


Figure 1 : overall structure of our context-aware middleware

2.2.1 Service description and service discovery

Service description and service discovery are two linked challenges: the description of a service is used during service discovery. These research challenges are not specific to context-aware mobile services.

In the context of ambient intelligence service discovery enables us to discover services that match the user needs. A service can be found in the vicinity of the user on the nearby devices or on a distant host. In CoDAMoS we focus on service located on the nearby devices.

In CoDAMoS, services are described using an ontology which is an extension of the OWL-s ontology [PVW+05]. This ontology enables the service developer to define:

- the required resources by the services;
- contracts, which provide guarantees on the service behavior;
- runtime adaptation, by defining for each component alternative components or whether this component is optional.

Service discovery is done by matching a service request with the service description.

2.2.2 Service adaptation

In CoDAMoS we are targeting ambient intelligence services. Ambient intelligence implies that the user is surrounded with different kind of devices, ranging from mobile phones to desktop computers. As outlined in the preceding scenario, services of ambient intelligence must be able to run on these different target devices. However, these devices have different resources limitations and input-output capabilities, so services must be adapted to these device specific constraints.

To reduce the costs of service development, we rely on a component-based methodology. With this methodology, a service is composed of several connected components. A component is a software black box that performs a specific function. Components are composed by means of their interfaces: they can provide interfaces to and require interfaces from other components. Component interfaces are reified into ports. Two components can be connected through ports, if

the corresponding interfaces match. We can consider that a component can provide and request services through its ports.

Component-oriented design is usually seen as a solution to code reuse. In the context of ambient intelligence, component-oriented design offers a simple and elegant way to develop services that can be quickly adapted to several kinds of devices. With our methodology, the service developer specifies for each components if it is optional or mandatory for the service. The developer can also specify alternative versions of the same component, each version using a different amount of resources. For example, on a resource constrained device, optional components may not be deployed, or a down-sized version of a component may be used to save resources. Thus, it is possible to adapt the service to different devices, while keeping the same global software structure for the service.

To support the development and deployment of component oriented services we have developed Draco, which runs both on mobile devices and desktop computers.

2.2.3 Service relocation

In the preceding section we have seen that using a component oriented methodology it is possible to develop several versions of the same service, each version fitting with the available resources of different devices.

In the context of ambient intelligence the user is always surrounded with different computing resources. To give the best quality of service, it may be needed to relocate the service to another device. In the scenario presented in section 2.1, to ensure service continuity, the conference client is relocated on the user's personal device. In this case, the user device has different capabilities compared to the device used in the conference room. Therefore, while relocating the conference client on the user device, it is necessary to adapt it to the capabilities of this device. In this case we perform a runtime adaptation.

To support runtime adaptation of services, Draco enables runtime component update and service relocation. In this way, it is possible to disconnect optional components or replace one or more components with alternative components, while the service is relocated on another host.

Draco also enables the relocation of only one or more component on another host. For example, if the user is close to a free wall display, the display component of the service he is currently using, can be relocated on the wall display to improve his experience. Component relocation is performed by the distribution module [RVL+05].

2.2.4 Context-aware adaptation of services

Once services are developed and ready to be adapted to different target devices we need a mechanism to detect when service migration is needed and how it must be done. This is the role of our context-awareness layer.

We have developed a context-awareness layer to enable a service to self-adapt to changes in the user's context. The user's context is modeled using an ontology [PB05, PVW+05]. Relying on this ontology, applications can define adaptation rules. For example, consider the rule: if the battery level is below 20% and if the user is close to a desktop computer then the service must be moved. Using the facts corresponding to the user location and the battery level, this rule enables the service to determine if it must move to another host or not.

Using this ontology the service developer can create rules which specify when the service must adapt. It is also possible to instantiate rules at runtime. For example, the user might specify in his preferences that he never wants to move the service, or that the service must always try to move to a nearby desktop computer.

The context-awareness layer is itself built up of components. Thus, services can get context-awareness capabilities by simply connecting their components to the context components.

2.3 Open research challenges

In the preceding section, we have presented the research challenges that we address in CoDAMoS. In this section, we present the open research challenges that are specific to context-aware service development.

2.3.1 Service adaptation

In CoDAMoS, possible service adaptations are service specific. We need more generic methodologies to develop self-adaptive services. An open research challenge is to create a language to define the adaptation capabilities of a service. This language should separate functionality and adaptation concerns. A possible approach could be to extend UML to express variability in the service model.

Another open research question is the guarantee of system consistency after adaptation. Indeed, when a service is adapted its functionality must remain unchanged. This challenge is directly linked to service monitoring.

2.3.2 Service composition

In CoDAMoS, services are built with components. With this approach, composing a new service on the basis of existing services is done simply by connecting the components of these existing services. Furthermore, Draco enables runtime connection of components, thus it is also possible to compose new services at runtime.

Our approach supposes that every service is built with our framework. We do not address the composition of services in a general way. For example, we do not address the composition of a Web service with a component-oriented service.

The global challenge of service composition is directly linked to service interoperability. Indeed, several kinds of services already exist, such as Web services or peer to peer services. Consequently, to address service composition in a general way, adaptation layers between each kind of service will have to be developed.

Service composition is also linked to service discovery and service description. The services that are composed are chosen according to their description. Service discovery is needed to compose services dynamically. In the context of ambient intelligence, we can envision that users will be provided with services composed with the services provided by the nearby devices.

2.3.3 Service monitoring

Service monitoring implies that the service's behavior is recorded and analyzed at runtime, in order to check that the behavior matches the description. The service provider and/or the service consumer can perform service monitoring. Service monitoring is challenging because each entity involved in a service has only a limited view of the service execution. This challenge is emphasized for composite services. A composite service is distributed across several service providers, so mechanisms have to be developed to merge the local observations into global monitoring.

Service monitoring also implies that actions are performed whether the service diverges from its description:

- Should we stop service delivery or just notify the user of service misbehavior?
- If monitoring concern only quality of service (QoS), should we renegotiate the QoS and continue service delivery?

Mobile and context-aware services add another dimension to service monitoring: context monitoring. Context monitoring is already used to find relevant services. However, in order to ensure that services remain relevant, context has to be monitored also during service provision.

As the user context usually contains personal data such as the user location, recording context naturally leads to privacy problems. Recording context-data on the user personal device can be a possible solution, but we have to keep in mind the resource-constrained nature of such devices.

3 CROSLOCIS: Creation of smart local city services

CROSLOCIS [Cro] is a new basic research project dedicated to the creation of a service architecture where new innovative local mobile services can easily be created, deployed and consumed by mobile users. The overall goal of this architecture is to lower the threshold for SME to set up their own mobile services and allow charging from the content or service provided in business-to-consumer or business-to-business scenarios.

The CROSLOCIS project is complementary to CoDAMoS, in the sense that its focus is broader, i.e. mobile services, and not just ambient intelligence environments. The focus of CROSLOCIS also differs from the focus of CoDAMoS: where CoDAMoS is looking at methodologies and techniques for adaptation, CROSLOCIS is focusing on architecture. Where context is used in CoDAMoS but is not a primary focus, context receives explicit attention in CROSLOCIS.

In CROSLOCIS we will address challenges one (service development) and challenge three (service interoperability).

In CROSLOCIS research is carried out by a research consortium of three Flemish universities, K.U. Leuven, U. Gent and U. Hasselt, and five companies Siemens, i-City, Microsoft, Concentra media and Androme.

3.1 *Identified inhibitors for the development of mobile services*

Today, third parties cannot easily introduce ubiquitous, innovative services to the general public or to targeted customer segments, on top of wireless network technologies for enterprises or public locations. We have identified a number of key inhibitors to this new market opportunity:

- **Heterogeneous service environments:** on the one hand we have the IT service environment, which is based on Web services in its latest generation. On the other hand we have mobile or fixed-mobile telecom environments (UMTS IMS or Tispan architecture), consisting of essentially different enabling services concepts, all on top of different middleware and network technologies. This heterogeneous environment results in too environment-specific enabling services, with only partial long-term requirements taken as a prerequisite, such as common formats and procedures for collecting, storing and exchanging identity, location, presence and profile data in a secure but application-independent way.
- **Unclear business and value chain roles of the various types of service providers,** needed to compose a more valuable service experience. Example issues are: subscriber data ownership, privacy, data federation across administrative boundaries, charging policies, revenue sharing models...
- **Lack of service management and administration,** needed in order to allow for maintaining service-user relationships and service inventories, performing resource management, and the enforcement of appropriate real-time charging.
- **Lack of service authoring tools that can leverage common service environment foundations** for building attractive and innovative services for end-users. Such tools should be perceived as appealing and efficient by third parties service developers, while hiding the inherent framework enabling services complexities.

To address these inhibitors the CROSLOCIS project aims to create:

1. A service architecture that spans the telecommunication and IT worlds;
2. A set of relevant enabling services to ease the creation of end-users services;
3. A set of tools to enable the framework users to easily create services;

3.2 A converged service architecture

With the need for more flexibility, the concept of Service Oriented Architecture (SOA) is emerging in the IT industry. A SOA provides the necessary communication tools and service composition mechanisms to build the composite service needed by the final business processes. Several layers of composition may be needed to create the business processes (see Figure 2). In this way, individual or collections of services can be combined and choreographed to produce new composite services that not only introduce new levels of reuse, but also allow the dynamic reconfiguration of business systems, while respecting service component “contracts” between businesses.

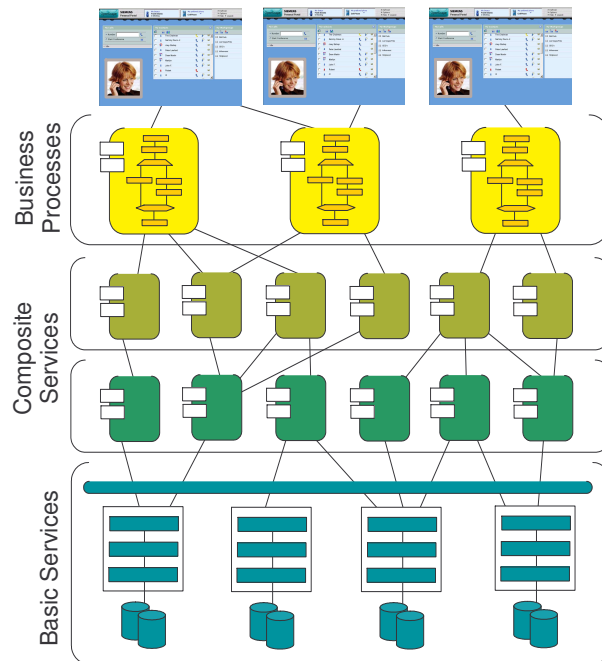


Figure 2: layered composition of business processes from basic technical services

The emerging SOA architecture principles are today being applied for “regular” (IT) business services, with B2B scenarios as the most challenging ones. The CROSLOCIS project will take the SOA state-of-the-art as a promising approach to go to a broader service architecture that is oriented towards public services and includes services incorporating real-time person-to-person communication services as typically realized on top of a SIP-based telecommunication environment, and several kinds of circumstantial information related to geographical location.

3.3 Research challenges tackled in CROSLOCIS

In the CROSLOCIS project the Distrinet group will address two main research challenges:

1. **Security:** in a service platform, security is needed to authenticate the user when he requests a service, and to ensure the non-repudiation of the charging contract between the user and the service provider.

2. **Collection and distribution of context data:** context data will enable service providers to tailor their services to the customer needs and preferences. However, this data is privacy sensitive, so the service platform must have mechanisms that enable the service providers to access context data while preserving the privacy of users.

3.3.1 Security architecture

The main objective is the inception of an access control and non-repudiation solution for the service architecture. We will start by studying the limitation of the existing PKI infrastructure used in the Belgian electronic identity card (e-id). The e-id has a classical bankcard format and contains two public private key pairs and the corresponding certificates, one for authentication signature, and one for non repudiation. These certificates and the PKI backend infrastructure are not only used by the government, but can also be used by commercial parties. It is even possible that in a future evolution all private PKI infrastructures will become obsolete and that banking applications, ATM, internet e-commerce.

Leveraging this evolution in the architecture for the creation of smart local city services is essential. However, this will require a much more powerful backend service-layer security architecture to cope with all different services relying on the e-ID card, than what a PKI infrastructure in itself is offering. For example, in the envisioned services, the e-ID card will be physically needed for multiple devices at the same time (e.g. if you need the card to logon to your PC, you need to take it out to make a phone call with a dedicated mobile device). Therefore, we will have to investigate possible ways to split the physical authentication function of the card via various devices from its impact on the security. Moreover, every single service transaction from every party able to act in the service framework must be authenticated and verified against the active policy rules in real-time.

3.3.2 Collection and distribution of context data

In CoDAMoS context was used as a tool to trigger adaptation of services. In CROSLOCIS, context will be used to provide enhanced services, typically services based on the user location and preferences.

Collection of context consists in capturing, storing and reasoning on the user context. In CROSLOCIS, we will use all the available sources of context information. To capture the user context, we will particularly rely on the information provided by the user terminal, but also on information provided by the user profile and the network, such as the cell-id. The user context will be modeled with an ontology. This ontology must encompass all the envisioned city services and remain open for future services. Using a reasoning system and this ontology we will be able to infer higher level context.

In CROSLOCIS, context distribution means that the service platform will enable third-party service providers to create enhanced services by allowing them to access context data. However, context data is privacy sensitive and several service providers may access to this data through the service platform. Therefore, to address context distribution, we will have to develop mechanisms that enable service providers to access context data without breaking users privacy. These mechanisms will have to prevent several service providers to enlarge their view on the users profile by sharing and aggregating their respective knowledge on the users. The mechanisms will also have to prevent unwanted propagation of context-data among service providers, typically when several services are chained among different service providers.

3.4 Open research challenges

CROSLOCIS is dedicated to city services, so the first open challenge is the generalization of the CROSLOCIS service platform to other service areas. This generalization would imply another

global iteration over the whole project, starting by refining business models and ending with a new version of the service platform.

CROSLOCIS addresses only part of the legal concerns of city services. We will identify and address only the main legal inhibitors for the deployment of city services, such as privacy. We will not address the legal aspects of service subscription, service provision.

CROSLOCIS does not address reliability. However, if users pay for services, services will have to be reliable. For the service area, reliability raises the following questions:

- How to determine the reliability of a composite service? The global reliability will have to be determined according to the role of each service in the composite service.
- How to relate reliability and the charging policy of services? We can envision that highly reliable services will be more expensive than less reliable services. We can also envision that the service's price could drop if the service has temporary reliability issues.

4 Related projects

In this section, we briefly present two related research projects¹ that are also investigating the research challenges of mobile services.

The Amigo [Ami] project is dedicated to ambient intelligence for networked home environment. Home networks start to emerge, but the lack of interoperability between devices, the complexity of configuration and the absence of compelling services prevent it to really change people's lives. To improve the usability and attractiveness of home networking for the end-user, Amigo's main objective is to research and develop open, standardized, interoperable middleware and intelligent user services for the networked home environment, which offer users intuitive, personalized and unobtrusive interaction by providing seamless interoperability of services and applications. Amigo clearly addresses the service creation challenge, with a strong focus on context-aware services. The Amigo middleware also enforces security and privacy policies.

The MADAM [Mad] project is strongly related to CoDAMoS. The overall objective of MADAM is to provide software engineers with modeling language extensions, tools and middleware that collectively foster the design, implementation and operation of innovative applications and services for the mobile user and worker. To achieve this objective MADAM will study the adaptivity requirements of mobile applications and develop a theory of adaptation. A set of reusable adaptation strategies and adaptation mechanisms, based on a dynamically reconfigurable component architecture will be developed. Modeling language extensions and tools will enable application designers to specify adaptation capabilities at design time.

5 Conclusion

In this paper we have presented the main research challenges of mobile and context-aware service development, through the research projects we are participating. We are contributing to the following research challenges:

- **Context-aware adaptation of services:** the CoDAMoS middleware enables services to self-adapt when the user context changes.
- **Service relocation:** the CoDAMoS enables a service to migrate on a nearby host when the available resources drop, to provide the best user experience.
- **Service description and discovery:** the CoDAMoS middleware relies on an extension of the OWL-s language to describe services. Services are discovered by matching a service request with its description.

¹Projects descriptions are taken from their respective website

- **Security architecture:** in CROSLOCIS we will develop a security architecture for user authentication and non repudiation of charging contracts between service providers and users.
- **Management of context data:** in CROSLOCIS we will develop mechanisms to enable service providers to access context data while preserving the users' privacy.

We have discussed the following open research challenges:

- **Service adaptation:** adapting services according to context changes needs a general language separating functional concerns and adaptation concerns.
- **Service composition:** we have seen that the service composition challenge is both linked to service discovery and service interoperability challenges.
- **Service monitoring:** for mobile and context-aware services, service monitoring is especially important to always propose relevant services to user; a service that diverges from its description may fail to support the user.
- **Service reliability:** if a user pays for a service, this service has to be reliable. We can also envision to link services reliability and services' price.

6 References

- [Ama] The Amazon Web Services: <http://www.amazon.com/gp/aws/landing.html>
- [Ami] The Amigo project: Ambient intelligence for the networked home environment. <http://www.hitech-projects.com/euprojects/amigo/>
- [Cod] The CoDAMoS project: Context Driven Adaptation of Mobile Services. <http://www.cs.kuleuven.be/~distrinet/projects/CoDAMoS/>
- [Cro] The Crolocis project: Creation of Smart Local City Services. <http://www.ibbt.be/site/index.php?id=160&L=1>
- [Goo] The Google APIs. <http://www.google.com/apis/>
- [Mad] The MADAM project: Mobility and ADaptation enAbling Middleware. <http://www.ist-madam.org/>
- [PB05] Davy Preuveneers and Yolande Berbers. Adaptive Context Management Using a Component-Based Approach. *International Conference on Distributed Applications and Interoperable Systems (DAIS 2005)*, June, 2005, Athens, Greece.
- [PVW+05] Davy Preuveneers, Jan Van den Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers and Koen De Bosschere. Towards an Extensible Context Ontology for Ambient Intelligence, *European Symposium on Ambient Intelligence (EUSAI 2004)*, November, 2004, Eindhoven, Nederland.
- [PVR+05] Davy Preuveneers, Yves Vandewoude, Peter Rigole and Yolande Berbers. Middleware Support for Component-Based Ubiquitous and Mobile Computing Applications. *International Middleware Conference (Middleware 2005) (demo)*, December, 2005, Grenoble, France.
- [RVL+05] P. Rigole, C. Vandervelpen, K. Luyten, Y. Vandewoude, K. Coninx, and Y. Berbers, A Component-based infrastructure for Pervasive User Interaction. *Workshop on Software Techniques for Embedded and Pervasive Systems (Pervasive 2005 workshop)*, May, 2005, Munich, Germany.
- [SM05] A-M. Sassen and C. Macmillan. The Service Engineering Area: an Overview of it Current State and a Vision of its Future. http://www.cordislu/ist/st/serv_eng.htm